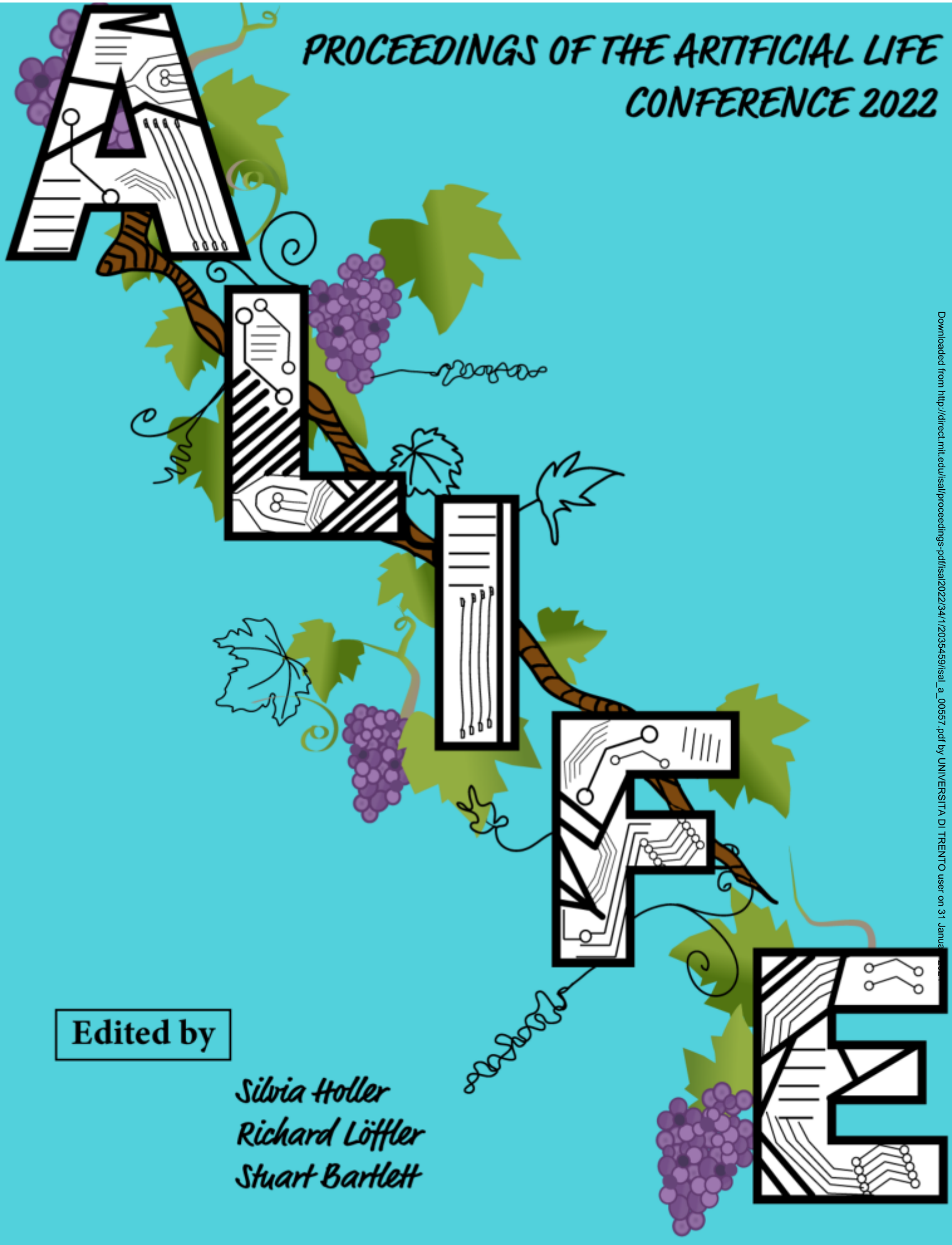


Proceedings of Artificial Life Conference 2022

Silvia Holler[1], Richar Löffler[1] and Stuart Bartlett[2]
[1]Centre of Cellular, Computational and Integrative Biology, Univeristy of Trento, TN, Italy
[2]California Institute of Technology, Pasadena, CA 91125

July 9, 2022

Preface

This volume presents the proceedings of the 2022 Conference on Artificial Life (ALIFE 2022) which took place in Trento, 18-22 July 2022 (https://2022.alife.org/). The conference was held virtually due to the ongoing COVID-19 pandemic.

ALIFE 2022 Theme

The ALIFE 2022 conference theme is 'La DOLCE vita. Discoveries on Life Complexity and Evolution for the improvement of real lives'. The conference theme explores how to improve the quality of real life using techniques and discoveries from the ALife field. This covers various topics including (but not limited to): the creation of artificial cells and organisms for health and technological applications, engineered ecosystems for improved environmental quality and sustainable agriculture, virtual/augmented reality creations with positive social impact, the well-being of our digital infrastructure, AI and ALife algorithms for equitable access to resources and accurate information, AI, ALife or robot assistance for those in need, AI, ALife or robot applications for food production and distribution, the regeneration, redistribution and reuse of everyday resources, microbial fuel cell systems for renewable energy, and other innovative technologies for social good.

The ALife 2021 Program

We received in total 129 submissions. The review process of ALIFE 2022 was double blind. All submissions were reviewed by three reviewers and Senior program committee members then performed a topic-wide meta-review to derive acceptance/rejection decisions. As a result 78 submissions were accepted as oral presentations. Of those 48 are full papers and 30 extended abstracts. Four special sessions are held for discussing more specific topics and in parallel to that, the conference will host ten workshops and three tutorials.
The conference program of this year includes the following items:

- Eight keynote presentations of internationally renowned speakers within a wide variety of topics:
  - **Rob Dunn**; North Carolina State University, *"The Past and Future of the Evolution of Intelligence in Nature*
  - **Stuart Bartlett**; California Institute of Technology, *"Searching for Lyfe, Genesity, Complexity and Emergent Learning"*
  - **Melanie Moses**; The University of New Mexico, *"Learning from COVID-19: The extraordinary success of the questionably alive"*
  - **Job Boekhoven**; Technical University Munich, *"Chemically fueled droplets; towards the synthesis of life"*
  - **Susanne Still**; University of Hawaii at Manoa, *"Inference, prediction and thermodynamic efficiency"*
  - **Dora Tang**; Max Planck Institute of Molecular Cell Biology and Genetics, *"Building life from scratch? Exploiting synthetic cellularity"*
  - **David Obura**; Coastal Oceans Research and Development – Indian Ocean *"Coral reefs, climate change and pathways into the future - a case study for future realities"*
  - **Masatoshi Funabashi**; Sony CSL, *"Life as it will be -The Coevolution of Artificial and Biological life in Anthropocene"*
- Special sessions:
  - Artificial Perception II, organized by Lana Sinapayen, Sofian Audry and Eiji Watanabe
  - Hybrid life: Approaches to integrate biological, artificial and cognitive systems, organized by Manuel Baltieri, Keisuke Suzuki and Olaf Witkowski
  - ALIFE and Society, organized by Peter Lewis, Imran Khan and Alex Penn
  - Artificial Life Journal Session, organized by Alan Dorin and Susan Stepney

- Workshops:
  - ABMHuB'22: 4th International Workshop on Agent-Based Modelling of Human Behaviour, organized by Soo Ling Lim and Peter J. Bentley
  - SB-AI 7: What can Synthetic Biology offer to Artificial Intelligence? Strategies and Perspectives for Embodied Chemical Approaches to AI, organized by Pasquale Stano, Luisa Damiano and Yutetsu Kuruma
  - Chemaliforms II: The Second Workshop on Chemistry and Artificial Life Forms, organized by Jitka Čejkova, Tan Phat Huynh and Richard Loeffler
  - OGD-CLEA: Origins of Goal Directedness: Ideas, Structures and Models, organized by Tomas Veloz
  - LIFELIKE 2022: Lifelike Computing Systems Workshop 2022, organized by Anthony Stein, Sven Tomforde, Jean Botev and Peter Lewis
  - WiDWS: Why it Didn't Work-Shop, organized by Lisa Soros, Lana Sinapayen and Nicholas Guttenberg
  - ERA: Emerging Researchers in Artificial Life, organized by Federico Pigozzi, Abraham J. Leite and Imy Khan
  - Web Hackathon: Developing Artificial Life Web Resources, organized by Emily Dolson
  - VCC workshop: The Virtual Creatures Workshop, organized by Kathryn Walker, Caitlin Grasso, Lana Sinapayen and Sam Kriegman
  - ALife Ethics: Should Artificial Systems Have Rights?, organized by Olaf Witkowski and Eric Schwitzgebel
  - TEMC 2020: 2nd International Workshop on Theoretical and Experimental Material Computing, organized by Susan Stepney, Matt Dale, Simon O'Keefe, Angelika Sebald, and Martin Trefzer
- Tutorials:
  - Introduction to Using Symbulation: An Agent-Based Model of Symbiosis Evolution, organized by Anya E. Vostinar, Emily Dolson, Piper Welch and Kai Johnson
  - The Langtons ant wave equation, organized by Graham Medland
  - Simulating pandemics with agent-based models, organized by Prof. Mikhail Prokopenko and Dr. Sheryl L. Chang

About the Editors

Slivia Holler (General, Program and Local Chair) is a postdoctoral researcher in the Cellular, Computational and Integrative Biology Department of the University of Trento. She obtained her PhD degree at the University of Trento in 2019. She is now a leader in the field of chemotactic droplets, vesicles and protocells, having demonstrated important and novel results in the experimental analysis of such systems. Her research is pioneering new discoveries that may elucidate the first steps in the transition from non-life to life. Furthermore, these results may prove to be generalizable, and apply to other planetary environments beyond Earth. Her research has huge potential for medical and industry applications and her droplet systems constitute a dynamic, configurable set of microscopic transport systems that could be used for focused disease attack and drug delivery. The treatment of cancer, viral and bacterial infections, and various chronic illnesses could be revolutionized by this technique, since the droplets would convey the necessary therapeutics only to the affected cells, and would harmlessly dissolve upon completion of the task.

Richard Löffler (Proceedings Chair) is a postdoctoral researcher in the Cellular, Computational and Integrative Biology Department of the University of Trento. He obtained his PhD degree in physical chemistry in 2021 at the Institute of Physical Chemistry, Polish Academy of sciences as part of a MSC COFUND action. His background is in Nanobioscience with interest in the fields of Biophysics, Nanomedicine, Origin of Life and Artificial life. During his PhD Richard worked on studying self-propelled motion of different non-biological systems and ordered spatial patterns which can appear as the result of different surface phenomena like dewetting or evaporation. During this time he developed and studied several novel systems that range from soft-matter droplets over waxy materials to solid particles. At his current position in Trento he continues to work with such systems with the goal of studying the intersection between chemistry, behavior and information and thereby getting closer to a system that is autonomously able to stay away from thermal equilibrium.

Stuart Bartlett (Submissions Chair) is a scientist exploring how life can begin, why and how it becomes more complex, and how it might be detected beyond Earth. He graduated with an MPhys degree in Physics from the University of Bath in 2008. After an assistantship at the Swiss Federal Institute for Snow and Avalanche Research (SLF) in Davos, Switzerland, he began a PhD in Complex Systems Simulation at the University of Southampton, graduating in 2014. He then took up a postdoctoral position in the Laboratory of Cryospheric Sciences at the Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland. In 2016 he was awarded a postdoctoral fellowship by the Earth-life Science Institute (ELSI) at the Tokyo Institute of Technology. This included research visits to the California Institute of Technology, where he now continues as a staff scientist.

Acknowledgements

I (Silvia Holler) would like to offer my heartfelt thanks to the entire organizing committee of ALIFE 2022. This conference would not have been possible without their hard work. They were always available to help me no matter the (local) time of day, late at night or before sunrise. I would like to thank Stuart Bartlett for having been the fastest and best organized submission chair and Richard Löffler not only for his help with the proceedings but also with web updates and social media posts. Without them my resting hours would have been significantly deprived. I would like to thank Jitka Čejkova for having initially created the ALIFE 2022 webpage and giving me the foundation from which I could make it my own. I would also like to thank those who helped me organize all the independent sessions: Special Sessions Chair Hiroki Sayama, and Workshop and Tutorials Chair Barbora Hudcová. Invaluable efforts were invested by Manuel Baltieri, who helped me contact keynote speakers and from Olaf Olaf Witkowski who was behind the organization of the students abstract and essay competition. I would like to especially thank Martin Hanczyc, my advisor, my mentor, but also the person who encouraged me to embark on this adventure. The organizing committee would like to thank all the reviewers and meta reviewers without whose hard work the submissions, proceeding and the whole conference wouldn't have been possible. We would like to especially thank all of those who were available for last minute reviews. We would like to thank all the authors and congratulate them for the incredible quality of their submissions. We wish to extend our sincerest thanks to ISAL and Connie James for registration and administrative help, to Darya Palchikova for her amazing logo and proceedings front page, to Charles Ofria for always being available for suggestions and useful advices, and Lisa Soros for the knowledge transfer for the proceedings creation.

Senior Program Committee

Kasper Stoy
Alexandra Penn
Tim Taylor
Pasquale Stano
Peter Andras
Takashi Ikegami
Randall Beer
Susan Stepney
Tom Froese
Charles Ofria
Lana Sinapayen
Peter Lewis
Wolfgang Banzhaf
Harold Fellermann
Martyn Amos
Manuel Baltieri
Julyan Cartwright

Program Committee

Ben Shirt-Ediss
Roberto Serra
Dennis Wilson
Keisuke Suzuki
Jaume Bacardit
Hiroyuki Iizuka
Ioannis Ieropoulos
Miguel A. Fortuna
Adam Gaier
Christoph Adami
Joseph Lizier
Thomas Schmickl
Frank Veenstra
Amine Boumaza
Alyssa Adams
Mikhail Prokopenko
Imran Khan
José M Cecilia
Martin Trefzer
Yuki Kubota
Ekaterina Sangati

Hiroki Kojima
Wailok Wook
Petr Švarný
Adam Stanton
Sara Kalvala
Alan Dorin
Sylvain Cussat-Blanc
Geoff Nitschke
Christopher Buckley
Juan-Carlos Letelier
Tomas Veloz
Carlos Gershenson
Nicolas Bredeche
Andrea Roli
Jan Feyereisl
Steen Rasmussen
Michael Crosscombe
Jim Torresen
Kai Olav Ellefsen
The Anh Han
Vito Trianni
Peter Dittrich
Marco Villani
Olaf Witkowski
Timoteo Carletti
Luisa Damiano
Ben Costello

Laura Grabowski
Martin Biehl
Kyrre Glette
Chrystopher L. Nehaniv
Alessandro Filisetti
Arend Hintze
Huw Lloyd
Ángel Goñi-Moreno
Claus Aranha
Alberto Antonioni
Simon Hickinbotham
Andres Faina
Inman Harvey
Charles Martin
Manuel Bedia
Reiji Suzuki
Taro Toyota
Simon Powers
Andrew Philippides
Stefano Nolfi
Alastair Channon
Antoine Cully
Robert Pennock
Atsushi Masumori
Mario Zarco
Miguel Aguilera
Nathaniel Virgo

Omer Markovitch
Mizuki Oka
Christoph Flamm
Sam Kriegman
Katarzyna Kozdon
Emily Dolson
Penelope Faulkner Rainford
Silvio Capobianco
Federico Rossi
Eduardo Izquierdo
Ali Tehrani-Saleh
Leonardo Bich
Eric Medvet
Nathanael Aubert-Kato
Takaya Arita
Jeremy Pitt
Daniel Polani
Pamela Knoll
Yasuhiro Hashimoto
Poramate Manoonpong
Gunnart Tufte
Anya Vostinar
Francisco C. Santos
Matthew Andres Moreno
Lisa Soros
Giovanni Iacca
Erik Hom

The ALIFE 2022 Proceedings Editors:
Silvia Holler
Richard Löffler
Stuart Bartlett

Conference Program

Efficiency Through GPU-based Co-Evolution of Control and Pose in Evolutionary Robotics

Kasper Stoy

IT University of Copenhagen
ksty@itu.dk

Abstract

A key challenge in evolutionary robotics is the computational cost of evolutionary runs. The high computational cost forces researchers to rely on power-hungry computer clusters and, even with these, researchers often are faced with long evaluation cycles that make development of evolutionary experiments a time consuming and tedious effort. In this paper we address this challenge on two fronts. We have developed an evolutionary robotic engine where all individuals are evaluated in parallel using a thread-based implementation on a graphical processing unit (GPU). This engine allows us to run an evolutionary robotics experiment in seconds on a modest laptop. The second avenue of exploration is that we have used this engine to study the role of initial robot poses in fitness evaluation. We find that if we co-evolve initial pose and controller competitively, we can reduce the evaluation period of individuals significantly. Combined the evolutionary robotics engine and the co-evolutionary approach are significant demonstrations of how to make evolutionary robotics more computationally efficient.

Introduction

In evolutionary robotics (Nolfi and Floreano, 2000; Doncieux et al., 2015) it is common to take advantage of the parallel processing power of graphics processing units (GPUs). However, often it is sub-components that are accelerated such as the simulator or maybe the evaluation of a neural network controller. However, individuals in a population are still predominately evaluated sequentially. Instead, what we propose is that all individuals in a population are evaluated in parallel on a GPU. This has drastic speedup potential because we can evaluate n individuals of a population on a single GPU. The enabling technology is that modern GPU allows for implementation of complex algorithms. Although one still must observe some restrictions such as avoiding branching in the code if possible. Doing this we can do full evolutionary runs in the order of seconds even on low-cost laptops making evolutionary robotics more accessible. We follow a general trend in machine learning where increased parallelisation leads to dramatic speed-ups. A prominent example of this is (Rudin et al., 2021) that demonstrate dramatic speed-ups on a reinforcement learning walking task.

Using this GPU-accelerated evolutionary robotics engine we continue our work started in (Stoy, 2021). The work is done in the context of the canonical task of evolving an obstacle avoidance behaviour for a simple differential-drive mobile robot. We use a challenging variation of this task where obstacles are rare due to the environment being a large, empty arena. The observation we build our work on is that it is important to consider where in the environment a robot is placed at the onset of fitness evaluation. If, for instance, an individual is placed far from obstacles either the evaluation will be unproductive, or the evaluation period must be so long that the robot actually encounters an obstacle as part of a fitness evaluation. As pointed out in our initial paper the solution is not to place the robot in a fixed position close to an obstacle, because this will lead to overfitting and may prevent the robot from handling obstacles approached from other angles or distances. Conventionally, the solution is to place the robot at a random position. While this works it also means that many fitness evaluations are counter-productive e.g., an individual scoring high fitness simply because it started far from obstacles and not because it has learned the desired behaviour. Instead, we proposed to co-evolve the initial position with the controller in a competitive setup. The idea is that the co-evolutionary system will try to find initial positions which exposes the weakness of the controller at a given time in the evolutionary process. We found in previous work that this is indeed the case. The contribution in this paper is that we show that we can reduce the evaluation period dramatically from 200 seconds of simulated time to 25 seconds of simulated time.

In conclusion we find that with the combination of a GPU-accelerated evolutionary robotics engine and a significant reduction in evaluation period due to co-evolution of initial pose and controller we can make a highly efficient evolutionary robotics setup.

Related work

While work on accelerating aspects of robotics such as simulators (Liang et al., 2018), visions systems, and neural network controllers (Pierson and Gashler, 2017) are abundant

the use of GPUs to accelerate a full evolutionary robotics setup is not explored. One step in this direction is in work by Makoviychuk et al. (2021) that demonstrates significant effect of putting both simulation and neural network controller on a GPU. However, compared to our work only one individual can be evaluated at a time. Another related piece of work is that of Ohkura et al. (2014) who implemented an evolutionary swarm robotics setup on a GPU. However, again here the individual in this case the swarm is evaluated sequentially.

Controlling the initial pose can be viewed as a simple way to control the difficulty of an evolutionary challenge. From this point of view the approach belongs to a small class of approaches that tries to make learning easier by ramping up complexity as the evolutionary process develops. The simplest is incremental evolution where the task itself is made more complex over evolutionary timescales (Gomez and Miikkulainen, 1997; Rossi and Eiben, 2014). Similarly, it is also possible to make the environment (Wang et al., 2019) or the robot's morphology (Bongard, 2011) change over evolutionary time. However, these approaches require encoding of the task or environment to be used by the evolutionary process and thus are complex, but arguably can also handle more complex, open-ended problems. In contrast, we only encode the pose which is simple and practical and well suited for problems where the task-environment is given.

An interesting development is that robots start to have GPUs on-board making it possible to potentially run evolutionary runs in real-time on on-board the robots themselves (Jones et al., 2015).

Parallel Implementation

We programmed our evolutionary robotics engine from scratch and based it on the Khepera IV robot (Soares et al., 2016) which has eight infrared sensors. For the environment we use a walled, square arena with varying side lengths. In the following we describe how elements of the evolutionary engine is assigned to threads on the GPU. The evaluation of an individual consists of running the following update loop:

Sensor update. The robot has eight sensors for each sensor a GPU thread is responsible for finding the nearest element of the environment (modelled as a polygon) and calculate the corresponding distance.

Controller update. The controller is a simple neural network which maps sensor inputs to motor outputs. A single thread is responsible for calculating this mapping.

Simulation update. A simple kinematic simulation of the differential drive robot is implemented which is calculated by one thread.

Collision update. A single thread checks for a collision.

This update loop is run in parallel for an entire generation of individuals meaning that for most update steps a number of threads equal to the number of individuals in a generation is used. The only exception is the sensor update which uses eight threads per individual. This means that for our 256 individuals to run in parallel we need 8 times 256 equals 2048 threads. This is easily achievable as even our Quadro T1000 laptop GPU has 896 cuda cores of 32 threads each which equals a total of 28672 threads. Hence, we have many more threads for further parallelism if desired. The parallel implementation allows us to update all our 256 individuals in parallel and thus cut computation time by 99.6%.

While not the focus here we have also implemented the evolutionary algorithm itself on the GPU. However, this is not critical as most of the computation time is spent in the update loop described above where the individuals are evaluated. The update loop runs for every $0.1s$ in our implementation which means that for a simulated time period of $200s$ the update loop is repeated 2000 times.

Experimental Setup

The evolutionary robotic engine follows that of Mondada and Floreano (1996). The controller consists of two perceptrons with eight inputs, two recurrent connections between the outputs and one bias unit. The corresponding chromosome consists of 22 real-valued genes encoding the weights of the neural network. For the initial configuration population we use a chromosome of 3 genes encoding position and orientation. The fitness function is summed for each time step and is calculated as:

$$fitness = V(1 - \sqrt{\Delta v})(1 - i) \quad (1)$$

Where V is the average velocity, Δv is the difference in velocities between left and right wheel, and finally i is the activity of the most active sensor (the sensor where an obstacle is closest). All parameters are normalised to be between 0 and 1.

The fitness function for the population of initial configurations is one minus the fitness of the controller. Hence, the system is a competitive co-evolutionary system.

For the evolutionary process we use a single point crossover with a probability of 0.6, a mutation with probability 0.02 which replaces a gene with a random number which for the weights is normalised to be between 8 and -8. We use an elitism of 8 and the rest of a generation is selected using tournament selection with a tournament size of 2. We use 256 individuals and run the process for 300 generations where each individual is evaluated. For the initial pose part, we use 128 individuals and an elitism of 4. The initial configuration is limited to be inside the arena and at least a robot diameter from the wall ($10cm$), which is just outside of sensor range which is $15cm$ from the centre of the robot. For the co-evolutionary experiment, we evolve the

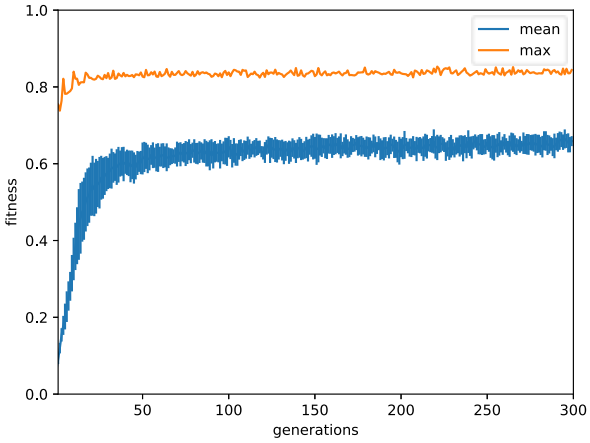


Figure 1: The fitness of the evolutionary runs based on random initial poses in the small environment. The figure shows the maximum fitness (top) and the average of ten evolutionary runs (bottom). The dark area corresponds to the standard deviation around the mean.

controller and the initial configuration alternately one generation at a time. For fitness evaluation the most fit individual from the other population is used. All code is implemented in CUDA (Nickolls et al., 2008) and all experiments are performed on a modest laptop with an Intel Core i7-10750H CPU running at 2.60GHz and a Quadro T1000 Graphical Processing Unit.

Results

Random Pose - Small Environment

The first experiment represents the base case and reproduces a result on evolving obstacle avoidance from literature. All individuals are evaluated in an arena with side length $1m$ and the evaluation period is $200s$. In Figure 1 the maximum and the average fitness of ten evolutionary runs can be seen. It shows that the algorithm consistently improves in the beginning and find a solution to the obstacle avoidance task. An example of a found solution can be seen in Figure 2 which shows that once a robot encounters the wall it turns sharply to get away from the wall. The experiment shows that in a small environment compared to the length of the evaluation period a standard evolutionary approach can find a solution and that our implementation works.

Random Pose - Timing

In the second experiment we measured the wall clock runtime of the evolutionary process as a function of the period each individual was evaluated. The result can be seen in Table 1. The Table shows that if each individual is evaluated for a simulated period of 400 seconds the evolutionary process takes 4.8 seconds. At the other end of the spectrum if each individual is evaluated for only 12 seconds of simulated time the evolutionary process takes 0.4 seconds. In other words, the GPU implementation allows us to run a full evolutionary run in 2.6 seconds for the run documented in the previous section where the evaluation period was 200 seconds. The table also documents that if we reduce the evaluation period needed to evaluate an individual, we have potential for further speed up.

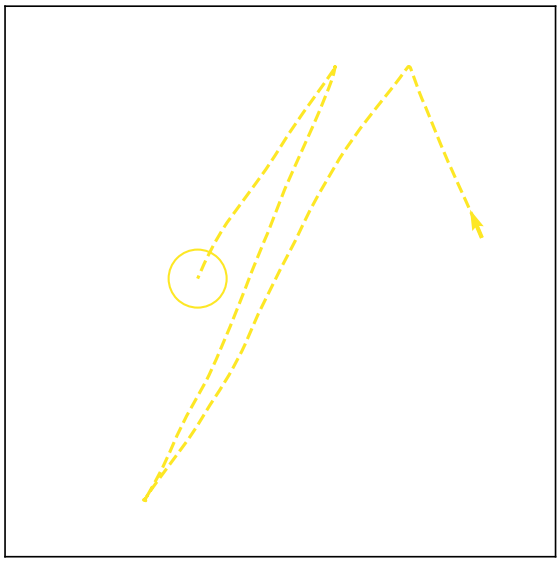


Figure 2: The robot arena (1m x 1m) and an example trajectory of the best individual of one of the evolutionary runs. The arrow represents the starting pose and the circle the final pose.

simulation time	12	25	50	100	200	400
run-time	0.4	0.5	0.8	1.4	2.6	4.8

Table 1: This table shows the run-time of the evolutionary process as a function of the evaluation time of individuals. Both are measured in seconds.

Random pose - Large Environment

We now rerun the first experiment with the only exception that we increase the size of the environment such that it has a side length of $8m$. However, while evolution strictly takes place in the larger environment, we additionally validate each individual in the small environment to make the absolute fitness values comparable to the first experiments. The validation fitness results are shown in Figure 3. The fitness graph shows that a solution cannot be found to the obstacle avoidance task in the large environment and inspection of the trajectory shows that the robot simply crashes

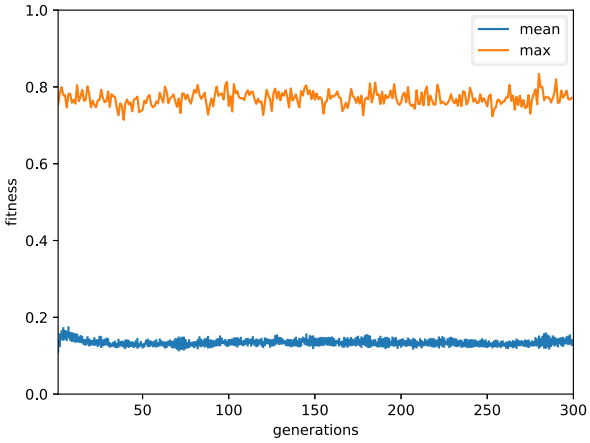


Figure 3: The fitness of the evolutionary runs based on random initial poses in the large environment The maximum fitness and the average with errors bars of ten standard evolutionary runs.

into the wall the first time it encounters it. The reason evolution fails is that a randomly placed individual in the large environment is unlikely to encounter an obstacle (the wall) and hence can score high fitness without having learned the correct behaviour. Furthermore, individuals which are randomly placed close to a wall will have a low fitness and be out-competed by individuals which were not placed close to a wall. Hence, the optimal solution from the evolutionary algorithms point of view is an individual that goes fast and straight.

Random Pose - Evaluation Period and Environment Size

In order to explore the potential for further speed-up, we now examine the sensitivity of the random pose-based evolutionary approach to arena size and evaluation period. For each set of parameters, we run the evolutionary approach ten times and count the number of times that the resulting controller can successfully avoid the wall in the validation environment (1m x 1m) for 200 seconds. If the robot within this time touches the wall the run is a failure. If not, it is a success. The results can be seen in Table 2. As the size of the arena increases (that is, the side length of the arena increases) the number of successful evolutionary runs decreases. Similarly, as the evaluation period of individuals decreases the number of successful evolutionary runs also decreases. In fact, we find that for large arena sizes and short evaluation periods evolution fails to find a solution. The only exception is a single successful run with an arena side length of 8m and an evaluation period of 50m. However, the other nine runs with these parameters fail hence we consider this

	1m	2m	4m	8m
12s	0	0	0	0
25s	0	0	0	0
50s	40	0	0	10
100s	100	90	0	0
200s	90	100	50	0
400s	70	100	100	50

Table 2: The percent of successful runs using the baseline evolutionary robotics setup as a function of arena side length and evaluation time of each individual.

	1m	2m	4m	8m
12s	30	20	30	50
25s	90	100	90	90
50s	70	100	90	80
100s	100	60	90	90
200s	100	90	80	100
400s	100	90	100	70

Table 3: The percent of successful runs using co-evolution of pose and controller as a function of arena side length and evaluation time of each individual.

an outlier. In general, the variation in the data stems from the random initialisation of weights and the random poses of individuals. Overall, the experiment shows that the baseline evolutionary approach does not converge on a solution for large arena sizes and short evaluation periods and thus we cannot obtain speed-up this way.

Co-evolved Pose - Evaluation Period and Environment Size

In the last set of experiments, we rerun the experiment from the previous section with the exception that we use the co-evolutionary approach where the initial pose is co-evolved with the controllers. In Table 3 we see the results of running the co-evolutionary setup using the same parameters as for the baseline setup. We can see that the number of successful runs is independent of arena size. This is because the co-evolutionary process finds initial poses close to the wall independent of the arena size. Hence, the performance of the co-evolutionary process is independent of arena size. If we look at the evaluation period, the process is also robust to changes of this again because the robot starts close to the wall and comparative fitness can quickly be established. First when the evaluation period becomes 12 seconds the process fails. This is because with the short evaluation time the robots do not have time to move away from the wall and obtain a higher fitness compared to an individual that does not move at all. In fact, from observing the trajectories we find that most of the individuals that succeeds in these runs tend to stay in one spot and hence does solve the task, but are low fitness scoring individuals.

Generally, we find that for the baseline evolutionary setup solutions can reliably be found if the arena size is less than 2m and the evaluation period is higher than 100s. While for the co-evolutionary setup we can find solutions for any arena size if the evaluation period is 25s or longer. Hence, in conclusion we find that the co-evolutionary process can find a solution to the obstacle task independent of arena size. Similarly, we find that we can reduce the evaluation periodic drastically without influencing the performance. This leads to much lower computational costs and thus faster run times of the evolutionary optimisation process.

Discussion

While the two efforts in this paper are directed at making evolutionary robotics more computationally tractable, it is worth to note that co-evolution of course is more computationally costly than a standard evolutionary setup because we both must do fitness evaluation of the controller and the pose. However, our results show that even though the algorithm is more time consuming the result is still that it overall is faster.

A short-coming of the work is that we have not considered the reality gap (Jakobi et al., 1995). However, we do feel on safe ground as related work has shown that for simple systems such as the one demonstrated here it is possible to overcome the reality gap.

An important question is if our approach generalises to more complex tasks. For the GPU acceleration we think this is the case because many simulators already take advantage of GPU acceleration. Hence, all that is needed is to allow several simulators to run in parallel on the same GPU. While this in theory is simple, the practical challenge may be significant. However, future simulator developers may well bear in mind that many simulations should be able to run in parallel.

The other generalisation question relates to whether the idea of co-evolution of initial pose and controller scales to more complex tasks or not. This is less clear. The challenge is that in many cases it is easy to come up with poses or in general configurations that makes solving a task for a controller impossible. E.g., for a walking task a specifically unfortunate configuration would be one where the robot has lost its balance and therefore is unrecoverable no matter what actions the controller decides to take. For these tasks, the configurations the co-evolutionary system should have access to, should likely be reduced which requires some hand-tuning.

Conclusion

In this paper we introduced the idea of fully parallelising an evolutionary robotics engine making it possible to run it on a GPU. This immediately led to drastic increase in performance. Specifically, we found for our relatively simple evolutionary setup that we could run a full evolutionary run in less than 2.6 seconds on a modest laptop. We used this engine to explore if co-evolution of robot pose and controller could reduce evaluation period and thereby further increase performance. We found this to be the case and was able to reduce the evaluation period to 25s for all arena sizes. Overall, we find the approaches to be promising and we think it is a timely topic for further research to understand how to reduce computational costs of evolutionary robotics setups.

References

Bongard, J. C. (2011). Morphological and environmental scaffolding synergize when evolving robot controllers: Artificial life/robotics/evolvable hardware. In *Proc. of the 13th Annual Conf. on Genetic and Evolutionary Computation*, page 179–186, New York, NY, USA. ACM.

Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. E. G. (2015). Evolutionary robotics: What, why, and where to. *Frontiers in Robotics and AI*, 2:4.

Gomez, F. and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, 5(3-4):317–342.

Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. *Lecture Notes in Computer Science*, 929:704–720.

Jones, S., Studley, M., and Winfield, A. (2015). Mobile GPGPU acceleration of embodied robot simulation. In *Artificial Life and Intelligent Agents*, pages 97–109, Cham. Springer International Publishing.

Liang, J., Makoviychuk, V., Handa, A., Chentanez, N., Macklin, M., and Fox, D. (2018). GPU-accelerated robotic simulation for distributed reinforcement learning. In *Proc. of The 2nd Conf. on Robot Learning, PMLR*, volume 87, pages 270–282.

Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., and State, G. (2021). Isaac gym: High performance gpu-based physics simulation for robot learning.

Mondada, F. and Floreano, D. (1996). Evolution and mobile autonomous robotics. *Towards Evolvable Hardware. Lecture Notes in Computer Science*, pages 221–249.

Nickolls, J., Buck, I., Garland, M., and Skadron, K. (2008). Scalable parallel programming with CUDA: Is CUDA the parallel programming model that application developers have been waiting for? *Queue*, 6(2):40–53.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press.

Ohkura, K., Yasuda, T., Matsumura, Y., and Kadota, M. (2014). GPU implementation of food-foraging problem for evolutionary swarm robotics systems. In *Swarm Intelligence*, pages 238–245, Cham. Springer International Publishing.

Pierson, H. A. and Gashler, M. S. (2017). Deep learning in robotics: a review of recent research. *Advanced Robotics*, 31(16):821–835.

Rossi, C. and Eiben, A. (2014). Simultaneous versus incremental learning of multiple skills by modular robots. *Evol. Intel.*, 7:119–131.

Rudin, N., Hoeller, D., Reist, P., and Hutter, M. (2021). Learning to walk in minutes using massively parallel deep reinforcement learning. *CoRR*, abs/2109.11978.

Soares, J. M., Navarro, I., and Martinoli, A. (2016). The Khepera IV mobile robot: Performance evaluation, sensory data and software toolbox. In *Robot 2015: 2nd Iberian Robotics Conf.*, pages 767–781. Springer.

Stoy, K. (2021). Co-evolution of Initial Configuration and Control in Evolutionary Robotics. In *ALIFE 2021: The 2021 Conference on Artificial Life*. 70.

Wang, R., Lehman, J., Clune, J., and Stanley, K. O. (2019). Paired open-ended trailblazer (POET): Endlessly generating increasingly complex and diverse learning environments and their solutions.

Heterogeneity and Robustness in Social Learning

Jonathan Lawry

Department of Engineering Mathematics,
University of Bristol,
Bristol, BS8 1TW, UK
Email: j.lawry@bristol.ac.uk

Abstract

Social learning is an important collective behaviour in many biological and artificial systems. We investigate a model of social learning which combines two distinct processes, one relating to how individuals adapt their beliefs as a result of interacting with their peers, and one relating to when they search for and how they learn directly from evidence. For each process we introduce conservative and open-minded behaviours and combine these to obtain four social learning behaviour types. A simple truth-seeking task is considered and a three-valued model of belief states is adopted. By means of difference equation models and agent-based simulations we then investigate the performance of the different learning behaviours. We show that certain heterogeneous mixtures of behaviours result in the most robust performance for a variety of learning rates and initial conditions, and that such mixtures are well suited for social learning in dynamic environments.

Introduction & Background

Social learning is fundamental to collective decision making in many biological systems and on social networks, as well as being crucial for applications in swarm robotics and multi-agent AI. In contrast to individual learning in which agents operate alone when collecting evidence and making inferences, in social learning individuals are also influenced by observation of, or interaction with, others in a socially connected population (Heyes, 1994). This means that population level consensus or polarisation is ultimately an emergent effect of individual learning that depends on learning behaviour as well as the nature of the problem, the level and type of connectivity between agents and the amount and quality of evidence available.

In applications such as swarm robotics it is common to consider homogeneous populations of simple robots, all cooperating to reach a common goal or to solve a shared problem (Brambilla, 2013). However, in natural systems some form of heterogeneity is common within social groups, resulting in different behaviours and divisions of labour. In this paper we focus on the effect of heterogeneous learning behaviours on the efficacy of social learning. More specifically, we will consider different approaches to learning which vary in terms of how conservative or open-minded they are in their treatment of evidence and also in they way that individuals learn from their peers. By defining difference equation models and through multi-agent simulations we will demonstrate that a heterogeneous mixture of learning behaviours can enhance both accuracy and robustness in a simple type of social learning task.

The effect of heterogeneity on emergent behaviour in complex systems has been studied across a number of disciplines. For example, dating back to (Smith, 1776) heterogeneity in the division of labour has been seen as a key feature of market economies. Similar divisions of labour have also been observed and modelled in biology, particularly in social insects (Buckingham, 1911; O'Shea-Wheller et al, 2020). In this context threshold-response models have been used to explain how a certain form of individual heterogeneity enables social insect societies to regulate their environmental conditions (Theraulaz et al, 1998). These models assume that each individual responds to a given stimulus when its stimulus intensity exceeds their threshold, and that thresholds vary between individuals in the population. For instance, threshold-response models have been applied to collective adaptive ventilation behaviour in European honey bees, where individual honey bees exhibit fanning behaviour when the local temperature exceeds their individual threshold. Heterogeneous thresholds then enable a graded overall response by the hive to external temperature changes (Peters et al, 2019). Other types of behavioural heterogeneity are also present in social insect species. This includes varying levels of boldness in social spiders such as *Stegodyphus Dumicol* (Hunt et al, 2019), which impacts on overall attack speed.

Emergent behaviour resulting in consensus or polarisation on social networks of interacting heterogeneous agents has been investigated using opinion dynamics models (Baronchelli, 2018). For example, thresholds can be used as part of the continuous belief Hegselmann-Krause model to capture different behaviour types where agents are more or less open-minded. More specifically, agents are influenced by those neighbours on their social network with beliefs sufficiently similar to their own. Here similarity is quantified

as a numerical measure between beliefs and two beliefs are deemed to be sufficiently similar if the degree of similarity between them exceeds a given threshold. In heterogeneous models, each agent makes these judgements based on their own threshold, and agents with lower thresholds can be considered as being more open minded. Opinion dynamics models aim to capture how agents on a social network influence each other's opinion over time, and hence they do not typically take account of the effect of evidence received directly from the environment. In contrast, in social learning individual agents must take account of both direct evidence and the beliefs of the other agents with whom they interact. This form of learning is important for collective decision making in biological systems such as social insects (Valentini et al, 2017), in swarm robotics and arguably in human societies e.g. as part of the scientific method (Douven & Kelp, 2011).

There is no reason to assume that for social learning homogeneous populations consisting of only one type of learning behaviour will always be optimal. For example, (Yaman et al., 2022) present simulation experiments for a simple multi-armed bandit problem indicating that best performance is achieved with a heterogeneous population consisting of independent (non-social) learners and two types of social learners, one type favouring learning from the most successful agents and the other tending to adopt the majority opinion. In this paper we argue that a heterogeneous mixture of behaviours can improve the robustness of social learning to varying initial conditions, and different rates of evidence and interactions between agents. It also enables social learning to be more adaptive in dynamic environments where the underlying state-of-the-world changes during the learning process (Prasetyo et al, 2019). We formulate learning in terms of two distinct but interacting processes; **evidential updating** according to which agents update their current beliefs based on direct evidence and **fusion** by which agents combine their beliefs with others with whom they are interacting. We describe two types of behaviour for each process resulting in four learning behaviours which agents can adopt in a social learning context.

Three-Valued Social Learning

We consider a simple social learning problem in which a population of agents attempt to determine the truth or falsity of a proposition relating to the environment in which they are operating. For instance, this could refer to the presence or absence of resources, e.g. food or shelter, in a certain location, or whether or not a particular option or choice is the best. Alternatively, in the context of swarm robotics applications such as search and rescue it could refer to the positions of casualties in a specified search area. A three-valued model of belief states is adopted where, at any time, an agent holds one of three beliefs; $\mathbf{f}$ indicating that they believe the proposition to be false, $\mathbf{u}$ indicating that they are uncertain or uncommitted, or $\mathbf{t}$ indicating that they believe the proposition is true. This approach has been shown to be an effective framework in social learning, which is robust to environmental noise and scalable to multiple propositions (Crosscombe & Lawry, 2017). Furthermore, for the best-of-n problem in swarm robotics the inclusion of the intermediate uncertain belief state $\mathbf{u}$ has been shown to improve robustness to the presence of malfunctioning robots in the swarm (Crosscombe et al, 2017). In addition, there is evidence that for some biological systems the presence of uncommitted individuals can help to facilitate population level consensus in decision making (Couzin et al, 2011).

In the following we propose different fusion and updating behaviours in the three-valued setting and formulate difference equation models for a totally collected well-mixed population in each case (Parker & Zhang, 2009). We consider time-stepped models where in each time-step agents attempt to undertake fusion followed by evidential updating. Difference equation models can then be formulated in terms of the proportion of agents in a large population holding each belief. More formally, we let:

$$\mathbf{P}_t = (P_t(\mathbf{f}), P_t(\mathbf{u}), P_t(\mathbf{t}))$$

denote the proportions of the three belief states in the agent population at time t. Macro-level learning can then be modelled as a difference equation of the form:

$$\mathbf{P}_{t\ 1}^{T} = U(F^{\mathbf{P}_t}\mathbf{P}_t^T) = (UF^{\mathbf{P}_t})\mathbf{P}_t^T$$

where U and $F^{\mathbf{P}}$ are matrices of transition probabilities for the updating and fusion processes respectively. These are 3×3 matrices of the form:

$$\begin{pmatrix} P(\mathbf{f}|\mathbf{f}) & P(\mathbf{f}|\mathbf{u}) & P(\mathbf{f}|\mathbf{t}) \\ P(\mathbf{u}|\mathbf{f}) & P(\mathbf{u}|\mathbf{u}) & P(\mathbf{u}|\mathbf{t}) \\ P(\mathbf{t}|\mathbf{f}) & P(\mathbf{t}|\mathbf{u}) & P(\mathbf{t}|\mathbf{t}) \end{pmatrix}$$

where, for example, $P(\mathbf{u}|\mathbf{f})$ denotes the probability that an agent currently in belief state $\mathbf{f}$ will transition to $\mathbf{u}$. In general, the transition probabilities for fusion are dependent on the current proportions of belief states in the population since they need to take account of how likely it is that a given agent will interact with another agent holding any of the different belief states. We denote this by the superscript in $F^{\mathbf{P}}$. On the other hand, for evidential updating, transition probabilities are independent of current belief state proportions.

Behaviour Types

Fusion Behaviours

We propose a simple three-valued model of a process whereby agents look to interact with other agents so as to be informed of and learn from their opinions. Specifically, within a time step each agent, acting as a receiving agent, attempts to interact with one other randomly selected agent,

	f	**u**	**t**
f	**f**	**f**	**u**
u	**f**	**u**	**t**
t	**u**	**t**	**t**

Table 1: Table for the adventurous operator. Each cell is the updated belief for a given current belief (row) and received belief (column)

	f	**u**	**t**
f	**f**	**u**	**u**
u	**u**	**u**	**u**
t	**u**	**u**	**t**

Table 2: Table for the cautious operator. Each cell is the updated belief for a given current belief (row) and received belief (column)

acting as a transmitting agent. Such an interaction occurs with probability σ (the fusion rate), quantifying limitations on communication within the population. All agents are able to adopt both receiving and transmitting roles. If no interaction takes place then the agent maintains their current belief as a default. As a result of such an interaction the receiving agent adapts their current belief state based on the belief state of the transmitting agent, according to a set of simple rules which can be represented in the form of a truth table. Here we propose two types of fusion behaviour, one more adventurous where agents are willing to adopt strong opinions expressed by their peers, and one more cautious in which agents are led to doubt their own opinions when others express uncertainty.

In **adventurous fusion** *certainty dominates over uncertainty* so that if the receiving agent with belief state $\mathbf{u}$ interacts with a transmitting agent with committed belief states $\mathbf{t}$ or $\mathbf{f}$, they adopt the latter according to table 1. Otherwise, inconsistency between the transmitting and receiving agents' truth values, i.e. $\mathbf{t}$ vs $\mathbf{f}$ or vice versa, results in the receiving agent changing their belief to uncertain. In this way strong disagreement between the transmitting and receiving agents results in the latter doubting their current belief. Consequently, we have the following transition probabilities for adventurous fusion:

$$F_A^{\mathbf{P}} = \begin{pmatrix} 1-\sigma P(\mathbf{t}) & \sigma P(\mathbf{f}) & 0 \\ \sigma P(\mathbf{t}) & 1-\sigma(P(\mathbf{t})+P(\mathbf{f})) & \sigma P(\mathbf{f}) \\ 0 & \sigma P(\mathbf{t}) & 1-\sigma P(\mathbf{f}) \end{pmatrix}$$

For illustrative purposes we derive one of these transition probabilities as follows: An agent currently in belief state $\mathbf{t}$ can remain in that state after adventurous fusion in one of two ways. They can fail to interact with any other agent and hence maintain their current belief as a default. This will occur with probability $1-\sigma$. Alternatively, they can succeed in interacting with another agent who either also has belief $\mathbf{t}$ or has belief $\mathbf{u}$ (see table 1). This has probability $\sigma(P(\mathbf{t})+P(\mathbf{u})) = \sigma(1-P(\mathbf{f}))$. Hence,

$$P(\mathbf{t}|\mathbf{t}) = 1-\sigma+\sigma(1-P(\mathbf{f})) = 1-\sigma P(\mathbf{f})$$

In **cautious fusion** *uncertainty dominates over certainty* so that if a receiving agent with committed belief state $\mathbf{t}$ or $\mathbf{f}$ interacts with a transmitting agent with uncertain truth state $\mathbf{u}$ they will abandon their committed position and change their belief to $\mathbf{u}$ according to table 2. Consequently, we have the following transition probabilities for cautious fusion:

$$F_C^{\mathbf{P}} = \begin{pmatrix} 1-\sigma+\sigma P(\mathbf{f}) & 0 & 0 \\ \sigma(1-P(\mathbf{f})) & 1 & \sigma(1-P(\mathbf{t})) \\ 0 & 0 & 1-\sigma+\sigma P(\mathbf{t}) \end{pmatrix}$$

Evidential Updating Behaviours

In this model, evidence corresponds to a statement of the truth value of the relevant proposition, either $\mathbf{t}$ or $\mathbf{f}$, resulting from a (hypothetical) investigation of the environment by the agent concerned, and influenced by the noise parameter ϵ. Without loss of generality, we assume here and throughout unless stated otherwise, that the correct truth value for the proposition is $\mathbf{t}$ and evidence will report this with probability $1-\epsilon$ while the erroneous truth value $\mathbf{f}$ will be reported with probability ϵ. On receiving evidence of this form an agent will update their current belief depending on one of two behaviour types. This updating only takes place if they find evidence once they decide to look, and this occurs with probability ρ (the evidence rate). If no evidence is found then the agent maintains their current belief as a default.

In **confident updating**, once agents are committed to belief state $\mathbf{t}$ or $\mathbf{f}$ they are sufficiently confident in their opinion that they cease looking for evidence. Therefore, since agents only look for evidence if they are in the uncertain state $\mathbf{u}$, evidential updating simply involves them changing their belief state to whichever is asserted by the evidence. This means that transition probabilities for the evidential updating process are constrained such that $P(\mathbf{t}|\mathbf{t}) = P(\mathbf{f}|\mathbf{f}) = 1$. See the flow diagram in figure 1a. The full matrix of transition probabilities for confident updating is given by:

$$U_C = \begin{pmatrix} 1 & \rho\epsilon & 0 \\ 0 & 1-\rho & 0 \\ 0 & \rho(1-\epsilon) & 1 \end{pmatrix}$$

According to **inquisitive updating** agents are always sufficiently curious to look for evidence no matter what is their current belief state. This means that evidence may be inconsistent with an agent's current belief i.e. $\mathbf{f}$ given $\mathbf{t}$ or vice versa. In this case the agent reverts to truth state $\mathbf{u}$. Otherwise, the agent simply adopts the truth state asserted by the evidence. See the flow diagram in figure 1b. This results in the following matrix of transition probabilities.

$$U_I = \begin{pmatrix} 1-\rho+\rho\epsilon & \rho\epsilon & 0 \\ \rho(1-\epsilon) & 1-\rho & \rho\epsilon \\ 0 & \rho(1-\epsilon) & 1-\rho\epsilon \end{pmatrix}$$

Again for illustration we derive one of the above transition probabilities. There are two ways that an agent currently in belief state $\mathbf{f}$ can remain in that state after inquisitive updating. They can fail to find evidence and this has probability $1-\rho$. Alternatively, since we are assuming that the proposition is actually $\mathbf{t}$, they can find erroneous evidence. This has probability $\rho\epsilon$. Hence,

$$P(\mathbf{f}|\mathbf{f}) = 1 - \rho + \rho\epsilon$$

From a certain perspective it is possible to view inquisitive and confident updating as special kinds of explore and exploit strategies respectively. Confident updating preserves the committed belief states of $\mathbf{t}$ and $\mathbf{f}$, and this can be exploited during fusion (especially adventurous fusion) to drive consensus across the population. On the other hand, inquisitive updating encourages exploration in the form of a constant search for evidence by all agents. In general, identifying a good trade-off between explore and exploit behaviours has been found to be beneficial in a variety of learning contexts including reinforcement learning (Schäfer et al, 2021), multi-armed bandits (Slivkins, 2019), and evolutionary algorithms (Črepinšek et al, 2013).

If we now consider social learning where agents learn both through fusion and evidential updating, then taking the conjunction of the behaviour types described above for both of these processes, naturally results in four behaviour types for the combined learning process; **adventurous & confident (AC)**, **cautious & confident (CC)**, **adventurous & inquisitive (AI)** and **cautious & inquisitive (CI)**. In the following section we investigate the learning efficacy of homogeneous populations comprising solely of agents of each of these four behaviour types.

Properties of Behavioural Types

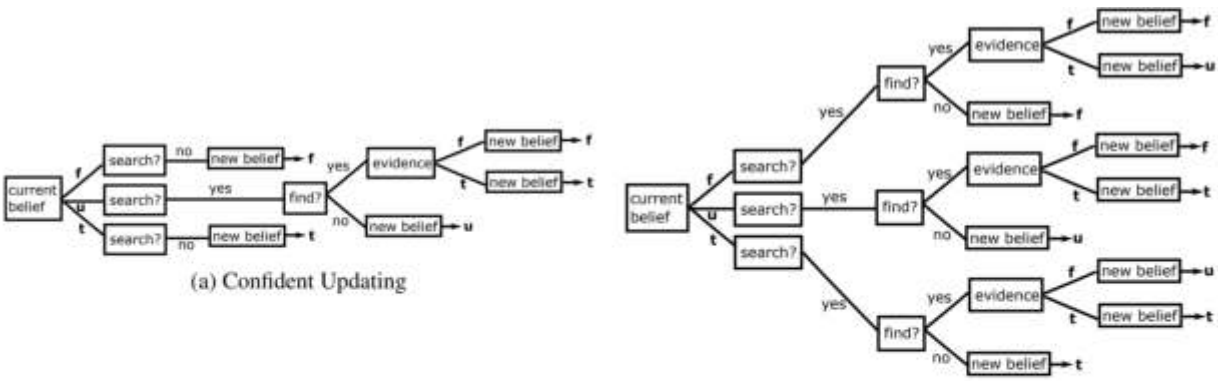
In this section we investigate the learning performance of the four behaviour types introduced above in a *homogeneous* setting, under significant noise $\epsilon = 0.3$, for varying fusion (σ) and evidence rates (ρ), and for different initial proportions of beliefs in the population. Figures 3 and 4 show heat maps of $P(\mathbf{t})$, i.e. the proportion of agents who have learnt the correct truth value, at $t = 1500$; evidence rates ρ and fusion rates σ vary in steps of 0.01 across the interval $(0, 1)$. Figure 3 is for a population of agents with initial beliefs such that 90% of agents begin the learning process committed to the incorrect belief (i.e., $\mathbf{f}$) and 10% to the correct belief i.e. $\mathbf{P}_0 = (0.9, 0, 0.1)$. This is clearly a challenging configuration of agent beliefs from which to initialise learning but we will argue that the capacity to learn effectively starting from this type of initial condition can be important when faced with dynamic environments where the underlying true state-of-the-world can change suddenly. Figure 4 assumes that all agents begin the learning process in the uncertain state $\mathbf{u}$ i.e. $\mathbf{P}_0 = (0, 1, 0)$. This scenario is particularly relevant for some applications of collective learning in robotics and autonomous systems when initial settings can be directly controlled.

For initial proportions $\mathbf{P}_0 = (0, 1, 0)$ it is clear from figure 4a that the best performance is obtained from a population consisting entirely of **AC** agents. Unsurprisingly, performance is better for initialisation $(0, 1, 0)$ than for $(0.9, 0, 0.1)$ for all four behaviour types, although the difference is small for cautious fusion agents (i.e. **CC** and **CI**). In general with the exception of **AI** initialised at $(0, 1, 0)$, all agents perform best when the evidence rate is relatively high compared to the fusion rate. Furthermore, although there are differences between the behaviour types regarding which combinations of σ and ρ give good performance, there are some interesting similarities, especially between **AC** and **CC**. In particular, figures 3a, 3b and 4b all show clear regions of good vs poor performance bounded by what seems to be the same functional relationship between ρ and σ. We can gain more insight into this by considering the stability of certain equilibrium points of the difference equation for these two behaviour types.

Note that both $(1, 0, 0)$ and $(0, 0, 1)$ are equilibrium (fixed) points of the **AC** and **CC** difference equations. Since these represent the situations in which the whole population of agents reach consensus about the incorrect and correct state-of-the-world respectively, it is therefore insightful to consider the stability of these equilibrium points under different noise conditions, fusion and evidence rates. This can be done by determining the Jacobian matrix for both difference equation models, evaluating it at the two fixed points, and considering the absolute values of the respective eigenvalues. If these values are all strictly less than 1 then the equilibrium point is stable, while it is unstable if any are strictly greater than 1. Accordingly, we find that for the **AC** and **CC** behaviour types there is a natural boundary dividing the (σ, ρ) parameter space into different stability regions given by:

$$\rho = \frac{\sigma}{1 + \sigma - 2\epsilon} = f(\sigma; \epsilon)$$

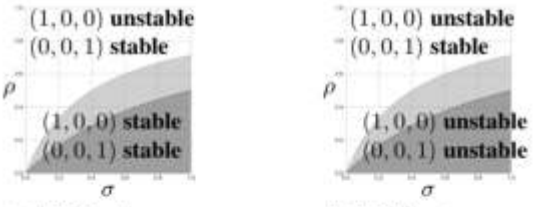
This then corresponds to the boundary for the regions observed in figures 3a, 3b and 4b. However, the exact nature of this division is different for **AC** and **CC**. For **AC**, $(0, 0, 1)$ is stable for all $\sigma, \rho \in (0, 1)$ and $\epsilon \in [0, 0.5)$, but $(1, 0, 0)$ is stable if $\rho < f(\sigma; \epsilon)$ and unstable if $\rho > f(\sigma; \epsilon)$ (see figure 2a). In contrast, for **CC** $(1, 0, 0)$ is unstable for all $\sigma, \rho \in (0, 1)$ and $\epsilon \in [0, 0.5)$, but $(0, 0, 1)$ is unstable if $\rho < f(\sigma; \epsilon)$ and stable if $\rho > f(\sigma; \epsilon)$ (see figure 2b). Hence, for both behaviour types with combinations of fusion and evidence rates such that $\rho > f(\sigma; \epsilon)$ we have that the incorrect consensus is an unstable equilibrium while the correct consensus is a stable equilibrium, and this helps to explain the good performance for both **AC** and **CC** in this region of the heat maps shown in figures 3a, 3b, 4a and 4b. For



(a) Confident Updating

(b) Inquisitive Updating

Figure 1: Flow diagram showing the evidential updating process for the confident and inquisitive behaviours.



(a) **AC** behaviour type (b) **CC** behaviour type

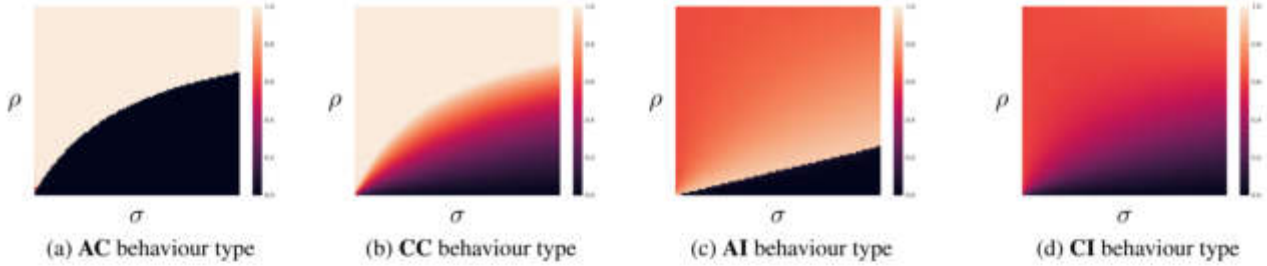
Figure 2: Stability of equilibrium points $(1, 0, 0)$ and $(0, 0, 1)$ with different combinations of fusion rate (σ) and evidence rates (ρ), for adventurous & confident and cautious & confident learning behaviour types. For both types there are two stability regions partitioned by $\rho = \frac{\sigma}{1+\sigma-2\epsilon}$, shown here for the noise values of $\epsilon = 0.3$ (light grey) and $\epsilon = 0$ (dark grey).

CC with combinations of fusion and evidence rates such that $\rho < f(\sigma; \epsilon)$ the instability of both equilibria seems to result in more mixed performance in this region of parameter space, especially for (σ, ρ) close to the boundary (see figures 3b,4b). In contrast, the stability of both equilibria in this region for **AC**, tends to result in convergence to $(1, 0, 0)$ (figure 3a) for initial condition close to that equilibrium point and to $(0, 0, 1)$ otherwise (figure 4a). Furthermore, notice that $f(0; \epsilon) = 0$ for all ϵ, and $f(\bullet; \epsilon) \leqslant f(\bullet, \epsilon')$ for $0 \leqslant \epsilon' \leqslant \epsilon < 0.5$. This monotonicity property implies that the area of the upper region of σ, ρ parameter space in which only the correct consensus is a stable equilibrium and which therefore tends to be associated with good performance, decreases as the noise increases, while the area of the lower region associated with mixed or poor performance increases. For example, contrast the light and dark grey regions in figure 2 corresponding to $\epsilon = 0.3$ and $\epsilon = 0$ respectively.

Space of Learning Behaviours

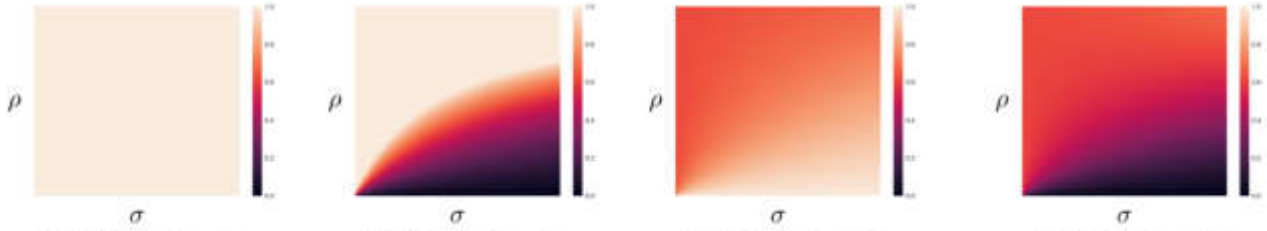
In the previous section we considered the collective learning performance of homogeneous populations consisting of each of the four main behaviour types. In this section we consider heterogeneity of behaviour types within a single population of agents. Initially, we fix the updating behaviour to be confident and consider a mixed population of **AC** agents, proportion w, and **CC** agents, proportion $1 - w$, for $w \in [0, 1]$. Figure 5 shows $P(\mathbf{t})$ at $t = 1500$ for initial condition $(0.9, 0, 0.1)$ for varying w and for different fusion and evidence rates. For the cases shown a $50/50$ mixture (i.e. $w = 0.5$) of **AC** and **CC** behaviours results in optimal performance. Indeed, the heat maps of $P(\mathbf{t})$ across σ and ρ shown in figure 6 indicate that this mixture of behaviour types performs consistently well for all combinations of fusion and evidence rate and for both initial proportions $(0.9, 0, 0.1)$ and $(0, 1, 0)$. The stability of the equilibrium points $(1, 0, 0)$ and $(0, 0, 1)$ also shed some light on why heterogeneous behaviour of this form is so effective, since for $w = 0.5$, $(0, 0, 1)$ is stable and $(1, 0, 0)$ is unstable for all $\sigma, \rho \in (0, 1)$ and $\epsilon \in [0, 0.5)$. However, for any mixture of only **AC** and **CC** behaviours the incorrect state-of-the-world $(1, 0, 0)$ is an equilibrium point and hence learning cannot take place for a population which begin with this incorrect consensus. In such cases, we hypothesise that including some inquisitive learners in the population may be helpful.

We now suppose that a proportion λ of the agent population are inquisitive regarding evidence while the remainder are confidence. Assuming that the behaviours for fusion and for evidential updating are allocated independently then this results in a mixture of the four behaviour types with proportions as given in table 3. Note that the allocation of behaviour types to agents can either be made at the beginning of the simulation and then fixed, or reallocated at every time step. For the latter we can think of agents independently



(a) **AC** behaviour type (b) **CC** behaviour type (c) **AI** behaviour type (d) **CI** behaviour type

Figure 3: Heat maps showing the proportion of correct beliefs with different combinations of fusion rate (σ) and evidence rates (ρ), for the four behaviour types initialised at $(0.9, 0, 0.1)$ i.e. with 90% of agents initially believing the wrong answer. Results are at $t = 1500$ and $\epsilon = 0.3$.



(a) **AC** behaviour type (b) **CC** behaviour type (c) **AI** behaviour type (d) **CI** behaviour type

Figure 4: Heat maps showing the proportion of correct beliefs with different combinations of fusion rate (σ) and evidence rates (ρ), for the four behaviour types initialised at $(0, 1, 0)$ i.e. with all agents initially uncertain. Results are at $t = 1500$ and $\epsilon = 0.3$.

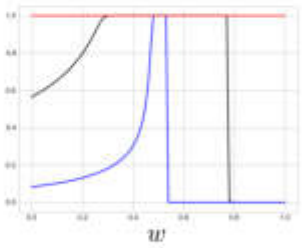
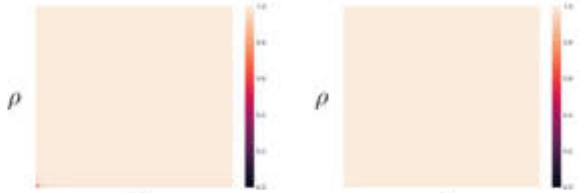


Figure 5: Proportion of correct beliefs for different proportions of **AC** (w) and **CC** ($1 - w$) behaviour types. Black line for $\sigma = 0.1$, $\rho = 0.1$, blue line for $\sigma = 0.9$, $\rho = 0.1$, green line for $\sigma = 0.1$, $\rho = 0.9$, and red line for $\sigma = 0.9$, $\rho = 0.9$. Results are for $\epsilon = 0.3$ at $t = 1500$ and for initial proportions $(0.9, 0, 0.1)$.



(a) Initialisation $(0.9, 0, 0.1)$. (b) Initialisation $(0, 1, 0)$.

Figure 6: Heat maps showing the proportion of correct beliefs with different combinations of fusion rate (σ) and evidence rates (ρ), for a $50/50$ mixture of **AC** and **CC**. Results are at $t = 1500$ and $\epsilon = 0.3$.

choosing a behaviour type at random according to the probabilities given in table 3 at every time step. Since we are assuming full mixing of totally connected agents the difference equation model does not distinguish between the two variations, but it will make a different to the implementation of the agent-based simulation in the following section.

Investigating performance across the space of parameter values for $(1, 0, 0)$ (figure 7) can now provide insight into which heterogeneous populations are robust to learning under the most challenging initial conditions. Figure 8 shows heat maps of $P(\mathbf{t})$ at $t = 1500$ for $\epsilon = 0.3$ and initial proportion $(1, 0, 0)$ for varying λ and w, and for different combinations of fusion and evidence rates. Taken together these suggest that good performance can be obtained for a low but non-zero proportion of inquisitive agents, e.g. $\lambda = 0.01$, and a $50/50$ split between adventurous and cautious agents, i.e. $w = 0.5$. In the next section, we show that this combination of agent behaviours is able to adapt to learning in a dynamic

	Adventurous w	Cautious $1-w$
Confident $1-\lambda$	$(1-\lambda)w$ **AC**	$(1-\lambda)(1-w)$ **CC**
Inquisitive λ	λw **AI**	$\lambda(1-w)$ **CI**

Table 3: Proportions of behaviour types in the population

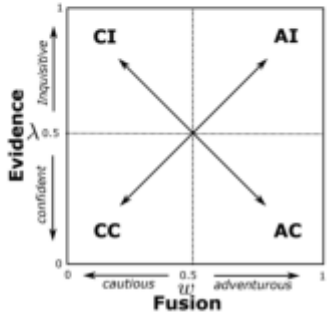


Figure 7: Space of parameter values (w, λ) representing different proportions of learning behaviour types.

environment in which the true state-of-the-world suddenly changes.

Agent-Based Model of Dynamic Environments

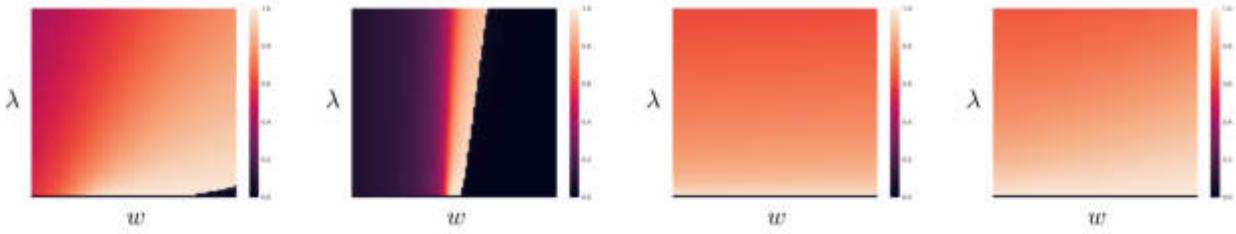
It is common for social learning to take place over a timescale during which the state-of-world may change (Prasetyo et al, 2019). Here we consider a simple scenario in which the true state-of-the-world changes suddenly from $\mathbf{f}$ to $\mathbf{t}$ at time step $t = 500$. This scenario is investigated using an agent-based simulation rather than difference equation model of the type discussed above. A population of 200 agents are all initialised as being uncertain, and at each time step each agent randomly selects a behaviour type according to probabilities given in table 3. Furthermore, at each time step each agent fuses their current truth-value with probability σ and no fusion takes place for that agent with probability $1-\sigma$. In the case that fusion occurs another agent is selected at random to act as the transmitting agent. Also, at each time step each agent decides whether or not to look for evidence according to their current behaviour type (see figure 1). If an agent searches then they receive evidence with probability ρ. For all combinations of parameter settings the results are averaged over 100 independent runs, each running for 1500 time-steps. Error bars then correspond to 90% percentiles for the data obtained from the independent runs.

Figure 9 shows the proportion of agents with truth-values $\mathbf{t}$ (black line), $\mathbf{u}$ (green line) and $\mathbf{f}$ (blue line) plotted against time, for three mixtures of behaviour types. The results are for $\sigma = 0.9$ and $\rho = 0.1$ which is a challenging scenario when the environment is dynamic since direct evidence is relatively scarce compared to the frequency of agent interactions, making it more difficult for the population to detect an underlying change in the state of the world. Figure 9a is for a homogeneous population consisting only of **AC** agents. In this case, the agents quickly reach consensus on what is initially the true state-of-the-world, i.e. $\mathbf{f}$, but given the subsequent dominance of this truth value in the population, agents are then unable to learn the new state of the world after the change at $t = 500$. There is similar behaviour shown in figure 9b which is for a $50/50$ mixture of **AC** and **CC** behaviours. In this case it takes the population slightly longer to reach consensus on the initial correct state $\mathbf{f}$, but again agents are unable to revise their truth-values at $t = 500$. We hypothesise that in both figures the population reaches consensus close to the proportions $(1, 0, 0)$ in the time period up to $t = 500$. This is an equilibrium point for both mixtures of behaviour types making subsequent updating impossible. In contrast, figure 9c show results for $\lambda = 0.01$ and $w = 0.5$ where there is a $50/50$ split between adventurous and cautious fusion behaviours, and where there is a small but non-zero proportion of inquisitive evidential updating behaviour in a population otherwise dominated by the confident behaviour type. In this case, we see that the population initially reaches consensus on $\mathbf{f}$, but at $t = 500$ we see a rise in the proportion of $\mathbf{u}$ agents followed by a gradual increase in the proportion of $\mathbf{t}$ agents until consensus is reached on the new state-of-the-world by $t = 1500$. In the first phase of learning for $t < 500$ figures 9c and 9b show almost identical results, but the small proportion of inquisitive agents that continue to collect evidence throughout are sufficient to allow the population to adapt when the state-of-the-world changes.

Conclusions & Future Work

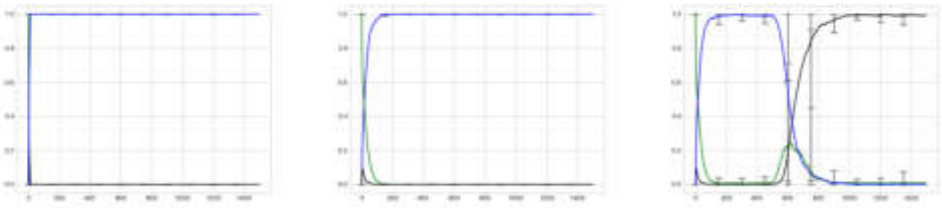
We have presented social learning as a combination of two processes; fusion in which agents change their beliefs under the influence of their peers and evidential updating in which agents learn directly from evidence. In this context we have considered four overall behavioural types generated by independently combining conservative and open-minded approaches to both processes. Different behaviour types have been shown to have different convergence and consensus properties. However, certain heterogeneous mixtures perform best in a range of different learning scenarios. In particular, a $50/50$ mix of adventurous and cautious fusion combined with a mix of 1% inquisitive and 99% confident evidential updating, is highly robust especially in dynamic environments.

In this paper we have assumed a fully connected, well-mixed population of interacting individuals. This is a strong assumption and there is growing evidence that it is not always optimal for social learning (Crosscombe & Lawry, 2022). Future work will investigate social networks with more constrained topologies in which connection between heterogeneous behaviour types may vary. Following



(a) $\sigma = 0.1$ and $\rho = 0.1$ (b) $\sigma = 0.9$ and $\rho = 0.1$ (c) $\sigma = 0.1$ and $\rho = 0.9$ (d) $\sigma = 0.9$ and $\rho = 0.9$

Figure 8: Proportion of correct beliefs with different combinations of parameter values w and λ and for different fusion and evidence rates. Results are for initial proportions $(1, 0, 0)$, i.e. all agents are wrong, $\epsilon = 0.3$ and at $t = 1500$



(a) $\lambda = 0$, $w = 1$ (b) $\lambda = 0$, $w = 0.5$ (c) $\lambda = 0.01$, $w = 0.5$

Figure 9: Average proportion of $\mathbf{t}$ (black line), $\mathbf{u}$ (green line) and $\mathbf{f}$ (blue time) belief states, plotted against time for different mixtures of behaviour types. Results are for 200 agents, $\sigma = 0.9$, $\rho = 0.1$ and $\epsilon = 0.3$ and are averages over 100 independent runs. The true state of the world switches from $\mathbf{f}$ to $\mathbf{t}$ at $t = 500$.

(Erbach-Schoenberg et al, 2011) it may also be interesting to consider heterogeneous updating rates e.g. where uncertain individuals update more often or more readily. Taking these ideas forward we will then explore their direct application to modelling social learning in biological and artificial systems.

Acknowledgements

This work was funded and delivered in partnership between Thales Group, University of Bristol and with the support of the UK Engineering and Physical Sciences Research Council, ref. EP/R004757/1 entitled "Thales-Bristol Partnership in Hybrid Autonomous Systems Engineering (T-B PHASE)."

References

A. Baronchelli: *The emergence of consensus: a primer*. R. Soc. open sci.5:172189. 2018

M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo: *Swarm robotics: a review from the swarm engineering perspective*. Swarm Intelligence. 7. 1–41. 2013

E. N. Buckingham: *Division of labor among ants*. Proc. Am. Acad. Arts Sci. 46. 425-508. 1911

M. Crosscombe, J. Lawry: *Exploiting vagueness for multi-agent consensus*. Multi-Agent and Complex Systems. 67-78. Springer. 2017

M. Crosscombe, J. Lawry, S. Hauert, M. Homer: *Robust distributed decision-making in robot swarms: Exploiting a third truth state*. IEEE/RSJ International Conference on Intelligent Robots and Systems. 4326-4332. IEEE. 2017

M. Crosscombe , J. Lawry: *The Impact of Network Connectivity on Collective Learning*. Distributed Autonomous Robotic Systems. DARS 2021. Springer Proceedings in Advanced Robotics. 22. Springer. 2022

I. D. Couzin, C. C. Ioannou, G. Demirel, T. Gross, C. J. Torney, A. Hartnett, L. Conradt, S. A. Levin, N. E. Leonard: *Uninformed Individuals Promote Democratic Consensus in Animal Groups*. Science. 334. 1578-1580. 2011

M. Črepinšek, S. H. Liu, M. Mernik: *Exploration and exploitation in evolutionary algorithms: A survey*. ACM Comput. Surv. 45. 3(35). 2013

I. Douven, C. Kelp: *Truth Approximation, Social Epistemology, and Opinion Dynamics*. Erkenntnis. 75. 271. 2011

E. Z. Erbach-Schoenberg, C. McCabe, S. Bullock: *On the interaction of adaptive timescales on networks*. Proceedings of the Eleventh European Conference on Artificial Life. 900-907. MIT Press. 2011.

G. Fu, W. Zhang: *Opinion Dynamics of Modified Hegselmann-Krause Model with Group-based Bounded Confidence*. Proceedings of the 19th World Congress The International Federation of Automatic Control. 9870-9874

C. M. Heyes: *Social learning in animals: categories and mechanisms*. Biol. Rev. 69. 207-231. 1994

E. R. Hunt, B. Mi, R. Geremew, C. Fernandez, B. M. Wong, J. N. Pruitt, N. Pinter-Wollman. *Resting networks and personality predict attack speed in social spiders*. Behavioral Ecology and Sociobiology. 73. 97. 2019

T. A. O'Shea-Wheller, E. R. Hunt, T. Sasaki: *Functional Heterogeneity in Superorganisms: Emerging Trends and Concepts*. Annals of the Entomological Society of America. XX(X). 1–13. 2020

C. A. C. Parker, H. Zhang: *Cooperative decision-making in decentralized multiple-robot systems: The best-of-n problem*. IEEE/ASME Transactions on Mechatronics. 14(2). 240–251. 2009

J. M. Peters, O. Peleg, L. Mahadevan: *Collective ventilation in honeybee nests*. J. R. Soc. Interface 16:20180561. 2019

J. Prasetyo, G. De Masi, · E. Ferrante: *Collective decision making in dynamic environments*. Swarm Intelligence. 13. 217–243. 2019

L. Schäfer, F. Christianos, J. Hanna, S. V. Albrecht. *Decoupling exploration and exploitation in reinforcement learning*. Unsupervised Reinforcement Learning (URL) Workshop in the 38th International Conference on Machine Learning.

A. Slivkins. *Introduction to Multi-Armed Bandits*. Foundations and Trends in Machine Learning. 12(1-2). 2019

A. Smith: *An inquiry into the nature and causes of the wealth of nations*. W. Strahan and T. Cadell, London. 1776

G. Theraulaz, E. Bonabeau, J-L. Deneubourg. *Response threshold reinforcement and division of labour in insect societies*. Proc. R. Soc. B. 265. 327-332. 1998

G. Valentini, E. Ferrante, M. Dorigo: *The Best-of-n Problem in Robot Swarms: Formalization, State of the Art, and Novel Perspectives* Front. Robot. AI. 4(9). 2017

A. Yaman, N. Bredeche, O. Caylak, J. Z. Leibo, S. Wan Lee: *Meta-control of Social Learning Strategies*. PLOS Computational Biology, 18(2). 2022

Symbiosis in Digital Evolution: A Review and Future Directions

Anya E. Vostinar[1*], Katherine G. Skocelas[2], Alexander Lalejini[3], and Luis Zaman[2]

[1]SymbuLab, Carleton College, Computer Science, Northfield, MN, USA
[2]Digital Evolution Lab, Michigan State University, Department of Computer Science,
Program in Ecology, Evolution, and Biology, BEACON Center for the Study of Evolution in Action, East Lansing, MI, USA
[3]ZE3 Lab, University of Michigan, Department of Ecology and Evolutionary Biology
& Center for the Study of Complex Systems, Ann Arbor, MI, USA
*vostinar@carleton.edu

Introduction

Symbiosis is a ubiquitous, vital biological dynamic (Paracer and Ahmadjian, 2000) that is difficult to experimentally study in DNA-based systems (Momeni et al., 2011). Since its inception, digital evolution has been used to study many types of symbiosis, and remains an area of active research in the field. Here, we summarize our recent review of symbiosis in digital evolution (Vostinar et al., 2021).

Symbiosis is often colloquially used to describe a mutually beneficial relationship. However, symbiosis actually encompasses any long-term and close relationship that occurs between organisms of at least two different species and that benefits at least one of the constituent partners. The spectrum of symbiotic relationships includes the extremes of both parasitism, in which one organism (the parasite) benefits, but the other organism (the host) is harmed, and mutualism, in which both organisms benefit. Historically, symbiosis has been understudied in biology, but there is growing recognition of the importance of better understanding this fundamental biological dynamic. Symbiosis, like many biological dynamics, is time- and resource-consuming to study at evolutionary timescales with traditional biological techniques (Sachs et al., 2011). It is also particularly challenging to model with traditional analytical mathematical modeling techniques, due to the importance of interactions between specific individuals of each species.

While most evolved symbiotic relationships on Earth are rife with idiosyncrasies (that may prohibit accurate generalizations from them), artificial life allows for the study of fundamental dynamics of symbiotic relationships, providing a lens to understand symbiosis both as it is and as it could be. In particular, digital evolution experiments are well-suited for studying the evolutionary causes and effects of symbiosis (Vostinar et al., 2021, Section 4), as depicted in Figure 1. Indeed, symbiotic relationships are woven throughout the history of digital evolution, even when the study of symbiosis was not the direct goal (Vostinar et al., 2021, Section 5). Here, we highlight the main findings regarding symbiosis in digital evolution, compare the software systems that support symbiosis, and discuss potential future directions of study.

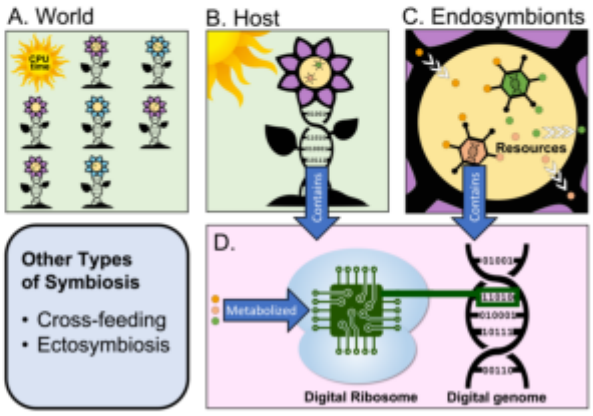


Figure 1: **Overview of the components of a hypothetical digital evolution system that supports endosymbiosis.** Figure originally published in (Vostinar et al., 2021) and further details found there.

Previous Findings

Digital evolution has been instrumental in several important findings regarding the evolutionary dynamics of symbiosis. These findings can be categorized as relating to 1) symbiosis' impact on population-level diversity over evolutionary time (Zaman et al., 2011; Fortuna et al., 2017; Zaman, 2018; Rocabert et al., 2017; Crombach and Hogeweg, 2009; Pachepsky et al., 2002), 2) symbiosis' impact on evolved complexity of one or both species (Colizzi and Hogeweg, 2016; Hickinbotham et al., 2021; Zaman et al., 2014), and 3) the evolution of symbiosis itself along the parasitism-mutualism spectrum (Vostinar and Ofria, 2019). The digital evolution software systems used to generate these results are summarized in Table 1, and the locations of the available codebases are detailed in (Vostinar et al., 2021, Table 3).

Future Directions

Because symbiosis has been historically understudied in both traditional biology and artificial life, there are an overwhelming number of open questions in this area. A subset

Table 1: A comparison of the digital evolution software systems that support symbiosis. Table originally published in (Vostinar et al., 2021).

System	Code available (Locations in (XX)	Ongoing Development	Parasitic Symbiosis	Mutualistic Symbiosis	Endo-symbiosis	Ecto-symbiosis	Graphical User Interface
Avida2	Yes	No	Yes	No	Yes	No	ncurses GUI
Symbulation	Yes	Yes	Yes	Yes	Yes	No	Web GUI
Evo2Sim	Yes	Yes	No	Yes	No	Yes	JavaScript GUI
Stringmol	Yes	Yes	Yes	No	No	Yes	No
Crombach & Hogeweg (2009)	No	No	No	Yes	No	Yes	No
Pachepsky et al. (2002)	No	No	No	Yes	No	Yes	No
Colizzi & Hogeweg (2016)	Yes	No	Yes	No	No	Yes	No

that are well-suited to digital evolution fall into two categories: 1) how does the presence of a co-evolving symbiont alter host evolutionary trajectories and vice versa?, and 2) what factors select for the *de novo* evolution of symbiosis, parasitism, and mutualism? There are many ways that symbiotic partners could alter each other's evolutionary trajectories, but digital evolution is well-suited to further exploring their impact on: 1) evolved complexity, 2) population diversity, 3) inter-species competition such as invasive species, 4) the transition from asexual to sexual reproduction, 5) the transition to eusociality, 6) range expansions and shifts of the symbiotic species, and 7) the possibility for open-ended evolution. Specific questions and further references for each of these topics are included in (Vostinar et al., 2021, Section 7). Here, we focus on the topic of open-ended evolution.

It is an open question whether symbiosis is necessary and/or sufficient for a system to achieve open-ended evolutionary dynamics. The definition of open-ended evolution is not settled, however the general idea is the presence of continuous evolution of "interesting" and novel organisms. This dynamic usually requires continuously changing selection pressures to avoid stagnation. As reported in the previous findings, coevolution in a symbiotic relationship can provide that continuous change, especially in parasitic symbiosis; however, mutualistic symbiosis may instead result in selection *against* rapid change. Because symbiotic systems can evolve along the parasitism-mutualism spectrum, the interplay of these dynamics raises questions, such as when symbionts will evolve to parasitism and provide the necessary selection pressure for open-ended evolution.

Further, the *de novo* evolution of symbiosis could possibly be viewed as *evidence* of a system being capable of open-ended evolution. The evolution of this trait could be considered a "state space" change (Adams, 2021), leading to continued emergence of truly novel organisms. Therefore, more exploration of the interplay between symbiotic dynamics and open-ended evolution is needed.

No current digital evolution software is able to investigate all (or even most) of the areas discussed here. Therefore, we conclude with a call for the next generation of digital evolution software to include support for: 1) both ecto- and endosymbiosis [1], 2) evolution along the parasitism-mutualism spectrum, 3) evolution between organisms with free-living and endosymbiont strategies as well as between those with facultative and obligate strategies, 4) multi-level symbiosis, such that an organism can be a host as well as a symbiont to a super-host, 5) multi-infection, such that endosymbionts are able to interact ecologically within a host, 6) ectosymbionts that can simultaneously interact with multiple hosts, 7) evolution between asexual and sexual reproduction strategies, and 8) evolution of endosymbionts with complex lifecycles of multiple hosts based on host predation.

Digital evolution, and the field of artificial life in general, have already made key contributions to the understanding of symbiotic evolutionary dynamics. However, the complexity and importance of symbiosis means that there is still much more work to be done. The field of artificial life has tremendous potential to play a pivotal role in determining the fundamental dynamics of symbiosis.

Acknowledgements

Funding sources are acknowledged in the full paper (Vostinar et al., 2021).

[1] Endosymbiosis is a symbiotic relationship in which one partner lives inside of the other. Ectosymbiosis is a symbiotic relationship in which that is not the case.

References

Adams, A. M. (2021). A graph-theoretic approach to understanding emergent behavior in physical systems. In *ALIFE 2021: The 2021 Conference on Artificial Life*. MIT Press.

Colizzi, E. S. and Hogeweg, P. (2016). Parasites sustain and enhance rna-like replicators through spatial self-organisation. *PLOS Computational Biology*, 12(4):1–17.

Crombach, A. and Hogeweg, P. (2009). Evolution of resource cycling in ecosystems and individuals. *BMC evolutionary biology*, 9(1):1–19.

Fortuna, M. A., Zaman, L., Wagner, A., and Bascompte, J. (2017). Non-adaptive origins of evolutionary innovations increase network complexity in interacting digital organisms. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1735):20160431.

Hickinbotham, S. J., Stepney, S., and Hogeweg, P. (2021). Nothing in evolution makes sense except in the light of parasitism: evolution of complex replication strategies. *Royal Society Open Science*, 8(8):210441.

Momeni, B., Chen, C.-C., Hillesland, K. L., Waite, A., and Shou, W. (2011). Using artificial systems to explore the ecology and evolution of symbioses. *Cellular and Molecular Life Sciences*, 68(8):1353–1368.

Pachepsky, E., Taylor, T., and Jones, S. (2002). Mutualism promotes diversity and stability in a simple artificial ecosystem. *Artificial life*, 8(1):5–24.

Paracer, S. and Ahmadjian, V. (2000). *Symbiosis: an introduction to biological associations*. Oxford University Press on Demand.

Rocabert, C., Knibbe, C., Consuegra, J., Schneider, D., and Beslon, G. (2017). Beware batch culture: Seasonality and niche construction predicted to favor bacterial adaptive diversification. *PLoS computational biology*, 13(3):e1005459.

Sachs, J. L., Skophammer, R. G., and Regus, J. U. (2011). Evolutionary transitions in bacterial symbiosis. *Proceedings of the National Academy of Sciences*, 108(Supplement 2):10800–10807.

Vostinar, A. E. and Ofria, C. (2019). Spatial structure can decrease symbiotic cooperation. *Artificial life*, 24(4):229–249.

Vostinar, A. E., Skocelas, K. G., Lalejini, A., and Zaman, L. (2021). Symbiosis in digital evolution: Past, present, and future. *Frontiers in Ecology and Evolution*, 9.

Zaman, L. (2018). Investigating open-ended coevolution in digital organisms. In *Artificial Life Conference Proceedings*, pages 258–259. MIT Press.

Zaman, L., Devangam, S., and Ofria, C. (2011). Rapid host-parasite coevolution drives the production and maintenance of diversity in digital organisms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO '11*, page 219, Dublin, Ireland. ACM Press.

Zaman, L., Meyer, J. R., Devangam, S., Bryson, D. M., Lenski, R. E., and Ofria, C. (2014). Coevolution Drives the Emergence of Complex Traits and Promotes Evolvability. *PLoS Biology*, 12(12):e1002023.

Automated Ligand Design in Simulated Molecular Docking

Rob Maccallum and Geoff Nitschke

Computer Science Department
University of Cape Town, South Africa
mccrob015@myuct.ac.za, gnitschke@cs.uct.ac.za

Abstract

The drug discovery process broadly follows the sequence of high-throughput screening, optimisation, synthesis, testing, and finally, clinical trials. We investigate methods for accelerating this process with machine learning algorithms that can automatically design novel ligands for biological targets. Recent work has demonstrated the viability of deep reinforcement learning, generative adversarial networks and auto-encoders. Here, we extend state-of-the-art deep reinforcement learning molecular modification algorithms and, through the integration of molecular docking simulations, apply them to automatically design novel antagonists for the adenosine triphosphate binding site of Plasmodium falciparum phosphatidylinositol 4-kinase, an enzyme essential to the malaria parasite's development within an infected host. We demonstrated that such an algorithm was capable of designing novel molecular graphs with better DSs than the best DSs in a set of reference molecules. There reference set here was a set of 1,011 structural analogues of napthyridine, imidazopyridazine, and aminopyradine.

Introduction

Drug discovery and design comprises three primary categories. First, *hit screening*, which is where classes of drugs or chemotypes which share a similar molecular scaffold are identified as having a notable binding affinity (BA) for a target receptor. Second, *hit-to-lead optimisation*, which is where hits obtained from the previous phase have their BA for the receptor increased through modification by experts. Third, *lead optimisation*, where leads optimised to have high BA for the target have their other physico-chemical properties such as solubility, selectivity, molecular polarizability, charge distribution, molecular weight and others customised for their intended application. The entire drug discovery and design process (including synthesis, testing and clinical trials), takes on average 10 years and costs an average of 2.6 billion US dollars (DiMasi et al., 2016). Also, drug discovery research productivity is on the decline, with average failure rates for clinical trials approaching 90% in all disease categories (Kadurin et al., 2017).

Related Work

In recent years there has been significant progress towards automating the hit-screening and lead optimisation phases through the use of *Evolutionary Algorithms* (EAs) (Eiben and Smith, 2015), *Generative Adversarial Networks* (GANs) (Guimaraes et al., 2017; You et al., 2018) and auto-encoders (Gómez-Bombarelli et al., 2018). For example, Supady et al. (2015) demonstrated that a population of random conformers of a given SMILES sequence could evolve into one containing only low energy conformers. Initially, a string was generated for each individual which represented a vector of torsion angles. The fitness function calculated each individual's conformational energy, estimated with Density Functional Theory (Atkins and de Paula (2010)), and modulated relative to the other individuals in the population. An EA was then used to generate low energy conformers of the initial random population.

Harel and Radinsky (2018) showed that VAEs could be applied to the generation of novel SMILES with optimised properties. Here one-dimensional recurrent convolutional layers were used to map from SMILES strings to a continuous vector encoding. Latent codes were decoded back to SMILES strings by mapping from the continuous vector to a probability distribution over the available characters. This was done iteratively for each character in the sequence until the terminal token was chosen. This method was expanded upon when Gómez-Bombarelli et al. (2018) used Bayesian optimisation with an auxiliary ANN to find latent codes which decoded to molecules with optimal properties. Also, Guimaraes et al. (2017) integrated GANs with RL to generate generate SMILES strings which, in terms of their constitution, resembled those of a set of reference molecules but also had improved solubility ($\log P$), SA and QED. Sanchez-Lengeling et al. (2017) extended this method to optimise the set of novel molecules for melting point, and photovoltaic conversion efficiency.

However, a key drug design goal is achieving suitable binding affinity (BA) for the target macromolecule, and automation of this phase of the drug design process (hit-to-lead optimisation) has received little research attention relative to the hit screening and lead optimisation phases.

In summary, the motivation for our method stems from the need to focus ML algorithms such as these on the hit-to-lead phase of drug design. In this context the goal is to generate novel compounds for which there are limited available reference sets. Supervised learning algorithms such as GANs and VAEs are consequently inappropriate as they require a large supply of empirically labelled training samples. In contrast, purely RL algorithms are capable of learning entirely unsupervised, improving their performance using only the feedback received from the environment through exploration and thus requiring no positive examples in order to learn to generate ligands of the required class. Therefore, in this work we explored the integration of RL algorithms with a docking simulator within which an autonomous agent could learn to design novel ligand candidates using only *a priori* laws of binding energy as a reward signal.

Research Objectives

Within the context of automating hit-to-lead optimisation, the focus of this study was the automatic design of novel ligands which are expected to yield high BA for a given target macromolecule. We coupled RL with an environment in which an autonomous agent could design molecular graphs and receive feedback about their docking scores (DSs). The result was a generative algorithm capable of correlating features of molecular structure with DS for a specified binding site of a target molecule. Such an algorithm must generate ligands with maximal DS for the receptor site and, when given a molecular scaffold, must return a ligand with greater DS for the target. Thus, our research question is:

> *How effective is RL as a method for automatically designing ligands possessing a high BA for specific macro-molecular targets?*

We address this question by evaluating a range of DSs that our algorithm attains versus DSs of ligands already available for given targets and the number of training episodes required to generate graphs with a DS above given task-performance thresholds.

Contributions

We contribute a simulation framework for training molecular docking agents to generate novel ligands with high BA, for target macromolecules, where BA is estimated using docking score (DS) as a proxy, for the receptor site of the macromolecule. The framework applies *double-deep Q-learning* (Mnih et al., 2013, 2015), in an environment comprised of a ligand to be modified and a target macromolecule. The agent designs molecular graphs and employs the Morgan algorithm (Rogers and Hahn, 2010) to convert graphs to molecular fingerprints. The environment state is the current molecular graph state and possible actions are graphs accessible from the current state, where the algorithm learns from environments with inconsistent action spaces. Docking between the ligand designed by the agent and the target macromolecule is simulated using the Autodock-GPU (Santos-Martins et al., 2021) package, and the DS calculated by Autodock-GPU defines the reward received by the agent at each step of a design episode.

Our framework potentially accelerates the *hit-screening* and *hit-to-lead* optimisation phases of drug design, since novel lead candidates may result from unintuitive graph arrangements identified by agent correlations between structural features and BA. Such a framework can thus guide and inform chemists during their selection of candidates for screening or when modifying hits to increase candidate BA. Also, the framework is extensible to *lead optimisation* by incorporating any physico-chemical property in rewards.

Methods

This study builds on previous work using *Q-learning* (Zhou et al., 2019) for automated drug design. Previous work explored the generation of molecules with optimised physico-chemical properties such as $\log P$, QED and SA; which were measured with the RDKit chemoinformatics package. However, to develop a ligand for a particular target receptor one needs to measure a given molecule's BA and specificity for that receptor. This can only be accurately measured experimentally, but estimated via simulations. Therefore, our methods incorporate simulations of molecular docking for a receptor identified as a target for a specific pathology into the reward function. Specifically, we use *deep Q-learning* to train agents within docking simulations to optimise a ligand's BA for a protein target, using DS as an estimate.

Agent

The agent was implemented as a pair of fully connected deep ANNs, and a memory buffer, where the input to the network was the concatenation of two vectors. The first was a 2048 bit ECFP_4 molecular fingerprint of each state (graph), accessible from the current state, where each bit in the fingerprint corresponded to a certain functional group. Fingerprint vectors were concatenated with the number of remaining steps in the design episode, so each accessible state was considered relative to number of steps the agent had left to modify the graph. Fingerprints of each state accessible from the current state, concatenated with episode steps remaining, were then presented to the *Q-network* in sequence and for each, the network calculated a *Q-value*.

This approach was taken at each step of the graph design process as the number and type of accessible states changes with the current state of the graph. The action-value distribution over the states accessible from the current state was thus constructed by evaluating each accessible state in sequence. An input layer of 2048 fingerprint neurons and 1 steps-remaining neuron was connected to 3 fully-connected hidden-layers with *Rectified Linear Unit* (ReLU) (Nair and Hinton, 2010) activation functions, comprising 1024, 512, and 128 neurons respectively. The final layer contained only a single output neuron with no activation function. This corresponded to the Q-value of each accessible state, given the steps remaining. This structure was the same for both the behavioural policy network (Q) and the target network (Q^{-}).

Environment

The training environment (simulated with *Autodock-GPU docking* (Santos-Martins et al., 2021)), combined the *ligand* (designed by the agent during its exploration phase over a maximum number of steps), and the target *protein*. Docking simulations then determined if the given ligand could form a stable complex with the target protein and if so what was the change in the free energy of the ligand-protein complex as a result of binding. This change in binding free-energy determined the reward an agent received at the end of each exploration episode. Agent designed molecules were two-dimensional molecular graphs (using the `.mol` format), where these graphs were presented to the agent as molecular fingerprints. Each episode step the agent was presented with the set of possible modifications it could make to the current graph (molecular fingerprint), choosing modifications according to its policy of finding a molecule with the highest DS for the receptor site of the target. After the modification episode, the agent designed ligand was presented to the target for docking.

Simulating Molecular Docking

Molecular docking was simulated with the Autodock-GPU package. Given a molecular graph and the crystal structure of a target protein, docking proceeded as follows.

The ligand's molecular graph was converted from 2D graph `.mol` format to 3D representation, including hydrogens, partial charges, and atomic coordinates using the `.pdbqt` format. Target protein crystal structure was obtained from the protein data in `.pdb` format (also converted to `.pdbqt` format). A 3D docking grid was then prepared which encapsulated the receptor site of the target protein. Preparation of the target protein and docking grid was done once prior to training whereas conversion of 2D ligand graphs to their 3D representations was performed after each molecular modification episode. Given the `.pdbqt` ligand and protein files and docking grid, Autodock used its search algorithm to explore conformational states of the given ligand, evaluating the ligand-protein interaction for each conformation. This conformational search was done by a *Lamarckian Evolutionary Algorithm* (EA) (Morris et al., 1999), which searched for the global minimum of equation 1. The conformation with the greatest corresponding release of binding-free energy was saved to a coordinate file.

Autodock implements a semi-empirical scoring function (differentiated from knowledge based and physics based), as a weighted sum of atomic interactions, tuned to structural data. Steric, hydrophobic, and hydrogen bonding interactions between atoms in the ligand, and atoms of the receptor within the docking grid are calculated. The weights of these terms were computed from a non-linear fit of the scoring function to structural data (Trott and Olson, 2010).

Free energy ΔG of a binding pose is given by equation 1.

$$\Delta G = \Delta H_{vdW} + \Delta H_{hbond} + \Delta H_{elec} + \Delta G_{desolv} + \Delta S_{tor} \tag{1}$$

Where, ΔH_{vdW}, ΔH_{hbond}, ΔH_{elec} are the enthalpy changes due to *Van Der Walls* interactions, hydrogen bonding and electrostatic interactions respectively, ΔG_{desolv} is the Gibbs free energy change due to desolvation and ΔS_{tor} is the change in ligand entropy due to the loss of rotatable degrees of freedom in the ligand as a result of binding. Energetic terms are approximated semi-empirically as follows:

$$\begin{aligned}
\Delta H_{vdW} &= W_{vdW} \sum_{i,j} \left(\frac{A_{ij}}{s(r_{ij})^{12}} - \frac{B_{ij}}{s(r_{ij})^{6}} \right) \\
\Delta H_{hbond} &= W_{hbond} \sum_{i,j} E(t) \left(\frac{C_{ij}}{s(r_{ij})^{12}} - \frac{D_{ij}}{s(r_{ij})^{10}} \right) \\
\Delta H_{elec} &= W_{elec} \sum_{i,j} \left(\frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}} \right) \\
\Delta G_{desolv} &= W_{desolv} \sum_{i,j} (S_i V_j + S_j V_i) e^{-r_{ij}^2 / 2\sigma^2} \\
\Delta S_{tor} &= W_{tor} N_{tor}
\end{aligned} \tag{2}$$

Where, sums $\sum_{i,j}$ are over all inter-molecular pairs of ligand-receptor atoms within the docking box. A_{ij}, B_{ij} are constants which depend on the modified Lennard-Jones potentials (Atkins and de Paula, 2010) between atoms i and j, and C_{ij}, D_{ij} are constants which depend on the hydrogen bonding potentials between i and j. S and V are salvation parameters and atom volume respectively and σ is set to 3.5. r_{ij} is the inter-atomic distance between atoms i and j and $s(r_{ij})$ is a smoothing function. $E(t)$ is a function which provides directionality for the hydrogen bond term based on the angle t. $\epsilon(r_{ij})$ is a dielectric function of r_{ij}. N_{rot} is the number of torsions in the receptor in its bound state. Weights W_{vdW}, W_{hbond}, W_{elec}, W_{desolv} and W_{rot} were

empirically set using linear regression on ligand-receptor complexes with known binding constants. Release of binding free-energy was then returned as the ligand's DS, which was an estimate of its BA. This DS then determined the agent's reward.

A single docking calculation was the evaluation of 10^6 to 10^8 scoring function evaluations during the EA run. Autodock-GPU (Santos-Martins et al., 2021) dramatically accelerates the run-time by exploiting the parallel nature of the docking algorithm, decomposing the population of candidate solutions into w work-groups, where work groups ran in parallel on a GPU compute unit.

Rewards and Shaping

The state of the environment (defined by the current molecular graph) is evaluated by the reward function on every step to determine if the goal state has been reached. The goal is that the agent discovers a novel molecular graph with high DS for the given target receptor, thus we cannot specify the goal molecular graph, rather we specify a property that the goal state should possess and the reward function is defined accordingly. Here, the desired property was high DS for the specified receptor site of a given target molecule. Therefore, the reward function returns the result of conformational search and the DS calculation performed by the docking package with its sign inverted, that is:

$$r(s_t) = -1 \times \Delta \text{ Binding Free Energy} \tag{3}$$

The reason for the inversion is that a greater reduction in binding free energy is of higher value. As a result, the Q-value (state-action value), of any molecular graph returned by the agent's behavioural network, equates to the expected cumulative DS expected in the remaining steps (after choosing the given action).

In this study, it was not possible to calculate the DS of each transition along a roll-out as this would have made the run-time of a complete training session infeasible. Hence, we calculated the DS of the final state along a roll-out. Given that a roll-out trajectory was composed of 40 transitions, only one of which receives an extrinsic reward, the problem was one of sparse rewards, and a shaping method was implemented to encourage learning. Given the reward for the terminal state of a roll-out trajectory, rewards for the preceding transitions were calculated using a method similar to that of potential-based reward shaping (Ng et al., 1999). Here the potential of the terminal state was taken as the DS received from the environment, and the potentials of the preceding states were estimated by linearly interpolating from the full DS in the final state to zero in the initial state (equation 4).

$$\phi(s_t) = r_f - \frac{r_f}{t_{max}} \times (t_i - 1) \tag{4}$$

Where, r_f is the reward of the terminal state, t_{max} is the number of steps in a trajectory and t_i is the steps remaining between state i and the terminal state.

Learning Algorithm

The RL algorithm (Algorithm 1) used to train the agent was double deep Q-networks (DDQN) (Mnih et al., 2013, 2015). First the agent was initialised with an empty molecular graph (line 4, Algorithm 1). The agent was given a maximum of 40 steps (chosen based on previous work (Zhou et al., 2019; You et al., 2018)) in which to construct a ligand. Each step corresponded to the potential addition or removal of an atom or bond, given a limit of 40 atoms as the maximum size of the ligands, excluding hydrogen, with a mass in the range of 500 to 1000 Daltons, common for small-molecule drugs (Chhabra, 2021). At each step of the modification phase, the environment generates all possible and valid molecular graphs that are accessible from the current state and returns these to the agent in a tensor.

The next states s_{t+1} are the actions a_t (Algorithm 1), so at each step of an episode the action space was defined by accessible states. The agent then uses its behavioural network to select a state (action) to move into (lines 5, 6, and 7, Algorithm 1). At each step, docking between the ligand and target receptor was simulated. The DS was calculated and returned to the agent as a reward (r_t in line 7, Algorithm 1), and the transition $(s_t, s_{t+1}, r_t, term)$ was stored in the agent's memory (line 9, Algorithm 1), where $term$ is a flag indicating if s_{t+1} was a terminal state. A trajectory of at most 40 such transitions (t_{max} in algorithm 1) constituted a single roll-out or exploration episode.

After T_{bp} roll-out episodes (backpropagation period), where data was sampled from the policy π_θ, the agent randomly sampled a mini-batch of transitions from the replay memory (line 12, Algorithm 1) and optimised its ANN approximation of the optimal action-value function Q_θ using stochastic gradient descent in backpropagation (lines 13, 14 and 15, Algorithm 1). The parameters of the target network θ^- were then updated with a fraction τ of the updates made to their counterparts in the behavioural network (line 16, Algorithm 1). Thus training alternated between T_{bp} sampling episodes e and mini-batch gradient descent in backpropagation. This loop continued for e_{max} episodes or until convergence in the policy network's loss function.

Experiments and Results

The learning framework was evaluated with the following three experiments, defined such that their results would answer our research questions (posed in the introduction).

1. Generation of ligands with maximal DS for the PfPI4k target receptor when the agent begins each training episode from an empty starting molecule.
2. Generation of ligands with maximal DS for the PfPI4k target receptor when the agent begins from three reference scaffolds known to be structural analogues of ligands with high BA, namely napthyridine, imidazopyridazine, and aminopyradine.
3. Docking of 1,011 structural analogues of napthyridine, imidazopyridazine, and aminopyradine with the PfPI4k target receptor.

In the first two experiments the agent's goal was to construct ligands through the addition or removal of atoms or bonds, such that the terminal state after 40 transitions received a high DS. When the environment was reset at the beginning of each new episode the state of the ligand was returned to its initial state. These experiments explored the impact of the initial state on the final agent DS. In the first experiment, the agent begins from a single carbon atom, whereas in the second experiment, the agent begins from one of three reference scaffolds. The purpose of the third experiment is to define a reference for comparing the scores of the ligands generated by the agent.

Generation from an Empty Molecule

The first part of experiment 1 (figure 1, blue curve) was to evaluate whether it was possible for the agent to achieve some degree of success with such sparse feedback from the environment. Figure 1 (left) shows the sparse and shaped reward curves over the course of training. Both curves correspond to training session where the agent begins from an empty molecule on each episode, where in the sparse reward training session (blue curve) only the terminal state received a DS reward, and in the shaped reward training session (red curve), reward shaping was used to estimate a reward for all transitions in the roll-out episode.

The impact of reward shaping was investigated since when rewarding the agent with a ligand's DS, it was not possible to return a reward at every step of a roll-out episode. This was because the time taken to calculate a single ligand's DS made the run-time for a full training session of 5000 episodes untenable. Thus, a DS was returned only for the terminal state of the roll-out episode, meaning preceding transitions were added to the replay memory with no associated reward.

The second part of experiment 1 (figure 1, red curve) created dense reward signals from the sparse extrinsic reward returned by the environment at the end of the agent's roll-out trajectory (equation 4). Instead of pushing the states received from the environment along with a zero reward to the agent's replay memory (as they were received), they were instead buffered until the episode's end.

Equation 4 was then used to calculate a non-zero reward for every preceding transition using the terminal state reward. The initial generation experiment where the agent received a reward only for the terminal state demonstrated that this is insufficient information from which to extract a useful policy as the DS of generated ligands failed to increase in 5,000 episodes. The terminal state DS after 4,500 training episodes was lower than the DSs of terminal states after random exploration in the first 500 episodes.

However, simple reward shaping that linearly extrapolates backwards from the final state, to calculate a reward for preceding states (equation 4), immediately enabled the agent to learn an effective design policy. The "shaped" plot (figure 1, left), shows that after starting at an initial DS of greater than -6 kcal/mol (from random exploration), the agent converged on a minimum of approximately -13 kcal/mol with the best candidates exceeding -15 kcal/mol.

The effectiveness of this shaping methods is likely due to the fact that the value of a given state is not determined by the state in isolation, but the terminal state of the path in which the state occurred. Thus, states are valuable if they are close to terminal states with high DSs. For example, if a given roll-out episode resulted in a terminal state which received a high DS, then the penultimate state along that path would also be valuable as it permits access to the high-scoring terminal state. The further back along the path one is from the terminal state, the less valuable the states become. The shaping function (equation 4) was designed to implement this logic, thus conditioning the agent to select features in the extended-connectivity fingerprints leading to high-scoring terminal states within the 40-step limit.

Given that the use of reward shaping enabled the agent to learn where it had previously failed in the sparse reward environment, we can assume that a more sophisticated sparse-reward amelioration strategy such as hindsight experience replay, which has been shown experimentally to outperform reward-shaping (Andrychowicz et al., 2017), should improve the algorithm's performance.

Algorithm 1: Double Deep Q-Learning (DDQN) (Mnih et al., 2013, 2015)

Randomly initialise action-value function Q_θ and target network Q_{θ^-} with parameters $\vec{\theta}$ and $\vec{\theta^-}$ respectively.
Initialise replay memory $\mathcal{D}$ to capacity $\mathcal{N}$
repeat
  Reset environment
  repeat
    With probability ϵ sample random action (next state) s_{t+1}
    Otherwise select $s_{t+1} = \underset{s_{t+1}}{\max} Q^*_\theta(s_t, s_{t+1})$
    Move into next state s_{t+1} and observe reward r_t
    Store transition $(s_t, s_{t+1}, r_t, term)$ in $\mathcal{D}$
  until t_{max}
  if $e \mod T_{bp} == 0$ **then** optimise behavioural and target networks
    Sample random minibatch of transitions $(s_j, s_{j+1}, r_j, term)$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{terminal } s_{j+1} \\ r_j + \gamma \underset{a'}{\max} Q(s_{t+1}, a') & \text{non-terminal } s_{j+1} \end{cases}$
    Update Q_θ via one step of gradient descent by backpropagation
    $\Delta\vec{\theta} = \nabla_\theta J(\theta) = \mathbb{E}\left[\left(r + \gamma \underset{a'}{\max} Q_{\theta^-}(s', a') - Q_\theta(s, a)\right)\nabla_\theta Q_\theta(s, a)\right]$
    $\theta^- \leftarrow \tau\theta$
  end
until e_{max}

Generation from Reference Series

This experiment explored the effect of focusing the search on regions of chemical space near to the structural features of chemical series known to have an affinity for the receptor. Here the variant of the algorithm incorporating reward shaping was applied to three runs, each composed of 5000 episodes starting from the *aminopyridine*, *napthyridine* and *imidazopyridazine* molecular backbones. hese chemical series are being investigated as derivatives of these three molecular backbones with varying substituents have demonstrated high inhibition potency against the PfPI4K target enzyme in phenotypic whole-cell screenings.

Plots of the terminal state DS over the course of training for these three runs are shown in figure 1 (right). The curves are colour-coded to indicate which scaffold the agent was modifying in each training session.

Whereas the previous experiment evaluated agent ability to find a path from an arbitrary point in chemical space (a single carbon atom) to a region of high DS for the receptor, this experiment evaluated the performance impact of simplifying the problem by beginning training from a region of chemical space known to contain high-BA structures.

Figure 1 (right) shows that when starting a 40-step rollout from each of these scaffolds the initial ligands generated by the agent (from random exploration), obtained a better terminal state DS than those obtained when beginning from only a single carbon atom, as the terminal state DSs from random roll-outs starting from napthyridine, aminopyradine and imidazopyridazine were approximately -9 kcal/mol.

After approximately 2,500 training episodes the agent converged on candidate ligands with an average terminal state DS of approximately -14 kcal/mol, with numerous candidates exceeding -15 kcal/mol and the best exceeding -16 kcal/mol. This result is comparable to the performance of MolDQN (Zhou et al., 2019), which converged on a score of 0.8 after 3,000 episodes when training to optimise QED.

In summary, when starting from scaffolds known to be structural analogues of ligands with high DS for the target, the agent begun training by finding (initially via random exploration), states with a better DSs than those discovered when beginning from only a single carbon atom, and converged on states with better DSs than those obtained when starting from only a single carbon atom. The agent likely converges on better molecules when building from known scaffolds since the search is now being constrained to the region of chemical space surrounding the given scaffold. Since the *aminopyradine*, *napthyridine* and *imidazopyridazine* scaffolds are the backbones of three chemical series known to contain high-BA ligands, by constraining the search to this region, the problem of finding high-BA candidates is simplified since the agent is more likely to discover rewarding states through exploration.

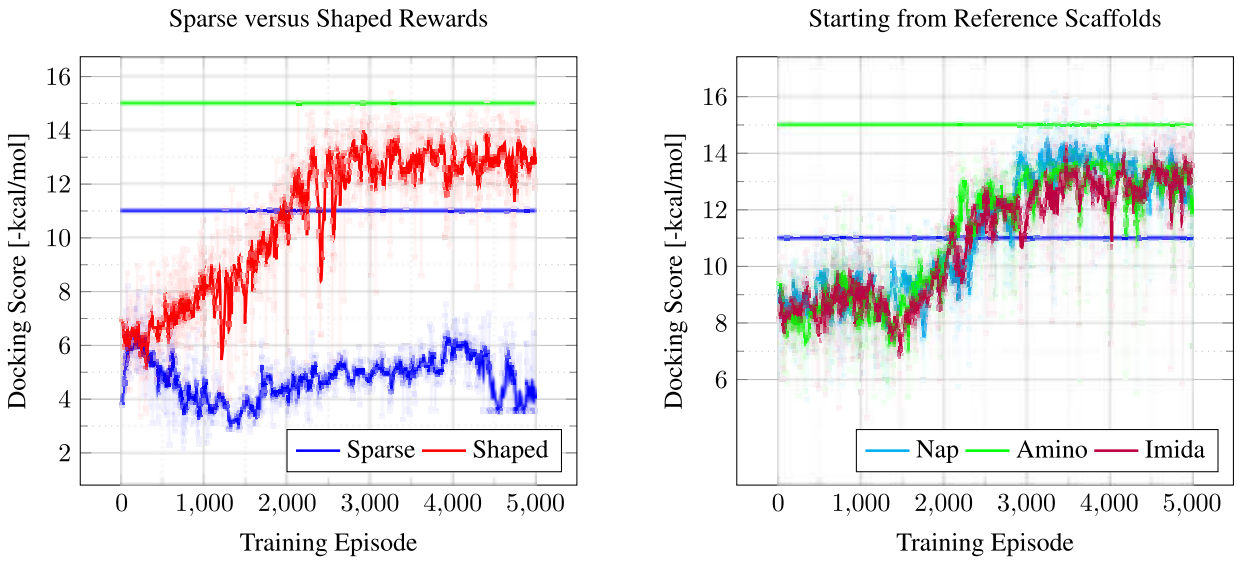


Figure 1: Training curves from the various DS rewards experiments showing the Autodock DS of the terminal state per training episode. The plots show the raw data as well as a that data smoothed with the following first-order IIR low-pass filter $f(x_t) = x_{t-1} * \alpha + (1-\alpha)x_t$ with $\alpha = 0.8$. Data are agent performance when learning from sparse rewards (left), using reward shaping (left), and starting from known reference molecules (right). Legend abbreviations (right) correspond to reference scaffolds *napthyridine*, *aminopyridine* and *imidazopyridazine*. Experiments starting from the reference molecules also incorporated reward shaping. The two horizontal lines show the mean and maximum DS of the ligands in the reference set (figure 2).

This is expected since structural analogues, (molecules sharing similar scaffolds but with different substituents and sufficiently similar molecular backbone), are candidates for functional analogues. These are two molecules with similar pharmacological properties. They exhibit similar biochemical or physiological effects on the human body, but with variations in efficacy and side effects (Bruice, 2011).

This result is also supported by other EAs using specially pre-initialised populations to boost the task performance of evolved solutions by evolving novel functional analogues in initial populations (Rupakheti et al., 2015; Brown et al., 2004; Lameijer et al., 2005). In other RL approaches the network has also been initially pre-trained structural analogues (Sumita et al., 2018) to boost task performance. The notion of structural analogues has also been incorporated in RL algorithms using *Tanimoto similarity* (Zhou et al., 2019), where the agent attempts to maximise similarity to a given scaffold when generating novel structure.

Comparison with Existing Ligands

In order to evaluate the algorithm's performance when generating ligands for the PfPI4K receptor, a reference point was needed. Therefore, a set of ligands currently being explored as potential antagonists for the PfPI4K enzyme were docked against the target receptor for comparison. The DSs of the reference ligands are plotted in figure 2.

Figure 2 shows the DSs of 1,011 structural analogues of the napthyridine, aminopyradine and imidazopyridazine scaffolds. These structures were docked with the same parameters used to reward the agent during the generation experiments. From the histogram we see that the mean of the set is approximately -11 kcal/mol with only a very small number of structures receiving scores better than -14 kcal/mol with none being better than -16 kcal/mol. Thus, figure 2 indicates the agent was able to design ligand candidates containing a substantial number of ligands with DSs better than the best DSs in a set of reference ligands.

Conclusion

This study sought to investigate the potential for applying RL to the development of generative algorithms for automated drug design. Our methods comprised a deep RL agent and simulation environment where the agent could construct molecular graphs bond-by-bond and dock the resulting ligands with the crystal structure of a target receptor. The experiments evaluated the agent and environment as the task of maximising DS for a given receptor, specifically the ATP binding site of the PfPI4K enzyme. Experiments investigated the impact of *sparse* versus *shaped* rewards, focusing the search on a particular region of chemical space, and comparing the ligands generated by the agent with those currently being evaluated in chemical laboratories.

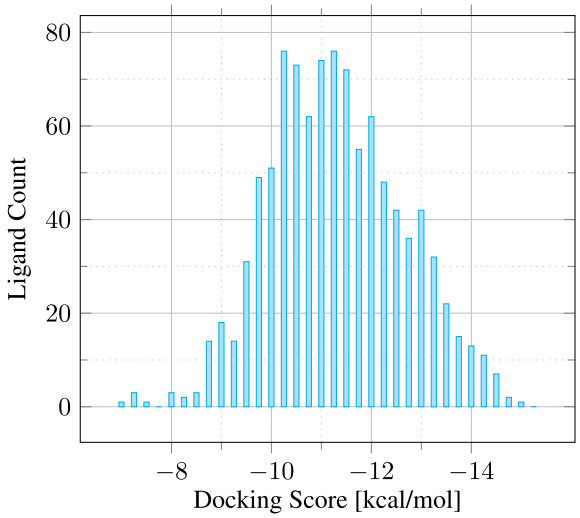


Figure 2: Result of docking 1011 structural analogues derived from *napthyridine*, *aminopyridine* and *imidazopyridazine* scaffolds against the PfPI4K receptor. x-axis: DS intervals of 0.25 kcal/mol. y-axis: Ligands in each interval. Histogram shows range and distribution of Autodock DSs for known ligands: benchmark for agent task-performance.

Rewarding only the terminal state during training significantly reduced training time and using reward shaping facilitated learning. Reward shaping was successful since the shaping function assumed the state value to be proportional to both the DS received by the (episode) terminal state and its distance from the terminal state. However, we are investigating the efficacy of other sparse reward strategies such as hindsight experience (Andrychowicz et al., 2017).

When beginning training episodes from reference scaffolds known to be analogues of ligands with high DSs, the agent was able to discover more molecules with higher DSs. This was due to the search space being focused to the region of chemical space where the *napthyridine*, *aminopyridine* and *imidazopyridazine* series are located, as this appears to simplify the search problem for the agent thus leading to the policy converging on ligands with a higher DSs.

Finally, when comparing the automatically designed ligands with those currently available, we observed that the agent was able to generate a substantial number of ligands with higher DSs than the best ligands in the reference set.

Future Work

The most apparent shortcoming of the current implementation is the number of impossible atomic arrangements which appear in the graphs designed by the agent, rendering the proposals impossible to synthesise even though they possess high DSs. This is a consequence of the reward function considering DS in isolation. The agent therefore searches for graphs which optimally fit the receptor site without any consideration for other properties of those graphs. There are a few methods which could potentially address this. The first would be to hard-code filters, which prevent specific modifications that would lead to these unrealistic structures, into the environment. Alternatively, instead of rules which filter out certain actions that would lead to undesirable features, the agent could instead be presented with modifications which change whole functional groups. For example, instead of choosing only between adding a single carbon, nitrogen or oxygen atom, the agent's choices would also include the options to add carobxylic acid, aldehyde, amine or phenyl functional groups. In combination with these methods, QED and SA could be incorporated into the reward function. The goal would then be to maximise QED and SA simultaneously with DS. In addition to having high DSs, the resulting ligands would then also have high QED and SA scores as well, thus improving their utility.

Finally, there is potential to avoid the inaccuracies and computational demands of simulations entirely by leveraging available IC50 data to develop surrogate models. These could replace the computationally demanding docking simulator to accelerate training. By first training a discriminative network to predict IC50 values from molecular fingerprints, this predictive model could be used as the reward function for a generative agent. In addition to accelerating training, this could potentially be more useful than DSs calculated from simulations, as IC50 measurements are obtained from whole-cell phenotypic screenings, and so they also account for off-target interactions within the cell. The reward returned from this predictive IC50 model could of course also be combined with QED and SA as previously described.

References

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. (2017). Hindsight Experience Replay. *arXiv*.

Atkins, P. and de Paula, J. (2010). *Physical Chemistry*. Oxford University Press, Oxford, 9 edition.

Brown, N., Mckay, B., and Gasteiger, J. (2004). The de novo design of median molecules within a property range of interest. *Journal of Computer-Aided Molecular Design*, 18(12):761–771.

Bruice, P. Y. (2011). *Organic Chemistry*. Pearson, sixth edition.

Chhabra, M. (2021). Biological therapeutic modalities. *Translational Biotechnology*, pages 137–164.

DiMasi, J. A., Grabowski, H. G., and Hansen, R. W. (2016). Innovation in the pharmaceutical industry: New estimates of R&D costs. *Journal of Health Economics*, 47:20–33.

Eiben, A. and Smith, J. (2015). *Introduction to Evolutionary Computing*. Natural Computing Series. Springer Berlin Heidelberg, Berlin, Heidelberg, 2nd edition.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2018). Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4(2):268–276.

Guimaraes, G. L., Sanchez-Lengeling, B., Outeiral, C., Farias, P. L. C., and Aspuru-Guzik, A. (2017). Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. *arXiv preprint arXiv:1705.10843*.

Harel, S. and Radinsky, K. (2018). Prototype-Based Compound Discovery Using Deep Generative Models. *Molecular Pharmaceutics*, 15(10):4406–4416.

Kadurin, A., Aliper, A., Kazennov, A., Mamoshina, P., Vanhaelen, Q., Khrabrov, K., and Zhavoronkov, A. (2017). The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, 8(7):10883–10890.

Lameijer, E. W., Bäck, T., Kok, J. N., and Ijzerman, A. P. (2005). Evolutionary algorithms in drug design. *Natural Computing*, 4(3):177–243.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. *CoRR*, abs/1312.5.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Morris, G. M., Goodsell, D. S., Halliday, R. S., Huey, R., Hart, W. E., Belew, R. K., and Olson, A. J. (1999). Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function. *Journal of Computational Chemistry*, 19(14):16391662.

Nair, V. and Hinton, G. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, ICML 2010, pages 807–814, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ng, A. Y., Harada, D., and Russell, S. J. (1999). Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 278–287, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Rogers, D. and Hahn, M. (2010). Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754.

Rupakheti, C., Virshup, A., Yang, W., and Beratan, D. N. (2015). Strategy to discover diverse optimal molecules in the small molecule universe. *Journal of Chemical Information and Modeling*, 55(3):529–537.

Sanchez-Lengeling, B., Outeiral, C., Guimaraes, G. L., and Aspuru-Guzik, A. (2017). Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC). *ChemRxiv*.

Santos-Martins, D., Solis-Vasquez, L., Tillack, A. F., Sanner, M. F., Koch, A., and Forli, S. (2021). Accelerating AutoDock 4 with GPUs and Gradient-Based Local Search. *Journal of Chemical Theory and Computation*, 17(2):1060–1073.

Sumita, M., Yang, X., Ishihara, S., Tamura, R., and Tsuda, K. (2018). Hunting for Organic Molecules with Artificial Intelligence: Molecules Optimized for Desired Excitation Energies. *ACS Central Science*, 4(9):1126–1133.

Supady, A., Blum, V., and Baldauf, C. (2015). First-Principles Molecular Structure Search with a Genetic Algorithm. *Journal of Chemical Information and Modeling*, 55(11):2338–2348.

Trott, O. and Olson, A. J. (2010). AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461.

You, J., Liu, B., Ying, R., Pande, V., and Leskovec, J. (2018). Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation. *Advances in Neural Information Processing Systems*, 2018-Decem(NeurIPS):6410–6421.

Zhou, Z., Kearnes, S., Li, L., Zare, R. N., and Riley, P. (2019). Optimization of Molecules via Deep Reinforcement Learning. *Scientific Reports*, 9(1):10752.

The benefits of credit assignment in noisy video game environments

Jacob Schoemaker[1] and Karine Miras[2]

[1]Rijksuniversiteit Groningen, Groningen, Netherlands
[2]Vrije Universiteit Amsterdam, Amsterdam, Netherlands
k.dasilvamirasdearaujo@vu.nl

Abstract

Both Evolutionary Algorithms (EAs) and Reinforcement Learning Algorithms (RLAs) have proven successful in policy optimisation tasks, however, there is scarce literature comparing their strengths and weaknesses. This makes it difficult to determine which group of algorithms is best suited for a task. This paper presents a comparison of two EAs and two RLAs in solving EvoMan - a video game playing benchmark. We test the algorithms both with and without noise introduction in the initialisation of multiple video game environments. We demonstrate that EAs reach a similar performance to RLAs in the static environments, but when noise is introduced the performance of EAs drops drastically while the performance of RLAs is much less affected.

Introduction

Machine Learning has achieved considerable success creating autonomous agents to play video games - referred to as Computational Intelligence in games (Lucas, 2008). Creating an agent that presents human-level performance in games has applications such as automatically testing the difficulty of (procedurally generated) opponents (Promsutipong and Kotrajaras, 2017), or generating adaptively interesting opponents for single-player games (Moriyama et al., 2014). Furthermore, the varied nature of games provides a great testbed for the capabilities of machine learning algorithms because they allow us to test the performance and dynamics of an algorithm in a variety of situations.

Two prominent methods in this field are Evolutionary Algorithms (EAs) (Ishikawa et al., 2020; Hausknecht et al., 2014) and Reinforcement Learning Algorithms (RLAs) (Crespo and Wichert, 2020; Givigi et al., 2010; LeBlanc and Lee, 2021). In game playing, EAs evolve a population of policies through mechanisms called selection, mutation, and crossover. RLAs on the other hand function by training a single policy, updating the policy step-by-step through feedback given by the environment. The differences between EAs and RLAs are fundamental and influence trade-offs when choosing one over the other. RLAs and EAs receive feedback at different points in time. EAs receive eventual feedback at the end of an episode and thus the feedback is over their entire performance. In contrast, RLAs receive immediate feedback after every action, which allows them to assign credit for rewards to specific actions. Delayed rewards can be a problem for RLAs (Sutton, 1992) since they rely on assigning credit to the actions most influential to the reward they gained. This is often addressed using discounted rewards, where some fraction of the total reward in the previous time step is added to the reward of the current time step. To EAs, delayed reward poses no problem since they do not receive their feedback until the end of an episode. However, this means that EAs are incapable of assigning credit to actions.

Although both EAs and RLAs have proven successful in policy optimisation tasks, there is relatively scarce literature assessing the strengths and weaknesses of their different characteristics. A few examples we are aware of are studies carried out by Rieser et al. (2011); Taylor et al. (2006); Drugan (2019). Therefore, this paper compares these classes of algorithms in two dimensions: nature (EA or RL) and complexity (simple and complex). As a testbed, we use EvoMan - a video game playing framework (da Silva Miras de Araújo and de França, 2016) created to facilitate experimentation using computational intelligence and to provide benchmarks. Importantly, no comparison of EAs with RLAs for the EvoMan benchmark has been found in the literature to this date.

In our experiments, we test these algorithms in four game environments twice: once with noise in the environment and once without noise. In particular, we seek to answer the following question: “What is the impact of noise on the performance of EAs and RLAs?”. Furthermore, this paper also contributes with an extension of the EvoMan framework that supports the use of RLAs, and it provides a baseline for this class of algorithms within EvoMan games.

Related Work

Due to the usefulness of games as benchmarks for understanding the behaviour of Artificial Intelligence (AI) algorithms, they have been extensively used in prior AI research. The Arcade Learning Environment (ALE) has been used as

a benchmark for performance of Computational Intelligence since it was first introduced by Bellemare et al. (2013). It consists of over 50 games originally designed for the Atari 2600, each of which provide a setting interesting enough to be representative of a real world scenario, free from the experimenter's bias as it has been created by a third party. Atari games are also simple enough so that it is possible to emulate them much faster than real time. With advances being made toward beating the human benchmark performance on the ALE, most notably by the AI agent Agent57 by Badia et al. (2020), researchers have been looking towards games from the next generation of consoles, the Nintendo Entertainment System (NES) (LeBlanc and Lee, 2021; Murphy, 2013).

One such game is Mega Man II, a challenging platforming game developed by CapCom in the 1980s for the NES. This game includes several one vs one combat scenarios with various mechanics. These scenarios have been emulated in a public domain clone of the original game Evoman (da Silva Miras de Araújo and de França, 2016). This clone serves as a testbed for one of these next-generation games, facilitating experimentation using computational optimisation techniques. EAs have yielded a good degree of success playing EvoMan games (da Silva Miras de Araujo and de Franca, 2016; Ishikawa et al., 2020).

EAs and RLAs have previously been compared in Rieser et al. (2011) and Taylor et al. (2006). Rieser et al. (2011) compared SARSA (RLA) and a simple, binary GA (EA), and found SARSA to perform significantly better than the GA when there was uncertainty in the environment. They also found that the more uncertain the environment became, the more of an advantage SARSA gained. In contrast, Taylor et al. (2006) found NEAT (EA) to perform better than SARSA in a partially observable task with noisy sensors. When the environment was made fully observable however, SARSA outperformed NEAT.

Environments

The EvoMan framework[1] is a reimplementation of the final-stage games of the video game MegaMan II, introduced by da Silva Miras de Araújo and de França (2016). Originally, the framework did not allow querying the states of the environments nor updating the agent's controller during the episodes but only at their end.[2] In the current paper, EvoMan has been adapted into an OpenAI Gym environment (Brockman et al., 2016), solving these limitations and thus supporting the use of RLAs. The code for the adapted framework is available here.

The player (or agent) can take 5 actions: *move right, move left, jump, shoot, and release*[3]. It is possible to take multiple actions simultaneously, e.g., go right and shoot. The goal of the player is to deplete the energy of the enemy by shooting at it. Meanwhile the player has to avoid the enemy and its projectiles (up to 8 at a time), which deplete the player's energy.

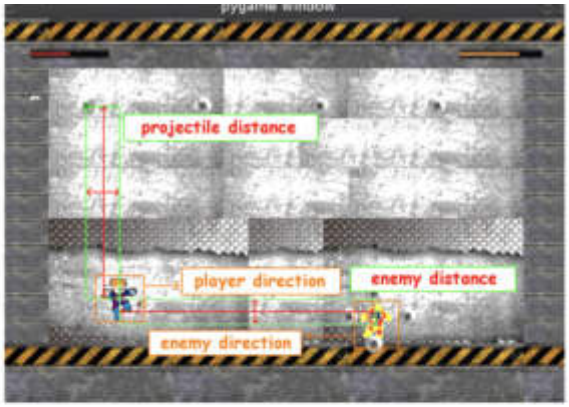


Figure 1: Sensors of the player agent.

The environment returns 20 sensor values at every timestep, representing the current state of the environment (Fig. 1). These values consist of the following:

- Enemy's x position relative to the player
- Enemy's y position relative to the player
- The direction the player is facing (represented as 1 or 0)
- The direction the enemy is facing (represented as 1 or 0)
- Hostile projectile's x position relative to the player (8×)
- Hostile projectile's y position relative to the player (8×)

Enemies

From EvoMan, 4 environments (enemies) were chosen for experimentation. These environments were chosen because they were deemed to present the most diverse set of game mechanics. Examples of learned agents playing against each of the enemies are available here. Notably, all enemies deal a small amount of damage to the player at every timestep the player is in (body) contact with the enemy.

AirMan AirMan was chosen because it is very easy to beat, and thus constitutes a good baseline benchmark. It is considered easy because the target is mostly static, and thus not challenging to aim at, and because the projectiles are always in the same location so that an avoidance move can easily be discovered.

[1] The documentation of the framework is available here.

[2] Additionally, the baselines available for EvoMan were only EAs.

[3] Release is equivalent to a human player letting go of the jump button. This allows for an early cut-off to the upwards momentum, resulting in a lower jump

BubbleMan BubbleMan was chosen because it has a special peculiarity in its arena, which makes for an interesting case. The difficulty of this environment is jumping over the projectiles, whilst releasing the jump in time in order to avoid the spikes which line the top of the arena.

FlashMan FlashMan was chosen because, though it is not particularly difficult to defeat, it is very difficult to obtain a high score on. The difficulty in this enemy is making sure the player is in the right place at the right time. The player needs to line themselves up with the enemy to be able to hit them with their projectile, but if they are lined up with the enemy at the wrong time, they could sustain a lot of damage whilst being powerless to avoid the projectiles in real-time.

HeatMan HeatMan was chosen because it was found to be difficult to optimise for (da Silva Miras de Araujo and de Franca, 2016). The difficulty of this enemy comes from the fact that the player needs to both jump over static particles and to dodge an enemy that is invulnerable during its time of movement.

Evaluation functions

The evaluation function was chosen such that the total reward gathered by an RLA over a single episode would be equivalent to the fitness assigned to an EA playing the exact same episode.

For the RLAs, the reward returned at each time step is the damage done to the enemy multiplied by a hyperparameter $e\_weight \times damageDone$, minus the damage taken by the player, multiplied by a hyperparameter $p\_weight \times damageTaken$, leading to Eq. 2. For the EAs, these rewards are summed over the course of a full episode to determine the fitness of the individual (Eq. 3).

During training, e_weight and p_weight were both set to 0.5, as this was empirically found to yield the best reward out of the tested values:

$$[e\_weight, p\_weight] = [\alpha, 1 - \alpha] \tag{1}$$

where $\alpha = [0.0, 0.1, ..., 1.0]$. This led to the equations

$$reward_n = 0.5 \times DD_n - 0.5 \times DT_n \tag{2}$$

$$fitness = \sum_{n=0}^{N} 0.5 \times DD_n - 0.5 \times DT_n \tag{3}$$

where DD is the damage dealt to enemy, DT is the damage taken by player, and n is the current timestep, and N is the length of the episode.

Noise

Noise can be introduced into the environment by having the position of the enemy randomised at the start of each episode (random initialization). When noise is enabled, the starting position of each enemy in the x-axis takes one of 4 values, sampled from a uniform distribution.

Algorithms

We use four different algorithms: 2 EAs and 2 RLAs - one simple and one complex from each class. All algorithms have been trained using a total budget of 2.5×10^6 timesteps. In preliminary testing, it was found that in most environments a player would either die or win within about 250 timesteps. For the EAs, which used a population and offspring size of 100 and were evolved for 100 generations, this came out to about $100 * 100 * 250 = 2.5 \times 10^6$ timesteps. Therefore, this value was also used as the (maximum) amount of timesteps for the RLAs. All algorithms optimize neural network controllers that receive the 20 sensors as inputs and that output the actions for the agent to take.

Evolutionary Algorithms

Both EAs perform neuroevolution of controllers for the player agent.

Genetic Algorithm For the simple EA, we used a self-designed EA, which we will refer to as "Genetic Algorithm" (GA) in this paper for simplicity sake. The GA evolves a population weights that plug into a fixed-topology neural network consisting of 20 inputs, followed by a fully connected hidden layer of 50 neurons, followed by a fully connected output layer of 5 neurons.

The initial population μ is generated with random values for each of the network's weights. At each generation: first, λ pairs of parents are selected via k-tournaments with $k = 2$; second, one child is generated per pair using whole arithmetic recombination, and each weight of the child has a 20% chance of being mutated with the increment of a value taken from a normal distribution with a mean of 0 and a standard deviation of 1; finally, a pool with $\mu + \lambda$ is formed, from which μ survivors are selected via fitness proportionate selection.

NeuroEvolution of Augmenting Topologies For the complex EA, we utilized NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen, 2002) - we consider it more complex because it evolves not only the weights but also the topology of neural networks. To achieve this goal, every individual starts off as a simple perceptron, and may grow into a more complex network. It makes use of speciation to protect topological innovation. Networks receive 20 inputs and have its final layer with 5 outputs.

Reinforcement Learning Algorithms

Both RLAs perform deep Reinforcement Learning.

Deep Q-Networks For the simple RLA, Deep Q-Networks (DQNs) were used, introduced by Mnih et al. (2013). DQNs are a Neural Network extension to Q-learning, with the high level idea to make Q-Learning prob-

lem look like a supervised learning problem. It employs two important ideas for stabilising Q-learning.

- Use a replay buffer, which stores a large amount of state transitions, which mini-batches can be sampled from and trained on.
- A secondary copy of the NN is kept and updated less frequently which is used to compute the target values. This is to keep the target function from changing too quickly, and avoid chasing a moving target.

The topology of the used network consists of the 20 inputs, followed by a fully connected layer of 64 neurons, followed by a fully connected layer of 32 (2^5) output neurons. All neurons in the network use ReLU activation. Each output neuron corresponds to a set of actions.

Proximal Policy Optimization For the complex RLA, we used Proximal Policy Optimization (PPO), introduced by Schulman et al. (2017). PPO is similar to TRPO in that it uses a trust region to avoid making too large of an update to the network at any one update, to avoid taking a bad step, which could ruin any further gathered data. It does this by clipping the commonly used objective function:

$$\hat{E}_t \left[r_t(\theta) \hat{A}_t \right] \tag{4}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$, such that

$$L(\theta) = \hat{E}_t \left[\min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), t-\epsilon, 1+\epsilon)\hat{A}_t) \right] \tag{5}$$

where ϵ is a hyperparameter. This removes the incentive to move r_t outside of the interval $[1-\epsilon, 1+\epsilon]$ (Schulman et al., 2017). The advantage $\hat{A}_t$ is calculated as:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + ... + (\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{6}$$

$$where \quad \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{7}$$

Besides using this clipped objective function, PPO also uses N (parallel) actors to gather data and accumulated updates (Mnih et al., 2016) - therefore, we consider it more complex than DQN. This improves training stability by exploring different parts of the environment at the same time. The optimisation of L is performed using Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions (Kingma and Ba, 2014).

The topology of the Actor network consists of the 20 inputs, followed by a fully connected layer of 64 neurons, which is fully connected to 5 output neurons[4]. The topology of the Critic network mirrors that of the Actor network, but it only has a single output neuron, which gives the expected value of the current state. All neurons in both networks use Tanh activation.

[4]Note that PPO needs fewer outputs than DQN because PPO supports continuous outputs. This is relevant because the agent can take simultaneous actions.

Experimental setup

Each of the four algorithms was trained on each of the four environments twice: once *with* noise and once *without* noise. At every 2.5×10^4 timesteps (1 generation - or stage), the progress was assessed by pausing the learning process and evaluating the agent for 25 episodes. For the EAs, the best performing individual from the population was used as the policy for this assessment. These 25 values were then averaged to get the performance of the agent for that stage/generation, and these are the values shown in the plots of the next section. The experiments were repeated 50 times (runs) independently for each algorithm.[5]

Results and Discussion

Figures 2 and 3 show comparisons among the final policies of each experiment. Figure 4 shows the results of the policies throughout the learning stages/generations. The main observations are I) The introduction of noise had a dramatically negative impact on both EAs in every environment, whilst the impact on the RLAs was either insignificant or much milder - the performance of EAs dropped severely in the face of noise while the performance of the RLAs was much less affected (Fig. 2); II) DQN performs poorly in all environments in comparison to the other algorithms. This is not surprising, since it has been shown before that DQN takes an extremely high number of steps (1×10^7 to 4×10^7) to find a solution better than random for the benchmark ALE (Schulman, 2017).

Environments without noise

AirMan As expected from the easy nature of this environment, all algorithms (but DQN) quickly learn how to beat AirMan consistently. NEAT evolves a strong policy after around 20 generations, and PPO steadily learns over time and manages to beat the enemy consistently after approximately 2×10^5 timesteps. Thereafter, PPO slowly improves its score over time to approach the GA's and NEAT's performance. GA does extremely well right from the beginning of the evolutionary process, and we hypothesise this is because in such a simple environment, out of a hundred random initialisations for the first generation, it is likely that one of them happens to start out with good parameters. This is unlikely to be the case for the RLs since they have a single solution per run, and unlikely to be the case for NEAT because all NEAT individuals start off as a single-layer perceptron.

[5]All the code to reproduce the experiments can be found here.

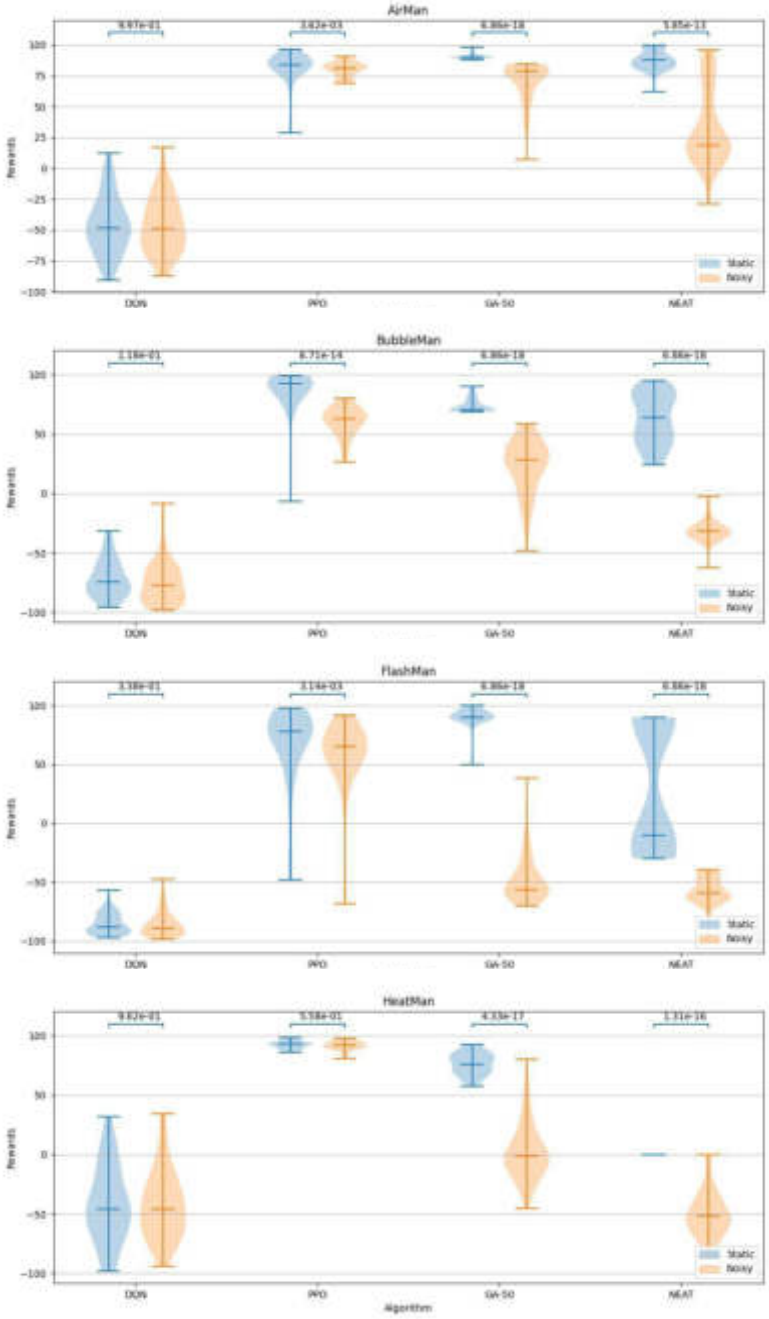


Figure 2: Rewards of final policies averaged among the independent runs - comparison between environments *with noise* (Noisy) and *without noise* (Static). A reward value above 0 means the amount of player-energy left when the player won, and a value below 0 means the amount of enemy-energy left when the player lost. For the EAs, the rewards of all individuals in the final population within each run were averaged to obtain the 'final policy' reward of the run. The plots are annotated with $p - values$ resulting from Wilcoxon Rank-Sums tests.

BubbleMan In the BubbleMan environment we consistently see a steady increase of performance until leveling out just under the maximum reward of 100 for PPO. GA once again starts off well, but levels out below the performance of PPO. It seems to be incapable of evolving past this point, possibly explained by the fewer hidden neurons it has compared to PPO. NEAT performed the worst out of these three algorithms during the time provided, though it

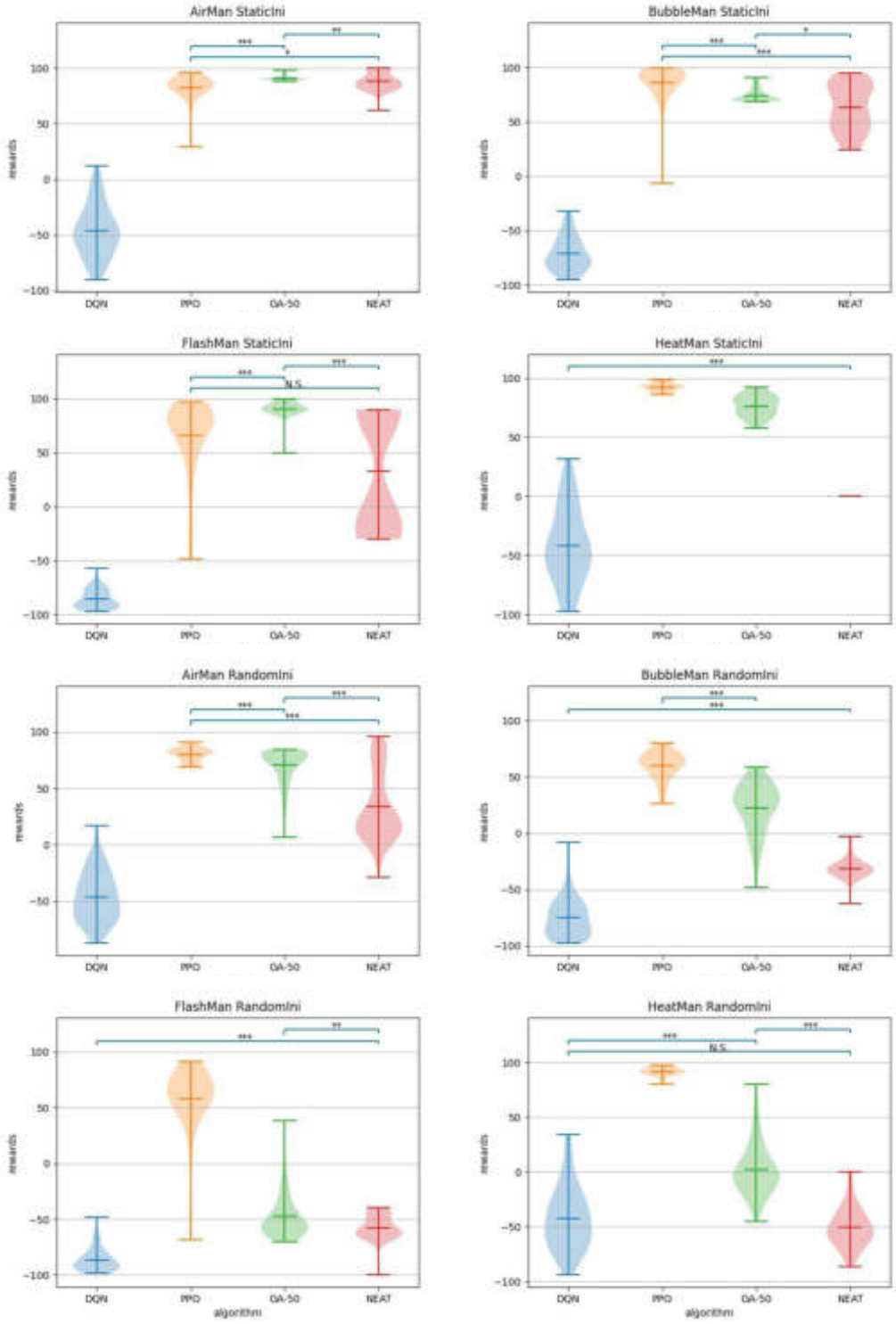


Figure 3: Rewards of final policies averaged among the independent runs for the environments *without noise* (StaticIni) and *with noise* (RandomIni). Definition of 'rewards' from Figure 2 applies. The plots are annotated with significance markers calculated with Wilcoxon Rank-Sums tests. N.S. = Not significant, $* = p < 0.05$, $** = p < 0.01$, $*** = p < 0.001$. Unlabeled means $p < 10^{-15}$.

is clearly still improving at the time of cut-off, so it likely would have reached a better score if allowed to run until convergence. NEAT is also by far the least consistent, which is likely explained by its lack of a complex topology starting out. Due to this is it reliant on chance to find a decent topology before it is actually capable of evolving good weights for the topologies it evolves.

FlashMan In the FlashMan environment, PPO learns a bit slower than in the AirMan and BubbleMan environments, but ends up consistently learning a good policy within 10^6 timesteps. It is also still improving at the cut-off time, so it is likely it could achieve even higher scores if allowed to run until convergence. GA also manages to consistently evolve a good policy. NEAT shows a very large IQR, with the 50th percentile towards the lower end of the search. This appears at first overall low performance, but in fact, 24 of the 50 runs actually reach a final score of 90 (Fig. 3). This suggests that there is a particular topological change that is essential to reaching a good score, and unless that connection is evolved, a good score can not be achieved in this environment. GA and PPO do not suffer from this limitation as they start off with multiple hidden neurons, and only have to find the correct weights for reaching a good score.

HeatMan In HeatMan we observe a very clear difference between the different algorithms. We can see that PPO very easily and consistently learns an almost ideal policy, and reaches convergence at approximately 10^6 timesteps. GA steadily evolves over time, and whilst consistently beating HeatMan, it does not seem to reach convergence within 100 generations. HeatMan is the only environment where DQN actually improves over time, but after 1.1×10^6 timesteps it regresses in performance. We are not sure why this happens. NEAT evolves to a reward of 0 and gets stuck there. After inspecting the amount of time the evaluated episodes ran for, we see all of them ran until the environment expired (1500 timesteps). This indicates that NEAT consistently evolved avoidance behaviour and got stuck in this policy.

Environments with noise

AirMan In the environment with random initial positions, all algorithms but DQN consistently found a policy that beats AirMan, however, they are less consistent with their score than in the case of static initial positions (higher variance among runs). GA finds a good policy from the start, but does not improve much thereafter. NEAT finds a policy to beat AirMan consistently after approximately 30 generations, but mostly stagnates after this, and does not reliably find a policy with a reward higher than 20. PPO improves steadily during training and whilst seemingly still improving at the cut-off point, gains significantly better rewards after the allotted time-frame.

BubbleMan Against BubbleMan, NEAT fails to find a policy sufficient for winning. This could be explained by a lack of neurons evolving to be able to determine when the agent should interrupt the jump. GA evolves a policy that wins from BubbleMan most of the time. This could be explained in two ways. Either GA takes most of its damage from the projectiles shot by, or contact with, BubbleMan, or GA avoids the spikes a bit over half the time and jumps into the spikes during the other evaluations. PPO once again steadily and reliably improves over time, and nearing the end of the training period finds a policy that is capable of defeating BubbleMan consistently.

FlashMan In the FlashMan environment, PPO is the one making any progress on learning the environment. NEAT evolves up to 20-40 generations, but this seems it is just learning to shoot in the right direction and not really avoiding and attacking the enemy. GA does not make any progress whatsoever and appears to just be stuck with policies that as effective as random button pressing.

HeatMan PPO finds an almost optimal policy against HeatMan quickly and consistently. Besides that, the results are very similar to that of BubbleMan.

Conclusion

The RLAs presented significantly greater capacity to deal with noise in all of the game environments than the EAs. One possible explanation for this is that: credit assignment is allowed in RLAs, therefore, the usefulness of each action in the face of a particular state is taken into consideration regardless of the success of the policy in the upcoming states; with the EAs on the other hand, if the actions taken at the beginning of the episode are inappropriate, it does not matter if the policy would perform well along most of the rest of the episode because the policy never arrives at the steps where it could show its strength. Furthermore, if a solution has good performance but its offspring is tested with an initial environmental condition that the parent could not cope with, this offspring will have poor performance. This may create strong selection pressure for solutions that "work well" on every starting position, leading the search to get stuck into a mediocre "do it all" local optimum.

It is noteworthy that though all four algorithms have been trained on the same amount of data, both EAs were trained in much less time than the RLAs. The RLAs needed about $10\times$ the amount of time that the EAs needed[6]. This means that for the cases in which efficacy was similar for RLAs and EAs, the efficiency of the EAs was higher.

The current study has experimented with noise only in the initialisation of the environments, and using flawless sen-

[6]This is related to differences in the nature of the two classes of algorithms investigated, e.g., calculating gradients can be more expensive than mutation operations.

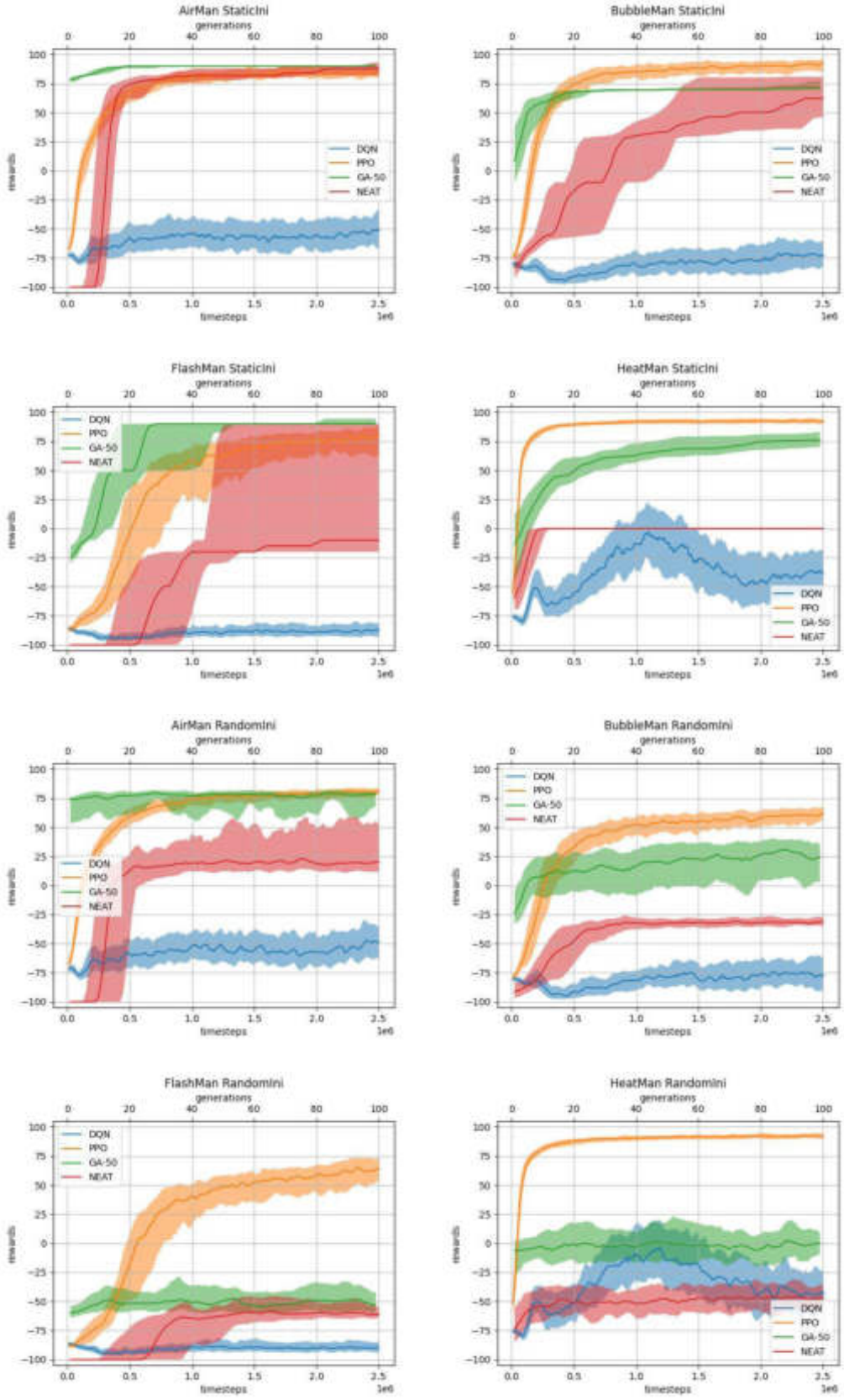


Figure 4: Rewards of policies across the stages/generations averaged among the independent runs for the environments *without noise* (StaticIni) and *with noise* (RandomIni). Lines are medians and shades are first and third quartiles. Definition of 'rewards' from Figure 2 applies.

sors. For future work, it would be interesting to investigate what happens with environments that suffer from noise all along the episode, with the use of noisy sensors, and with high-dimensional problems.

Acknowledgements

This research was funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research[7], grant number 024.004.022.

References

Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., and Blundell, C. (2020). Agent57: Outperforming the atari human benchmark. *CoRR*, abs/2003.13350.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

Crespo, J. and Wichert, A. (2020). Reinforcement learning applied to games. *SN Applied Sciences*, 2(5):824.

da Silva Miras de Araújo, K. and de França, F. O. (2016). An electronic-game framework for evaluating coevolutionary algorithms. *CoRR*, abs/1604.00644.

da Silva Miras de Araujo, K. and de Franca, F. O. (2016). Evolving a generalized strategy for an action-platformer video game framework. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1303–1310.

Drugan, M. M. (2019). Reinforcement learning versus evolutionary computation: A survey on hybrid algorithms. *Swarm and evolutionary computation*, 44:228–246.

Givigi, S. N., Schwartz, H. M., and Lu, X. (2010). A reinforcement learning adaptive fuzzy controller for differential games. *Journal of Intelligent and Robotic Systems*, 59(1):3–30.

Hausknecht, M., Lehman, J., Miikkulainen, R., and Stone, P. (2014). A neuroevolution approach to general atari game playing. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(4):355–366.

Ishikawa, F., Trovões, L. Z., Carmo, L., França, F. O. d., and Fantinato, D. G. (2020). Playing mega man ii with neuroevolution. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2359–2364.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeBlanc, D. G. and Lee, G. (2021). General deep reinforcement learning in nes games.

[7] https://www.hybrid-intelligence-centre.nl

Lucas, S. M. (2008). Computational intelligence and games: Challenges and opportunities. *International Journal of Automation and Computing*, 5(1):45–57.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *CoRR*, abs/1602.01783.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Moriyama, K., Branco, S. E. O., Matsumoto, M., Fukui, K.-i., Kurihara, S., and Numao, M. (2014). An intelligent fighting videogame opponent adapting to behavior patterns of the user. *IEICE TRANSACTIONS on Information and Systems*, 97(4):842–851.

Murphy, T. (2013). The first level of super mario bros. is easy with lexicographic orderings and time travel... after that it gets a little tricky.

Promsutipong, P. and Kotrajaras, V. (2017). Enemy evaluation ai for 2d action-platform game. In *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6.

Rieser, V., Robinson, D. T., Murray-Rust, D., and Rounsevell, M. (2011). A comparison of genetic algorithms and reinforcement learning for optimising sustainable forest management. In *Proc. 11th Int. Conf. GeoComput.*, pages 20–24.

Schulman, J. (2017). *Deep Reinforcement Learning Bootcamp Lecture 6: Nuts and Bolts of Deep RL Experimentation*. AI Prism.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.

Sutton, R. S. (1992). *Introduction: The Challenge of Reinforcement Learning*, pages 1–3. Springer US, Boston, MA.

Taylor, M. E., Whiteson, S., and Stone, P. (2006). Comparing evolutionary and temporal difference methods in a reinforcement learning domain. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1321–1328.

Voluntary safety pledges overcome over-regulation dilemma in AI development: an evolutionary game analysis

The Anh Han[1], Francisco C. Santos[2], Luís Moniz Pereira[3], Tom Lenaerts[4,5]
[1] School of Computing, Engineering and Digital Technologies, Teesside University (Email: t.han@tees.ac.uk)
[2]INESC-ID and Instituto Superior Técnico, Universidade de Lisboa
[3] NOVA Laboratory for Computer Science and Informatics (NOVA LINCS), Universidade Nova de Lisboa
[4] Machine Learning Group, Université Libre de Bruxelles
[5] Artificial Intelligence Lab, Vrije Universiteit Brussel

Abstract

With the introduction of Artificial Intelligence (AI) and related technologies in our daily lives, fear and anxiety about their misuse, as well as the hidden biases in their creation, have led to a demand for regulation to address such issues. Yet, blindly regulating an innovation process that is not well understood may stifle this process and reduce benefits that society might gain from the generated technology, even under the best of intentions. Starting from a baseline game-theoretical model that captures the complex ecology of choices associated with a race for domain supremacy using AI technology, we show that socially unwanted outcomes may be produced when sanctioning is applied unconditionally to risk-taking, i.e., potentially unsafe behaviours. As an alternative to resolve the detrimental effect of over-regulation, we propose a voluntary commitment approach, wherein technologists have the freedom of choice between independently pursuing their course of actions or else establishing binding agreements to act safely, with sanctioning of those that do not abide to what they have pledged. Overall, our work reveals for the first time how voluntary commitments, with sanctions either by peers or by an institution, leads to socially beneficial outcomes in all scenarios that can be envisaged in the short-term race towards domain supremacy through AI technology.

Introduction

Rapid technological advancements in Artificial Intelligence (AI), together with the growing deployment of AI in new application domains such as robotics, face recognition, self-driving cars, genetics, are generating an anxiety which makes companies, nations and regions think they should respond competitively (Armstrong et al., 2016; Baum, 2017; Bostrom, 2017; Cave and ÓhÉigeartaigh, 2018; Lee, 2018). AI appears for instance to have instigated a race among chip builders, simply because of the requirements it imposes on the technology. Governments are furthermore stimulating economic investments in AI research and development as they fear of missing out, resulting in a racing narrative that increases further the anxiety among stake-holders (AI-Roadmap-Institute, 2017; Cave and ÓhÉigeartaigh, 2018; Apps, 2019).

Races for supremacy in a domain through AI may however have detrimental consequences since participants to the race may well ignore ethical and safety checks in order to speed up the development and reach the market first. AI researchers and governance bodies, such as the EU, are urging to consider together both the normative and the social impact of major technological advancements concerned (Declaration, 2018; Jobin et al., 2019; European Commission, 2020; Future of Life Institute, 2019). However, given the breadth and depth of AI and its advances, it is not an easy task to assess when and which AI technology in a concrete domain needs to be regulated. This issue was, among others, highlighted in the recent EU White Paper on AI (European Commission, 2020) an the UK National AI strategy.

Several proposals for mechanisms on how to avoid, mediate, or regulate the development and deployment of AI, have been made (Baum, 2017; Cave and ÓhÉigeartaigh, 2018; Geist, 2016; Shulman and Armstrong, 2009; Han et al., 2019; Vinuesa et al., 2020; Nemitz, 2018; Taddeo and Floridi, 2018; Askell et al., 2019; O'Keefe et al., 2020; Cimpeanu et al., 2022). Essentially, regulatory measures such as restrictions and incentives are proposed to limit harmful and risky practices in order to promote beneficial designs (Baum, 2017). Examples include financially supporting the research into beneficial AI (McGinnis, 2010) and making AI companies pay fines when found liable for the consequences of harmful AI (Gurney, 2013).

Although such regulatory measures may provide solutions for particular scenarios, one needs to ensure that they do not overshoot their targets, leading to a stifling of novel innovations, hindering investments into the development into novel directions as they may be perceived to be too risky (Hadfield, 2017; Lee, 2018). Worries have been expressed by different organisations and academic societies that too strict policies may unnecessarily affect the benefits and societal advances that novel AI technologies may have to offer (EDRI, 2021). Regulations affect moreover big and small tech companies differently: A highly regulated domain makes it more difficult for small new start-ups, introducing an inequality and dominance of the market by a few big players (Lee, 2018). It has been emphasised that neither over-regulation nor a laissez-faire approach suffices

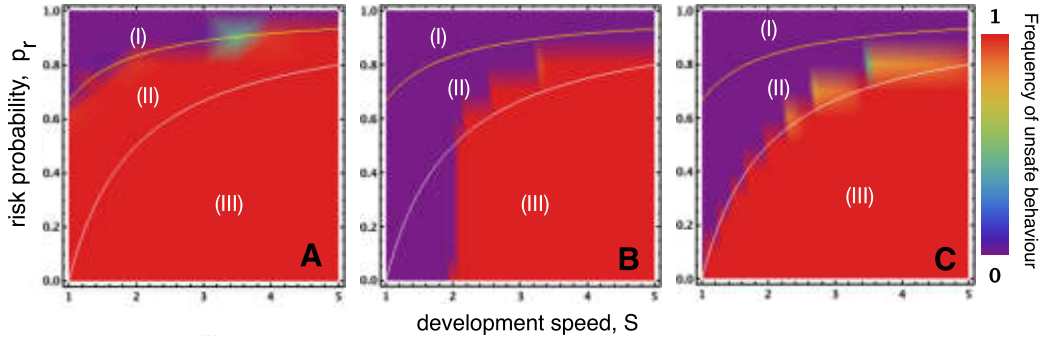


Figure 1: Frequency of unsafe behaviour as a function of development speed (s) and the disaster risk (p_r). **Panel A, in absence of incentives** (Han et al., 2020), the parameter space can be split into three regions. In regions (I) and (III), safe and unsafe/innovation, respectively, are the preferred collective outcome also selected by natural selection, thus no regulation being required. Region (II) requires regulation as safe behaviour is preferred but not the one selected. **Panel B, when unsafe behaviour is sanctioned unconditionally** (Han et al., 2021), while unsafe behaviour is reduced in region II, over-regulation occurs in region III, reducing beneficial innovation. **Panel C, unsafe behaviour is sanctioned only in presence of a voluntary commitment** (Han et al., 2022), unsafe behaviour is significantly reduced in region II while avoiding over-regulation.

when aiming to regulate AI technologies (Dawson et al., 2019). In order to find a balanced answer, one clearly needs to have first an understanding of how a competitive development dynamic actually could work and how governance choices impact this dynamic, a task well-suited for dynamic systems or agent-based models.

Here, we highlight main results from our recent work (Han et al., 2022) examining this problem theoretically, using methods from Evolutionary Game Theory (Sigmund, 2010), see Figure 1. It resorts to a baseline model describing a development competition where technologists can choose a safe (SAFE) vs risk-taking (UNSAFE) course of development (Han et al., 2020). Namely, it considers that to reach domain supremacy through AI in a certain domain, a number of development steps or technological advancement rounds are required (Han et al., 2020). In each round the technologists (or players) need to choose between one of two strategic options: to follow safety precautions (the SAFE action) or ignore safety precautions (the UNSAFE action). Because it takes more time and more effort to comply with precautionary requirements, playing SAFE is not just costlier, but implies slower development speed too, compared to playing UNSAFE. Moreover, there is a probability that a disaster occurs if UNSAFE developments take place during this competition (see (Han et al., 2020) for a full description).

We first demonstrate that unconditional sanctioning will negatively influence social welfare in certain conditions of a short-term race towards domain supremacy through AI technology (Han et al., 2021), leading to over-regulation of beneficial innovation (see Figure 1B). Since data to estimate the risk of a technology is usually limited (especially at an early stage of its development or deployment), simple sanctioning of unsafe behaviour (or reward of safe behaviour) could not fully address the issue.

To solve this critical over-regulation dilemma in AI development, we propose an alternative approach (Han et al., 2022), which is to allow technologists or race participants to voluntarily commit themselves to safe innovation procedures, signaling to others their intentions (Han et al., 2015; Nesse, 2001; Han, 2022). Specifically, this bottom-up, binding agreement (or commitment) is established for those who want to take a safe choice, with sanctioning applied to violators of such an agreement. It is shown that, by allowing race participants to freely pledge their intentions and enter (or not) in bilateral commitments to act safely and avoid risks, accepting thus to be sanctioned in case of misbehavior, high levels of the most beneficial behaviour, for the whole, are achieved in all regions of the parameter space, see Figure 1C. These results are directly relevant for the design of self-organized AI governance mechanisms and regulatory policies that aim to ensure an ethical and responsible AI technology development process.

Acknowledgements

This work was supported by a Future of Life Institute AI grant (RFP2-154).

References

AI-Roadmap-Institute (2017). Report from the ai race avoidance workshop, tokyo.

Apps, P. (2019). Are China, Russia winning the AI arms race? [Reuters; Online posted 15-January-2019].

Armstrong, S., Bostrom, N., and Shulman, C. (2016). Racing to the precipice: a model of artificial intelligence development. *AI & society*, 31(2):201–206.

Askell, A., Brundage, M., and Hadfield, G. (2019). The Role of Cooperation in Responsible AI Development. *arXiv preprint arXiv:1907.04534*.

Baum, S. D. (2017). On the promotion of safe and socially beneficial artificial intelligence. *AI & Society*, 32(4):543–551.

Bostrom, N. (2017). Strategic implications of openness in AI development. *Global Policy*, 8(2):135–148.

Cave, S. and ÓhÉigeartaigh, S. (2018). An AI Race for Strategic Advantage: Rhetoric and Risks. In *AAAI/ACM Conference on Artificial Intelligence, Ethics and Society*, pages 36–40.

Cimpeanu, T., Santos, F. C., Pereira, L. M., Lenaerts, T., and Han, T. A. (2022). Artificial intelligence development races in heterogeneous settings. *Scientific Reports*, 12(1):1–12.

Dawson, D., Schleiger, E., Horton, J., McLaughlin, J., Robinson, C., Quezada, G., Scowcroft, J., and S, H. (2019). Artificial Intelligence: Australia's Ethics Framework. Technical report, Data61 CSIRO, Australia.

Declaration, M. (2018). The montreal declaration for the responsible development of artificial intelligence launched. https://www.canasean.com/the-montreal-declaration-for-the-responsible-development-of-artificial-intelligence-launched/.

EDRI (2021). Civil society calls for AI red lines in the European Union's Artificial Intelligence proposal. Technical report, European Commission. Accessed January-29-2021.

European Commission (2020). White paper on Artificial Intelligence – An European approach to excellence and trust. Technical report, European Commission.

Future of Life Institute (2019). Lethal autonomous weapons pledge. https://futureoflife.org/lethal-autonomous-weapons-pledge/.

Geist, E. M. (2016). It's already too late to stop the ai arms race: We must manage it instead. *Bulletin of the Atomic Scientists*, 72(5):318–321.

Gurney, J. K. (2013). Sue my car not me: Products liability and accidents involving autonomous vehicles. *U. Ill. JL Tech. & Pol'y*, page 247.

Hadfield, G. K. (2017). *Rules for a flat world: why humans invented law and how to reinvent it for a complex global economy*. Oxford University Press.

Han, T. A. (2022). Institutional incentives for the evolution of committed cooperation: ensuring participation is as important as enhancing compliance. *Journal of The Royal Society Interface*, 19(188):20220036.

Han, T. A., Lenaerts, T., Santos, F. C., and Pereira, L. M. (2022). Voluntary safety commitments provide an escape from over-regulation in ai development. *Technology in Society*, 68:101843.

Han, T. A., Pereira, L. M., and Lenaerts, T. (2019). Modelling and Influencing the AI Bidding War: A Research Agenda. In *Proceedings of the AAAI/ACM conference AI, Ethics and Society*, pages 5–11.

Han, T. A., Pereira, L. M., Lenaerts, T., and Santos, F. C. (2021). Mediating Artificial Intelligence Developments through Negative and Positive Incentives. *PLOS ONE*, 16(1):e0244592.

Han, T. A., Pereira, L. M., Santos, F. C., and Lenaerts, T. (2020). To Regulate or Not: A Social Dynamics Analysis of an Idealised AI Race. *Journal of Artificial Intelligence Research*, 69:881–921.

Han, T. A., Santos, F. C., Lenaerts, T., and Pereira, L. M. (2015). Synergy between intention recognition and commitments in cooperation dilemmas. *Scientific reports*, 5(9312).

Jobin, A., Ienca, M., and Vayena, E. (2019). The global landscape of AI ethics guidelines. *Nature Machine Intelligence*, pages 1–11.

Lee, K.-F. (2018). *AI superpowers: China, Silicon Valley, and the new world order*. Houghton Mifflin Harcourt.

McGinnis, J. O. (2010). Accelerating AI. *Nw. UL Rev.*, 104:1253.

Nemitz, P. (2018). Constitutional democracy and technology in the age of artificial intelligence. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2133):20180089.

Nesse, R. M. (2001). *Evolution and the capacity for commitment*. Foundation series on trust. Russell Sage.

O'Keefe, C., Cihon, P., Garfinkel, B., Flynn, C., Leung, J., and Dafoe, A. (2020). The windfall clause: Distributing the benefits of ai for the common good. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 327–331.

Shulman, C. and Armstrong, S. (2009). Arms control and intelligence explosions. In *7th European Conference on Computing and Philosophy (ECAP), Bellaterra, Spain, July*, pages 2–4.

Sigmund, K. (2010). *The Calculus of Selfishness*. Princeton University Press.

Taddeo, M. and Floridi, L. (2018). Regulate artificial intelligence to avert cyber arms race. *Nature*, 556(7701):296–298.

Vinuesa, R., Azizpour, H., Leite, I., Balaam, M., Dignum, V., Domisch, S., Felländer, A., Langhans, S., Tegmark, M., and Nerini, F. F. (2020). The role of artificial intelligence in achieving the sustainable development goals. *Nature Communications*, 11(233).

On the Trajectories of Planetary Civilizations: Asymptotic Burnout vs. Homeostatic Awakening

Michael L. Wong[1] and Stuart Bartlett[2]

[1] Earth and Planets Laboratory, Carnegie Institution for Science, Washington, DC 20015, USA
[2] Division of Geological and Planetary Sciences, California Institute of Technology, Pasadena, CA 91125, USA
mwong@carnegiescience.edu

Abstract

Previous studies show that city metrics having to do with growth, productivity, and overall energy consumption scale superlinearly, attributing this to the social nature of cities. Superlinear scaling results in crises called "singularities," where population and energy demand tend to infinity in a finite amount of time, which must be avoided by ever more frequent "resets" or innovations that postpone the system's collapse. Here, we place the emergence of cities and technological civilizations in the context of major evolutionary transitions. With this perspective, we hypothesize that once a planetary civilization transitions into a state that can be described as one virtually connected global city, it will face an "asymptotic burnout," an ultimate crisis where the singularity-interval timescale becomes smaller than the timescale of innovation. If a civilization develops the capability to understand its own trajectory, it will have a window of time to affect a fundamental change to prioritize long-term homeostasis and well-being over unyielding growth—a consciously induced trajectory change or "homeostatic awakening." We propose a new resolution to the Fermi paradox: civilizations either collapse from burnout or redirect themselves to prioritizing homeostasis, a state where cosmic expansion is no longer a goal, making them difficult to detect remotely.

Introduction

The evolution of life has been characterized as a series of "major transitions" in units of selection, information processing, and energy transduction (e.g., Szathmáry & Maynard-Smith 1995; Judson 2017). These transitions are not limited strictly to biological evolution but can also be extended to encapsulate advancements of human society, culture, and the dataome (Scharf 2021).

Bettencourt et al. (2007) offered a quantitative explanation for the accelerating pace of innovations, specifically in the development of cities. They found that city metrics having to do with growth, productivity, and overall energy consumption obey scaling laws where the scaling exponent $\beta > 1$ (unlike in purely biological systems, where $\beta < 1$) and attribute this to the *social* nature of cities. Systems where $\beta > 1$ will trend towards crises called "singularities," where population and energy demand tend to infinity in a finite amount of time. For any chance of long-term survival, these singularities must be avoided by "resets," which correspond to innovations that postpone the system's collapse. Singularities can be avoided so long as the timescale between singularities, t_{cycle}, is greater than the timescale of innovation, $t_{\mathrm{innovation}}$. However, the cadence of the unavoidable singularities and necessary resets increases in frequency over time.

Civilization Burnout

Scharf (2021) defines the "dataome" as the recording and processing of information that life performs external to its biology. The dataome encompasses books, architecture, computers, etc., as well as the coevolution of those infological organisms atop of a collection of biological organisms. Due to how deeply intertwined the dataome and human biome have become, it is possible that we are in the midst of another major informational phase transition: one that pushes civilization into a state where the physical colocation of humans in cities is no longer the dominant constraint on human interaction. Because it is human interactions that appear to give rise to the $\beta > 1$ scaling laws of cities, if civilization transitions into a state that can be described as one virtually connected globalized city, it is likely that such an organizational structure will exist in the same universality class as cities. In other words, we conjecture that a technologically connected civilization's productivity, growth, and resource consumption would be characterized by scaling laws with a scaling exponent $\beta > 1$ (Wong & Bartlett 2022).

As advances in artificial intelligence are made, it is also possible that human–human interactions may become less important than human–technology, and eventually technology–technology interactions. While human–human interactions are constrained by time and cognitive capacity, the ability for technological agents to interact with one another in an ever-growing digital planetary network could be boundless. What effect these near-future shifts may have on the scaling coefficient of a globalized city remains speculative, but we find it plausible that such transitions could result in an even larger β.

Like cities, planetary civilizations may naturally set themselves on trajectories towards singularities and experience an ultimate crisis that we call "asymptotic burnout," where the singularity-interval timescale becomes smaller than the timescale of innovation. Additionally, as information processing and free energy–harnessing capabilities grow, the magnitude of a civilization's internal fluctuations also increases; examples of past, current, and future fluctuations driven by free energy and informational expansions include: the oxygenation of the atmosphere, anthropogenic climate change, nuclear warfare, and a "disinformation catastrophe." Thus, with time, the singularity timescale, t_{cycle}, decreases while potentially harmful fluctuations become more likely to derail innovations. Once

t_{cycle} becomes short enough that internal fluctuations or external perturbations can cause $t_{innovate} > t_{cycle}$ with some non-negligible probability, collapse/regression may be inevitable.

Homeostatic Awakening & Reorientation

A civilization may have a window of time, Δt_{window}, between when they develop the capability to understand their own trajectory and when they reach burnout (Fig. 1). During this potentially slim window of opportunity, perhaps a civilization can affect a fundamental change to prioritize long-term homeostasis and well-being over unyielding growth and cycles of necessary innovation—a consciously induced trajectory change that we call "homeostatic awakening." A homeostatic reorientation would require rewriting of the fabric of global civilization so that unbounded growth is no longer the priority, or at least no longer the outcome (Wong & Bartlett 2022).

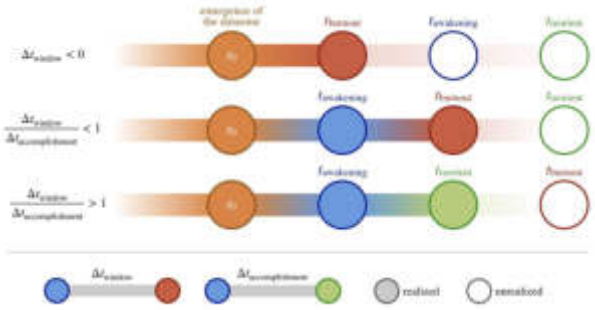


Figure 1: Three scenarios for the evolution of a civilization on a burnout trajectory. *Top*: $\Delta t_{window} < 0$. In this scenario, the civilization does not realize its trajectory before it suffers from burnout. *Middle*: $\Delta t_{window} > 0$, but $\Delta t_{window}/\Delta t_{accomplishment} < 1$. In this scenario, the civilization realizes it is on a trajectory but is unable to accomplish a reorientation towards homeostasis before burnout. *Bottom*: $\Delta t_{window} > 0$, and $\Delta t_{window}/\Delta t_{accomplishment} > 1$. In this scenario, the civilization is able to both understand that it is on a burnout trajectory and is able to reorient towards prioritizing homeostasis.

Reasons for optimism include: historical "mini-awakenings" (e.g., the banning of CFCs); non-expansionist national policies (e.g., Bhutan's policy of maximizing "Gross National Happiness" instead of gross domestic product); and previous evolutionary transitions (e.g., the emergence of regulatory mechanisms in cell-to-cell communication that allow cells to cooperate towards organ-level and organism-level homeostasis). Self-awareness-driven reprioritization towards homeostasis may be the next transcendence that life takes (or must take) after civilization as we know it (Frank et al. 2022).

Implications for the Fermi Paradox

We propose a new "resolution" to the Fermi paradox: the reason we do not observe a galaxy teeming with evidence of extraterrestrial civilizations is that civilizations either collapse from burnout or redirect themselves to prioritizing homeostasis, a state where cosmic expansion is no longer a goal, making them difficult to detect remotely (Figure 2). If the burnout–awakening hypothesis does indeed describe the fate of civilizations across the cosmos, then the lifetime of planetary civilizations may have a *bimodal* distribution.

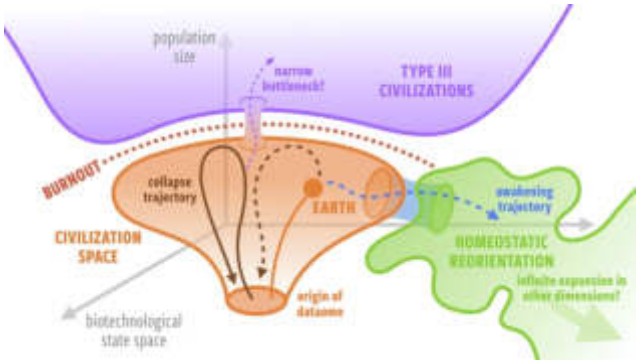


Figure 2: Perhaps hypothetical "Type III" civilizations are in an inaccessible (or difficult to access) region of biotechnological–population size state space because civilization trajectories are bounded by a "burnout horizon" and long-lived civilizations have consciously reoriented their trajectories away from growth in population size and length scales to explore other dimensions of biotechnological state space.

The burnout–awakening hypothesis does not preclude the remote detection of exo-civilizations via planetary-scale technosignatures. In fact, civilizations that are near burnout may be the *most* detectable exo-civilizations, as they would be altering their environments and dissipating free energy in a wildly unsustainable manner—fluctuations on the planetary scale that exhibit the largest signal-to-noise. This presents the possibility that a good many of humanity's initial detections of extraterrestrial life may be of the *intelligent*, though not yet *wise*, kind. Observing such burnouts (provided humanity is long-lived enough to do so) would provide potential confirmation of part of our hypothesis. On the other hand, persistent civilizations that transition through homeostatic awakening may be difficult or impossible to detect.

Conclusions

We have outlined a hypothesis that planetary civilizations, virtually connected by their dataomes, may grow along trajectories toward asymptotic burnout. As burnout approaches, civilizations may attain the cognitive horizon to understand their trajectory and affect a reprioritization towards homeostasis. Either outcome—homeostatic awakening or civilization collapse—would be consistent with the observed absence of Type III civilizations (Wong & Bartlett 2022).

Acknowledgements

We thank Caleb Scharf, Robert M. Hazen, Nadia Drake, Nathalie Cabrol, Ann Marie Cody, Jill Tarter, and Desun Oka for insightful discussions.

References

Bettencourt, L. M., Lobo, J., Helbing, D., Kühnert, C., & West, G. B. (2007). Growth, innovation, scaling, and the pace of life in cities. *PNAS*, 104(17), 7301-7306.

Frank, A., Grinspoon, D., & Walker, S. (2022). Intelligence as a planetary scale process. *International Journal of Astrobiology, 21*(2), 47-61. doi:10.1017/S147355042100029X

Judson, O. P. (2017). The energy expansions of evolution. *Nature Ecology & Evolution*, 1(6), 1-9.

Scharf, C. (2021). *The Ascent of Information*. Penguin Publishing Group.

Szathmáry, E., & Smith, J. M. (1995). The major evolutionary transitions. *Nature*, 374(6519), 227-232.

Wong, M. L., & Bartlett, S. (2022). Asymptotic burnout and homeostatic awakening: a possible solution to the Fermi Paradox? *J. R. Soc. Interface* 19: 20220029. https://doi.org/10.1098/rsif.2022.0029

Evolving Unbounded Neural Complexity in Pursuit-Evasion Games

Thomas Willkens[1] and Jordan Pollack[1]

[1] Dynamical and Evolutionary Machine Organization Lab
Brandeis University, Waltham, MA, USA
twillkens@brandeis.edu, pollack@brandeis.edu

Abstract

We study the conditions in which the unbounded growth of complexity – measured in terms of expressed genome size – can be observed in coevolving populations of neural agents involved in different classes of interactions. To reproduce the results of prior work on the dynamics of open-ended evolution, we introduce a simple pursuit-evasion scenario that allows for the development of increasingly intricate strategies. It is shown that for some configurations of our game, fitness-proportionate selection leads to stagnation while more sophisticated coevolutionary methods produce apparently unbounded complexity growth. Analysis of behavioral patterns sheds some light on the evolutionary pressures introduced by the model. Our findings replicate many features of previously reported work; however, we observe particular dynamics that differ in important respects, challenging prior conclusions, creating new opportunities, and highlighting the need for further investigation of this domain.

Introduction

Here we build on the work of Moran and Pollack (2019), which examines the dynamics of artificial organisms coevolving within *ecosystem topologies*, which can be understood as networks of interactions between species. Using deterministic finite state machines and a simple prediction game with cooperative and competitive varieties, they claim to show that purely competitive and purely cooperative ecosystems lead to stable plateaus, while certain combinations of the two spur the apparently unbounded growth of ever-larger genotypes. These findings were cited as a promising research direction in Stanley (2019) and believed to yield some insight into the dynamics of *open-endedness*, an area currently of great interest to the artificial life community (Adams et al., 2016; Soros, 2018; Taylor, 2019; Dolson et al., 2019; Guttenberg et al., 2019; Stepney, 2021).

Their results are intriguing, but much remains unclear. Are the observed dynamics generally applicable to other domains, or specific to those particular representations and methods? What light do they shed on the nature of open-endedness, and what use might they have for further experimentation and application? This study seeks to answer these questions through the replication and examination of such dynamics in a different substrate. We show that the phenomenon of unbounded complexity growth is reproducible using evolved neural networks within an abstract pursuit-evasion scenario. Moreover, ecosystem topologies are shown to play a powerful role in the dynamics of this growth, and preliminary analysis supports their potential to produce complex behavioral patterns.

However, our results challenge key conclusions of the prior work. Dynamics previously believed to be of great importance – such as the driving power of mixed cooperative/competitive interactions, the accelerated growth rates of highly cooperative species, and the degeneracy of purely competitive systems – in fact appear to be domain-dependent. And unlike in the previous work, fitness-proportionate selection (Goldberg and Deb, 1991) leads to mediocre plateaus while the *Discovery of Search Objectives* (DISCO) method (Liskowski and Krawiec, 2017) engenders growth in topologies previously considered stagnant. We conclude that this negative result may ultimately afford greater opportunity and flexibility in the design of unbounded systems.

The Linguistic Prediction Game

In Moran and Pollack (2019), members of different populations are tasked with playing a *linguistic prediction game*, similar to the Iterated Prisoner's Dilemma (Axelrod, 1984). At each timestep, both players simultaneously emit either a 0 or a 1 bit. If the interaction is *cooperative* and the bits match, both players receive a point. However, if the two species are in a *competitive* relationship, their goals are misaligned: One player receives a point if the bits match, while the other receives a point if the bits mismatch. Crucially, neither player knows their relationship with the other and must infer it through the patterns of bits they produce. [1]

Unlike typical two-population competitive coevolution scenarios, here multiple populations may coevolve via a variety of interactions. The overall network of interactions constitutes the *ecosystem topology*. Moran introduces

[1] The game has a symmetric variation: an interaction is also cooperative if both players are rewarded for mismatching their bits.

ecosystems such as *Two-Species Cooperative*, where members of two populations must cooperatively match the other's bits, and *Three-Species Mixed*, in which a "host" species must try to match with a symbiotic species while also dealing with a mismatching parasitic species.

In their experiments, organisms are represented as *deterministic finite state machines* (DFSMs). The value of each state may be either 0 or 1, corresponding to the bit the player emits at that state. Each state has two transition links, labeled 0 and 1, corresponding to the bit emitted by the other player and leading to another state. While the game is infinite horizon, advantage is taken of the discrete nature of the model for evaluation: The simulation ends once a loop is detected, and a final score is calculated for each player by taking their average score among timesteps within the loop.

The first generation of organisms begin with just a single state. After each round of all-versus-all play, asexual reproduction occurs through *fitness-proportionate* selection and mutation, allowing the potential for growth over time. The complexity of each DFSM is measured as its state count after minimization using a variation of *Hopcroft's algorithm* (Hopcroft, 1971). Various ecosystem topologies were analyzed over long evolutionary horizons, yielding a number of salient observations:

1. Ecosystems with solely cooperative or competitive interactions reach stable plateaus in terms of complexity.
2. Purely competitive ecosystems demonstrate the least growth and highest degree of stability.
3. Ecosystems with mixtures of both interactions tend to exhibit rapid, unbounded complexity growth among one or more species.
4. Unbounded growth is most pronounced among species involved in cooperative interactions.
5. Ecosystem *ladders* with many species and interactions appear to ratchet up the velocity of growth.
6. Clusters of mutually competitive interactions tend to stifle growth in more intricate configurations, resulting in *degenerate* ecosystems.

The trends were so persistent that they appeared to be evidence of a more general dynamic. Theoretical justification was left for future work, but Moran hypothesized that high-frequency mutual adaptation between competing populations prompts cooperative species to evolve more complex mechanisms to adapt to the rapid changes in their partner's behavior.

The Collision Game

We introduce a new interaction domain aiming to capture the core dynamics of the linguistic prediction game. The mapping is imperfect, but that is useful for our aim of testing the generality of the predicted dynamics. The *collision game* is a parameterized finite-horizon two-player game that takes place on a real-valued one-dimensional number line. On timestep zero, each agent is assigned a position on the line such that they are some distance apart. In our case, the starting distance is 5.0 units: The *lesser agent* is assigned a starting position of -2.5, while the *greater agent* is given the position 2.5. At every timestep, each agent emits two real-valued outputs between -1.0 and 1.0. The first is a *movement* value, which is summed with the agent's current position value to determine its position at the next timestep, while the second is a *communication* value.

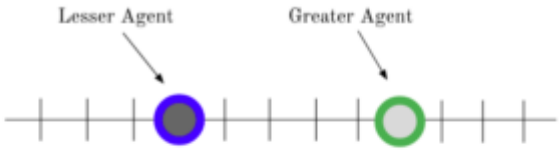


Figure 1: Sample state for agents in the collision game. The dark color of the lesser agent corresponds to a communication output close to 1.0, while the light gray of the greater agent corresponds to communication closer to -1.0.

At the start of the next timestep, each player is provided with the distance between their two respective positions along with the communication value of the other agent. Each then determines new action and communication values, and so on. The episode ends upon one of two conditions: If the position of the lesser agent becomes either greater than or equal to the position of the greater agent, the game is over and the interaction is classified as a *collision*; however, if the agents fail to collide by some specified timestep (here it is timestep 128), it is an *evasion*. Thus our game is somewhat analogous to a mating dance or courtship ritual – two parties begin separated in space and ignorant of the other's qualities; depending on nature of the display, one or both may choose to narrow the distance.

The particular parameter values chosen here for the initial distance, episode length, etc., have some useful features. The episode length is short enough to be computationally tractable, but long enough for nontrivial behavior to develop, while the action space can conveniently be handled by neural networks. With our setup, if two agents take actions uniformly at random from $[-1.0, 1.0]$, then there is a 56% chance they will collide by the end of the episode, which is a useful property for analysis. However, these particular values are ultimately arbitrary, and we hypothesize that different choices could result in significantly different evolutionary landscapes.

The linguistic prediction game and the collision game dif-

	Collision	*Evasion*
Affinitive	(1, 1)	(0, 0)
Avoidant	(0, 0)	(1, 1)
Adversarial	(1, 0)	(0, 1)

Table 1: Payoffs for the collision game

fer in various ways: The former is infinite-horizon, with discrete actions and a one-dimensional action space; meanwhile, the latter has finite-horizon episodes, real-valued actions, and a two-dimensional action space. Moreover, the collision game features three classes of interaction:

- *Affinitive*: If the episode terminates with a collision, both agents are rewarded with a single point; otherwise neither agent is rewarded. We classify this as a *cooperative* relationship as both interests are aligned.
- *Avoidant*: Both agents are rewarded if no collision occurs and are given nothing otherwise. This also is a *cooperative* relationship, but one incentivizing a different pattern of behavior.
- *Adversarial*: One species is assigned the role of predator, the other the prey. If the episode ends with a collision, the predator is awarded a point while the prey receives nothing, and vice-versa. This is a *competitive* relationship due to the misalignment of their goals.

We note that the reward in the linguistic prediction game is real-valued, while in the collision game it is discrete. But the similarities between the two games are also strong. Our game can be easily mapped to various ecosystem topologies, agents begin with ignorance of their partner's identity, different flavors of competitive and cooperative interactions are involved, and there is room for diverse strategies, secret codes, and methods of deception to emerge.

Ecosystems

We study six different ecosystem topologies of the collision game. Many of the ecosystems investigated here are quite analogous to those in Moran, mapping the "competitive" interaction to the "adversarial" and the "cooperative" to the "affinitive." But we shall see that inherent differences between the two domains lead to some variation in their observed dynamics.

We first consider the CONTROL case, a trivial single-population baseline with which to compare later results. The next, 2-COMP and 2-COOP, are simple two-population ecologies for the study of the affinitive and adversarial interactions respectively. As a simple biological metaphor, we use the term *parasite* to refer to species that act aggressively, *symbiote* for those that are largely affinitive, and *host* for species with mixed interaction sets. The letters H, P, and S in Figure 2 stands for these terms, while C stands for control.

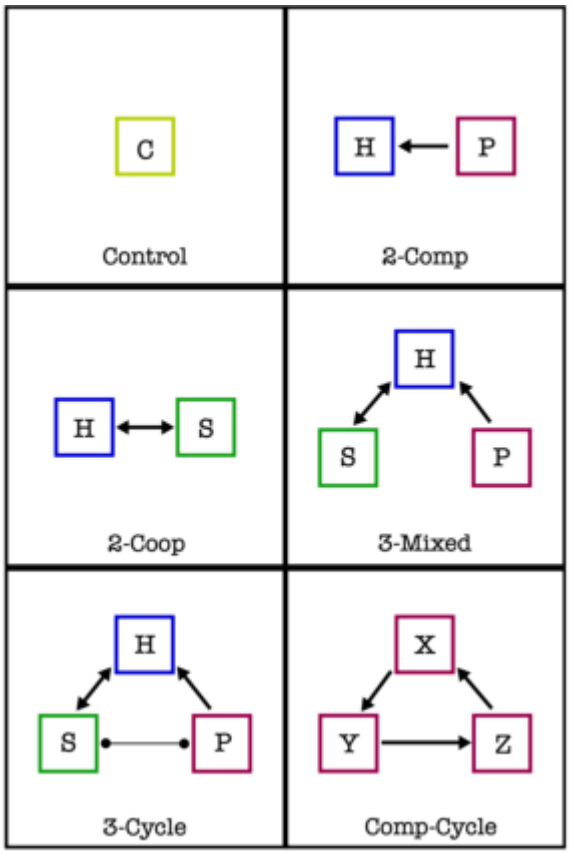


Figure 2: The six ecosystem topologies investigated in this paper. The double-headed arrow represents an *affinitive* interaction; the single-headed arrow denotes an *adversarial* interaction, with the arrow pointing to the prey; and the circle-ended line shows an *avoidant* interaction.

The 3-MIXED ecosystem introduces a third population and mixed interactions. This was the first to exhibit open-ended behavior in the prior work. The next, 3-CYCLE, features the avoidant interaction, which is unique to the collision game. The last, COMP-CYCLE, is an adaptation of Moran's *Three-Comp* that better suits the dynamics of our domain. As all members are competitive, we just call them X, Y, and Z.

Organism

Moran's DFSM model was inspired by the recurrent GNARL networks described in Angeline et al. (1994), and we use a variation of this architecture as our organism model. A predecessor to the NEAT networks of Stanley and Miikkulainen (2002), GNARL networks also evolve both their weights and topologies starting from a minimal state, albeit through asexual reproduction as opposed to crossover. Our networks have two input nodes corresponding to the distance and communication value of the other agent, one bias

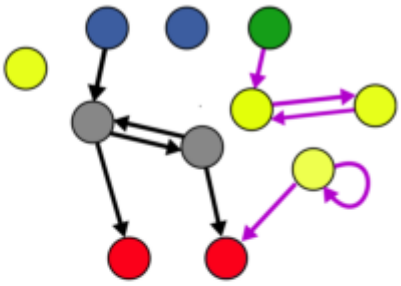


Figure 3: Example GNARL-like network. Blue are input nodes, green is the bias node, grey are useful hidden nodes, and red are output nodes. Yellow nodes and purple connections are unneeded for output and would be excluded from the count. This organism would receive a complexity score (number of expressed connections) of five.

node, and two output nodes corresponding to the movement and communication actions. The first generation has zero hidden nodes and zero connections.

Each node output is the weighted sum of all incoming connections passed through a nonlinear function. We use the *tanh* activation function for all connections so that output is bound in the range $[-1.0, 1.0]$. Each node retains this output value as a hidden state; in the case of a self-loop, the state value from the prior activation is weighted and likewise included in the sum. The movement action of the *greater* (or rightmost) agent is negated in simulation, so that an output of 1.0 always corresponds with moving towards the other agent and -1.0 with retreat. In these experiments, the distance between the agents is first divided by $128 + 5 = 133$, so as episodes must end by 128 timesteps, the distance input is bounded roughly in the range of $[0, 2.0]$.

Mutation

Reproduction occurs through asexual mutation and consists of the following steps. First, a clone is made of the parent and each of its existing connections is injected with Gaussian noise drawn from $\mathcal{N}(0, 0.1)$. (There is no notion of temperature in this model.) Then one of four structural mutations is performed, each with $\frac{1}{4}$ probability [2]:

1. *Add node*: A new node is added to the network with no incoming or outgoing connections.

2. *Add connection*: Two random nodes, an origin node and a destination node, are selected from the network and a connection is added between them with a weight value of zero. As in Angeline et al. (1994), an input node cannot serve as a destination node, and an output node may not serve as an origin node.

[2] We note that addition and deletion of nodes and connections occur with equal probability.

3. *Delete connection*: An existing connection is selected at random and deleted.

4. *Delete node*: A hidden node is selected at random and deleted. We must decide what to do with the connections attached to the deleted node. To ensure that the number of expected *add connection* and *delete connection* operations remains the same, we randomly reassign such connections to other nodes and set their weights to zero.

Complexity Metric

It is well acknowledged that complexity is ambiguous and difficult to quantify; however, certain definitions can be useful depending on the context. When gauging the complexity of our networks, we appeal to a similar logic as Moran and Pollack (2019) and Lecun et al. (1989): A neural network with many millions of connections may be considered more complex than one with only a few. But if the majority of those connections are disconnected from the output or produce identical (or random) output regardless of input, we would not call it complex.

As a proxy we define a network's genomic complexity as the number of connections with nonzero weight capable of influencing the output, first tracing the network graph to identify spurious connections and nodes. This approach may be less reliable in absolute terms than the state count of a minimized DFSM, but it should serve as a useful heuristic, especially if correlated with nonrandom, nontrivial behavior. We further test this metric via heuristic pruning in the Experiments.

Selection

In Moran's experiments, each species comprises fifty individuals reproducing through simple fitness-proportionate selection without elitism. Preliminary experiments in our domain suggested that populations of fifty are unstable, while those of two hundred exhibit more rapid growth but are costlier to simulate. Here we use populations of one hundred organisms. Each generation consists of a round of all-against-all play between each member of a species and those of another as dictated by the ecosystem's interaction set.

Once scores are recorded, a round of reproduction occurs. Two different forms of selection are studied. We first use *fitness-proportionate* selection with elitism of 50%. The top fifty fittest individuals are retained for the next generation, with roulette wheel selection among them determining the parents of the next fifty children. We refer to this set of trials as the ROULETTE group. The primary reason for including elitism is to allow for better comparison with our second approach, the *Discovery of Search Objectives* (DISCO) method introduced by Liskowski and Krawiec (2017), which uses elitism by default. More rigorous study of population size and elitism effects remains a direction for future work.

The collision game can be interpreted as a *test-based problem* as defined in de Jong and Pollack (2004), with each

entity serving as a possible solution and its interaction partners as tests to be passed. The application of simple fitness-based *scalar evaluation* to test-based coevolutionary problems is prone to a number of pathologies leading to premature convergence (Watson and Pollack, 2002). DISCO tackles this issue through the automatic identification of different "skills" required to pass related tests within a population, yielding a set of *derived objectives* that can be explored using multi-objective optimization. The derived objectives may be obtained by executing a standard clustering algorithm on the columns of the *interaction matrix* gathered from the population of tests and solutions following evaluation. The method has produced impressive results on classic coevolutionary benchmarks with relatively little overhead.

A full review of DISCO is beyond the scope of this work, but we briefly outline the parameters employed in our experiments. We use *X-Means clustering* on the interaction matrix with no limit to the number of possible clusters, then perform *NSGA-II* on the population with respect to the derived objectives with elitism of 50% and a tournament size of three to select the parents of the next generation (Pelleg and Moore, 2002; Deb et al., 2002).

Experiments

For each of the experiments detailed below, twenty separate trials were run simultaneously for 25,000 generations. Simulation code was written in Julia and executed in parallel on an HPC cluster.[3,4] All of the data examined here result from taking the average of each statistic over all members of a population, then taking the median with lower and upper quartiles over all twenty trials.

CONTROL

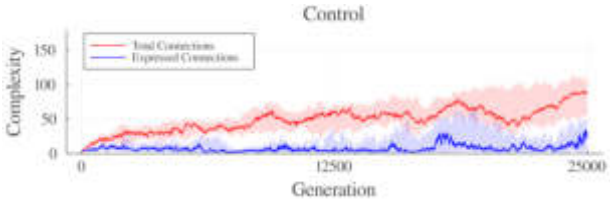


Figure 4: CONTROL complexity growth

In the CONTROL experiment, no games are played or evaluations are performed. A single population is allowed to reproduce with parents chosen at random and with no fitness input. Because the number of connections is bounded from below at zero, it is expected for the number of connections to rise slightly. We plot the number of expressed connections as well as the total number of connections, regardless of their involvement with output. It can be seen that the number of the total connections continues to rise, but without pressure to produce meaningful output, the growth of expressed connections remains at a relatively stable plateau. For all the ecosystems that follow, the total number of connections is omitted and only the expressed connections are plotted.

COOP

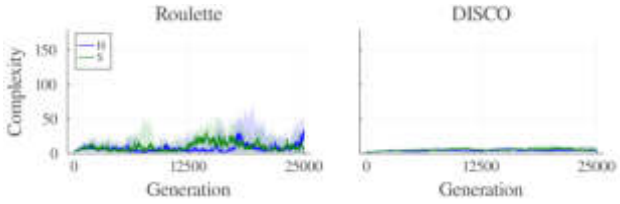


Figure 5: COOP complexity growth

As in Moran, neither of the COOP groups show signs of breakaway growth, but there is a marked difference between ROULETTE and DISCO selection. The number of connections in the COOP: ROULETTE group tends to fluctuate beneath fifty. Both species in the COOP: DISCO group, however, are much smaller and remarkably static. This provides early evidence for the increased efficacy of DISCO. The COOP ecosystem presents its inhabitants with a trivial task: To achieve maximum reward, both agents must simply move towards one another at maximum speed for three timesteps. This requires maintenance of a single connection from the bias node to the movement node – any extraneous connections thus only introduce risk. The minimal size and variance of the COOP: DISCO group already suggests that the method possesses sharper discriminatory capabilities.

COMP

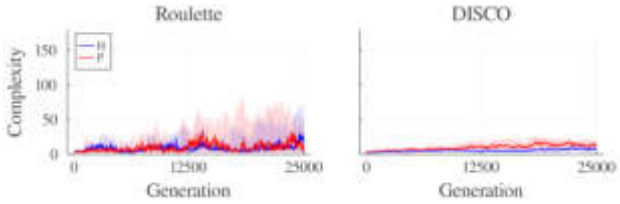


Figure 6: COMP complexity growth

The results here are largely similar to the COOP group, but with a few notable differences. In Moran, the trend in complexity among COMP agents was static, lower even than the control, while here we see more fluctuation and slightly higher parasite complexity. This may be due to the dynamics of our game, wherein the parasite could try some strategy

[3]Source code and additional resources may be found at https://github.com/twillkens/collision/

[4]We acknowledge computational support from the Brandeis HPCC which is partially supported by the NSF through DMR-MRSEC 2011846 and OAC-1920147.

to lure fleeing mutant hosts. Again the COMP: DISCO group remains largely static with minimal complexity. This likely reflects the method's capacity to screen for deleterious mutations among the hosts. In any case, as with Moran, we observe no evidence of unbounded behavior.

3-MIXED

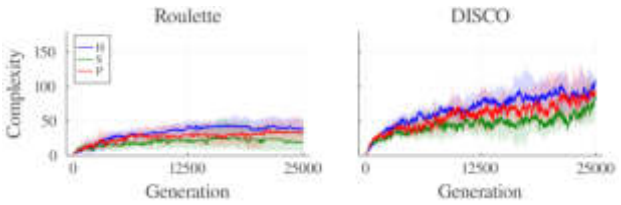


Figure 7: 3-MIXED complexity growth

This experiment presents us with our first surprise. The 3-MIXED topology includes both a competitive and cooperative interaction. Assuming the dynamic predicted by Moran to be universal, we would expect to observe unbounded growth of the symbiote. However, it can be seen that around halfway through the trial the 3-MIXED: ROULETTE group stagnates, reaching a stable configuration.

In contrast, the 3-MIXED: DISCO group gives us our first inklings of open-endedness. The complexity scores rise at a much higher rate, with the hosts in the lead. A clue as to why might be found in the fitness data, and Figure 8 shows very different performance patterns between the groups. In early generations, hosts of the ROULETTE group are very willing to collide, as evidenced by the high fitness of both the symbiote and parasite groups. But we see that the hosts gradually evolve to become more cautious, causing the fitnesses of the other populations drop. In the 3-MIXED: DISCO group, we see no such shift in outcomes, instead noting that hosts appear generally more willing to take risks, and that parasites are still generally at a disadvantage but one not so severe. The two selection methods lead their populations to traverse the fitness landscape in very different ways.

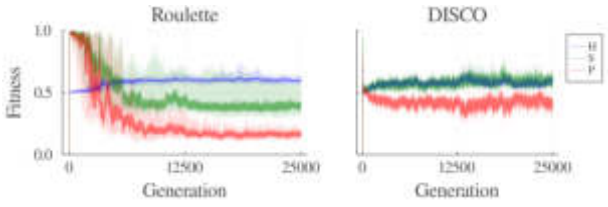


Figure 8: Comparison of fitness trends (median with lower and upper quartiles) between both selection methods within the 3-MIXED ecosystem

We might still claim evidence of the generality of Moran's dynamics: that the inclusion of both competitive and cooperative interactions is responsible for the growth. One could argue that the failure of the ROULETTE trial to reproduce the phenomenon is due to the higher dimensionality and continuous nature of our game. However, the growth trends differ substantially. In the prior work, only the symbiote grew without bound; in ours, all three rise with the host leading the group. This already contradicts Moran's hypothesis regarding high-frequency mutual adaptation. Instead we see a mutually reinforcing dynamic among all populations that seems to follow different rules.

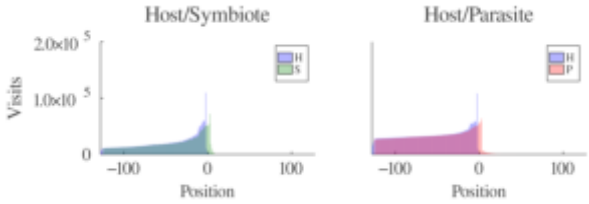


Figure 9: Total position visitation counts following interactions between the elite 50% at generation 25,000 for each trial. In each plot, the first species in the title is cast as the lesser agent, while the second acts as the greater agent.

We expected to observe a variety of movement behavior in this scenario, but histograms of position visitation counts (see Figure 9) show that movement patterns were highly constrained. These diagrams are created by simulating all interactions between all members of two species. The positions visited on the number line by each player of a species are fed into a histogram with 225 bins, and the two histograms are plotted together. In the Host/Symbiote plot, we see that the two histograms almost completely overlap. This shows that the hosts are strongly biased to retreat, while each symbiote and parasite pursues with full speed. Evidently, the host only allows itself to be overtaken if the other's pattern of communication passes the test.

3-CYCLE

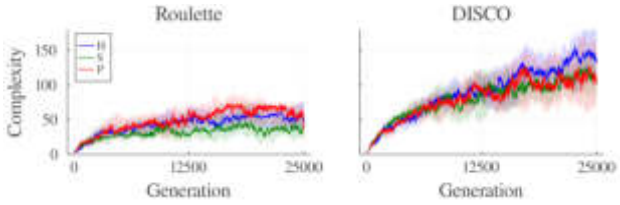


Figure 10: 3-CYCLE complexity growth

This lack of behavioral variety motivated the design of the 3-CYCLE ecosystem. We hypothesized that there was little evolutionary incentive for the symbiote and parasite in 3-MIXED to alter their movement patterns as there was no risk in moving forward, only possible reward. The growth

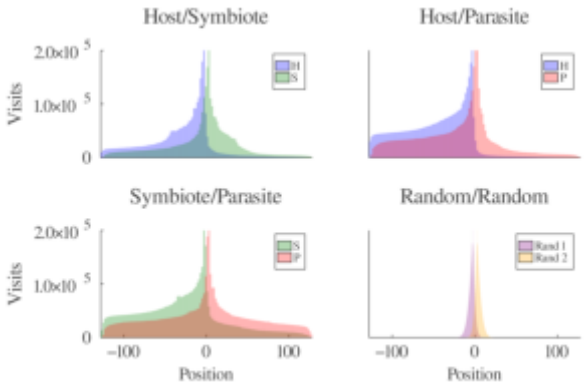


Figure 11: Total position visitation counts for the elite 50% at generation 25,000 for each trial. The Random/Random subplot results from random simulations using the same collision game parameters.

of their communication faculties may proceed unbounded, but a whole dimension of phenotypic expression is left unexplored. We hypothesized that complexity growth could be accelerated if varied movements were incentivized.

To provide an element of risk, we added an *avoidant* interaction between the parasite and the symbiote. This topology has no immediate analogue in Moran's work; however, as with the affinitive interaction, avoidant species are ultimately in a cooperative relationship as their goals are aligned. The 3-CYCLE: ROULETTE group again fails to achieve unbounded growth. But 3-CYCLE: DISCO shows improvement over the 3-MIXED: DISCO topology in terms of complexity, accelerating the growth rates of all species.

Figure 11 shows that variety was indeed added to the movement patterns of the organisms. Comparison with simulations of random agents performing actions drawn uniformly from $[-1, 1]$ indicates that the outputs of our networks are nonrandom. Moreover, we can spot an interesting trend: Certain species appear more biased to aggression as the frequency plots remain somewhat lopsided, most evidently in the Host/Parasite pairing.

	Random	*Host*	*Symbiote*	*Parasite*
Random	56%	30%	58%	75%
Host	30%	-	72%	40%
Symbiote	58%	72%	-	24%
Parasite	75%	40%	24%	-

Table 2: Average collision rates between the elite 50% of species at generation 25,000 of the 3-CYCLE: DISCO ecosystem.

To explore this, we played the elite members of each 3-CYCLE: DISCO species against a random agent twenty times, then again against each other following the interactions of 3-CYCLE. We also play a random agent against itself for one thousand games. We see in Table 2 that the parasite is far more aggressive than random chance, while the host is more circumspect. There is some evidence of effective communication: A symbiote is far less likely to collide with a parasite and far more likely to collide with a host. Despite the high aggression of the parasite, we see it is 50% less active versus the symbiote. And a host is more likely than random to collide with a parasite, suggesting the some members of the latter group posses a seductive power.

Returning to Figure 9, we see that the hosts often succeed in compelling the parasite to retreat. Closer study is needed, but this could be evidence of a manipulative strategy by the host, or perhaps part of a more elaborate convention.

One might ask whether this genotypic growth is a meaningful accumulation of complexity, or simply bloat introduced by the implementation. We performed an experiment to test the impact of neural network pruning on organism fitness, similar to that in Nadizar et al. (2022). As we incrementally raise the *pruning rate*, we sort and remove connections from each network according to one of three different *criteria*: (1) random, (2) absolute value of the connection weight, and (3) the absolute signal mean, i.e., the mean of the absolute value of all outputs produced by the connection over its simulated lifetime. Each of the three pruned organisms then interacts with the unchanged members of the other species, and their fitnesses are recorded.

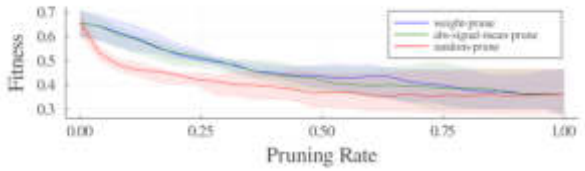


Figure 12: Effects of pruning on organism fitness. The initial networks are taken from the elite 50% of all species at generation 25,000 of 3-CYCLE: DISCO, already stripped of unreachable or zero-weight connections.

We see in Figure 12 that for all three methods, fitness decreases as the pruning rate increases, with random pruning producing the sharpest decline. If the networks were primarily bloat, we would not expect pruning to have such an impact on performance. This suggests that the networks possess meaningful structure and that our complexity metric may have utility in this domain.

While the dynamics are quite different between Moran's results and ours, the changes produced by our modification to 3-MIXED, along with the pruning results, supports their broader argument that ecosystem topologies are an effective means of inducing complexity growth and exploring the

space of meaningful genomic and behavioral possibilities.

COMP-CYCLE

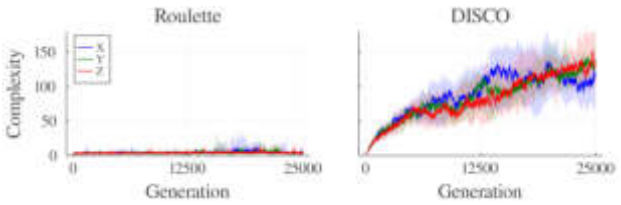


Figure 13: COMP-CYCLE complexity growth

So far our results have mostly supported the conclusions of the prior work. The dynamics may differ in various regards, but we still see that purely cooperative and purely competitive interactions lead to stagnation while the mixture of both results in unbounded complexity growth. The final test is to examine the dynamics of a purely competitive ecosystem with multiple interactions.

In Moran, competitive clusters are depicted as a troublesome antagonists. Attempts to automatically generate larger ecosystems reportedly failed due to their *convention chasing* dynamics (Ficici and Pollack, 1998). These dynamics allegedly produce a kind of "simplicity pollution," exposure to which poisons open-ended potential and leads to *degenerate ecosystems*. This prompted Moran to develop hand-designed ecosystem *ladders* instead, which explicitly exclude such clusters.

We first experimented with a model where X competes with Y, while Y competes with Z. However, this necessarily leads to trivial dynamics in our game: Z always will learn to flee Y, and once Y learns the futility of chasing, they will also flee to avoid being caught by X. We speculated that a cycle of competitive interactions could prove more interesting. In the COMP-CYCLE ecosystem, X pursues Y, while Y pursues Z, and finally Z pursues X. A single-minded strategy carries the risk of being punished; the question is whether convention-chasing stifles any possibility of complexity growth.

The results of the COMP-CYCLE: ROULETTE group seem to answer in the affirmative. Like in Moran, the complexity scores are very low; in fact, COMP-CYCLE has the most stable complexity out of any of the ROULETTE trials. However, the COMP-CYCLE: DISCO results offer a remarkable contrast, rising equally as fast as 3-CYCLE: DISCO. This demands an explanation: How does the topology most correlated with simplicity and stagnation in the prior work produce such a dramatic rise in complexity?

Discussion

For confidence in our verdict it is necessary to revisit the linguistic prediction game and perform a similar comparative analysis, but we shall offer a hypothesis. Scalar evaluation methods such as fitness-proportionate selection are especially prone to *coevolutionary pathologies*. Moran believes that the stagnation of purely competitive topologies is caused by convention chasing, and this is likely the case. However, the DISCO method's focus on combinations of interaction outcomes broadens the "evaluation bottleneck" between the fitness function and the evolutionary search. This could allow agents with valuable skills but lower fitness a greater chance to reproduce, alleviating the risk of premature convergence.

Within our context, it is empirically shown that convention-chasing is not insurmountable and that a different selection method unlocks complexity growth in topologies previously considered intractable. We finally argue that the interaction-level dynamics observed in Moran – such as the unique driving power of mixed interactions – are unlikely to be truly general. They instead may be an artifact of the selection method and the specific dynamics of the linguistic prediction game. This does not imply that the dynamics of the collision game are necessarily general; moreover, the efficacy of DISCO may prove to be domain-dependent as well. The necessary conditions for such "coevolutionary updrafts" remain obscure and still serve as an interesting object of study.

Conclusion

Our work shows that the phenomenon of unbounded complexity growth described in Moran and Pollack (2019) is reproducible using neural networks in a substrate with very different characteristics. Ecosystem topologies are shown to induce rises in complexity and variation in behavior, and the utility of coevolutionary methods such as DISCO is shown in convincing fashion. On the other hand, our results contradict the specific dynamics predicted by the prior work, supporting the argument that such dynamics are in fact domain dependent. This may prove to be a blessing, as combinations of different methods could relieve us from designing ecosystems by hand and allow for rapid automatic exploration of certain domains.

There are many possible directions in which to take future work. Returning to the linguistic prediction game, different reproduction methods should be compared to test the sensitivity of the underlying dynamics. Within our new domain, more intricate ecosystems could be explored, along with deeper study of the network structures and evolutionary drivers. Metrics should be devised to test for the presence of open-endedness in the wider sense suggested by Stepney (2021), and deeper analysis of behavior and communication is needed, possibly in terms of statistical complexity (Crutchfield, 1994; Sinapayen and Ikegami, 2017). Finally, other interaction domains should be designed and tested to better understand the general features of the phenomenon.

References

Adams, A. M., Zenil, H., Davies, P. C., and Walker, S. I. (2016). Formal definitions of unbounded evolution and innovation reveal universal mechanisms for open-ended evolution in dynamical systems.

Angeline, P., Saunders, G., and Pollack, J. (1994). An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(1):54–65.

Axelrod, R. (1984). *The Evolution of Cooperation*. Basic, New York.

Crutchfield, J. P. (1994). The calculi of emergence: computation, dynamics and induction. *Physica D: Nonlinear Phenomena*, 75:11–54.

de Jong, E. and Pollack, J. (2004). Ideal evaluation from coevolution. *Evolutionary computation*, 12:159–92.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Dolson, E. L., Vostinar, A. E., Wiser, M. J., and Ofria, C. (2019). The MODES Toolbox: Measurements of Open-Ended Dynamics in Evolving Systems. *Artificial Life*, 25(1):50–73.

Ficici, S. and Pollack, J. (1998). Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states.

Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. volume 1 of *Foundations of Genetic Algorithms*, pages 69–93. Elsevier.

Guttenberg, N., Virgo, N., and Penn, A. S. (2019). On the potential for open-endedness in neural networks. *Artificial Life*, 25:145–167.

Hopcroft, J. E. (1971). An n log n algorithm for minimizing states in a finite automaton.

Lecun, Y., Denker, J., and Solla, S. (1989). Optimal brain damage. volume 2, pages 598–605.

Liskowski, P. and Krawiec, K. (2017). Online discovery of search objectives for test-based problems. *Evolutionary Computation*, 25:375–406.

Moran, N. and Pollack, J. (2019). Evolving complexity in prediction games. *Artificial Life*, 25:74–91.

Nadizar, G., Medvet, E., Huse Ramstad, H., Nichele, S., Pellegrino, F. A., and Zullich, M. (2022). Merging pruning and neuroevolution: towards robust and efficient controllers for modular soft robots. *The Knowledge Engineering Review*, 37:e3.

Pelleg, D. and Moore, A. (2002). X-means: Extending k-means with efficient estimation of the number of clusters. *Machine Learning, p*.

Sinapayen, L. and Ikegami, T. (2017). Online fitting of computational cost to environmental complexity: Predictive coding with the ϵ-network. In *ECAL*.

Soros, L. B. (2018). Necessary conditions for open-ended evolution.

Stanley, K. O. (2019). Why Open-Endedness Matters. *Artificial Life*, 25(3):232–235.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.

Stepney, S. (2021). Modelling and measuring open-endedness. In *OEE4 workshop,at ALife 2021, Prague, Czech Republic (online), July 2021*.

Taylor, T. (2019). Evolutionary Innovations and Where to Find Them: Routes to Open-Ended Evolution in Natural and Artificial Systems. *Artificial Life*, 25(2):207–224.

Watson, R. and Pollack, J. (2002). Coevolutionary dynamics in a minimal substrate. *Morgan Kaufmann*.

Endosymbiosis or Bust: Influence of Ectosymbiosis on Evolution of Obligate Endosymbiosis

Kiara Johnson[1], Piper Welch[1], Emily Dolson[2] and Anya E. Vostinar[1]

[1]SymbuLab, Carleton College, Northfield, MN, 55057
[2]ECODE Lab, Michigan State University, East Lansing, MI 48824
anya.vostinar@gmail.com

Abstract

Endosymbiosis, symbiosis in which one symbiont lives inside another, is woven throughout the history of life and the story of its evolution. From the mitochondrion residing in almost every eukaryotic cell to the gut microbiome found in every human, endosymbiosis is a cornerstone of the biological processes that sustain life on Earth. While endosymbiosis is ubiquitous, many questions about its origins remain shrouded in mystery; one question in particular regards the general conditions and possible trajectories for its evolution. Modern science has hypothesized two possible pathways for the evolution of mutualistic endosymbiosis: one where an obligate antagonism is co-opted into an obligate mutualism (Co-Opted Antagonism Hypothesis), and one where a facultative mutualism evolves into an obligate mutualism (Black Queen Hypothesis). We investigated the viability of these pathways under different environmental conditions by expanding on the evolutionary agent-based system Symbulation. Specifically, we considered the impact of ectosymbiosis on *de novo* evolution of obligate mutualistic endosymbiosis. We found that introducing a facultative ectosymbiotic state allows endosymbiosis to evolve in a more diverse set of environmental conditions, while also decreasing the evolution of endosymbiosis in conditions where it can evolve independently.

Introduction

Endosymbiosis has played a crucial role in the evolutionary history of eukaryotes, as well as the evolution of life as a whole (Martin et al., 2015).[1] In particular, the evolution of endosymbiosis drove the major evolutionary transitions involving plastids (de Vries and Archibald, 2017) and mitochondria. The endosymbiotic acquisition of mitochondria provided so much chemical energy that it encouraged a wide expansion of the eukaryotic clade (Archibald, 2015; Zachar and Boza, 2020). Furthermore, humans are hosts to many endosymbionts; hence, analyzing their evolution and interaction with hosts is necessary to understanding the human system (Eloe-Fadrosh and Rasko, 2013; Perotti et al., 2007). Whether antagonistic or mutualistic, endosymbiotic relationships impact the population diversity and the complexity achieved by host species (Vostinar et al., 2021) as involved members undergo coevolution (Lazcano and Peretó, 2017).

[1]Symbiosis is a close and sustained relationship between individuals of different species (Lewin, 1982). Endosymbiosis is a specific form of symbiosis in which one organism lives inside the body or cells of the other.

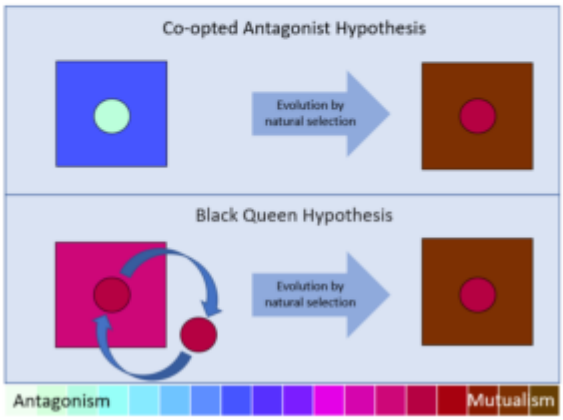


Figure 1: **Two of the posited hypotheses for the evolution of endosymbiosis.** Squares are host organisms and circles are symbionts. Color indicates whether the organism is antagonistic (pale green to blue) or mutualistic (purple to brown) towards its partner.

While the precise origins of every obligate mutualistic endosymbiosis necessarily remain unclear, two of the major hypothesized pathways are: the Co-Opted Antagonist Hypothesis (Johnson et al., 2021) and the Black Queen Hypothesis (Morris et al., 2012) (Figure 1). The Co-Opted Antagonist Hypothesis proposes that an obligate mutualism evolves when an antagonistic relationship is co-opted into a relationship that benefits both the host and endosymbiont, whereas the Black Queen Hypothesis suggests that a pre-existing facultative mutualism evolves into an obligate mutualism because one or both partners lose functionality required to remain independent. It is important to note that the relationship between these hypotheses is not an obligate dichotomy and it is likely that they have both contributed to the evolution of various endosymbioses. However, it is un-

known which path is more likely and how they may interact in a co-evolving population.

The timescales and resources required to observe co-evolutionary dynamics in a traditional laboratory environment inherently hinder their investigation. Even the fastest evolving microbial systems still require weeks, months or years to achieve the necessary evolutionary timescales. Further, current technology lacks the ability to perfectly control every potential confounding variable and perform data collection at the level of each individual organism. However, interactions between individual organisms in complex and varied populations are necessary to investigate symbiotic dynamics, making traditional population-level analytical modeling also insufficient. Therefore, we utilized and expanded upon Symbulation – an evolutionary agent-based platform designed to explore symbiotic relationships – to investigate the trajectories of co-evolving populations during the *de novo* evolution of endosymbiosis.

Specifically, the question that this investigation is centered around is: what are the conditions under which a mutualistic obligate endosymbiotic relationship can evolve and how do facultative and antagonistic intermediate stages impact that evolution? We determined that ectosymbiosis, symbiosis in which the parasite lives on the host's body surface, 1) expands the conditions in which endosymbiosis can evolve and 2) decreases the evolution of endosymbiosis in conditions where it would independently evolve.

Methods

To investigate the *de novo* evolution of obligate endosymbiosis, we used Symbulation, an open-source agent-based modeling platform for the study of symbiosis (Vostinar, 2021) that is built upon Empirical (Ofria et al., 2020). As shown in Figure 2, we created a virtual world with the following:

1. the necessary elements for evolution via natural selection (time, variation, competition, inheritance) for a population of 'hosts' and a population of 'symbionts' (whether they engaged in symbiosis or not),
2. the possibility of an interaction between an individual host and symbiont that was anywhere along a spectrum between parasitism/antagonism and mutualism,
3. the possibility for a free-living symbiont to infect a host and become an endosymbiont, and
4. a limit of at most one symbiont able to interact with each host.

Specifically, each experiment began with a full population of 10,000 hosts and a population of approximately 7,000 free-living symbionts. As shown in Figure 2, hosts and free-living symbionts exist in distinct but parallel populations with corresponding locations. This representation is necessary because limited space (only 10,000 locations are in the world) is the main source of competition (resources are set to unlimited), however for the questions of interest in this work, hosts and free-living symbionts should not compete directly with each other (in the same way that humans are rarely directly competing with bacteria for limited resources). This world structure allows for the hosts and symbionts to not compete with each other for space in the world. Instead, hosts compete only with other hosts, and symbionts with other symbionts. Additionally, hosts and symbionts can exist completely independently of each other, enabling us to also explore the possibility that, in certain environmental conditions, endosymbiosis will not evolve at all.

During these experiments, each organism receives a set amount of resources per timestep from the world, which varies by treatment. Upon accruing sufficient resources, both hosts and symbionts can reproduce with a chance of mutation. Hosts reproduce once they have collected 600 resources, free-living symbionts when they have collected 300 resources, and the resource quantity required for endosymbiont reproduction varies depending on the *transmission mode*, as discussed in the following paragraph. All reproduction is asexual and an offspring inherits its genome from its parent with mutations. A mutation of some kind will occur 100% of the time, but the size of the mutation varies. The mutation size is selected from a Gaussian distribution with a mean of 0 and standard deviation of 0.05.[2] Reproduction by hosts and free-living symbionts sends the offspring to a random world position. If an organism was occupying that space in the world, the offspring kills the former inhabitant (and its endosymbiont, if it has one). Both hosts and symbionts can also die of old age, 60 timesteps for hosts and 30 timesteps for symbionts, regardless of whether they are endosymbionts or free-living.

Endosymbionts have two possible transmission modes. First, when a host reproduces, the endosymbiont might *vertically transmit* an offspring based upon the user-configured *vertical transmission rate*, which is 50% by default. When a host reproduces, a random number between 0 and 1 is checked against the user-configured vertical transmission rate. In our experiments, we varied vertical transmission rate from 0 to 100% at 10% intervals. If that number is less than the vertical transmission rate and the endosymbiont has sufficient resources (200), the endosymbiont will also reproduce and transmit its offspring directly into the host offspring before the host offspring is dispersed as normal. The other possibility is *horizontal transmission*, which occurs when an endosymbiont acquires enough resources to reproduce without the help of its host (300). The endosymbiont's offspring exits the host and becomes a free-living symbiont in a random location of the world; the offspring can later attempt to become an endosymbiont by infecting a host. Note that a high vertical transmission rate

[2] If the mutation causes the trait to go outside of the fixed bounds, the trait is set to the nearest bound.

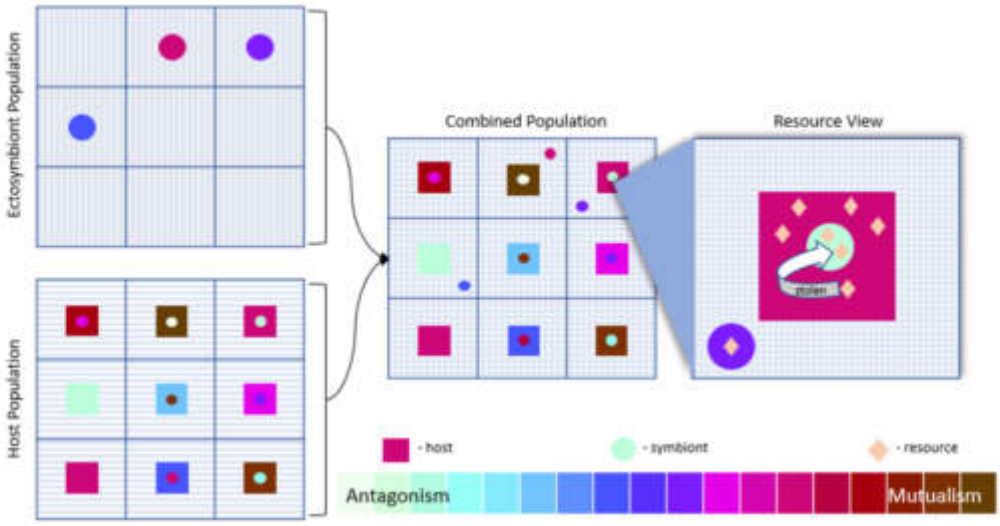


Figure 2: **A general overview of the Symbulation system.** Opaque squares are host organisms, circles are symbionts, and diamonds are resources. Color of hosts and symbionts indicate phenotype, ranging from antagonistic (pale green to blue) to mutualistic (purple to brown) towards a potential partner. The system is implemented with parallel populations of hosts and free-living symbionts, such that hosts and symbionts do not compete for limited space between species. However, free-living symbionts are able to infect hosts in the corresponding location of the parallel population.

does not mean that endosymbionts will exclusively transmit vertically – in situations where the endosymbionts reach 300 resources more quickly than the hosts reach 600 resources, such as when highly parasitic symbionts infect weakly antagonistic hosts (and steal most of the host's incoming resources), they will still horizontally transmit. In mutualistic relationships, however, if vertical transmission rate is high then the principle transmission mode will almost certainly be vertical, as hosts will accrue resources more quickly than their endosymbionts.

Each organism has a single floating-point number that represents its behavior on the antagonism to mutualism spectrum, which we will refer to as the 'interaction value.'[3] All hosts and symbionts begin every experiment with interaction values of 0, which assumes that they have not previously co-evolved together. Interaction values can span from -1 – representing antagonism (parasitism/defensiveness) – to 1 – representing mutualism. The further their interaction value is from 0, the more extreme the behaviour they exhibit.

An antagonistic host spends a portion of its incoming resources (based on its interaction value) on defense, while an antagonistic symbiont attempts to steal resources from its host. Symbionts that are less antagonistic than their hosts (*i.e.* if the host interaction value is more negative) fail to steal any resources and their hosts retain whatever proportion of resources they didn't spend on defense. If, however, a symbiont is more antagonistic than a host, it successfully overpowers the host's defenses and steals a proportion of the resources that weren't already spent on defense. The proportion stolen is based on the difference between the symbiont's interaction value and the host's. For example, if a symbiont has an interaction value of -1 and a host has an interaction value of -0.1, and the host receives 100 resources per update, then the host would spend 10 resources on defense (and those resources would be unavailable for either the symbiont or host to use), leaving 90 resources. The symbiont would then steal 81 of the remaining resources, and the host would keep the final 9 resources for its own reproduction.

When mutualistic, a host donates a portion of its resources to its symbiont based on the host's interaction value, while a mutualistic symbiont sends a portion of its resources back to its host based on the symbiont's own interaction value; the resources returned to the host by the symbiont are multiplied by a user-configured synergy factor of 5. For example, a mutualistic host with an interaction value of 0.5 might have a mutualistic symbiont with an interaction value of 0.5 as well. The host would receive 100 resources and donate 50 of them to the symbiont. The symbiont would then keep half (25) of the resources and donate back the other half, with the donated portion multiplied by the synergy (5). Therefore, at the end of the resource distribution process, the host

[3]Previous work using Symbulation used the term 'resource behavior value' instead.

would have 175 resources, and the symbiont would have 25 resources.

By default, symbionts can only interact with hosts through endosymbiosis, *i.e.* they must have infected the host already to interact with it. However, in parts of this work we also allow for *ectosymbiosis*. Ectosymbiosis will occur between a host and a symbiont in corresponding locations of their respective populations, but only if the host does not have an endosymbiont (this restriction is to remove the confounding factor of a host being able to have two symbionts when ectosymbiosis is enabled, but is configurable and could be relaxed in future work). In an ectosymbiotic relationship, resource distribution (mutualistic and parasitic/defensive behavior) unfolds identically to endosymbiosis.

If hosts or free-living symbionts have no partner, they will still spend resources attempting their symbiotic behavior (attempting to steal, investing in defense, or donating resources out) but incur no benefit. Thus, unless symbiosis is beneficial, an interaction value of 0 is optimal. This penalty decreases random drift of the interaction value in the absence of the partner species and means that interaction values that deviate from 0 are likely meaningful. Note that ectosymbionts can be considered facultative because they are able to survive outside of and without a host, though they may still suffer a fitness penalty if they have evolved to rely on a host.

We implemented an additional symbiont trait, *infection chance*, governed by another floating point number that can be between 0 (never try to infect) and 1 (always try to infect). At the beginning of an experiment, infection chance is 0 for all symbionts, but is under the same inheritance and mutation regime as the interaction value.

At each timestep, each free-living symbiont has a chance to attempt to infect a host based on its infection chance. If it decides to attempt infection, it attempts to enter the host with the matching location to its own in the host population. The infection can still fail if there isn't a host at that location, the host already has an endosymbiont (because only one endosymbiont is allowed per host in these experiments), or based on the user-configured infection failure rate. If infection is unsuccessful the aspiring endosymbiont is killed and removed from the symbiont population.

All experiments were run for 100,000 timesteps with 31 replicates per treatment. We used R (R Core Team, 2020) and the ggplot2 (Wickham, 2016) and viridis (Garnier et al., 2021) packages for all plots. For all statistical analysis we used Wilcoxon rank-sum tests and Bonferroni corrections for multiple comparisons. All code to recreate the experiments and analysis, along with data and supplementary materials, are available at `https://github.com/anyaevostinar/Evolution-of-Endosymbiosis-Paper`.

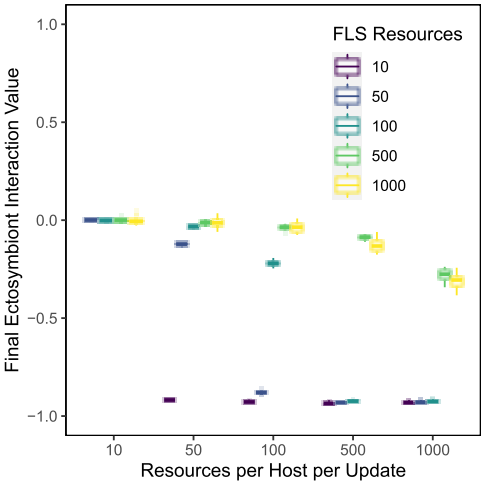


Figure 3: **Free living ectosymbiont interaction value at final timestep when endosymbiosis was prohibited.** FLS Resources are the resources distributed to free-living symbionts each timestep. When all organisms received 10 resources/timestep, the free symbiont population went extinct.

Results and Discussion

To investigate the *de novo* evolution of symbiosis we conducted three sets of experiments: 1) determining the degree to which ectosymbiosis evolves when endosymbiosis is prohibited, 2) investigating the evolution of endosymbiosis directly from a free-living ancestor, and 3) determining how the possibility of ectosymbiosis impacts the evolution of endosymbiosis. In each set of experiments, we started with a population of hosts and free-living 'symbionts.' Note that we refer to the two species as 'host' and 'symbiont' even when they are not engaged in a symbiosis for the sake of clarity. Due to the possible effect of resource availability, in all experiments, we varied the amount of resources received by hosts and free-living symbionts at each timestep. The resource amounts were 10, 50, 100, 500, or 1000 resources per organism per update. We tested each pairwise combination of resource amounts for each species.

Evolution of Ectosymbiosis in the Absence of Endosymbiosis

We first investigated whether our system would evolve significant ectosymbiosis in the absence of the possibility of endosymbiosis. We ran simulations were the endosymbiont limit was set to 0, therefore, ensuring that no endosymbiosis was possible. We determined the amount of ectosymbiosis based on the interaction values of the hosts and symbionts. If the organisms evolved to rely on ectosymbiosis, their interaction values would deviate from 0.

As shown in Figure 3, the amount of ectosymbiosis that evolved depended on the resource amounts received by the

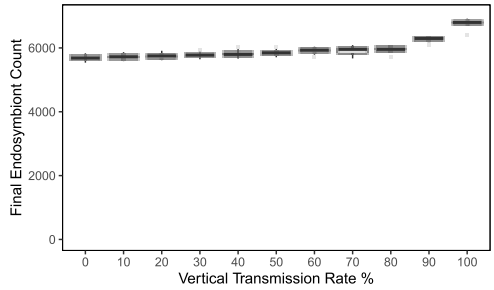


Figure 4: **Endosymbiont counts at final timestep when ectosymbiosis was prohibited across vertical transmission rates.** Hosts received 100 resources per organism per update and free-living symbionts received 50 resources per organism per update.

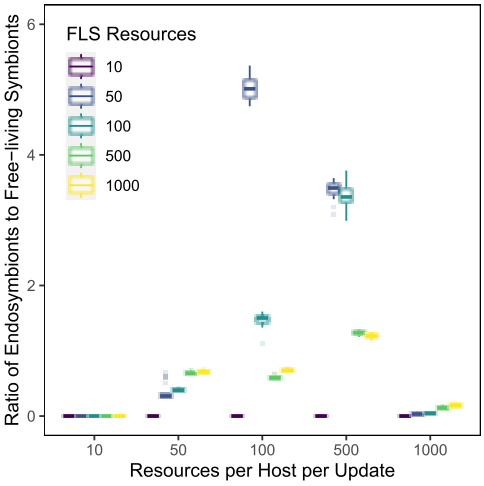


Figure 5: **Ratios of endosymbionts to free-living symbionts at final timestep when ectosymbiosis was prohibited.** When free-living symbionts received 10 resources per update, all symbiont populations went extinct.

hosts and symbionts. When hosts received only 10 resources per update, no meaningful amount of symbiosis evolved, and the symbionts went extinct when they also received only 10 resources per update. Increasing the amount of resources given to hosts generally led to increased levels of parasitism among the symbionts. For example, when symbionts received 100 resources per update, they evolved to be significantly more parasitic when hosts received 100 resources per update than when the hosts received only 10 resources per update ($p < 0.005$).

These results indicate that the evolution of ectosymbiosis in this system depends on the amount of resources available to both hosts and symbionts. They also indicate that, in the absence of vertical transmission through endosymbiosis, mutualistic symbiosis does not evolve in this system.

Endosymbiosis Can Evolve Directly From Free-Living Ancestor

We next determined which environmental factors favor the evolution of *de novo* endosymbiosis, by running simulations with 50% vertical transmission and varying the resources received by free-living symbionts and hosts.

We conducted two control treatments where endosymbiosis was not beneficial. In the first control we held the host's interaction value at -1, meaning that hosts invested all of their resources into defense (and therefore we also prevented them from dying of old age because they were unable to reproduce). In this control, symbiont interaction values remained at 0 as expected (data in supplemental materials). In the second control, we set the infection failure chance at 100%, meaning symbionts could never successfully infect a host and engage in symbiosis. As expected, we again found that interaction values and infection chance remained at 0 (data in supplemental materials).

We next examined the impact of vertical transmission rate on the evolution of endosymbiosis. The evolved interaction value of endosymbionts agreed with previous work (Vostinar and Ofria, 2019). However, as shown in Figure 4, vertical transmission rate did not have a meaningful effect on the final number of endosymbionts.

We then determined how the amount of resources received by hosts and free-living symbionts impacts the evolution of endosymbiosis when ectosymbiosis is not possible. As shown in Figure 5, the amount of resources accrued by hosts and free-living symbionts impacts the relative amount of endosymbionts compared to free-living symbionts. In general, intermediate amounts of resources for both hosts and free-living symbionts lead to the highest ratio of endosymbionts to free-living symbionts. Specifically, when free-living symbionts receive 50 resources per symbiont per update and hosts receive 100 resources per organism per update, the ratio of endosymbionts to free-living symbionts is 5.02, while the treatment with the next highest ratio is significantly lower at 3.47 endosymbionts/free-living symbionts when hosts receive 500 resources per update and symbionts still receive 50 resources/update/organism ($p < 0.005$)

The amount of resources received by hosts and free-living symbionts also impacted the final behavior of the endosymbionts, as shown in Figure 6. At resource amounts of 10, 50 or 100 for endosymbionts and 50 or 100 for hosts, endosymbionts generally remained neutral to the host, whereas at resource/update amounts of 50 for free-living symbionts and 500 for hosts, endosymbionts evolved to mostly be parasitic, with an average interaction value of -0.767. When the hosts received 500 resources/update and free-living symbionts received 500 or 1000 resources/update, endosymbionts evolved to be significantly more mutualistic, with average interaction values of 0.293 and 0.325, respectively

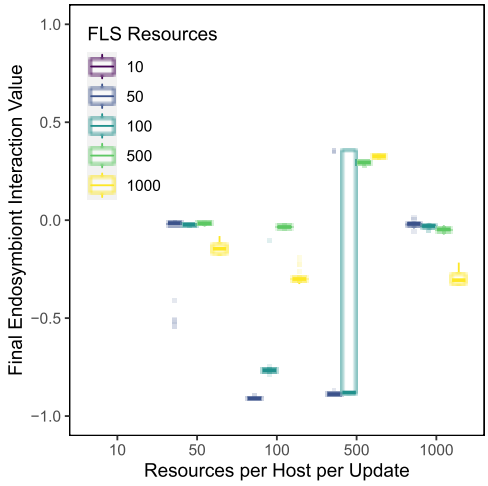


Figure 6: **Endosymbiont interaction value at final timestep when ectosymbiosis was prohibited.** When hosts were given 10 resources per update, the symbiont population died out.

($p < 0.00005$ for both).

These results show that host resource levels significantly impact the evolution of *de novo* endosymbiosis, and that there is an ideal intermediary resource level that favors the strongest levels of endosymbiosis. When resource levels are too low (10), symbionts go extinct, and so endosymbiosis does not evolve. Conversely, when resource levels are too high (1000), hosts are able to undergo rapid proliferation, making endosymbiosis unfavorable; thus it does not evolve. In addition, the amount of resources received by both species influences the nature of the symbiotic relationship, with a limited range of values selecting for mutualism. The mechanisms underlying this range are worthy of future study.

Does the option of ectosymbiosis increase the evolution of endosymbiosis?

To investigate if the possibility of ectosymbiosis increases the evolution of endosymbiosis, we ran simulations with both ectosymbiosis and endosymbiosis permitted with the same range of resources received by both species.

As shown in Figure 7, the effect of ectosymbiosis on the evolution of endosymbiosis varies dramatically across environmental conditions. For example, when free-living symbionts receive 50 resources/update and hosts receive 100 resources/update, ectosymbiosis significantly *decreases* the rate of endosymbiosis ($p < 0.005$). Conversely, when hosts receive 1000 resources per host per update, permitting ectosymbiosis significantly increases endosymbiosis across all free-living symbiont resource levels (all p-values < 0.0005)

These results indicate that ectosymbiosis enables the survival of symbionts in 1) conditions that are not ideal for symbionts, such as when free-living symbionts receive a low amount of resources, and 2) conditions that are not ideal for endosymbiosis, such as when hosts are able to reproduce rapidly. When the symbionts can survive in these conditions, they can then evolve towards endosymbiosis. However, counter to our hypothesis, in more ideal conditions, when endosymbiosis is able to evolve without ectosymbiosis, the option of ectosymbiosis decreases the degree to which endosymbiosis evolves. These results therefore suggest that endosymbioses will evolve in a more diverse set of conditions when ectosymbiosis is possible, however endosymbiosis may evolve to a lesser degree in ideal conditions.

What path do endosymbionts take to mutualism?

Finally, we investigated which of the two hypothesized pathways mutualistic endosymbionts took during evolution. To answer this question, we measured the complete phylogeny of the population, defining taxonomic units based on a discretization of the space of possible interaction values into 4 distinct bins. For a more thorough discussion of our phylogeny tracking methodology, see (Dolson et al., 2020). We then extracted the full lineage of the dominant (*i.e.* most numerous) taxonomic unit at the end of each replicate run. Finally, we compared the dominant lineages under our three experimental conditions: (1) endosymbiosis only, (2) ectosymbiosis only, and (3) endosymbiosis and ectosymbiosis both possible. Note that the Black Queen Hypothesis pathway was only possible when ectosymbiosis was enabled.

As shown in Figure 8, when only ectosymbiosis was permitted, all dominant symbiont lineages were intermittently parasitic. Conversely, when only endosymbiosis was permitted, most dominant symbiont lineages ended in a mutualistic phenotype, but spent some evolutionary time somewhat parasitic. However, the degree to which the dominant symbiont lineage was parasitic depended on the amount of resources each free-living symbiont received at each timestep. Specifically, when free-living symbionts received 500 resources/organism/update, no symbiont lineage spent any time in the extremely parasitic phenotype, but when the free-living symbionts received 1000 resources/organism/update, most (26/31 replicates) symbiont lineages were extremely parasitic for a period of their evolutionary history.

Finally, when ectosymbiosis and endosymbiosis were both possible, all symbiont lineages spent time in the somewhat parasitic phenotype space during their evolution. However, the degree of parasitism again depended on the amount of resources received by the free-living symbionts. When free-living symbionts received 500 resources/organism/update, 1/31 lineages spent evolutionary time in the extremely parasitic phenotype state. Conversely, when free-living symbionts received 1000 resources/update/organism, 19/31 symbiont lineages spent time in the extremely parasitic state.

These results suggest that the co-opted antagonist hypothesis is the dominant evolutionary pathway towards mutual-

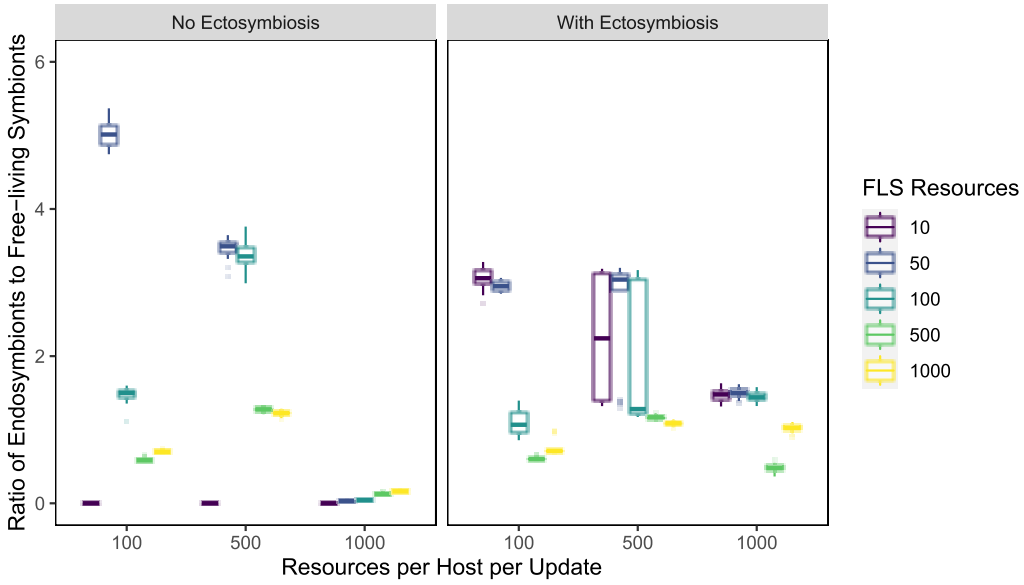


Figure 7: **Ratios of endosymbionts to free-living symbionts at final timestep with and without ectosymbiosis.** Resources distributed to hosts and free-living symbionts each timestep were varied at 10, 50, 100, 500, and 1000 for each species (host resource/update amounts of 10 and 50 were not meaningfully different and so are not shown here but are included in the supplemental material).

ism under these conditions. All final dominant symbiont lineages were historically parasitic to some degree. However, when symbionts receive more resources from the world, they are more likely to engage in stronger parasitism, and for a longer period of evolutionary time. Further, the possibility of an ectosymbiotic intermediary stage reduces the duration of evolutionary time that lineages spend in a strongly parasitic state, and hastens the evolution of mutualism. Presumably, the presence of ectosymbiosis means that a population can undergo lower-commitment mutations regarding symbiotic behavior while still receiving resources from the world. Thus such populations are less reliant upon coevolution from their symbiotic partners, in support of the Black Queen Hypothesis. However, while the presence of endosymbiosis almost always selects for mutualism, when only ectosymbiosis is possible the dominant lineages are likely to be somewhat parasitic, though selection is weaker overall.

Conclusion

We have shown that introducing the possibility of an ectosymbiotic intermediary stage into the evolution of endosymbiosis 1) diversifies the environmental conditions in which endosymbiosis is able to evolve, and 2) lessens the evolution of endosymbiosis in conditions where it can evolve independently. Further, within the conditions where we conducted extended phylogenetic analysis, most mutualistic symbionts descended from parasitic ancestors. However, adding the capacity for ectosymbiosis in addition to endosymbiosis promotes faster evolution of mutualism, but ectosymbiosis alone does not enable mutualism to evolve. Therefore, this work supports the Co-Opted Antagonist Hypothesis as the dominant evolutionary trajectory in our system, with Black Queen dynamics present to a lesser extent.

There are many other factors that influence the evolution of endosymbiosis and should be explored in future work. Specifically, many symbiotic systems allow for multiple symbionts (endo- and ecto-) to infect and interact with the same host. Symbulation would be an ideal system to further expand for the investigation of the effect of multi-infection on the *de novo* evolution of endosymbiosis. Further, many systems have elements of host and symbiont partner choice, which likely also would impact the evolutionary trajectory of both partners. Finally, Symbulation will be a valuable system for developing and testing metrics for analyzing the phylogenetic structure of mutualistic endosymbionts.

In this work, we have uncovered some of the conditions necessary for *in silico* endosymbiosis to evolve, elucidating possible evolutionary pathways that *in vivo* endosymbionts may have taken. Therefore, this work contributes to the story of why Earth's multicellular terrestrial life has persisted fruitfully, predictions of extraterrestrial life, and predictions of future evolution of vital endosymbionts.

Acknowledgements

This work was supported by NSF grant No. 1750125, and Carleton College's Towsley Endowment.

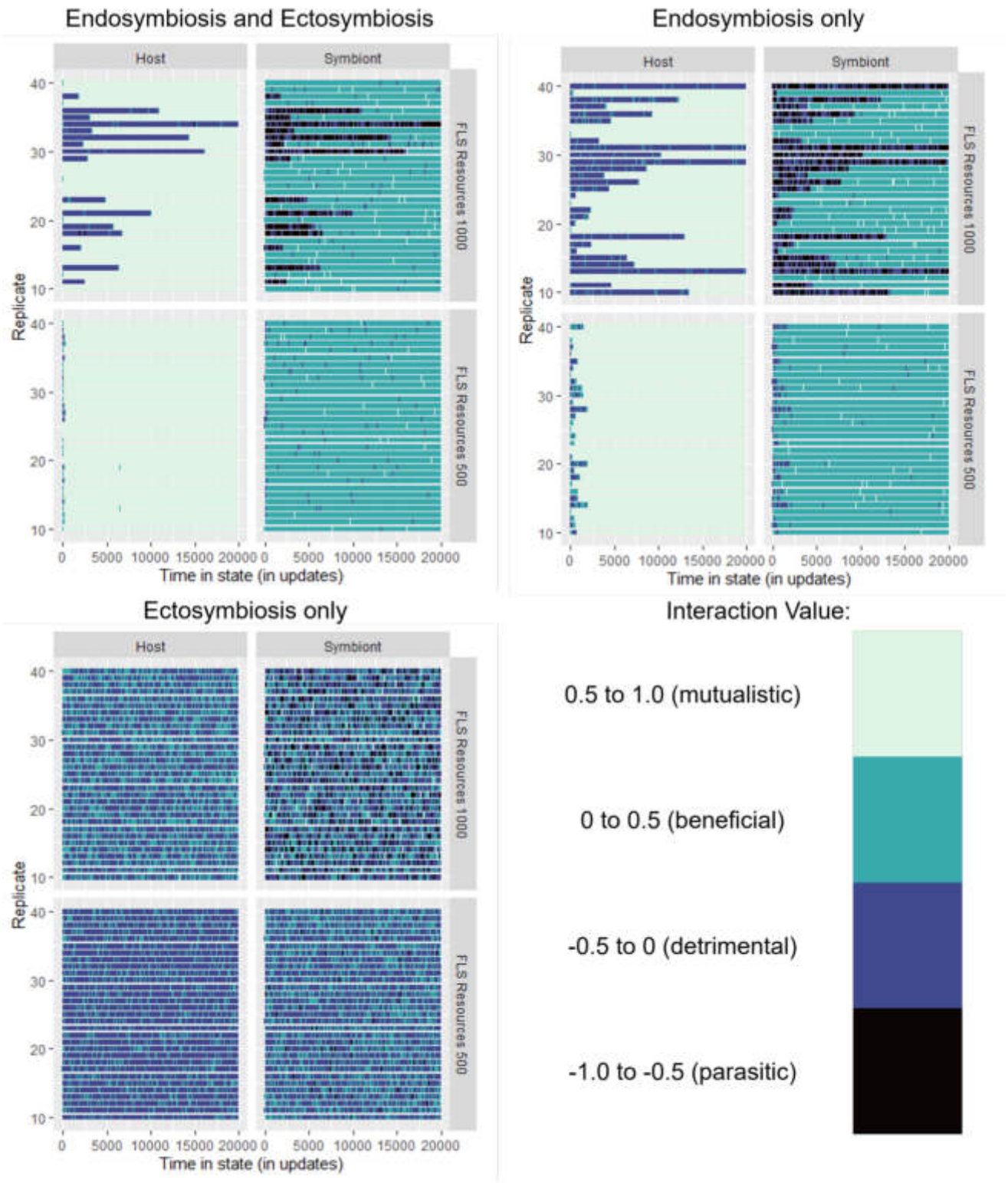


Figure 8: **Dominant lineage state sequences.** The dominant lineage is the sequence of ancestors (*i.e.* line of descent) of the most populous phenotype at the end of each experiment. Hosts were given 500 resources per update, and free living symbionts were distributed either 500 or 1000 resources. Interaction value categories are lower-bound inclusive, upper bound exclusive (except for the 0.5 to 1.0 state, which is 1.0-inclusive).

References

Archibald, J. M. (2015). Endosymbiosis and eukaryotic cell evolution. *Current Biology*, 25(19):R911–R921.

de Vries, J. and Archibald, J. M. (2017). Endosymbiosis: did plastids evolve from a freshwater cyanobacterium? *Current Biology*, 27(3):R103–R105.

Dolson, E., Lalejini, A., Jorgensen, S., and Ofria, C. (2020). Interpreting the Tape of Life: Ancestry-based Analyses Provide Insights and Intuition about Evolutionary Dynamics. *Artificial Life*, 26(1):1–22.

Eloe-Fadrosh, E. A. and Rasko, D. A. (2013). The human microbiome: from symbiosis to pathogenesis. *Annual review of medicine*, 64:145–163.

Garnier, Simon, Ross, Noam, Rudis, Robert, Camargo, Pedro, A., Sciaini, Marco, Scherer, and Cédric (2021). *viridis - Colorblind-Friendly Color Maps for R*. `10.5281/zenodo.4679424`.

Johnson, C. A., Smith, G. P., Yule, K., Davidowitz, G., Bronstein, J. L., and Ferrière, R. (2021). Coevolutionary transitions from antagonism to mutualism explained by the co-opted antagonist hypothesis. *Nature communications*, 12(1):1–11.

Lazcano, A. and Peretó, J. (2017). On the origin of mitosing cells: A historical appraisal of lynn margulis endosymbiotic theory. *Journal of theoretical biology*, 434:80–87.

Lewin, R. (1982). Symbiosis and parasitism—definitions and evaluations. *BioScience*, 32(4):254–260.

Martin, W. F., Garg, S., and Zimorski, V. (2015). Endosymbiotic theories for eukaryote origin. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1678):20140330.

Morris, J. J., Lenski, R. E., and Zinser, E. R. (2012). The black queen hypothesis: evolution of dependencies through adaptive gene loss. *MBio*, 3(2):e00036–12.

Ofria, C., Moreno, M. A., Dolson, E., Lalejini, A., rodsan0, Fenton, J., perryk12, Jorgensen, S., hoffmanriley, grenewode, Edwards, O. B., Stredwick, J., cgnitash, theycallmeHeem, Vostinar, A., Moreno, R., Schossau, J., Zaman, L., and djrain (2020). devosoft/Empirical: Before directory reorganization. `https://doi.org/10.5281/zenodo.4141943`.

Perotti, M. A., Allen, J. M., Reed, D. L., and Braig, H. R. (2007). Host-symbiont interactions of the primary endosymbiont of human head and body lice. *The FASEB Journal*, 21(4):1058–1066.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. `https://www.R-project.org/`.

Vostinar, A. E. (2021). Symbulation. `https://doi.org/10.5281/zenodo.5062147`.

Vostinar, A. E. and Ofria, C. (2019). Spatial structure can decrease symbiotic cooperation. *Artificial life*, 24(4):229–249.

Vostinar, A. E., Skocelas, K. G., Lalejini, A., and Zaman, L. (2021). Symbiosis in digital evolution: Past, present, and future. *Frontiers in Ecology and Evolution*, 9.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. `https://ggplot2.tidyverse.org`.

Zachar, I. and Boza, G. (2020). Endosymbiosis before eukaryotes: mitochondrial establishment in protoeukaryotes. *Cellular and Molecular Life Sciences*, pages 1–21.

Keep Your Frenemies Closer: Bacteriophage That Benefit Their Hosts Evolve to be More Temperate

Alison Cameron[1], Seth Dorchen[1], Sarah Doore[2], and Anya E. Vostinar[1]

[1]SymbuLab, Carleton College, Computer Science, Northfield, MN, 55057
[2]University of Florida, Microbiology & Cell Science, Gainesville, FL 32611
anya.vostinar@gmail.com

Abstract

Bacteriophages, also known as phages, are viruses that infect bacteria. They are found everywhere in nature, playing vital roles in microbiomes and bacterial evolution due to the selective pressure that they place on their hosts. As obligate endosymbionts, phages depend on bacteria for successful reproduction, and either destroy their hosts through *lysis* or are maintained within the host through *lysogeny*. Lysis involves reproduction within the host cell and ultimately results in the disruption or bursting of the cell to release phage progeny. Alternatively, lysogeny is the process by which phage DNA is incorporated into the host DNA or maintained alongside the host chromosome, and thus the phage reproduces when their host reproduces. Recent work has demonstrated that phages can exist along the parasitism-mutualism spectrum, prompting questions of how phage would evolve one reproductive strategy over the other, and in which conditions. In this work, we present an agent-based model of bacteriophage/bacterial co-evolution that enables lysogenized phage to directly impact their host's fitness by using the software platform Symbulation. We demonstrate that a viral population with beneficial lysogenic phage can select against lytic strategies. This result has implications for bottom-up control of vital ecosystems.

Introduction

Bacteriophages, viruses that specifically infect bacteria, are found everywhere in nature (Chevallereau et al., 2022). Phages play vital roles in the construction of their ecosystems as a result of selective pressure that they place on their hosts (Hobbs and Abedon, 2016; Chevallereau et al., 2022). The interactions between phage and bacterial hosts have demonstrated co-evolutionary dynamics such as the Red Queen hypothesis, Lotka-Volterra, arms races, bet hedging, etc. (Maslov and Sneppen, 2015; Stern and Sorek, 2011; Rohwer and Segall, 2015; Weitz and Dushoff, 2008) and they have particular importance in many microbial communities, including the human gut and digestive system (Subramanian et al., 2020; Bäckhed et al., 2005; Allison and Verma, 2000). The rise in antibiotic-resistant bacteria has led to renewed interest in phage therapy, the use of bacteriophages to combat bacterial infections, driving further research regarding phage-bacterial evolutionary dynamics (Lin et al., 2017; Lu and Koeris, 2011; Neu, 1992; Sulakvelidze et al., 2001). Additionally, bacterial populations can develop resistance to phage, and phage can increase bacterial cooperation in eukaryotic cells (Obeng et al., 2016). Finally, some bacteriophages can transfer genes between bacteria or disrupt genes upon integration into the chromosome, leading to phenotypic changes of the host (Miller, 2001). Because of these dynamics and their medical relevance, increasing our understanding of phage-bacteria co-evolution is crucial.

Phages have been historically considered strictly-harmful obligate endosymbionts because they depend on their bacterial hosts for successful reproduction and often cause the death of their host. However, there is growing evidence that phages can also confer beneficial traits to their hosts (Anderson et al., 2014; Harrison and Brockhurst, 2017; Owen et al., 2021; Obeng et al., 2016). As such, it is clear that phage have the potential to exist along a parasitism-mutualism spectrum.

Where a species lands on that parasitism-mutualism spectrum is largely influenced by its reproductive strategy (Yamamura, 1993; Ewald, 1987; Vostinar and Ofria, 2019). That is, endosymbionts that employ *horizontal transmission* are less likely to be mutualistic than their counterparts who rely on *vertical transmission*. Horizontal transmission is the process where an endosymbiont reproduces and its offspring is released into the world, independent of host reproduction. Vertical transmission, on the other hand, involves the transfer of a symbiont (or symbiont's offspring) from the parent host to the host offspring near the moment of reproduction. This method thereby provides selective pressure for the symbiont to increase their host's fitness in order to increase the likelihood of the symbiont's own reproduction. Thus vertical transmission is a mechanism that directly fosters mutualistic relationships between organisms, unlike horizontal transmission.

Bacteriophages mainly use two reproductive strategies, which can be considered approximate parallels to horizontal and vertical transmission: lysis and lysogeny (Hobbs and Abedon, 2016). Upon successful infection of a bacterial host, a bacteriophage will either enter the lytic cycle or be-

come a temperate lysogenic phage. Lytic phage redirect the bacterium's replication machinery and metabolic processes to mass produce phage particles, ultimately releasing those particles into the environment upon death of the host cell. This process is similar in dynamic to other methods of horizontal transmission, where endosymbiont offspring spread to other hosts in a population, and therefore the endosymbiont is not under selective pressure to increase host fitness. Lysogenic phage, however, can enter the host's genome and become an integrated prophage that may benefit or harm the bacterium's ability to function, or they may be completely inert (Anderson et al., 2014; Harrison and Brockhurst, 2017; Owen et al., 2021; Obeng et al., 2016). In this state, lysogenic phage do not produce virulent offspring but instead are maintained along with the bacterium's own reproductive process. Thus, lysogeny is a vertical transmission strategy, since endosymbiont offspring are transmitted vertically to host offspring: when the host reproduces, so too does the prophage. These are not the only two ways for phages to reproduce; however, the majority of phage reproduction likely exists along a continuum between these two methods (Mäntynen et al., 2021).

Phage replication through either the lytic or lysogenic cycle can dramatically affect host cells at both the individual and population levels. Evaluating this relationship between reproductive strategy and eco-evolutionary dynamics has been challenging, yet critical to our understanding of phage-host interactions. With phages now being intentionally used to treat antibiotic-resistant bacterial infections, the ability to predict potential outcomes of these interactions is becoming more crucial. Specifically, phage therapy relies on obligately lytic bacteriophage to kill bacteria, but their infection cycle is typically evaluated in controlled, pure culture systems. However, the environment in which they are introduced has multiple organisms, including temperate phages within the pathogenic hosts they may be targeting. It is unknown whether or how the obligately lytic phages may transition to a temperate infection cycle, or vice versa.

Previous mathematical modeling research shows that a regular influx of naive or uninfected hosts selects for phage to be lytic, because there are sufficient hosts for phage offspring to infect (Wahl et al., 2019; Sinha et al., 2017). Conversely, a lack of naive hosts selects for phage to be temperate, with a higher chance of lysogeny, because the number of available hosts for their offspring to infect is limited. However, most analytical models necessarily assume that lysogenic phage are dormant and do not have an impact on their host. There is growing evidence that that is not necessarily the case, and that lysogenized phage can either harm or help the infected bacterium (Anderson et al., 2014; Harrison and Brockhurst, 2017; Owen et al., 2021; Obeng et al., 2016). The factors determining the switch or preference for phages to be lytic or lysogenic is therefore likely more complicated than host population density alone.

For example, many strains of the species *Shigella flexneri* harbor prophages and parasitic genes in their genomes. Although it is a close relative of *Escherichia coli*, *Shigella flexneri* is an intracellular human pathogen that proliferates in intestinal epithelial cells (Labrec et al., 1964). The genus *Shigella* causes approximately 167 million cases of bacillary dysentery annually (Troeger et al., 2018). Its virulence has been attributed to the fact that some of the prophages disrupt functional avirulence genes, while others contribute to virulence or immune evasion (Nakata et al., 1993; Maurelli, 2007). For example, part of what determines the virulence of *Shigella flexneri*, its survival in the intestinal environment, and its ability to evade the human immune system is its serotype West et al. (2005). A molecule on the outer surface of the bacterium–the O-antigen–determines the serotype, with at least 20 serotypes described thus far (Muthuirulandi Sethuvel et al., 2017). This property is highly evolvable, with new serotypes regularly emerging Livio et al. (2014); Muthuirulandi Sethuvel et al. (2017). Sequencing has revealed that a majority of genes involving serotype modification originated from bacteriophages (Knirel et al., 2015). While some of these phages are now defunct prophages, others still persist as functional viruses. Thus, phages can provide a benefit to the host through serotype modification or conversion, by facilitating immune evasion of the *Shigella flexneri* host.

While a phage may be able to provide a benefit to its host bacterium, lysogenized phage still have the potential to induce and enter the lytic cycle either spontaneously or when in stressful environmental conditions (Nanda et al., 2014; Bruneaux et al., 2022). This possibility leads to conflicting selective pressures for bacteria and phage, where prophage that benefit their hosts may be under increased selection towards lysogeny, but hosts that evolve to rely on such benefits are then highly susceptible to eventual induction and lysis. It is therefore an open question (1) under what environmental conditions temperate phage are under selective pressure towards more frequent lysogeny if they are able to impact host fitness, (2) whether more frequent lysogeny then selects for temperate phage to be more beneficial to their hosts, and (3) whether these phenomena might reinforce one another.

These questions are difficult to experimentally test in wet-lab systems due to the necessary control of the relevant traits, the time required to observe evolutionary timescales, and the cost of materials and labor. Further, they would be challenging to create traditional analytical models to investigate due to the complex interactions between individual bacterium and phage. However, agent-based modeling enables us to overcome those challenges as well as investigate the general principles potentially governing these systems (Vostinar et al., 2021). Therefore, we turned to the field of artificial life and expanded the open-source agent-based modeling platform Symbulation to simulate bacteria/phage coevolutionary dynamics and determine the effect of beneficial prophage on

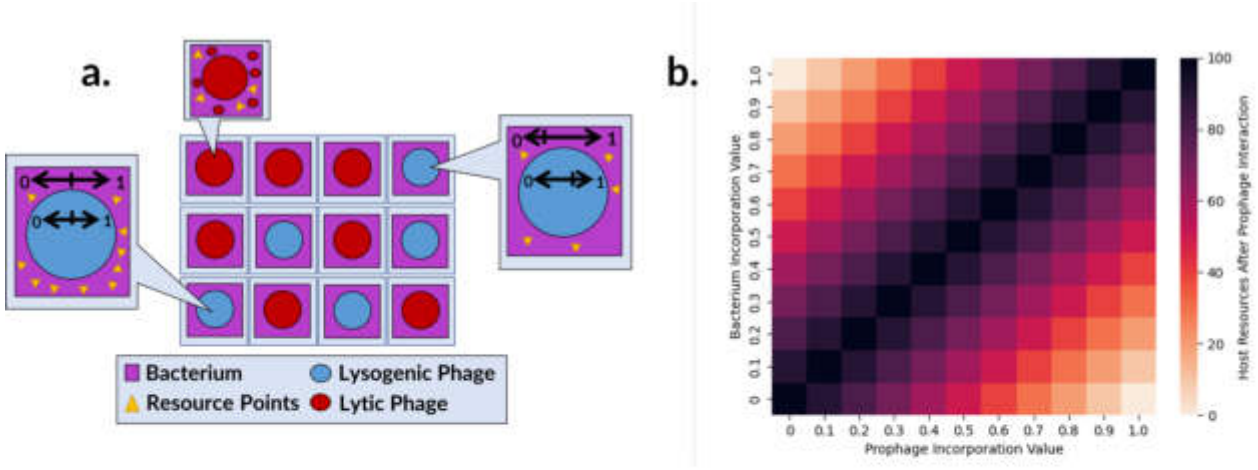


Figure 1: **Overview of the simulation.** (a) A population of hosts (purple squares) can be infected by up to one phage (blue or red circles). Phage and hosts each have an *incorporation value* (slider between 0 and 1). Lytic phage hijack host resources and produce viral particles, while lysogenic phage influence host resource levels depending on the similarity of incorporation values. (b) The amount of resources a host receives depends on the similarity between the host and prophage's incorporation value. The more closely they match, the more resources the host receives. The prophage influence ranges from depleting to doubling the 50 resources given to the hosts at each timestep.

the evolution of lysogeny.

We determined that if the prophage population provides a benefit to hosts on average, the population will also evolve higher rates of lysogeny, even to the point of evolving to be temperate when they otherwise would have evolved to be lytic. These results indicate that these co-evolutionary dynamics need to be explored in wetlab systems and their implications considered in phage therapy and related applications.

Methods

For this work, we expanded Symbulation, an agent-based modeling platform (Vostinar, 2021). Symbulation enables symbiotic relationships between agent populations and can track the evolution of genes and characteristics over time at an individual level. In Symbulation, a population of simulated bacteria are able to compete for resources, reproduce with mutation, and therefore evolve. In addition to bacterial hosts, the virtual world also supports a population of bacteriophage, as shown in Figure 1a. Each phage can survive outside of a host, but must infect a bacterial host to reproduce. Upon infection, a phage enters the lytic or lysogenic life cycle, depending on their genome.

Host/Bacterium Characteristics

Each host and symbiont in Symbulation has a set of characteristics, which can be viewed as the subset of interest of its genome. The bacterial host genome consists of an *interaction value* (between -1 and 1) and an *incorporation value* (between 0 and 1). The interaction value determines the degree to which it will defend against a potentially harmful symbiont. In this work, the interaction value is always negative, though able to evolve, due to the hostile interaction between bacterial hosts and phage symbionts. The incorporation value determines how successful a lysogenic bacteriophage is at incorporating itself into the bacterial host genome. At each simulated timestep, a bacterium collects 50 resources from the environment, and is able to reproduce when it accrues 600 resources. Thus, an uninfected bacterium will reproduce every 12th time step.

However, an infected bacterium may spend some of its resources on defense against its symbiont and will therefore reach the reproduction threshold at a slower rate. This is an abstraction of the many ways that bacteria defend against bacteriophage (Ofir and Sorek, 2018) by imposing a trade-off between reproductive speed and defensive capability. The amount of resources spent on defense is proportional to the host's interaction value. For example, a bacterium with an interaction value of -0.3 will spend 15 resources (30% of its collected 50 resources) on defense at each timestep. The resources remaining after defense spending can be further augmented or disrupted by the behaviors of its symbiont, leading to each infected bacterium reaching the reproduction threshold at its own rate.

When a bacterium reaches the reproduction threshold, it will create a copy of itself as its offspring (along with any lysogenized prophage), however the copying process is imperfect and mutations occur in both values of its genome. The size of a mutation is taken from a normal distribution with a mean of 0 and standard deviation of 0.02. The bacte-

rial offspring (along with lysogenized prophage), after mutation, is then placed into a random position in the world. If the selected position is already occupied by another bacterium, the previous occupant is killed and replaced by the new offspring. By this mechanism, the organisms that accrue resources the quickest, and thus reproduce the fastest, will eventually dominate the population.

A bacterium that is infected with a bacteriophage may be helped or harmed by the phage and thus reproduce at a different rate, if at all. In particular, a host's resource amounts are influenced differently if the phage is lytic or lysogenic. Lytic phage typically steal incoming resources from (and eventually kill) their hosts, while the impact of a lysogenic phage varies depending on both phage and host genomes, and ranges from destroying all new resources to doubling them, as shown in Figure 1b.

Symbiont/Bacteriophage Characteristics

The bacteriophage genome consists of an *interaction value* (between -1 and 1), an *incorporation value* (between 0 and 1), a *chance of lysis* (between 0 and 1), and a *chance of induction* (between 0 and 1). What a phage does at each timestep differs for phage living outside of a host, lytic phage, and lysogenic phage.

Free-Living Phage At each time step, each freely-living phage (those that are outside of a host) attempts to infect a nearby host to begin the process of reproduction. If the targeted host is not already hosting a phage, the freely-living phage will infect said host with a 100% success rate. Upon infection of a host, the phage will then begin either the lytic or lysogenic life cycle, with a probability based on its chance of lysis. However, if the targeted host for infection is already hosting a phage, the freely-living phage will die upon attempted infection. The freely-living phage cannot collect resources, reproduce, or move their position in the world without an available susceptible host, making them obligate endosymbionts that can survive temporarily outside of a host. Thus the successful infection of a bacterial host is essential for their evolutionary survival.

Lytic Phage Lytic phage behavior follows the mechanics of the lytic cycle, where the phage redirects the bacterium's resources to produce phage offspring until the host cell is eventually burst during lysis. At each timestep, the lytic phage attempts to steal resources from its host. To be successful, the phage's interaction value must be more negative, or smaller, than its bacterial host's interaction value. Otherwise, the phage will be unable to steal any resources and therefore will be unable to reproduce. If the phage's genome will allow it to steal resources, then the amount stolen from its host will be proportional to the difference between the bacterium and bacteriophage interaction values. For example, if the bacterium interaction value is -0.3, then it will have 35 resources leftover after spending 15 resources on defense. Then if the phage interaction value is -0.7, it will successfully steal an additional 14 resources ($|(-0.3) - (-0.7)| * 35 = 14$). Therefore the lytic bacteriophage will be able to use 14 resources for phage reproduction, and its host is left with only 21 resources for bacterial reproduction.

Once the bacterium resources have been successfully redirected, the lytic phage will use these resources to create as many phage offspring as possible (each of which uses 10 resources) before bursting its host cell. Once offspring are created, they are dormant in the bacterial host until the cell bursts. The time at which a lytic phage will burst its host cell is determined by the phage's burst timer, which starts at 0 upon injection to their host. At each time step, the burst timer is incremented by a random number pulled from a normal distribution centered around 1 with a standard deviation of 1 (the addition of some noise to the timer is necessary to prevent artifacts from perfectly synchronized phage populations). Once the burst timer reaches a value of 100, the bacterial host cell will burst. Upon bursting, each phage offspring is released into the world where they become freely-living phage. The bacterial host and lytic phage then both die.

Lysogenized Prophage The lysogenic phage cycle includes 1) a potential interaction with the host resources, 2) a chance for the phage to induce and begin the lytic cycle, and 3) a chance that the phage is killed off - simulating the degradation of prophage DNA. The lysogenic phage in Symbulation have an active life cycle, which relaxes assumptions from previous work that prophage are dormant. In this model, lysogenic phage have the ability to interact with their host's resources, thereby affecting rates of reproduction and creating an environment in which lysogenic phage and bacterial hosts may develop a (tenuous) mutualistic relationship. The *incorporation* mechanism described here is an abstraction of the many ways that a lysogenized prophage can positively or negatively impact its host's fitness through the genetic material that the prophage contains and how that genetic material complements or interferes with the host's genes. In addition, the location that the prophage incorporates into the host's genome can directly change the host's gene expression, which can have varying impacts on the host fitness. The mechanism that we implemented in this work captures the core dynamics of this interaction, specifically the importance of how matched or mismatched the prophage and host genomes are to each other and the possibility that a phage type may have a positive impact on one host but a negative impact on another host, depending on chance and the hosts' genomes.

Incorporation Value The ability of prophage to influence host resources is controlled by a configuration setting and was enabled for only some experiments in this study. If the

direct effects of lysogenic phage on host resources is enabled, it proceeds as follows.

At each time step, the lysogenic phage's host may have spent some of its resources on defense, as detailed previously. A lysogenic phage will then be able to influence the host's remaining resources, ranging from removing all resources to doubling host resources. The amount of resources left for the host after the lysogenic phage has interacted with them is proportional to the difference between the phage's incorporation value and the bacterial host's incorporation value. More specifically, the resources left for the host will be equal to $r * (1 - |i_b - i_p|) * s$, where r is the host resources remaining after defense spending, i_b is the bacterial host incorporation value, i_p is the phage incorporation value, and s is the synergy value (which is set to 2 for all experiments). Therefore, the closer together the incorporation values are, the more resources the host accrues. For example, if a bacterium's interaction value is -0.3, it will first use 15 resources on defense and have 35 remaining resources. Then, if the bacterium's incorporation value is 0.8 and the lysogenic phage's incorporation value is 0.6, the bacterium will have 56 resources ($35 * (1 - |0.8 - 0.6|) * 2 = 56$). Because 56 resources for the host is more than 35 resources, the presence of the prophage would lead to a faster rate of reproduction for the bacterial host, as well as the prophage by vertical transmission.

Induction After influencing their host's resources, lysogenic phage have a chance of inducing back to the lytic cycle, determined by the probability based on their inherited chance of induction. The ability for an incorporated lysogenic phage to induce back into the lytic cycle means that any mutualistic relationship between bacterial host and bacteriophage is unstable and could be exploited by the phage.

Prophage Loss Last in the process of a lysogenic phage is the possibility of prophage loss, or DNA degradation. The probability of prophage loss is based on a global setting and remains constant for the entirety of an experiment. If the prophage is lost, it is immediately removed from the bacterial host, leaving it uninfected.

Configuration Settings

Each experiment described below begins with 1000 bacterial hosts and 500 bacteriophages, leading to a multiplicity of infection of 0.5. The world has a carrying capacity of 10,000 bacteria and has no spatial structure - meaning offspring are placed randomly in the world upon birth. We assume that only one phage can infect each host at a time, and any subsequent infection attempts lead to the death of the second phage, following the assumptions made by previous models. All experiments were run for 10,000 timesteps, and were replicated 30 times with varying random seeds. For all statistical significance tests, we conducted Wilcoxon rank-sum tests and applied a Bonferroni correction for multiple comparisons to all p-values.

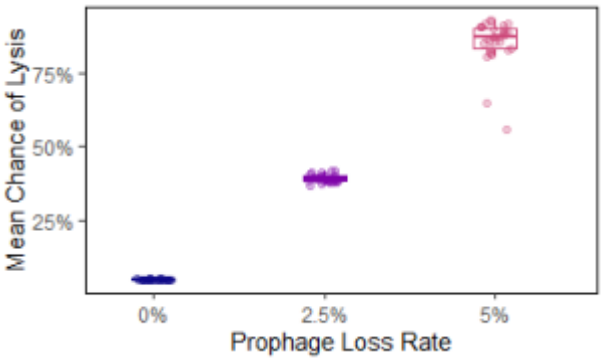


Figure 2: **Final average chance of lysis at three prophage loss rates.** Induction and prophage interaction were prevented to remain consistent with previous work.

Code for generating all the data and supplemental figures, including configuration settings, are available at `https://github.com/anyaevostinar/Evolution-of-Lysogeny-Paper`. All plots were created with R (R Core Team, 2020) and ggplot2 (Wickham, 2016) using the Viridis color library (Garnier et al., 2021). Symbulation is available at `http://www.symbulation.org` (Vostinar, 2021) and is built on the Empirical platform, which is available at `https://github.com/devosoft/Empirical` (Ofria et al., 2020).

Results and Discussion

To investigate the effect of beneficial or harmful prophage on the evolution of lysogeny, we conducted three sets of experiments: 1) verification that higher prophage loss rate selects for lysis when prophage are strictly dormant, 2) determination of the effect of a chance of induction on the evolution of lysogeny when prophage were still strictly dormant, and 3) determination of the effect of prophage that can be beneficial or harmful on the evolution of lysogeny.

Higher prophage loss rate selects for lysis

Previous work has shown that a higher density of uninfected hosts selects for phage with a higher propensity for lysis (Wahl et al., 2019; Sinha et al., 2017). In the previous work, the population of uninfected hosts came into the population through either migration or loss of prophage.

We first verified that our system was consistent with these previous findings by running experiments with varying amounts of prophage loss. Increased chance of prophage loss leads in turn to a higher proportion of naive/uninfected hosts, and so should select for increased chance of lysis in the phage population. We examined the resulting phage

population for propensity to enter the lytic or lysogenic life cycle upon infection. In these experiments, induction and prophage interaction were both prevented to match the assumptions of previous models.

As shown in Figure 2, when the prophage loss rate was 0%, bacteriophage populations evolved to have a final average of 4.87% chance of lysis. At a higher prophage loss rate of 5%, the chance of lysis evolved to a significantly higher rate of 85.64% ($p < 0.005$). At an intermediate prophage loss rate of 2.5%, the chance of lysis evolved to a value of 39.19%. These results agree with previous work that the density of naive hosts has a large impact on the evolution of phage reproductive strategies. Specifically, lysis is not beneficial when there are not enough uninfected hosts for the offspring to successfully infect and spread, because the offspring generally die. Therefore, when prophage loss rate is low, a lysogenic strategy is more successful. However, when prophage loss rate is higher, there are more uninfected hosts, making rapid spread through lysis a more fit strategy.

Effect of induction back into the lytic cycle

Most previous work modeling the evolution of the lysis/lysogeny switch has made the simplifying assumption that once a phage becomes lysogenic, it stays that way. However, in most bacteriophage, there is a small chance that a lysogenic phage will induce back into the lytic cycle when under certain kinds of stress, as discussed previously. Therefore, we determined whether the possibility of induction back to the lytic cycle would significantly change the evolution of lysis and lysogeny.

To determine the effect of induction, we enabled the possibility for lysogenic phage to induce and begin the lytic cycle, and repeated the same experiments with varying prophage loss rates. Because the starting chance of induction could influence the evolutionary trajectory, we conducted two separate treatments: one where the population of phage started with a 0% chance of induction and another where they started with a 10% chance of induction.

As Figure 3c shows, the final average chance of induction evolved to be at or below 10% regardless of prophage loss rate and the final average chance of induction was not meaningfully impacted by the starting chance of induction for all treatments. However, the final average chance of induction was impacted by different prophage loss rates. Specifically, when the chance of induction started at 0% and prophage loss rate (PLR) was either 2.5 or 5%, the chance of induction evolved to an average of 4.78% and 7.49%, respectively, both significantly above the rate of 1.56% when PLR was 0% (both $p < 0.005$). These results show that the common modeling assumption that prophage are not able to induce is not in agreement with selection pressures, however the impact may be small.

Further, as shown in Figure 3b, the possibility of the induction chance evolving significantly impacts the evolution of the lysis/lysogeny decision at only some prophage loss rates. When the prophage loss rate is 0%, the possibility of induction (starting at 0% or 10% probability) led to probabilities of lysis that are not meaningfully different than when induction is not possible. Specifically, with a prophage loss rate of 0% and without induction, the final average chance of lysis was 4.88%. With the possibility of induction, the final average chance of lysis was 4.92% and 4.93% with starting induction chances of 0% and 10% respectively. The similarity in these values demonstrate that the ability for prophage to induce and begin the lytic cycle does not have a meaningful effect on the lysis/lysogeny decision when the prophage loss rate is 0%, most likely because the lytic cycle is not beneficial.

However, the chance of lysis at a prophage loss rate of 2.5% is significantly impacted by the possibility of induction, with final average chance of lysis of 13.23% and 19.23% when chance of induction started at 0 and 10% respectively, compared to 39.19% when induction was not possible (both $p < 0.005$). In addition, the prophage loss rate of 5% showed meaningful difference in the evolution of the lysis/lysogeny decision when the chance of induction started at 10%, but not 0%. Without any induction to the lytic cycle, the bacteriophage evolved to have an average chance of lysis of 85.64% and when the chance of induction started at 0%, the chance of lysis evolved to 82.20%. This difference in the chance of lysis was also insignificant with $p = 0.302$. With induction starting at 10%, however, the chance of lysis evolved to 54.81%, significantly more than without the possibility of induction ($p < 0.005$).

These results indicate that the prophage inducing into the lytic cycle does have some effect on the evolution of the lysis/lysogeny decision, especially when other selection pressures are not as strong, such as when there are a limited number of uninfected hosts. Specifically, induction back into the lytic cycle has little impact in stable environmental conditions as it does not significantly affect the evolutionary dynamics between bacterial host and phage when there are other strong selective pressures. However, future work should investigate whether the ability to induce would have a significant impact on evolutionary dynamics in unstable environments, such as when there are varying numbers of uninfected hosts.

Beneficial Prophage Select for Increased Lysogeny

The historical assumption has been that lysogenic phage are dormant and have little to no effect on their bacterial hosts. However, recent work has shown that lysogenic phage have the potential to actively influence their hosts (Anderson et al., 2014; Harrison and Brockhurst, 2017; Owen et al., 2021; Obeng et al., 2016), and in some cases can confer beneficial traits such as serotype conversion (Knirel et al., 2015). This benefit largely depends on how well a phage can incorporate itself into the host DNA and how compati-

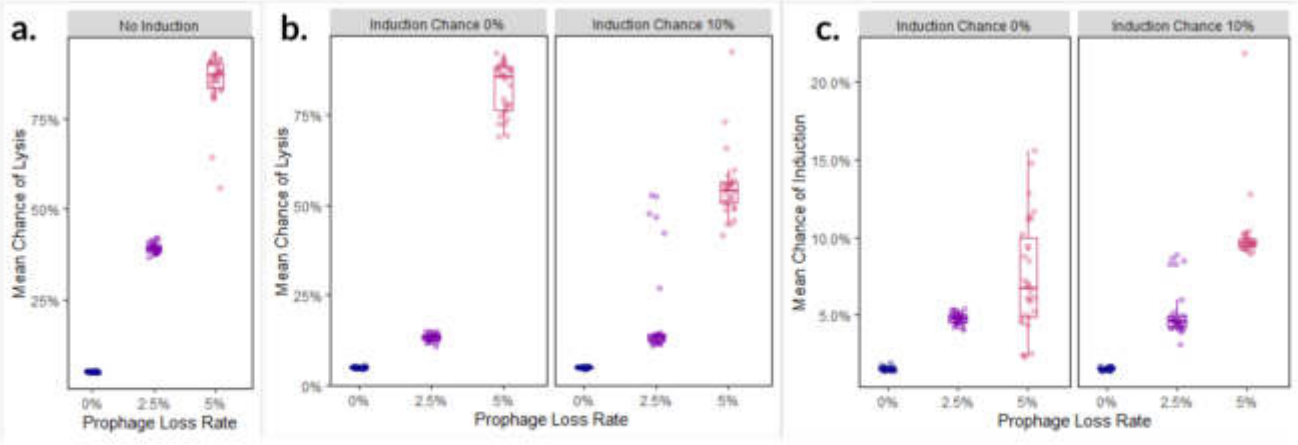


Figure 3: **Final average chance of lysis for (a) no induction enabled and (b) induction enabled. Final average chance of induction (c) when induction is enabled. All panels were tested across prophage loss rates.** Prophage were not able to impact host fitness. Induction chance in all phage for (b) and (c) was initialized to either 0% or 10% and then allowed to evolve. Note that the data in (a) is the same as Figure 2 and that the y-axis of (b) is from 0% to 25% chance.

ble these genomes are.

To determine the effects of active prophage, we expanded upon Symbulation to allow lysogenic phage to influence host metabolism, as detailed in the methods. Based on the compatibility between a host and phage, the host resources can range from being eliminated to doubled (Figure 1b). In this experiment, we enabled this direct interaction between bacteria and phage and did not enable the lysogenic phage to induce back to the lytic cycle. We had three classifications of starting populations: beneficial phage (doubling host resource), neutral phage (no effect on host resources), and harmful phage (cancelling out host resources). Once these populations were initialized, their compatibility genes, or *incorporation values*, were permitted to evolve. Once again, we tested these conditions across varying levels of prophage loss rate to investigate how non-dormant prophage would influence the lysis/lysogeny decision in known environmental settings.

As shown in Figure 4b, in populations of harmful phage, the lytic reproductive strategy is highly conserved. At a prophage loss rate of 5%, the chance of lysis evolved similarly for populations of harmful phage and populations of dormant phage, with average probabilities of 90.76% and 85.64%, respectively. However, at a more intermediate PLR of 2.5%, a population of harmful phage led to a significantly higher propensity for lysis (94.59%) than dormant phage (39.19%) ($p < 0.005$).

Conversely, in populations of beneficial phage, the lysogenic reproductive strategy is highly conserved. That is, at a prophage loss rate of 5%, populations of beneficial phage evolved to have a 38.19% chance of lysis, while populations of dormant phage evolved to a significantly higher 85.64% chance of lysis ($p < 0.005$). A similar significance follows with an intermediate PLR of 2.5%, where the average chance of lysis for beneficial phage and dormant phage evolved to 5.54% and 39.19%, respectively ($p < 0.005$).

These results indicate that the active influence of prophage over host fitness (whether harmful or beneficial) has significant impacts on the lysis/lysogeny decision. Populations of harmful phage are far more likely to evolve to lysis, while populations of beneficial phage are far more likely to evolve to lysogeny. Notably, even in environmental conditions that would typically select for lysis (prophage loss rate of 5%), a starting population of beneficial phage leads to a primarily temperate and lysogenic phage population (Figure 4b). However, for a prophage loss rate of 0%, the lysogenic strategy was conserved for all populations of phage. This result indicates that a lack of naive hosts is a stronger selection pressure for lysogeny than the active influence of prophage over host fitness.

The relationship between the chance of lysis and the phage/host compatibility is likely to be a reinforcing dynamic. Figure 4c shows how phage/host compatibility evolved across varying starting phage populations. As explained in the methods, the closer the incorporation values between host and phage, the higher the host-phage compatibility, and thus the more beneficial the impact of prophage on host fitness. In environmental conditions that led to higher chances of lysis (harmful phage with PLR 2.5% and 5%), the incorporation values evolved to be far apart - leading to very low compatability. Note that a highly incompatible phage does not gain more benefit from the host. Conversely, in conditions that lead to lower chances of lysis and thus higher chances of lysogeny (beneficial phage at all

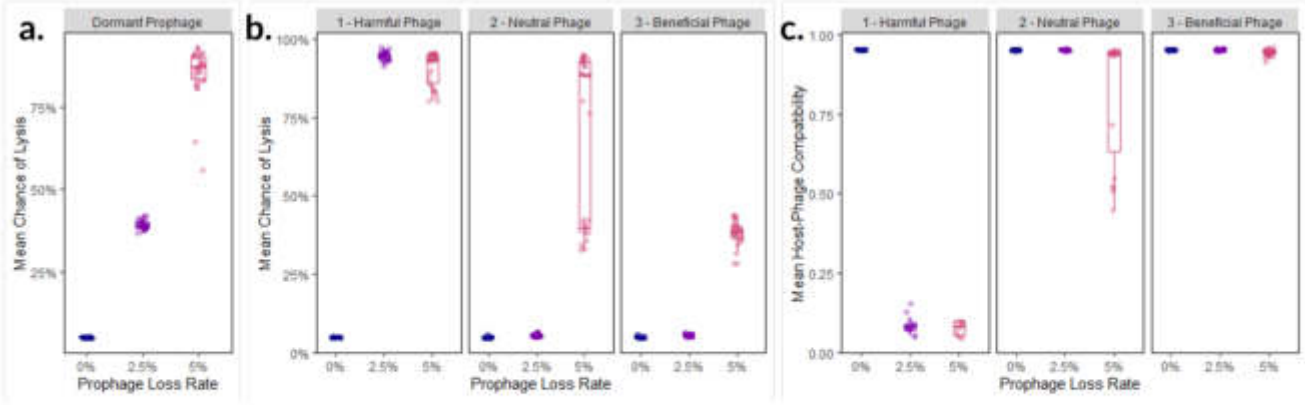


Figure 4: **Final average chance of lysis for (a) dormant prophage and (b) prophage impacting host resources. Final average host-phage compatibility (c) when prophage impact host resources. All panels were tested across prophage loss rates.** Prophage were not able to induce into the lytic cycle. Incorporation values for phage and host in (b) and (c) were initialized such that the phage were either harmful, neutral, or helpful, and then allowed to evolve. Note that the data in (a) is the same as Figure 2, and 100% compatibility indicates that the incorporation values were the same for a host-phage pair.

PLRs), the incorporation values evolved to be quite similar - leading to very high compatibility.

Therefore, environmental conditions and starting populations that select for lysis also select for low host-phage compatibility. Similarly, the conditions that select for lysogeny also select for high host-phage compatibility. These results imply that the lysis/lysogeny decision and compatibility between host and phage may be reinforcing phenomena. That is, a population of harmful phage may select for lysis and a population of lytic phage may select for incompatibility. Similarly, a population of beneficial phage may select for lysogeny and a population of lysogenic phage may select for strong compatibility. Furthermore, this implies that it is not evolutionarily advantageous for phage to be highly lytic, while maintaining traits that would allow it to be beneficially lysogenic, or vice versa.

Conclusion

In this work, we expanded upon the Symbulation software to investigate the effects on the lysis/lysogeny decision for (1) lysogenic induction into the lytic cycle and (2) non-dormant prophage. The system was first calibrated to match the assumptions of previous mathematical modeling work by showing that a regular influx of naive hosts leads to a more lytic population of phage. Then, we relaxed previous assumptions beyond what is possible in the mathematical frameworks by allowing for induction and for the prophage to have an active influence on host metabolism. From these experiments, we concluded that induction has some effect on the lysis/lysogeny decision and that, when prophage can impact host fitness, populations of harmful phage evolve towards lysis, while populations of beneficial phage more often evolve towards lysogeny. Furthermore, there is a negative relationship between the probability of lysis and host-phage compatibility in non-dormant phage. These results imply that both induction and the influence of prophage on host fitness play an active role in the evolution of the lysis/lysogeny decision, making it a vital area for further discovery.

Future work exploring these dynamics should be conducted both *in vitro* and *in silico*. Specifically, the above results and general evolutionary dynamics should be thoroughly tested in wetlab, for example with the *Shigella flexneri* system (Doore et al., 2021; Subramanian et al., 2020). Further computational research should address the limitations and assumptions of our work. First, the association demonstrated between the propensity for lysis and host-phage compatibility should be investigated to determine if there is a causal relationship in either direction. Second, the impact of multi-infection (more than one phage infecting a host) and various spatial structures should be explored to determine how they change these dynamics. And last, the model should be expanded to allow for more open-ended prophage incorporation dynamics.

A more complete understanding of the lysis/lysogeny decision is vital to our conceptions of evolutionary trajectories for both bacteriophage and bacteria. Because of the prevalence of phage and bacteria in the natural world, the co-evolutionary dynamics and potential mutualistic relationships between them has serious implications for human and environmental health as highlighted in this work.

Acknowledgements

This work was supported by NSF grant No. 1750125, and Carleton College's Towsley Endowment.

References

Allison, G. E. and Verma, N. K. (2000). Serotype-converting bacteriophages and o-antigen modification in shigella flexneri. *Trends in microbiology*, 8(1):17–23.

Anderson, R. E., Sogin, M. L., and Baross, J. A. (2014). Evolutionary strategies of viruses, bacteria and archaea in hydrothermal vent ecosystems revealed through metagenomics. *PloS one*, 9(10):e109696.

Bäckhed, F., Ley, R. E., Sonnenburg, J. L., Peterson, D. A., and Gordon, J. I. (2005). Host-bacterial mutualism in the human intestine. *science*, 307(5717):1915–1920.

Bruneaux, M., Ashrafi, R., Kronholm, I., Laanto, E., Örmälä-Odegrip, A.-M., Galarza, J. A., Chen, Z., Kubendran Sumathi, M., and Ketola, T. (2022). The effect of a temperature-sensitive prophage on the evolution of virulence in an opportunistic bacterial pathogen. *bioRxiv*.

Chevallereau, A., Pons, B. J., van Houte, S., and Westra, E. R. (2022). Interactions between bacterial and phage communities in natural environments. *Nature Reviews Microbiology*, 20(1):49–62.

Doore, S. M., Subramanian, S., Tefft, N. M., Morona, R., Ter-Avest, M. A., and Parent, K. N. (2021). Large metabolic rewiring from small genomic changes between strains of shigella flexneri. *Journal of bacteriology*, 203(11):e00056–21.

Ewald, P. W. (1987). Transmission modes and evolution of the parasitism-mutualism continuum a. *Annals of the New York Academy of Sciences*, 503(1):295–306.

Garnier, Simon, Ross, Noam, Rudis, Robert, Camargo, Pedro, A., Sciaini, Marco, Scherer, and Cédric (2021). *viridis - Colorblind-Friendly Color Maps for R*. `https://sjmgarnier.github.io/viridis/`.

Harrison, E. and Brockhurst, M. A. (2017). Ecological and evolutionary benefits of temperate phage: what does or doesn't kill you makes you stronger. *BioEssays*, 39(12):1700112.

Hobbs, Z. and Abedon, S. T. (2016). Diversity of phage infection types and associated terminology: the problem with 'lytic or lysogenic'. *FEMS microbiology letters*, 363(7).

Knirel, Y., Sun, Q., Senchenkova, S., Perepelov, A., Shashkov, A., and Xu, J. (2015). O-antigen modifications providing antigenic diversity of shigella flexneri and underlying genetic mechanisms. *Biochemistry (Moscow)*, 80(7):901–914.

Labrec, E. H., Schneider, H., Magnani, T. J., and Formal, S. B. (1964). Epithelial cell penetration as an essential step in the pathogenesis of bacillary dysentery. *Journal of bacteriology*, 88(5):1503–1518.

Lin, D. M., Koskella, B., and Lin, H. C. (2017). Phage therapy: An alternative to antibiotics in the age of multi-drug resistance. *World journal of gastrointestinal pharmacology and therapeutics*, 8(3):162.

Livio, S., Strockbine, N. A., Panchalingam, S., Tennant, S. M., Barry, E. M., Marohn, M. E., Antonio, M., Hossain, A., Mandomando, I., Ochieng, J. B., et al. (2014). Shigella isolates from the global enteric multicenter study inform vaccine development. *Clinical Infectious Diseases*, 59(7):933–941.

Lu, T. K. and Koeris, M. S. (2011). The next generation of bacteriophage therapy. *Current opinion in microbiology*, 14(5):524–531.

Mäntynen, S., Laanto, E., Oksanen, H. M., Poranen, M. M., and Díaz-Muñoz, S. L. (2021). Black box of phage–bacterium interactions: exploring alternative phage infection strategies. *Open biology*, 11(9):210188.

Maslov, S. and Sneppen, K. (2015). Well-temperate phage: optimal bet-hedging against local environmental collapses. *Scientific reports*, 5(1):1–11.

Maurelli, A. T. (2007). Black holes, antivirulence genes, and gene inactivation in the evolution of bacterial pathogens. *FEMS microbiology letters*, 267(1):1–8.

Miller, R. V. (2001). Environmental bacteriophage-host interactions: factors contribution to natural transduction. *Antonie Van Leeuwenhoek*, 79(2):141–147.

Muthuirulandi Sethuvel, D., Devanga Ragupathi, N., Anandan, S., and Veeraraghavan, B. (2017). Update on: Shigella new serogroups/serotypes and their antimicrobial resistance. *Letters in applied microbiology*, 64(1):8–18.

Nakata, N., Tobe, T., Fukuda, I., Suzuki, T., Komatsu, K., Yoshikawa, M., and Sasakawa, C. (1993). The absence of a surface protease, ompt, determines the intercellular spreading ability of shigella: the relationship between the ompt and kcpa loci. *Molecular microbiology*, 9(3):459–468.

Nanda, A. M., Heyer, A., Krämer, C., Grünberger, A., Kohlheyer, D., and Frunzke, J. (2014). Analysis of sos-induced spontaneous prophage induction in corynebacterium glutamicum at the single-cell level. *Journal of bacteriology*, 196(1):180–188.

Neu, H. C. (1992). The crisis in antibiotic resistance. *Science*, 257(5073):1064–1073.

Obeng, N., Pratama, A. A., and van Elsas, J. D. (2016). The significance of mutualistic phages for bacterial ecology and evolution. *Trends in microbiology*, 24(6):440–449.

Ofir, G. and Sorek, R. (2018). Contemporary phage biology: from classic models to new insights. *Cell*, 172(6):1260–1270.

Ofria, C., Moreno, M. A., Dolson, E., Lalejini, A., rodsan0, Fenton, J., perryk12, Jorgensen, S., hoffmanriley, grenewode, Edwards, O. B., Stredwick, J., cgnitash, theycallmeHeem, Vostinar, A., Moreno, R., Schossau, J., Zaman, L., and djrain (2020). *devosoft/Empirical: Before directory reorganization*. `https://doi.org/10.5281/zenodo.4141943`.

Owen, S. V., Wenner, N., Dulberger, C. L., Rodwell, E. V., Bowers-Barnard, A., Quinones-Olvera, N., Rigden, D. J., Rubin, E. J., Garner, E. C., Baym, M., et al. (2021). Prophages encode phage-defense systems with cognate self-immunity. *Cell host & microbe*, 29(11):1620–1633.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. `https://www.R-project.org/`.

Rohwer, F. and Segall, A. M. (2015). A century of phage lessons. *Nature*, 528(7580):46–47.

Sinha, V., Goyal, A., Svenningsen, S. L., Semsey, S., and Krishna, S. (2017). In silico evolution of lysis-lysogeny strategies reproduces observed lysogeny propensities in temperate bacteriophages. *Frontiers in microbiology*, 8:1386.

Stern, A. and Sorek, R. (2011). The phage-host arms race: shaping the evolution of microbes. *Bioessays*, 33(1):43–51.

Subramanian, S., Parent, K. N., and Doore, S. M. (2020). Ecology, structure, and evolution of shigella phages. *Annual review of virology*, 7:121–141.

Sulakvelidze, A., Alavidze, Z., and Morris Jr, J. G. (2001). Bacteriophage therapy. *Antimicrobial agents and chemotherapy*, 45(3):649–659.

Troeger, C., Blacker, B. F., Khalil, I. A., Rao, P. C., Cao, S., Zimsen, S. R., Albertson, S. B., Stanaway, J. D., Deshpande, A., Abebe, Z., et al. (2018). Estimates of the global, regional, and national morbidity, mortality, and aetiologies of diarrhoea in 195 countries: a systematic analysis for the global burden of disease study 2016. *The Lancet Infectious Diseases*, 18(11):1211–1228.

Vostinar, A. E. (2021). *Symbulation*. `https://doi.org/10.5281/zenodo.5062147`.

Vostinar, A. E. and Ofria, C. (2019). Spatial structure can decrease symbiotic cooperation. *Artificial Life*, 24(4):229–249.

Vostinar, A. E., Skocelas, K. G., Lalejini, A., and Zaman, L. (2021). Symbiosis in digital evolution: Past, present, and future. *Frontiers in Ecology and Evolution*, 9.

Wahl, L. M., Betti, M. I., Dick, D. W., Pattenden, T., and Puccini, A. J. (2019). Evolutionary stability of the lysis-lysogeny decision: why be virulent? *Evolution*, 73(1):92–98.

Weitz, J. S. and Dushoff, J. (2008). Alternative stable states in host–phage dynamics. *Theoretical Ecology*, 1(1):13–19.

West, N. P., Sansonetti, P., Mounier, J., Exley, R. M., Parsot, C., Guadagnini, S., Prévost, M.-C., Prochnicka-Chalufour, A., Delepierre, M., Tanguy, M., et al. (2005). Optimization of virulence functions through glucosylation of shigella lps. *Science*.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. `https://ggplot2.tidyverse.org`.

Yamamura, N. (1993). Vertical transmission and evolution of mutualism from parasitism. *Theoretical Population Biology*, 44(1):95–109.

Dirty Transmission Hypothesis: Increased Mutations During Horizontal Transmission Can Select for Increased Levels of Mutualism in Endosymbionts

Claire Schregardus[1], Michael Wiser[2] and Anya E. Vostinar[1]

[1]Carleton College, Northfield, MN, USA
[2]Michigan State University, East Lansing, MI, USA
anya.vostinar@gmail.com

Abstract

A mutualistic symbiosis occurs when organisms of different species cooperate closely for a net benefit over time. Mutualistic relationships are important for human health, food production, and ecosystem maintenance. However, they can evolve to parasitism or breakdown all together and the conditions that maintain and influence them are not completely understood. Vertical and horizontal transmission of mutualistic endosymbionts are two factors that can influence the evolution of mutualism. Using the artificial life system, Symbulation, we studied the effects of different rates of mutation during horizontal transmission on mutualistic symbiosis at different levels of vertical transmission. We propose and provide evidence for the "Dirty Transmission Hypothesis", which states that higher rates of mutation during horizontal transmission can select for increased mutualism to avoid deleterious mutation accumulation.

Introduction

Mutualistic endosymbiosis — close and long term cooperation between species where one organism lives inside of another – is a widespread and well-established phenomena in the biological world (de Vries and Archibald, 2017; Archibald, 2015; Zachar and Boza, 2020; Lazcano and Peretó, 2017; Johnson et al., 2021). These mutualistic relationships impact humans in a number of ways, including human health, food production, and the maintenance of ecosystems around the world (Toby Kiers et al., 2010). Common examples of mutualistic endosymbiosis include the human gut microbiome as well as the root-nodule bacteria of legumes (Drew et al., 2021; Trivedi et al., 2020).

While mutualism can be rewarding, there are risks to engaging in it. There is always a chance that one partner in a mutualistic relationship will cheat, increasing its own fitness to the detriment of the other and potentially shifting into parasitism or causing the mutualism to breakdown completely (Jones et al., 2015; Moran et al., 2008). Further, mutualistic endosymbiosis is a particularly tight relationship, changing the environment of one of the species completely and often leading to host species' dependence on its endosymbionts (O'Malley, 2015). Thus, there is the question of under what conditions mutualistic endosymbioses emerge and what factors influence them the most. Previous research has shown that mutualistic endosymbioses can be influenced by the rate of vertical transmission (Vostinar and Ofria, 2019; Bruijning et al., 2021; Shapiro and Turner, 2014). Vertical transmission is when a host reproduces and its offspring are infected by its symbiont (Fine, 1975). There is also a second mode of symbiont transmission, horizontal transmission. Horizontal transmission is transmission that is not linked to reproduction. (Ewald, 1987).

Symbionts can incur mutations in their genomes, which could in turn impact their fitness and relationships with their hosts (Drake, 1991; Drake et al., 1998; Drake and Holland, 1999; Sanjuán et al., 2010). It is possible that, in the short term, symbionts may accumulate more mutations during horizontal transmission because they must leave their hosts and expose themselves to the environment, potentially leaving them open to more damage to their genome, such as due to ultraviolet light (Marais et al., 2008; Witkin, 1969). In addition, some symbionts may experience further decreased mutation rates during vertical transmission due to host repair mechanisms. For example, temperate bacteriophage specifically can have the benefit of host genetic repair mechanisms while lysogenized, potentially decreasing their realized mutation rate when vertically transmitted compared to when horizontally transmitted through lysis (Duffy et al., 2008). Over the longer term, the effects of vertical vs. horizontal transmission can become more complicated, as mutation rates become more heavily influenced by changes in bottleneck size and/or recombination rates (Russell et al., 2020).

In such a system with a higher mutation rate during horizontal transmission and an intermediate chance of vertical transmission (as found in many natural systems (Bruijning et al., 2022)), a symbiont that has evolved to rely on horizontal transmission could have more offspring than a symbiont evolved to rely on vertical transmission. However, if most of the offspring transmitted horizontally acquire deleterious mutations, the symbiont with the vertical transmission strategy could actually have higher fitness. Symbionts with a vertical transmission strategy should then also be under selection to be more mutualistic to improve their host's fitness

and therefore their own.

To our knowledge, there is no previous research on how a higher mutation rate during horizontal transmission might impact the evolution of mutualistic relationships. Controlling and detecting the mutation rates during different transmission modes is challenging if not impossible in most biological systems (Peck and Lauring, 2018). Therefore, to test this hypothesis, we used an artificial life system called Symbulation, where a population of hosts and endosymbionts are able to co-evolve between antagonistic and mutualistic behavior (Vostinar, 2021; Vostinar et al., 2021). Using this system, we were able to test how horizontal transmission-associated mutation impacts mutualistic relationships. We determined that at intermediate vertical transmission rates, a higher relative mutation rate during horizontal transmission selects for a stable mutualism where otherwise parasitism dominates. These results support the "Dirty Transmission Hypothesis" and thereby provide an additional mechanism that could tip the balance towards mutualism when an endosymbiotic relationship is first evolving.

Methods

For this investigation, we used the Symbulation platform (Vostinar, 2021) to enable endosymbiotic relationships that could evolve between parasitism and mutualism. The evolutionary agent-based simulation consists of hosts and endosymbionts that each have their own genome consisting of one value, the interaction value. As shown in Fig. 1, this value dictates the amount of cooperation or antagonism that that organism will engage in and ranges from -1 to 1. We expanded Symbulation, as shown in Fig. 2, such that endosymbionts have an additional trait, their efficiency value, which determines how effective they are at processing resources into a usable form for themselves, and is an abstraction of the many traits that can contribute to endosymbiont fitness other than how they interact with their host.

At each time step, every host receives 100 resources that can be used for reproduction, defense, or distribution to its endosymbiont (if it has one). Each host can have up to one endosymbiont, restricting multiplicity of infection to 1 or less. Endosymbionts can receive or steal resources from hosts, as well as donate resources back to hosts. These behaviors are dependent on host and symbiont interaction values[1].

Interaction Value

Interaction values below 0 indicate antagonism between partners. An endosymbiont with a negative interaction value will attempt to steal that proportion of resources from its host, while a host with a negative value will invest that proportion of its resources into defense. When resources are

[1]This trait was referred to as *resource behavior value* in previous work. Here we use the term *interaction value*.

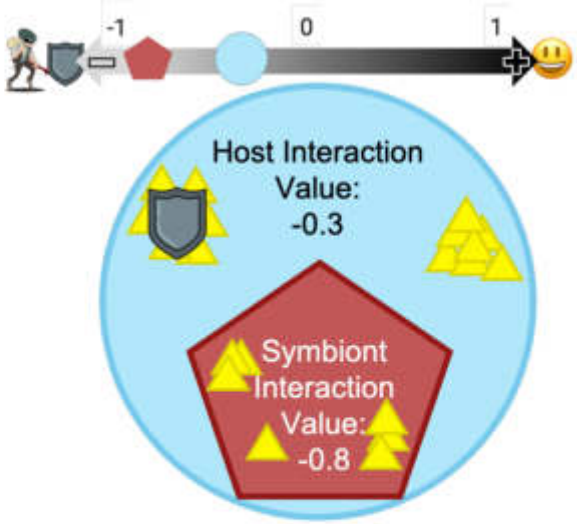


Figure 1: **Overview of host and symbiont interaction.** Each host can have up to one symbiont. The behavior of both organisms is determined by their interaction value. A negative interaction value indicates antagonistic behavior whereas a positive interaction value indicates mutualistic behavior.

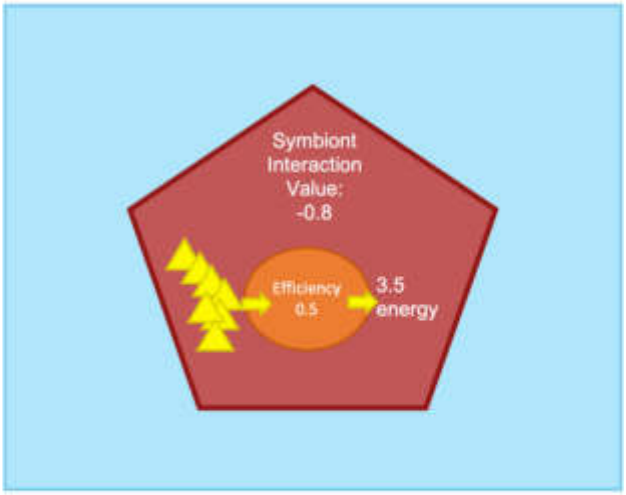


Figure 2: **Overview of the efficiency trait of symbionts.** Each symbiont has an efficiency trait, which determines how effective they are at using resources regardless of how they receive those resources. A lower efficiency value means a symbiont is able to glean less energy (for reproduction) from the resources it has.

Table 1: Results for host and symbiont interaction values

Host IV	Symbiont IV	Result
$X > 0$	$Y > 0$	Host donates proportion X to symbiont, symbiont donates proportion Y back, which is multiplied by 2
$X < 0$	$Y < 0$	Host invests proportion X in defense, symbiont steals proportion X - Y, host gets what remains
$X > 0$	$Y < 0$	Host donates proportion X to symbiont, symbionts steals additional proportion Y, host gets what remains
$X < 0$	$Y > 0$	Host invests X in defense, symbiont has no resources to donate, host gets remaining resources

used for defense they are no longer available to be used for reproduction or transmission. The amount of resources stolen from the host is the difference between the endosymbiont and host interaction values, assuming the endosymbiont's interaction value is more negative. If the host value is positive and the symbiont value is negative, the host donates the appropriate proportion of resources and the symbiont steals a further proportion of resources from those that the host attempted to keep for its own reproduction. Conversely, if the symbiont value is positive and the host value is negative, the host invests in defense and the symbiont receives no resources. All possible impacts of interaction value combinations are shown in Table 1, adapted from (Vostinar and Ofria, 2019).

Interaction values above 0 indicate cooperation between partners. A host with a positive interaction value will donate that proportion of its resources to its endosymbiont. An endosymbiont with a positive value will donate that proportion of resources back to the host, multiplied by a synergy factor of 2. The synergy factor is meant to represent the benefit of participating in mutualism and sharing resources. Previous research has evaluated the use of division of labor across multiple resources instead of an artificial synergy factor and found similar results (Vostinar and Ofria, 2019).

Reproduction

Both host and endosymbiont interaction values and the endosymbiont's efficiency value are subject to mutation upon reproduction and transmission. A host can reproduce after it accumulates 1000 resources, at which point its offspring is placed at a random location in the world, killing any organisms already existing in that space. The offspring's interaction value has a chance of mutating based on the mutation rate. If the interaction value mutates, a random number is generated from a normal distribution with a mean of 0 and a standard deviation of 0.002 and the value is changed by that amount. In all experiments in this work, the host mutation rate is fixed at a 10% chance. Note that because there is only a single value per genome, and genomes are haploid, there is no recombination in this system.

Transmission of symbionts can occur under two circumstances. First, when a host reproduces, its symbiont has a set chance between 0 and 100% of vertically transmitting a symbiont offspring to the host offspring. Second, a symbiont can horizontally transmit its offspring after accumulating 100 resources. A random host is selected for the symbiont offspring to infect; if that host is already infected with a symbiont, the offspring will die. Because both host and symbiont offspring (through horizontal transmission) are placed in random locations in the world, the environment is spatially unstructured and therefore akin to a well-mixed liquid environment. Hosts can only have one symbiont, and that symbiont cannot be removed.

Symbiont Mutation Rates

Mutations to the interaction and efficiency values of the symbiont can occur during both vertical and horizontal transmission. For this work, we controlled the mutation rates during horizontal and vertical transmission separately for both interaction and efficiency value. For the interaction value, this means that when transmission occurs, there is a chance for mutation of the interaction value dependent on what type of transmission is occurring. We held the vertical transmission-associated mutation rate constant at 10% for both traits. We then tested the degree of mutualism, measured by the interaction value, when changing the *horizontal transmission-associated mutation rate (HTMR)* for 1) both the efficiency and interaction value, 2) only the interaction value, and 3) only the efficiency value across the full spectrum of vertical transmission rates.

Experimental Settings

Experiments had 30 replicates, ran for 10,000 time steps, and had a population limit of 10,000 hosts. The environment was a 2D well-mixed torus, and experiments began with a full population of hosts and symbionts with randomly generated interaction values.

Symbulation is built on the Empirical library (Ofria et al., 2020) and all code and scripts

for this work are available under the MIT license at https://github.com/anyaevostinar/Dirty-Transmission-Hypothesis-Paper.

Statistical Analysis

All plots were created in RStudio (R Core Team, 2020) using the ggplot2 package (Wickham, 2016) and the Viridis package (Garnier et al., 2021). For all significance tests, we conducted Wilcoxon rank-sum tests. We applied a Bonferroni correction for multiple comparisons to all p-values.

Results and Discussion

To determine the effect of a higher mutation rate during horizontal transmission (HTMR) on the evolution of mutualism, we enabled hosts and endosymbionts to evolve when the mutation rates during horizontal and vertical transmission were both 10% and when the mutation rate during horizontal transmission was 50%. We tested the effect at vertical transmission rates from 10-100% at 10% intervals. We then also explored whether the effect of a higher mutation rate during horizontal transmission would increase when the mutation rate during horizontal transmission was raised further to 100%. All treatments started with populations of hosts and symbionts with random starting interaction values and symbiont's had starting efficiency of 100% and evolution proceeded for 10,000 timesteps. Hosts were restricted to having at most one endosymbiont, keeping the multiplicity of infection to at most 1.

Increased HTMR Selects for Increased Mutualism at Intermediate Vertical Transmission Rates

We first determined the effect of an increased horizontal transmission mutation rate by comparing the degree of mutualism that symbionts evolve when HTMR is 10 and 50% and all other mutation rates are held at 10% across vertical transmission rates.

As shown in Figure 3, when the vertical transmission rate is low (10-20%) or high (70-100%), a higher HTMR does not have a meaningful impact on the final degree of mutualism that evolves in the symbionts (some treatments have a significant difference, however the effect size is not meaningful). The result is probably due to the fact that the dominant selection pressure at these extreme vertical transmission rates is from either rarely or usually vertically transmitting. Specifically, in agreement with previous work (Vostinar and Ofria, 2019), when vertical transmission rate is high, symbionts evolve to donate nearly all of their resources to their hosts, losing the ability to horizontally transmit and thus negating any effect of a higher mutation rate during horizontal transmission. Conversely, when vertical transmission rate is quite low, even though vertical transmission may be beneficial, it is such a rare occurrence that symbionts are selected to be extremely parasitic anyway.

However, at intermediate vertical transmission rates of 30%, 40%, 50%, and 60%, an HTMR of 50% results in significantly more mutualistic symbionts than when the HTMR is the same as the other mutation rates at 10% ($p < .005$ for all comparisons).

As shown in Figure 4, for all treatments, the mean efficiency values of the symbionts remains above 95%, indicating that these results are not due to mutational breakdown. In agreement with the theory around a lack of purifying selection on endosymbionts at very high vertical transmission rates (O'Fallon, 2008), the lowest mean efficiency values are actually found at the highest vertical transmission rates. These results demonstrate that the Dirty Transmission Hypothesis does not conflict with the predictions of endosymbiont 'de-evolution' due to a lack of purifying selection at high vertical transmission rates.

These results indicate that when the chance of vertical transmission is near 50%, and therefore the selection pressure from vertical transmission or lack thereof is weaker, a higher HTMR can tip the balance towards mutualism, supporting the Dirty Transmission Hypothesis. Specifically, when the vertical transmission rate is 30%, a higher HTMR makes mutualism possible where it otherwise wouldn't be and when the vertical transmission rate is 50 or 60%, a higher HTMR pushes mutualism from a possibility to a near certainty.

Impact of 100% HTMR on Evolution of Mutualism

To explore the full effects of a higher mutation rate during horizontal transmission, we also determined the effect of a 100% HTMR. We focused on the vertical transmission rates below 60% due to the lack of meaningful impact above that rate due to the low amount of horizontal transmission. We again started host and symbiont populations at random interaction values, enabled evolution for 10,000 timesteps, and measured the average interaction value of the symbionts after evolution.

As shown in Figure 3, an HTMR of 100% does not lead to a significant difference in the amount of mutualism evolved compared to an HTMR of 50% ($p >= 0.05$). Note that when the vertical transmission rate is 40%, the individual treatment difference appears significant, however it does not remain significant when corrected for multiple comparisons, as discussed in the Methods. This result indicates that the effect of an increased mutation rate during horizontal transmission does not necessarily depend on the amount of mutation rate increase.

Differential Effects of Increased HTMR on Host-Associated Traits

An increased rate of mutation during horizontal transmission impacts both the symbiont's interaction value (i.e. its host-associated traits) and its efficiency value (i.e. its adaptive traits that do not impact its interaction with the host). To

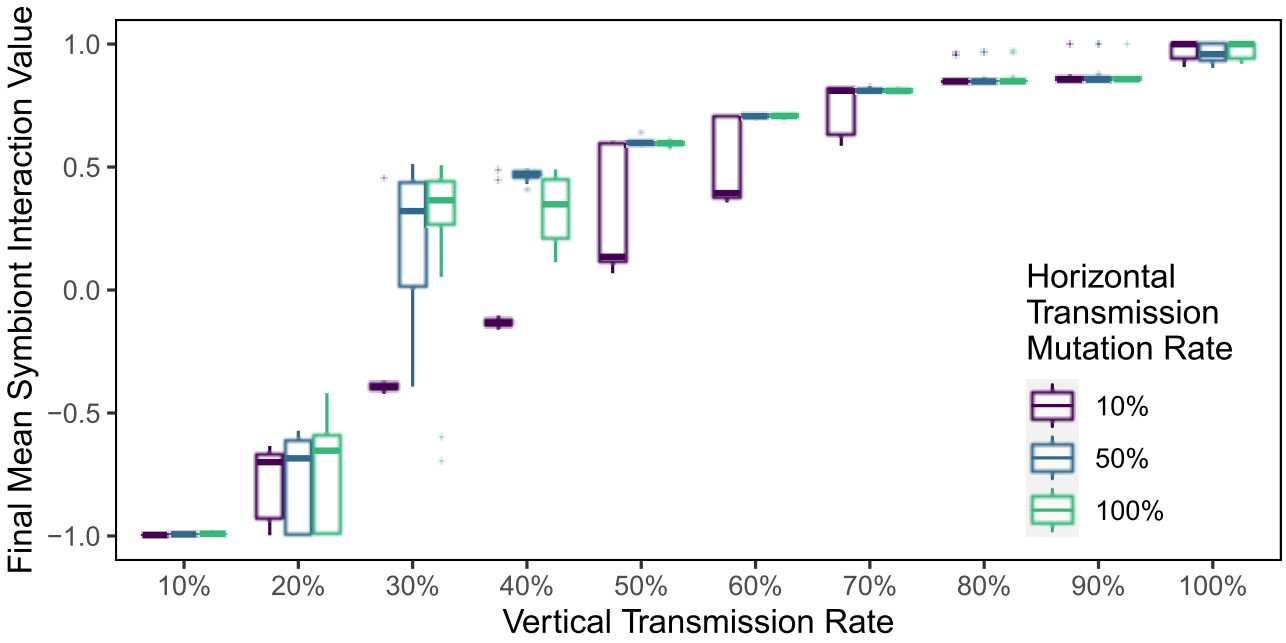


Figure 3: **Mean symbiont interaction value across vertical transmission rates when the rate of mutation during horizontal transmission was increased.** The mutation rate during vertical transmission and host reproduction was held at 10%. The difference between HTMR 10% and 50% is significant at vertical transmission rates of 10, 30, 40, 50, and 60% ($p < 0.005$ for all comparisons after Bonferroni correction). The difference between HTMR 50% and 100% is not significant at any vertical transmission rates after correction for multiple comparisons.

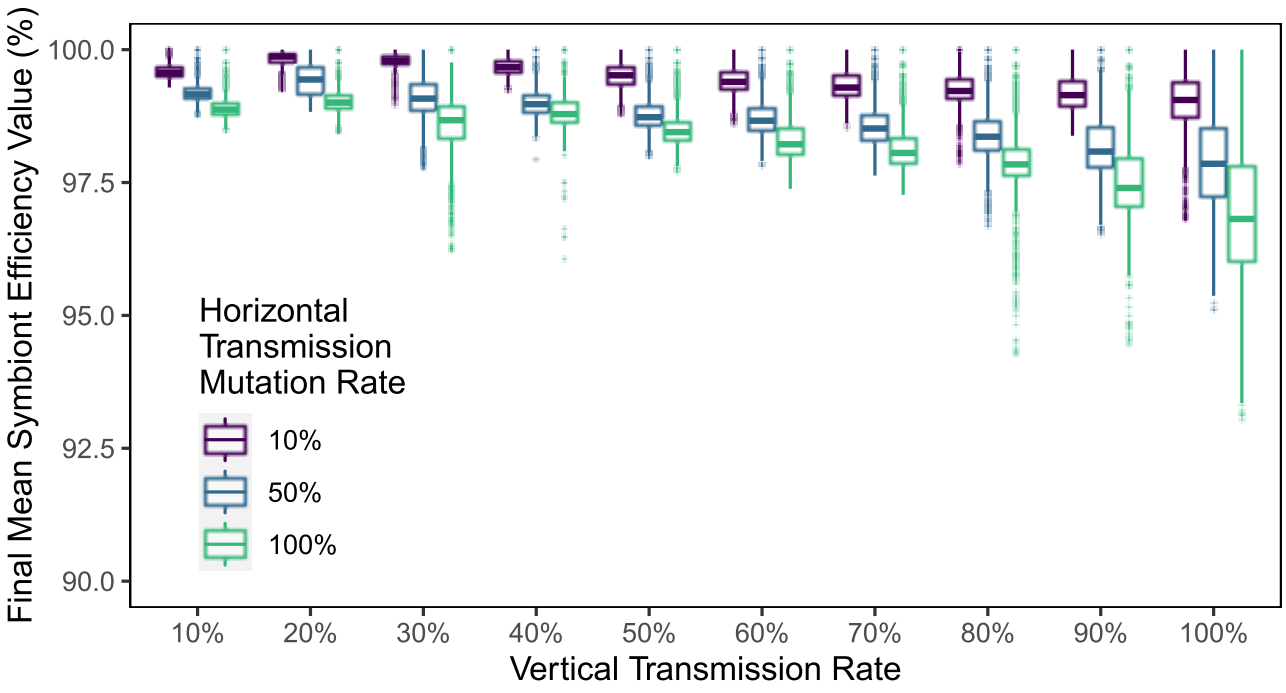


Figure 4: **Mean efficiency value of symbionts at final timestep.** Efficiency value is the percentage of resources a symbiont is able to convert to energy for use in reproduction. In all treatments, average efficiency value did not decrease below 90%.

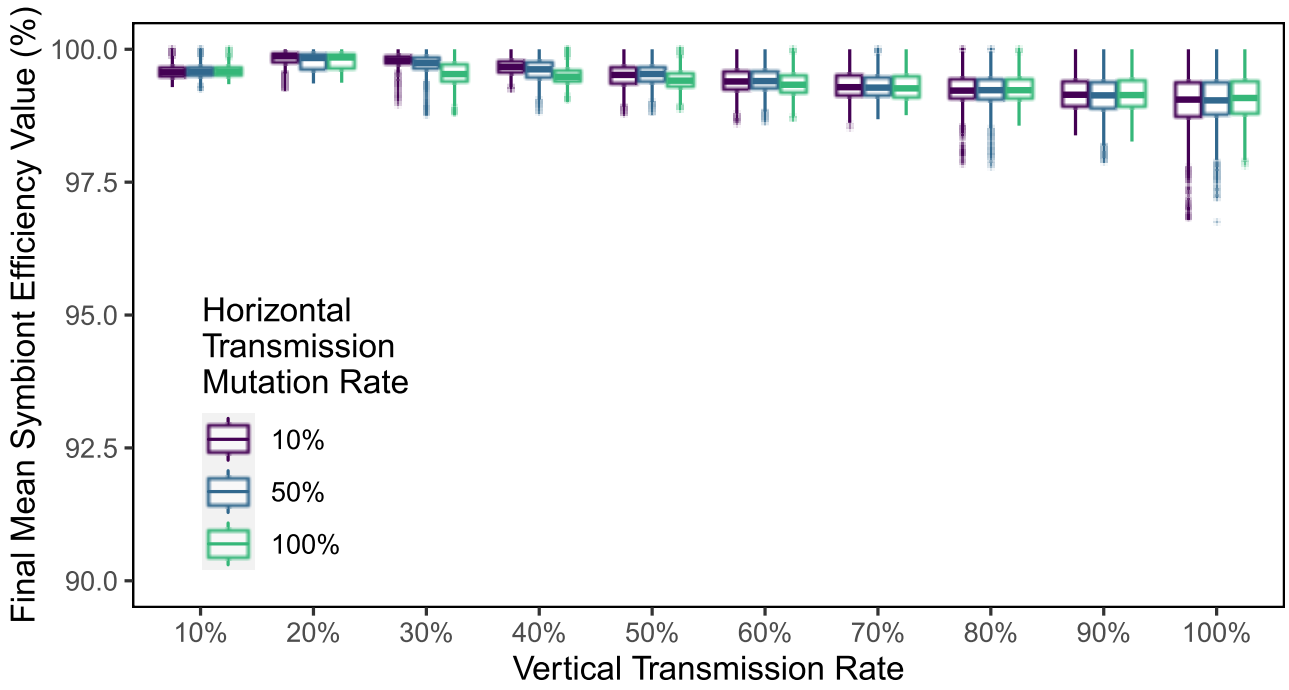


Figure 5: **Mean efficiency value of symbionts at the final timestep when only the interaction value was under an increased mutation rate during horizontal transmission and the HTMR of efficiency value was held at 10%.** In all treatments, average efficiency value did not decrease below 95%.

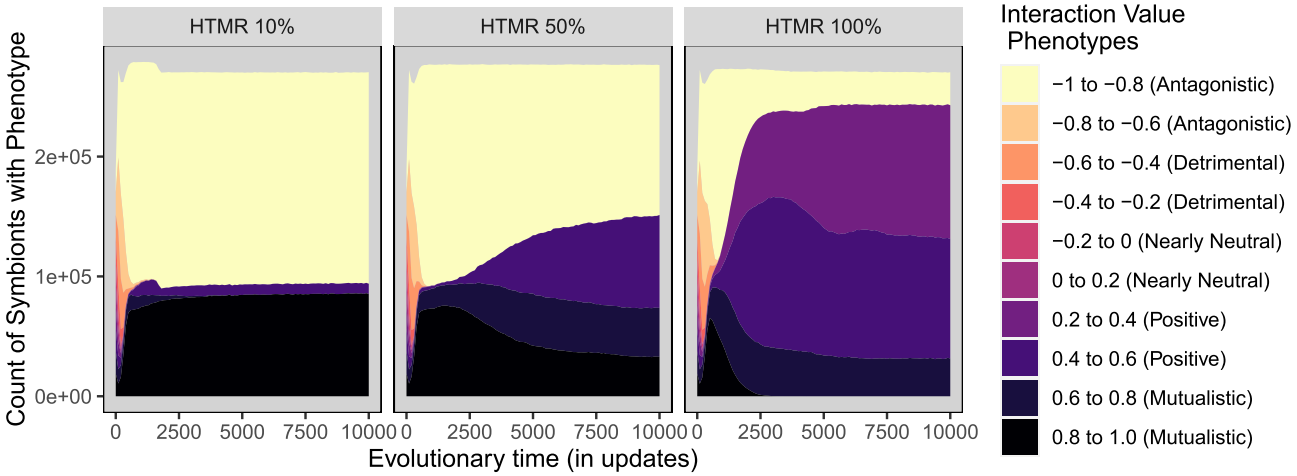


Figure 6: **Count of symbiont phenotypes over time at a vertical transmission rate of 30% and when only the interaction value is subject to increased mutation during horizontal transmission.** All other mutation rates were held constant at 10%.

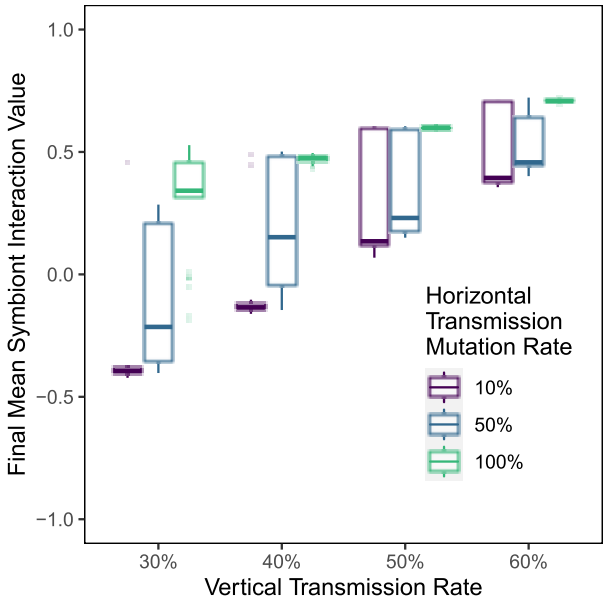


Figure 7: **Mean symbiont interaction value at intermediate vertical transmission rates when the rate of mutation during horizontal transmission was increased only for the interaction value.** The mutation rate for symbiont efficiency value, host traits, and the mutation rate during vertical transmission was held at 10%.

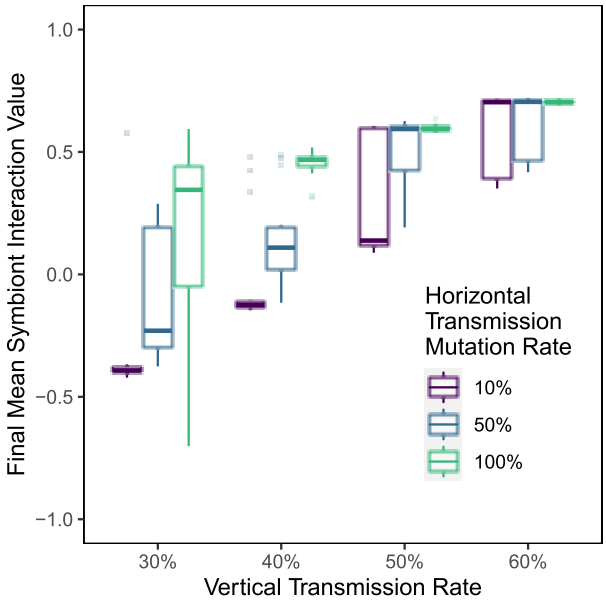


Figure 8: **Mean symbiont interaction value across vertical transmission rates when the rate of mutation during horizontal transmission was increased only for the efficiency value.** The mutation rate for symbiont interaction value, as well as during vertical transmission and host reproduction, was held at 10%.

determine the impact of increased HTMR on each of these traits individually, we repeated the previously described experiments with the mutation rate of the efficiency value held constant at 10%. Therefore, only the symbiont's interaction value was subject to the increased mutation rate during horizontal transmission.

As expected, and shown in Figure 5, when the efficiency value is not under increased HTMR, the final evolved efficiency values remain above 95% in all treatments. However, as shown in Figure 7, the final interaction value of symbionts is still impacted by the increased HTMR at intermediate vertical transmission rates. When the increased HTMR only effects the symbiont's host-associated trait (interaction value), 100% HTMR selects for a significantly higher final median symbiont interaction value at vertical transmission rates of 30, 40, 50, and 60% (all $p < 0.05$ after correction for multiple comparisons). As an example, Figure 6 shows the distribution of symbiont phenotypes over time at each HTMR when vertical transmission is 30%, demonstrating that the populations are stably dominated by mutualistic symbionts when HTMR is 100%, but not at the lower HTMR values. Note that the final median interaction values are not significantly different between the following treatments when the HTMR is 30% and the HTMR is 50%: 1) when both traits are subjected to increased HTMR and 2) only the interaction value is ($p > 1$), meaning that the difference seen in this treatment is not due to a change in the degree of mutualism at 50% HTMR. These results indicate that when only host-associated traits are impacted by increased HTMR, a further increase from 50% to 100% does have an impact at intermediate vertical transmission rates. They also demonstrate that the effect on the host-associated trait of the interaction value contributes to the overall increased rate of mutualism, but does not fully explain it.

Effects of Increased HTMR on Non-Host Associated Traits

Finally, we investigated the effect of the increased mutation rate during horizontal transmission on the symbiont's non-host associated trait: the efficiency value. We held the HTMR of the symbionts' interaction value constant at 10% and conducted the same experiments with HTMR levels of 10, 50, and 100% on the efficiency value.

As shown in Figure 8, the effects of increased HTMR on the efficiency value are generally qualitatively consistent with the effects of overall increased HTMR when vertical transmission rates are 60% or above, or 20% or below. However, at 30 and 40% vertical transmission rates, 100% HTMR leads to significantly more mutualistic symbionts than at 50% HTMR (all $p < 0.05$ after correction

for multiple comparisons). Specifically, when vertical transmission rate is 30%, an HTMR of 100% leads to a median interaction value of 0.34, whereas when HTMR is 50%, the median interaction value is -0.21. These results, when combined with the previous section, indicate that the increased mutualism evolved during higher HTMR at intermediate vertical transmission rates is due to both the effect on the interaction value and the efficiency value. However, when both traits are under an increased mutation rate at 50% HTMR, the combined effect is qualitatively equivalent to a HTMR of 100% on only one of the traits. This means that if only a host-associated or non-host-associated trait is under increased mutational load, there can be increased selection for mutualism at the most extreme HTMR.

Conclusion

In this work, we presented a novel mechanism for the evolution of mutualism, termed the Dirty Transmission Hypothesis. Specifically, we demonstrated that high mutation rates associated with horizontal transmission can select for higher levels of mutualism when vertical transmission rates are at intermediate values. We also examined the effect of extreme mutation rates during horizontal transmission and the contributing effects of increased mutation rates on host-associated and non-host-associated symbiont traits. We demonstrated that the increased mutualism that is evolved when the rate of mutation during horizontal transmission increases is due to the combined effects on symbiont traits that are associated with the host and symbiont traits that are independent of its interaction with the host.

There are many future directions to explore regarding the effect of the Dirty Transmission Hypothesis. As with all models, this work made necessary simplifying assumptions and to our knowledge, the effect of increased mutation rate during horizontal transmission has not previously been measured in any lab or natural system. Therefore, observation and experimentation in lab and natural systems will be needed in the future. Further, this work focused on single-infecting obligate endosymbionts, however multi-infection and symbionts that are capable of surviving outside of the host are common occurrences in natural systems and therefore fertile ground for further exploration.

Many natural systems have vertical transmission rates that appear insufficient to select for mutualistic behavior (Bruijning et al., 2022) and yet mutualism is found in those systems. There are many mechanisms that can lead to increased selection for mutualism, however they often require organisms capable of complex behavior. Here, we have experimentally demonstrated that the simple environmental effect of higher mutation rate during horizontal transmission can directly select for increased mutualism at realistic vertical transmission rates. This work contributes to our understanding of under what conditions mutualism can be expected to evolve and persist, and indicates how we may be able to predict and control its evolutionary trajectory in symbiotic systems vital to human health and society.

Acknowledgements

This work was supported by NSF grant No. 1750125.

References

Archibald, J. M. (2015). Endosymbiosis and eukaryotic cell evolution. *Current Biology*, 25(19):R911–R921.

Bruijning, M., Henry, L. P., Forsberg, S. K., Metcalf, C. J. E., and Ayroles, J. F. (2021). Natural selection for imprecise vertical transmission in host–microbiota systems. *Nature ecology & evolution*, pages 1–11.

Bruijning, M., Henry, L. P., Forsberg, S. K., Metcalf, C. J. E., and Ayroles, J. F. (2022). Natural selection for imprecise vertical transmission in host–microbiota systems. *Nature ecology & evolution*, 6(1):77–87.

de Vries, J. and Archibald, J. M. (2017). Endosymbiosis: did plastids evolve from a freshwater cyanobacterium? *Current Biology*, 27(3):R103–R105.

Drake, J. W. (1991). A constant rate of spontaneous mutation in dna-based microbes. *Proceedings of the National Academy of Sciences*, 88(16):7160–7164.

Drake, J. W., Charlesworth, B., Charlesworth, D., and Crow, J. F. (1998). Rates of spontaneous mutation. *Genetics*, 148(4):1667–1686.

Drake, J. W. and Holland, J. J. (1999). Mutation rates among rna viruses. *Proceedings of the National Academy of Sciences*, 96(24):13910–13913.

Drew, G. C., Stevens, E. J., and King, K. C. (2021). Microbial evolution and transitions along the parasite–mutualist continuum. *Nature Reviews Microbiology*, pages 1–16.

Duffy, S., Shackelton, L. A., and Holmes, E. C. (2008). Rates of evolutionary change in viruses: patterns and determinants. *Nature Reviews Genetics*, 9(4):267–276.

Ewald, P. W. (1987). Transmission modes and evolution of the parasitism-mutualism continuum a. *Annals of the New York Academy of Sciences*, 503(1):295–306.

Fine, P. E. (1975). Vectors and vertical transmission: an epidemiologic perspective. *Annals of the New York Academy of Sciences*, 266(1):173–194.

Garnier, Simon, Ross, Noam, Rudis, Robert, Camargo, Pedro, A., Sciaini, Marco, Scherer, and Cédric (2021). *viridis - Colorblind-Friendly Color Maps for R.* `10.5281/zenodo.4679424`.

Johnson, C. A., Smith, G. P., Yule, K., Davidowitz, G., Bronstein, J. L., and Ferrière, R. (2021). Coevolutionary transitions from antagonism to mutualism explained by the co-opted antagonist hypothesis. *Nature communications*, 12(1):1–11.

Jones, E. I., Afkhami, M. E., Akçay, E., Bronstein, J. L., Bshary, R., Frederickson, M. E., Heath, K. D., Hoeksema, J. D., Ness, J. H., Pankey, M. S., et al. (2015). Cheaters must prosper: reconciling theoretical and empirical perspectives on cheating in mutualism. *Ecology letters*, 18(11):1270–1284.

Lazcano, A. and Peretó, J. (2017). On the origin of mitosing cells: A historical appraisal of lynn margulis endosymbiotic theory. *Journal of theoretical biology*, 434:80–87.

Marais, G. A., Calteau, A., and Tenaillon, O. (2008). Mutation rate and genome reduction in endosymbiotic and free-living bacteria. *Genetica*, 134(2):205–210.

Moran, N. A., McCutcheon, J. P., and Nakabachi, A. (2008). Genomics and evolution of heritable bacterial symbionts. *Annual review of genetics*, 42:165–190.

O'Fallon, B. (2008). Population structure, levels of selection, and the evolution of intracellular symbionts. *Evolution: International Journal of Organic Evolution*, 62(2):361–373.

Ofria, C., Moreno, M. A., Dolson, E., Lalejini, A., rodsan0, Fenton, J., perryk12, Jorgensen, S., hoffmanriley, grenewode, Edwards, O. B., Stredwick, J., cgnitash, theycallmeHeem, Vostinar, A., Moreno, R., Schossau, J., Zaman, L., and djrain (2020). devosoft/Empirical: Before directory reorganization. `https://doi.org/10.5281/zenodo.4141943`.

O'Malley, M. A. (2015). Endosymbiosis and its implications for evolutionary theory. *Proceedings of the National Academy of Sciences*, 112(33):10270–10277.

Peck, K. M. and Lauring, A. S. (2018). Complexities of viral mutation rates. *Journal of virology*, 92(14):e01031–17.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. `https://www.R-project.org/`.

Russell, S. L., Pepper-Tunick, E., Svedberg, J., Byrne, A., Ruelas Castillo, J., Vollmers, C., Beinart, R. A., and Corbett-Detig, R. (2020). Horizontal transmission and recombination maintain forever young bacterial symbiont genomes. *PLOS Genetics*, 16(8):e1008935. Publisher: Public Library of Science.

Sanjuán, R., Nebot, M. R., Chirico, N., Mansky, L. M., and Belshaw, R. (2010). Viral mutation rates. *Journal of virology*, 84(19):9733–9748.

Shapiro, J. W. and Turner, P. E. (2014). The impact of transmission mode on the evolution of benefits provided by microbial symbionts. *Ecology and evolution*.

Toby Kiers, E., Palmer, T. M., Ives, A. R., Bruno, J. F., and Bronstein, J. L. (2010). Mutualisms in a changing world: an evolutionary perspective. *Ecology letters*, 13(12):1459–1474.

Trivedi, P., Leach, J. E., Tringe, S. G., Sa, T., and Singh, B. K. (2020). Plant–microbiome interactions: from community assembly to plant health. *Nature reviews microbiology*, 18(11):607–621.

Vostinar, A. E. (2021). *Symbulation*. `https://doi.org/10.5281/zenodo.5062147`.

Vostinar, A. E. and Ofria, C. (2019). Spatial structure can decrease symbiotic cooperation. *Artificial life*, 24(4):229–249.

Vostinar, A. E., Skocelas, K. G., Lalejini, A., and Zaman, L. (2021). Symbiosis in digital evolution: Past, present, and future. *Frontiers in Ecology and Evolution*, 9.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. `https://ggplot2.tidyverse.org`.

Witkin, E. M. (1969). Ultraviolet-induced mutation and dna repair. *Annual Review of Genetics*, 3(1):525–552.

Zachar, I. and Boza, G. (2020). Endosymbiosis before eukaryotes: mitochondrial establishment in protoeukaryotes. *Cellular and Molecular Life Sciences*, pages 1–21.

Testing the Efficiency of a Genome-Wide Association Study on a Computational Evolutionary Model

Arend Hintze [1,2] and Yasir Imam[1] and Lars Rönnegård[1,3]

[1]Dalarna University, Department of MicroData Analytics, Dalarna, 79188, Sweden
[2]Michigan State University, BEACON Center for the Study of Evolution in Action, East Lansing, 48824, United States of America
[3]Swedish University of Agricultural Sciences, Department of Animal Breeding and Genetics, Uppsala, 75007, Sweden
email of corresponding author: ahz@du.se

Abstract

Genome-wide association studies (GWAS) are a powerful tool for identifying genes. They exploit the standing genetic variation and correlate phenotypic diversity to genetic markers close to or with genes of interest. However, their power is limited when it comes to complex phenotypes caused by highly epistatically interacting genes. To improve GWAS and to develop new methods, a computational model system could prove invaluable. In the computational model system presented here, the functionality of all genes in question can be identified using knockouts. This allows the comparison between the quantitative genetics results and the functional analysis. Here the goal is to perform a pilot study to investigate to which degree such a computational model can serve as a positive control for a GWAS. Surprisingly, even though the model used here is relatively simple and uses only a few genes, the GWAS struggles to identify all relevant genes. The advantages and limitations of this approach will be discussed to improve the model for future comparisons.

Introduction

The phenotype of a knockout mutant is the ultimate tool to identify the function of a gene. However, a systematic knockout analysis can only be performed for very few organisms. Specifically where the genetic manipulation is simple, and the organism only has a comparably low number of genes (Giaever and Nislow, 2014). Redundant genes, pleiotropic effects, and differential gene expression further complicate this approach. Often only the first phenotype of a gene can be seen as later effects might be obscured by earlier effects. For example, a gene might have an essential function in early cell divisions, such that the developing embryo never divides in the first place. Consequently, the genes' potential role in organ differentiation remains unknown. Further, weak effects are overlooked easily – regardless, this method remains the golden standard for determining the function of a gene.

Genome-wide association studies present a complementary approach. Instead of complex genetic manipulations, the diversity of the population itself, together with knowledge about genetic markers, can be used. In this quantitative approach, the distribution of a specific trait is *correlated* to the distribution of genetic markers (where the term correlated here covers a much wider range of complex mathematical methods). In the simplest case, a single site in the genome for which only two alleles can be found perfectly correlates with a binary trait. Either this genetic variation is directly causing the phenotypic difference – very much like in Mendelian genetics – or the actual mutation responsible for the phenotypic difference is close to the genetic marker in question.

This approach, highly related to quantitative trait mapping, also works for more complex traits and multiple genes. Human height is one of the most well-studied traits, with hundreds of significant genomic locations detected (Wood et al., 2014; Patxot et al., 2021). Unfortunately, it seems as if GWAS are limited in their ability to unveil the relation between phenotypes and genes for many complex diseases, and the missing heritability (or dark matter) not captured by GWAS has been thoroughly discussed (Manolio et al., 2009).

Various reasons for the inefficiency of GWAS have been identified before (Manolio et al., 2009; Uffelmann et al., 2021; Tam et al., 2019), such as:

- Insufficient modeling of genetic interactions (epistasis)
- Most complex traits are governed by a large number of genes, each with a small effect (known as infinitesimal or polygenic effects)
- Extreme significant thresholds are required to account for the massive multiple testing problem where a P-value is computed for each tested genomic position
- A causative gene needs to be in linkage disequilibrium with a tested marker unless all positions along the genome are tested.

Furthermore, several methods have been developed to account for population structure and relatedness between individuals (Amin et al., 2007; Sul et al., 2018; Rönnegård et al., 2016), since an important assumption of the standard GWAS method is that the observations from different individuals can be treated as independent.

While all of them seem plausible, it remains unclear how to further narrow down this list beyond simply performing more GWAS studies and trying more advanced models. What is needed is a testbed or positive control such that we can first explore how much a GWAS can actually unveil and, secondly, which changes in analysis strategy are actually effective. The ideal would be a biological organism for which all gene functions are known. However, this creates the notorious chicken or egg conundrum because one would need a GWAS to identify all genes and their functions in the first place to accomplish such a task. Instead, using a computational model would be much more convenient, as it saves the cost for molecular manipulations, while at the same time, the number of samples one could take to satisfy statistical requirements is only bound by computational resources. The problem is that the computational model can not just be a randomly created mapping between genes and traits but should closely resemble the functional complexity of a natural organism. Here we propose to use a Markov Brain for that purpose. Markov Brains are evolvable neural controllers which form an artificial neural network. The connections of a Markov Brain between sensor inputs, hidden states, and motor outputs are formed by computational units. These units relay information and perform computations, analogous to how neuronal cells do the same. Each computational unit is encoded (specified) by a gene. Consequently, Markov Brains possess a genome made from none coding regions interspersed with coding genes. Its computational units, like proteins, are either directly controlling behavior or can perform complex computations by exchanging information between them.

While Markov Brains are clearly a much-simplified abstraction when compared to a real biological organism, they share key features relevant to the task at hand. Like biological genomes that mutate from generation to generation, Markov Brains experience the same type of mutations. Specifically, here point mutations, gene duplications, and deletions are applied. At the same time, because the computational units of the Markov Brain exchange information and perform computations on them to control a virtual agent, a wide range of interactions between them can be observed. Like in natural organisms, where the epistatic interactions between genes make it hard for GWAS to identify which genes control what phenotypic traits, in a Markov Brain, the link between the actions a virtual agent takes and how that is linked to the computational components could be equally obfuscated. The question is if a Markov Brain is a sufficiently complex model or if a GWAS can easily identify all genes and how they relate to different traits, and if, therefore, this system can be used as a testbed for further studies?

The results from the GWAS performed in Markov Brains still need to be compared to an objective truth about genes mapping to phenotypic traits. For that, we use a systematic knockout analysis. For each Markov Brain in the population, the phenotypic effect of all possible single knockout is determined. Consequently, the result of a KO analysis can be compared to the results of a GWAS. Ideally, the results from the GWAS should perfectly match those from the knockout analysis. Given the previously suggested weaknesses of GWAS, one might expect to observe the same shortcomings in this computational comparison. However, if Markov Brains turn out to be insufficiently complex, the GWAS could perform flawlessly, identifying all genes and how they relate to the traits observed.

The simplicity of this computational model suggests that the few genes that define the behavior of an agent, and with it, its phenotypic traits, should be identified easily. Further, each trait, due to the high number of replicate measurements, can be determined precisely. Lastly, the complete genomic information about all individuals, which also is noise-free, creates the perfect basis for a GWAS to run flawlessly.

At the same time, there are two properties the model is different from its biological counterpart. Here, the total population is not only rather small (10.000) but thus also highly related, which can limit the efficiency of a GWAS (Amin et al., 2007). Further, there are no physical or physiological properties used to describe traits but purely behavioral. While in humans, for example, obesity (Loos and Yeo, 2014) or height (Wood et al., 2014; Patxot et al., 2021), are well-described traits that are obvious targets for a GWAS, here we selected behavioral traits that might sound arbitrary. For example, how often an agent turns left might not sound as characteristic or important as baldness (Pirastu et al., 2017). However, the traits we are interested in studying in humans or animals are equally subjective and arbitrary as the ones we collect here.

Considering all the above, it remains hard to predict the outcome of this artificial GWAS, and since this kind of analysis[1] has not been done before, we consider this an exploratory endeavor. We simply want to identify possible shortcomings or peculiarities of this approach before we commit more computational resources and research time to repeating this experiment using more complex virtual organisms with more genes and much larger population sizes.

Materials and Methods

Computational Model and Task

Markov Brains are here evolved to control the behavior of virtual agents performing a cooperative task. Specifically, here, four agents start in the corners of a fully enclosed rectangular room. An agent would need 14 forward steps to cross the room. Agents can further turn left or right by 90 degrees or stand still. The room is filled with boxes that should be collected by the group of agents. However, instead of rewarding each agent according to the number of boxes collected, here the task is made harder by first identifying

[1] to our knowlege

which of the four agents collected the least amount of boxes and then awarding each agent exactly this reward. Ensuring that agents receive a minimal reward, we hope to encourage cooperative behavior. For the purpose here, it is irrelevant if this cooperation actually works; we are only interested in complex behavior. Other tasks can be easily imagined and implemented in our modeling framework (Bohm et al., 2017). An agent can further send a beeping sound that can be heard by all other agents, who can also identify which other agent beeped. Agents can also hand over previously collected boxes. When one agent faces another, instead of moving, at every update, an agent can pass one box. When agents try to hand over previously collected boxes, but no agent is in front of them, they just place the box in front of them if possible. If they face another box or the wall, that box disappears. Lastly, agents have a set of sensors so that they can see what is in front of them (empty, box, wall, or another agent), and they have a sensor reporting the number of boxes they have collected so far.

At the beginning of the evolutionary run, a random population of 100 agents is created. At each generational update, each agent needs to be evaluated and its fitness determined. Since this is a cooperative task, each agent gets cloned to form a group of four. This group is then tested, and it is the performance of the group that defines the fitness of the original agent. After testing, the three superfluous clones are removed again – one might think of this as an extended phenotype or genetically identical swarm. Selection is performed using standard roulette wheel selection. We found 500 generations to be sufficient to allow agents to evolve to us interesting behaviors as well as proper performance on the task. However, we allow the population of 100 agents to evolve for 5000 generations. Thereafter, the population size is increased by 100 at every generation to increase the population size to 10.000. After that size is reached, another 1000 generations of evolution continue to allow the population to equilibrate after the growth phase. This procedure saves a lot of computational resources but might also affect the evolutionary and population dynamics.

Markov Brains were encoded by a genome as described before, but to simplify the later genomic analysis, the genome size was fixed to 10000 sites, with each site being a 32-bit integer. Markov Brains were allowed to use deterministic, probabilistic, and mathematical computational units (Hintze et al., 2019). At the end of evolution, genomes were translated into sequences of the letters A, C, G, and T to resemble DNA nucleotides. Because each of the 10000 sites was a 32-bit integer, this translation resulted in 160000 nucleotides long genomes in a DNA format. Further, the phenotype of each agent in the population was recorded by testing it working on the task together with three identical copies of itself. Fifteen different behaviors were recorded, including scores of everyone as well as the group, number of beeps sent and received, how often agents moved, turned, did nothing, picked up boxes, handed over boxes, and how many boxes, walls, nothing, or other agents each one saw. Each agent was tested 50 times, and the average across those samples was used to characterize each behavior. Theoretically, other behaviors could be recorded, but these seemed to be the most relevant for this task.

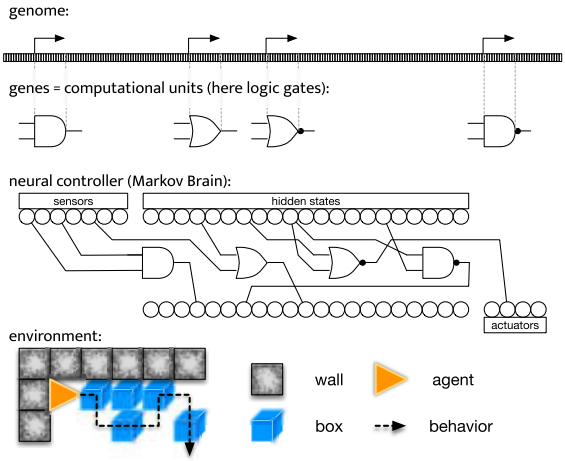


Figure 1: Illustration of a Markov Brain and how it controls an agent. At the top is the genome of 160.000 nucleotides. Specific sequences indicate start codons, here four (arrows). Start codons define genes, which encode the type and connectivity of computational units. Here four logic gates are shown (from left to right: AND, OR, NOR, and NAND). The neural controller for the virtual agents is a Markov Brain neural network, with input nodes (sensors), hidden nodes, and actuators. The logic gates perform computations and update hidden states (which are recurrent) and send signals to the actuators. The agent (shown at the bottom) perceives the environment and, due to the interactions of logic gates (genes), produces complex behavior. This behavior is quantified and characterizes the phenotypic traits of the agent.

GWAS: Correlational Analysis and ANOVA to Identify Genes

Two separate GWAS analyses were performed. To resemble a GWAS analysis using Single Nucleotide Polymorphisms (SNPs), we first coded each position to be a binary value; 1 if the individual had the most common allele at that position and 0 otherwise. These values were subsequently correlated to the phenotype, and a P-value was computed at every position along the genome using a t-test. In a second analysis, the information from all four types of alleles (A, C, G, and T) was fitted using ANOVA, and a P-value was computed at every position along the genome using an F-test.

A significance level of 5×10^{-8}, which corrects for multiple testing (Fadista et al., 2016), was used corresponding to a $-\log_{10}$(P-value) of 7.3. The $-\log_{10}$(P-value)-threshold

was adjusted for any general inflation in P-values along the genome by finding the 99.9% quantile, say q, for all P-values, and subsequently multiplying the threshold with $q/3$ (i.e. a rough estimate of the "genomic inflation factor", see e.g. Bacanu et al. (2000)) .

Knockout Analysis

For the Markov Brains, the location of each gene is known, and genes can be manipulated easily. For the knockout analysis, each gene for each agent in the final population was performed, and the behavior for each knockout phenotype was recorded as described earlier. The effect for each gene was then determined as the average change in each trait compared to the wild type (no knockout). This can be done in two ways because, here, not all organisms have all genes. As we will see later, due to a high degree of diversity within the final population, we find subsets of agents sharing genes that the majority of others do not. The effect of a knockout of those genes can be computed for the subset of organisms that have the gene and averaging their effect. Or by pretending all other organisms experienced a knockout of the same region in the genome, without an effect, and then averaging over all organisms. Both methods will be compared later.

Genetic Diversity of the Population

Typically in a GWAS, not every individual of a population is used in the analysis, but rather a sample of the entire population. Here, however, the entire population is known, and a subset of individuals needs to be selected. This can either happen randomly or by taking genetic diversity into account. Thus, either highly related individuals or those that are maximally different from each need to be selected. An advantage of using those that are maximally different is that the phenotypic values from the different individuals can be treated as (close to) independent and thereby minimize any effect of population substructure.

For that, the number of sites that differ between all $N = 10.000$ genotypes was counted. That would mean here $N(N-1)$ comparisons for 160.000 nucleotides (genome length). To reduce this number, only every 32. nucleotide was compared, such that this comparison took only $\sim$2h, as opposed to 70 days. This analysis resulted in a distance matrix that could now be clustered agglomeratively, joining closely related genotypes into groups. To identify the closest related group, this clustering was done until one cluster reached the size of 1000. The individuals in this group represent the closest related group within the entire population. In order to identify the most diverse group, clustering was performed until 1000 independent clusters were found. One individual from each cluster was then randomly selected. This method guarantees a high degree of diversity within that group (for an illustration, see 2).

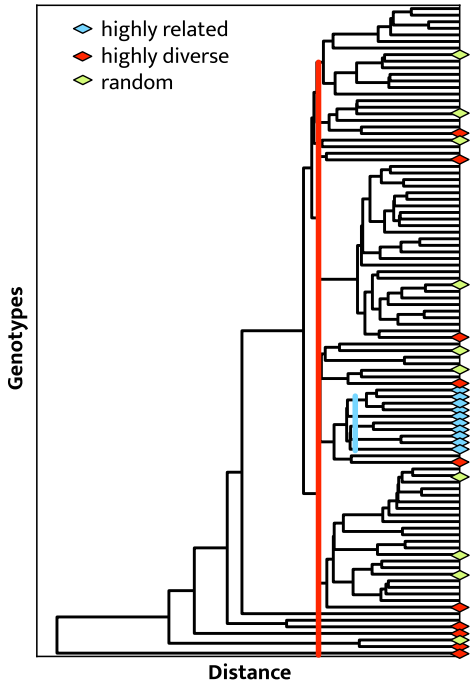


Figure 2: Illustration of how groups with different degrees of genetic diversity can be selected. The dendrogram presented here for 100 individuals represents their genetic diversity. If a single cluster is selected (blue line), all individuals (blue diamonds) of that cluster are highly related. When instead individuals from different clusters are selected (red line and diamonds), the selected subset is highly diverse genetically. Green diamonds illustrate a random sampling of individuals.

Epistasis

Epistasis (ϵ) between two genes (here a and b) can be calculated if their wildtype fitness (W_0), each gene's mutant or knockout phenotypic effect W_a and W_b, as well as, their double mutant effect W_{ab} is known, using the following equation 1 (Østman et al., 2012):

$$\epsilon = \log \frac{W_0 W_{ab}}{W_a W_b} \tag{1}$$

The computational model not only allows us to test all pairs of genes, but all possible sets of genes can be knocked out, and their effect on any of the phenotypic traits can be measured. This allows us to consider not only the two-way epistatic effect between two genes but, for example, three and more degrees of interaction. This "n-way" epistasis is

computed using the following equation 2 as a direct extension of equation 1:

$$\epsilon = \log \frac{W_0 W_{a \to n}}{\prod_{x=a}^{n} W_x} \quad (2)$$

Observe that traits W can have a mathematical value of 0.0, indicating, for example, that an agent never "moved". This would result in an undefined ϵ. Thus, a pseudo count of 1 was assumed for all computations involving knockouts. Also, phenotypic effects were normalized such that $W_0 = 1.0$.

Results

Two independent computational evolutionary experiments were run to have at least one replicate for comparisons (called A and B). Using multi-threading on a machine supporting 50 parallel threads took about 4 hours per experiment. Then the phenotype of all traits of all organisms was determined, as well as the phenotype of all knockouts, including all combinations of all possible knockouts to determine n-way epistatic relationships later. The genetic distance between each member of the finally evolved population was determined to select 1000 highly related, highly diverse, or random individuals (see Figure 2). For all selected populations, a GWAS was performed using either a correlation analysis using binary marker information or an ANOVA exploiting all the information in the DNA code at the tested genomic location. From the 13 possible phenotypic traits, only 11 showed sufficient diversity (apparently, agents did not evolve behavior to hand over previously collected boxes, and neither did they ever drop boxes). Figure 3 shows the results for experiment A, including the functional contribution of each site.

While the above shows how correlations of traits map to sites on the genome, we can ask how many genes each of the different methods (SNP correlation and ANOVA) identifies, given the three different criteria to select individuals (maximally diverse, highly related, or random). Typically a genome-wide significance threshold of 7.3 is used as the most rigorous threshold to identify genes. Alternatively, values above a threshold of 5.0 (e.g. Zayats et al., 2015) can also be considered a potential site. However, it is possible that an even lower threshold still positively identifies genes without identifying sites that do not contain genes. For that, we lower the threshold below 5.0 until just before the point where false positives appear. Table 1 shows the result of these analyses and how many genes were positively identified given each method. For the case of highly related individuals, the ratio of true and false positives is lowest, as expected, since relatedness induces elevated values along the entire genome in a GWAS (Amin et al., 2007).

Table 1: Positive and false positively identified genes for each experiment A (top) and B (bottom). Before the dash is the number of correctly identified genes, while the number after the dash indicates sites identified that were outside of genes. Observe that mutations suggesting a strong effect could be close to genes actually responsible for the effect (hitchhiking mutations). For each computational experiment, the genetic variation was correlated to different traits using two different methods (ANOVA and SNP). Populations of 1000 agents were sampled using either maximum diversity (divers), maximal relatedness (related), or randomly sampled (random). To identify positive correlations, three thresholds were used, 7.3 as the most stringent, 50 as an acceptable lower limit, and for the one labeled "best," the threshold was lowered just so that no extra gene was falsely identified.

A [71 genes]		7.3	5.0	best
ANOVA	diverse	0/1	3/6	1/1
	related	0/12	6/99	3/9
	random	1/1	2/12	1/0
SNP	diverse	6/28	9/56	8/25
	related	0/9	1/28	0/9
	random	5/8	6/43	6/8

B [90 genes]		7.3	5.0	best
ANOVA	diverse	2/14	8/38	4/9
	related	2/38	3/95	1/24
	random	1/0	4/17	3/0
SNP	diverse	3/19	4/41	5/18
	related	0/11	1/33	1/7
	random	1/1	4/9	3/1

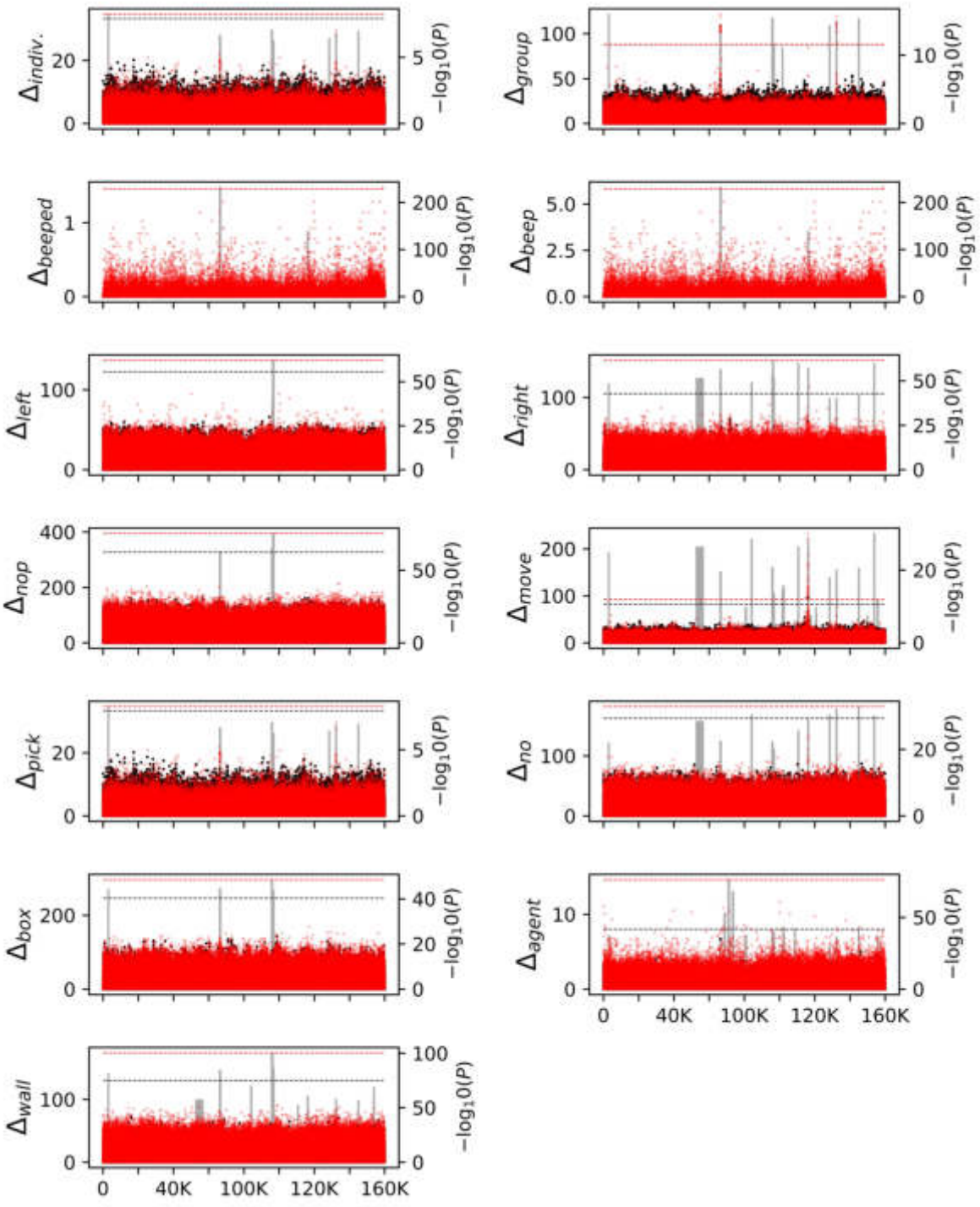


Figure 3: Significance ($-\log_1 0$(P-value)) for all 160.000 sites (x axes) and 11 phenotypic traits (see label on the left y-axis). In red result for binary SNP correlation analysis and in black results from ANOVA, the average effect of knock-outs in gray in the background. Dashed lines show the 7.3 threshold for genome-wide significance (red for SNP correlation and black for ANOVA). The traits handing over or dropping boxes were omitted as they neither had a knockout effect nor a signal in the GWAS analysis.

Surprisingly, we only find, at best, 8 out of 71 (experiment A), or 5 out of 90 (experiment B), genes suggesting less than 10% efficiency. However, we are potentially dealing with highly diverse individuals, and the model system might present us with some strange phenomena. Clearly, the 71 (or 90) genes are not found in every individual but instead represent all genes that can be found amongst all 10.000 individuals of either population. As it turns out, the majority of genes are only found in a few individuals, while only a few genes are common to all individuals in the population (see Figure 4). Further, the mean effect on fitness each knockout has, and thus by proxy, the genetic variance of mutants, is thus low for the majority of uncommon genes. Or in other words, only a few genes (about 10 to 15) are found in all organisms and also show a strong phenotype in the knockout analysis. The GWAS analysis still does not find all of them, but around 30% to 50%, depending on the choice of significance threshold.

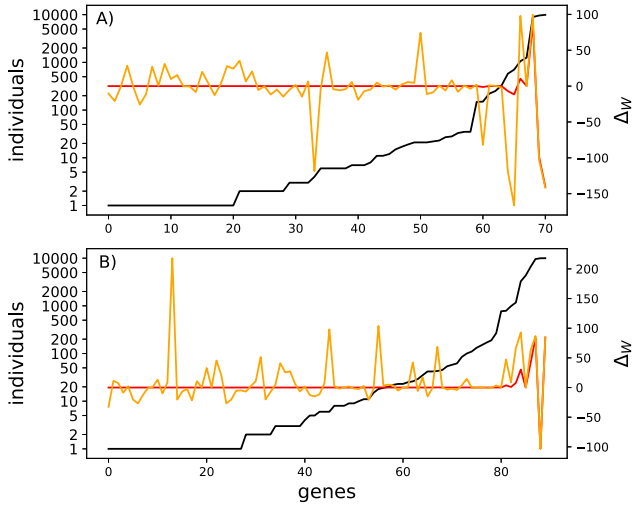


Figure 4: Genes and their average effect on fitness when knocked out. In black, the number of individuals that share the same gene. Genes on the x-axis are sorted by the number of individuals that share them (log scale). In red, for the same genes, their average fitness effect Δ_W when knocked out (right-hand y-axis). In orange, the mean fitness effect for each knockout, but only considering the group of organisms that share the same gene.

Epistasis

Another option why the GWAS did not find all or most of the genes could be epistatic interactions. Genes could be redundant, phenotypic variations could be masked, or their signal obfuscated due to other gene functions. Similarly, a mutation in one gene could be neutralized by a mutation in another, which co-segregates. This is highly expected since here, haploid none sexually recombining organisms are modeled. Anyways, to illustrate the effect of epistasis, the effect of all knockouts and all possible interactions between the powerset of all knockouts was quantified for experiment A. We find that, indeed all genes have at least some, but most have many interaction partners (see Figure 5). Here, only their epistatic effects with respect to fitness (group score) were investigated, but of course, it is possible to do for all traits.

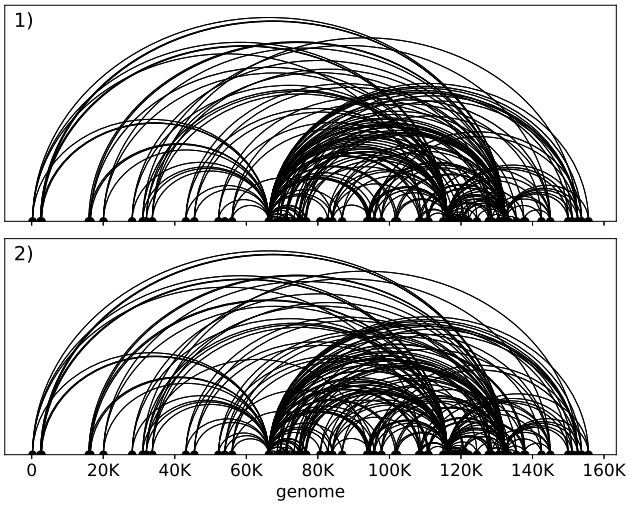


Figure 5: Epistatic effects between all genes. The x axis shows the 160.000 nucleotides and the relative location of each gene. The top panel 1) shows the mean positive epistatic effect between all gene pairs as arches. The width of the arch is proportional to $\bar{\epsilon}$. On the bottom panel 2), the same for negative epistatic effects.

Discussion

This project attempted to perform a GWAS on computationally evolved model organisms. The main goal was to determine how many genes could be identified using exact measurements of behavioral traits and to correlate them to all possible genetic sites. While it was possible to identify some genes, the vast majority of genes remained undetected. This is, in part, explained by an artificially high gene count. Within the population of 10.000 evolved virtual agents, many genes can only be found in one or a limited number of individuals. Only 10 to 15 genes can be found in more than half of the agents. Consequently, only this smaller set actually delivers a detectable signal. However, while some were found, others remained obscure. We can now ask what factors or procedures could be improved to get a more realistic model, while at the same time, some constraints might not be a problem of the computational model but a shortcoming of GWAS in principle.

Population Size Limits

One might argue that a natural population is much larger than 10.000 organisms and may have undergone evolutionary processes with varying population sizes different from our simulated population. This could fairly easily be considered in future studies since increasing the population size only increases the computation time linearly. Right now, an entire experimental run, including the knockout analysis, took on a machine supporting 50 parallel threads for only 4 hours. This also means that not only larger populations but also evolution over more generations or for more complex tasks can easily be conducted.

Limited Number of Genes

The evolved Markov Brains only needed about six genes to function properly for this task. This number is vanishingly small compared to natural organisms. However, more complex tasks, but also physical properties, or different genetic encodings are possible. For example, instead of having one gene encode one logical unit, multiple genes could all contribute to the same function. However, this would also automatically increase the degree of epistatic interactions. It remains an open question to what degree this would hamper the success of the GWAS or allow for more detectable phenotypic variation?

Trait Selection

The phenotypic traits here were mostly selected based on convenience. Counting turns, beeps, or how often an agent sees a wall are easy to quantify. To what degree they are meaningful descriptors of evolved behavior remains questionable. Traits that are directly selected and are thus most critically shaped by evolution would, of course, ideal. However, with natural organisms, we humans also handpick the traits we are interested in, and not necessarily those aligning with selection pressures. One might argue that such selection is equally artificial and subjective. Still, particularly if we assume that agents performing different tasks should be tested in the future, the selection of traits will remain a critical issue. This type of experiment can help distinguish between traits suitable for a GWAS performed in natural organisms in the future.

Epistasis

Extensive efforts have been made to model epistasis in quantitative trait mapping and GWAS, with considerable success in model organisms (Mackay, 2014) such as yeast (Forsberg et al., 2017) and *Arabidopsis thaliana* (Lachowiec et al., 2015). One of the major challenges is, however, to account for the massive increase in multiple testing when combinations of interacting genes are considered. A possible solution could be to investigate not only the differences in phenotypic means between genotypes but also differences in phenotypic variance, which is expected as a consequence of epistasis (Rönnegård and Valdar, 2011). We expect that simulations based on Markov Brains can be a powerful tool to investigate the feasibility of alternative methods to detect epistasis in applied genetic studies.

Simplicity of the Genetic Model

Here, haploid asexual organisms were modeled, while GWAS are very often performed on sexually recombining diploids. Fortunately, Markov Brains can be built from diploid genomes and can perform recombination. The software framework MABE already supports such experiments.

Conclusion

In conclusion, we think that the use of a computational model to test and improve the accuracy and efficiency of a GWAS is promising. While the model is simple, it already provides sufficient complexity to overwhelm the GWAS performed here. Further, while the diversity of the model was surprising and hampered efficiency, it might also present an additional factor to take into account. Clearly, the evolutionary dynamics could possibly confound results, but also the ability of a computational model to control for them and consequently learn how they affect natural systems and our ability to analyze them presents new opportunities we hope to explore in the future.

Acknowledgements

This work was supported by the Uppsala Multidisciplinary Center for Advanced Computational Science SNIC 2020-15-48, and the BEACON Center for the Study of Evolution in Action. LR was supported by Formas - a Swedish Research Council for Sustainable Development (ID: 2019–02276 and 2019–02111).

References

Amin, N., Van Duijn, C. M., and Aulchenko, Y. S. (2007). A genomic background based method for association analysis in related individuals. *PloS one*, 2(12):e1274.

Bacanu, S.-A., Devlin, B., and Roeder, K. (2000). The power of genomic control. *The American Journal of Human Genetics*, 66(6):1933–1944.

Bohm, C., CG, N., and Hintze, A. (2017). MABE (modular agent based evolver): A framework for digital evolution research. *Proceedings of the European Conference of Artificial Life*.

Fadista, J., Manning, A. K., Florez, J. C., and Groop, L. (2016). The (in) famous gwas p-value threshold revisited and updated for low-frequency variants. *European Journal of Human Genetics*, 24(8):1202–1205.

Forsberg, S. K., Bloom, J. S., Sadhu, M. J., Kruglyak, L., and Carlborg, Ö. (2017). Accounting for genetic interactions improves modeling of individual quantitative trait phenotypes in yeast. *Nature genetics*, 49(4):497–503.

Giaever, G. and Nislow, C. (2014). The yeast deletion collection: a decade of functional genomics. *Genetics*, 197(2):451–465.

Hintze, A., Schossau, J., and Bohm, C. (2019). The evolutionary buffet method. In *Genetic Programming Theory and Practice XVI*, pages 17–36. Springer.

Lachowiec, J., Shen, X., Queitsch, C., and Carlborg, Ö. (2015). A genome-wide association analysis reveals epistatic cancellation of additive genetic variance for root length in arabidopsis thaliana. *PLoS genetics*, 11(9):e1005541.

Loos, R. J. and Yeo, G. S. (2014). The bigger picture of fto—the first gwas-identified obesity gene. *Nature Reviews Endocrinology*, 10(1):51–61.

Mackay, T. F. (2014). Epistasis and quantitative traits: using model organisms to study gene–gene interactions. *Nature Reviews Genetics*, 15(1):22–33.

Manolio, T. A., Collins, F. S., Cox, N. J., Goldstein, D. B., Hindorff, L. A., Hunter, D. J., McCarthy, M. I., Ramos, E. M., Cardon, L. R., Chakravarti, A., et al. (2009). Finding the missing heritability of complex diseases. *Nature*, 461(7265):747–753.

Østman, B., Hintze, A., and Adami, C. (2012). Impact of epistasis and pleiotropy on evolutionary adaptation. *Proceedings of the Royal Society B: Biological Sciences*, 279(1727):247–256.

Patxot, M., Banos, D. T., Kousathanas, A., Orliac, E. J., Ojavee, S. E., Moser, G., Holloway, A., Sidorenko, J., Kutalik, Z., Mägi, R., et al. (2021). Probabilistic inference of the genetic architecture underlying functional enrichment of complex traits. *Nature communications*, 12(1):1–16.

Pirastu, N., Joshi, P. K., De Vries, P. S., Cornelis, M. C., McKeigue, P. M., Keum, N., Franceschini, N., Colombo, M., Giovannucci, E. L., Spiliopoulou, A., et al. (2017). Gwas for male-pattern baldness identifies 71 susceptibility loci explaining 38% of the risk. *Nature communications*, 8(1):1–10.

Rönnegård, L., McFarlane, S. E., Husby, A., Kawakami, T., Ellegren, H., and Qvarnström, A. (2016). Increasing the power of genome wide association studies in natural populations using repeated measures–evaluation and implementation. *Methods in ecology and evolution*, 7(7):792–799.

Rönnegård, L. and Valdar, W. (2011). Detecting major genetic loci controlling phenotypic variability in experimental crosses. *Genetics*, 188(2):435–447.

Sul, J. H., Martin, L. S., and Eskin, E. (2018). Population structure in genetic studies: Confounding factors and mixed models. *PLoS genetics*, 14(12):e1007309.

Tam, V., Patel, N., Turcotte, M., Bossé, Y., Paré, G., and Meyre, D. (2019). Benefits and limitations of genome-wide association studies. *Nature Reviews Genetics*, 20(8):467–484.

Uffelmann, E., Huang, Q. Q., Munung, N. S., de Vries, J., Okada, Y., Martin, A. R., Martin, H. C., Lappalainen, T., and Posthuma, D. (2021). Genome-wide association studies. *Nature Reviews Methods Primers*, 1(1):1–21.

Wood, A. R., Esko, T., Yang, J., Vedantam, S., Pers, T. H., Gustafsson, S., Chu, A. Y., Estrada, K., Kutalik, Z., Amin, N., et al. (2014). Defining the role of common variation in the genomic and biological architecture of adult human height. *Nature genetics*, 46(11):1173–1186.

Zayats, T., Athanasiu, L., Sonderby, I., Djurovic, S., Westlye, L. T., Tamnes, C. K., Fladby, T., Aase, H., Zeiner, P., Reichborn-Kjennerud, T., et al. (2015). Genome-wide analysis of attention deficit hyperactivity disorder in norway. *PloS one*, 10(4):e0122501.

On the Mutual Influence of Human and Artificial Life: an Experimental Investigation

Stefano Furlan, Eric Medvet, Giorgia Nadizar, and Federico Pigozzi

Evolutionary Robotics and Artificial Life Lab, Department of Engineering and Architecture, University of Trieste, Italy
stefanofurlan6@gmail.com, emedvet@units.it, giorgia.nadizar@phd.units.it, federico.pigozzi@phd.units.it

Abstract

Our modern world is teeming with non-biological agents, whose growing complexity brings them so close to living beings that they can be cataloged as artificial *creatures*, i.e., a form of Artificial Life (ALife). Ranging from disembodied intelligent agents to robots of conspicuous dimensions, all these artifacts are united by the fact that they are designed, built, and possibly trained by humans taking inspiration from natural elements. Hence, humans play a fundamental role in relation to ALife, both as creators and as final users, which calls attention to the need of studying the mutual influence of human and artificial life. Here we attempt an experimental investigation of the reciprocal effects of the human-ALife interaction. To this extent, we design an artificial world populated by life-like creatures, and resort to open-ended evolution to foster the creatures adaptation. We allow bidirectional communication between the system and humans, who can observe the artificial world and voluntarily choose to perform positive or negative actions towards the creatures populating it; those actions may have a short- or long-term impact on the artificial creatures. Our experimental results show that the creatures are capable of evolving under the influence of humans, even though the impact of the interaction remains uncertain. In addition, we find that ALife gives rise to disparate feelings in humans who interact with it, who are not always aware of the importance of their conduct.

Introduction and related works

In the 1990s, the commercial craze of "Tamagotchi" (Clyde, 1998), a game where players nourish and care for virtual pets, swept through the world. Albeit naive, that game is a noteworthy instance of an Artificial Life (ALife) (Langton, 1997), i.e., a simulation of a living system, which does not exist in isolation, but in deep entanglement with human life. It also reveals that ALife is not completely detached from humans, who might need to rethink their role and responsibilities toward ALife. We already train artificial agents by reinforcement or supervision: trained agents are notoriously as biased as the datasets we feed them (Kasperkevic, 2015), and examples abound[1]. For instance, chatbot Tay shifted from lovely to toxic communication after a few hours of interaction with users of a social network (Hunt, 2016). The field of robotics is no exception to the case, and while robots, a relevant example of ALife agents, are becoming pervasive in our society, we—the creators—*define* and influence them (Pigozzi, 2022). One day in the future, a robot could browse for videos of the very first robots that were built, eager to learn more about its ancestors. Suppose a video shows up, displaying engineers that ruthlessly beat up and thrust a robot in the attempt of testing its resilience (Vincent, 2019). How brutal and condemnable would that act look to its electric eyes? Would our robotic brainchildren disown us and label us "a virus" as Agent Smith (the villain, himself an artificial creature) does in the "Matrix" movie (Wachowski et al., 1999)? At the same time, how would such responsibility affect the creators themselves?

Broadly speaking, when dealing with complex systems involving humans and artificial agents, whose actions are deeply intertwined, what results from the mutual interaction of humans and ALife? In particular, do artificial agents react to the actions of humans, displaying short-term adaptation in response to stimuli? Do these actions influence the inherited traits of artificial creatures, steering their evolutionary path and long-term adaptation? And, conversely, are humans aware of their influence on ALife? Do they shift their conduct accordingly?

We consider a system that addresses these questions in a minimalist way. We design and implement an artificial world (Figure 1), populated by virtual creatures that actively search for food, and expose it to a pool of volunteer participants in a human experiment. We consider three design objectives: (a) interaction, that is bidirectional between human and ALife; (b) adaptation, of creatures to external stimuli, including human presence; (c) realism, of creatures to look "familiar" and engaging for participants. Participants interact with the creatures through actions that are either "good" (placing food) or "bad" (eliminating a creature): we then record the participants' reactions. At the same time, creatures can sense human presence. We achieve long-term adaptation through artificial evolution, and, for the sake of realism, we design the creatures to be life-like. As a result, the goodness or badness of human actions can potentially

[1] https://github.com/daviddao/awful-ai

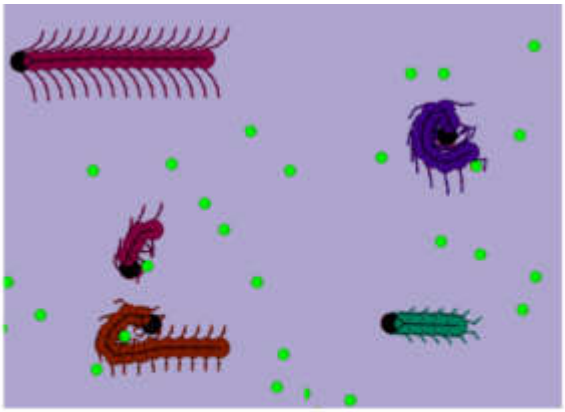


Figure 1: Our artificial world: worm-like agents are creatures that search for food (the green dots).

affect the evolutionary path of creatures, as well as their relationship with humans. Humans, on the other side, can feel emotions in the process. Participants thus play the role of a "superior being", absolute from any conditioning authority (Milgram, 1963), with power of life and death upon the creatures. Whether their actions will be good or bad is up to them: a philosophical debate on human nature that goes back to Thomas Hobbes (1651) and Jean-Jacques Rousseau (1755), with their opposing views propagating through history.

Other studies crafted artificial worlds, e.g., Tierra (Ray, 1992), PolyWorld (Yaeger et al., 1994), and Avida (Ofria and Wilke, 2004), with several different goals: they mostly investigate questions related to evolutionary biology (Lenski et al., 2003), ecology (Ventrella, 2005), open-ended evolution (Soros and Stanley, 2014), social learning (Bartoli et al., 2020), or are sources of entertainment and gaming (Dewdney, 1984; Grand and Cliff, 1998). Albeit fascinating, none of these addresses the main research question of this paper, i.e., the mutual influence of human life and ALife. Our work also differs from multi-agent platforms, whose focus is on optimizing multi-agent policies for a task (Suarez et al., 2019; Terry et al., 2021).

The work that is the most similar to ours pivots around the "Twitch Plays Robotics" platform of Bongard et al. (2018). While paving the way for crowdsourcing robotics experiments, it is, rather than an artificial world, an instance of "interactive evolution" (with participants issuing reinforcements to morphologically-evolving creatures), and does not detail the influence of creatures on participants.

We instead concentrate on the bidirectionality of interaction, and branch into two complementary studies: the first aimed at quantifying the effects of human interaction on artificial creatures, and the second focused on surveying how humans perceive and interface themselves with ALife. Concerning the former, we simulate human actions on the system and analyze the progress over time of some indexes, whereas for the latter we perform a user study involving a pool of volunteer participants interacting with the creatures. The experimental results confirm the importance of focusing on the bidirectionality of human-ALife interaction, and open a way towards more in depth analyses and studies in the field. Not surprisingly, we find that an artificial world subjected to human influence is capable of evolving, yet the real impact of human behavior on it, be it positive or negative, remains enigmatic. In addition, we discover two main currents of thought among people who interface themselves with ALife: those who feel involved and are aware of the consequences of their actions on an artificial world, and those who perceive ALife as a not attention-worthy farfetched artifact.

The artificial world

Objectives

The aim of this work is to investigate the mutual influence of human life and ALife. We introduce an artificial world, populated by virtual creatures, that is suitable for such an investigation. We consider three objectives:

Interaction. In order to study any bidirectional impact between human life and ALife, the artificial world must support interaction. Moreover, interaction follows two design principles: (a) ergonomics, and (b) characterization. The former makes interaction easy and accessible for humans, while the latter is concerned with mapping an interaction back to the human behavior that generated it, and, in this study, classifying the interaction as either "good" or "bad". Last but not least, we remark that influence between human life and ALife must be bidirectional. Thus, for any human influence on ALife to happen, we require virtual creatures to be able to sense the presence of a human observer.

Adaptation. Second, we require the creatures inhabiting the artificial world to have the potential for adaptation to the environment and over time. During their life, virtual creatures undergo exposure to a set of stimuli, both "endogenous" and "exogenous": the former arise from the simulation itself (e.g., presence of food), while the latter arise from interaction with humans (e.g., good or bad actions). In order to evaluate any impact of human life on ALife, creatures should show adaptation to those stimuli, both in the short- and in the long-term, the first being a form of action-reaction, and the latter involving the development of more favorable traits.

Realism. To incentive interaction, humans should be able to relate with familiar entities. Creatures should then have a realistic look, possibly resembling natural organisms. In particular, they should have an appearance of "life" and engage in life-like activities, in order to elicit any notion of AL-

ife in the observers. At the same time, creatures should not be too realistic, or even human-like, to avoid the notorious "uncanny valley" problem witnessed with highly-realistic robots (i.e., uneasiness and revulsion in the observers) (Mori et al., 2012).

Environment and creatures

The artificial world introduced in this paper is visually two-dimensional, simulated in discrete time and continuous space, enclosed within an impassable rectangle of size 420×240 m. The colorful virtual creatures that populate it actively explore the space and hunt for food units lying on the ground (Figure 1). Each creature is endowed with a certain amount of energy, which dissipates at every time step and replenishes once the creature eats food. If a creatures depletes all of its energy, it dies and is removed from the world; then, a new creature is born by mutating one of the surviving creatures. Human observers may interact with the creatures by nourishing them, i.e., placing food in their proximity, (a "good" action) or eliminating some of them (a "bad" action).

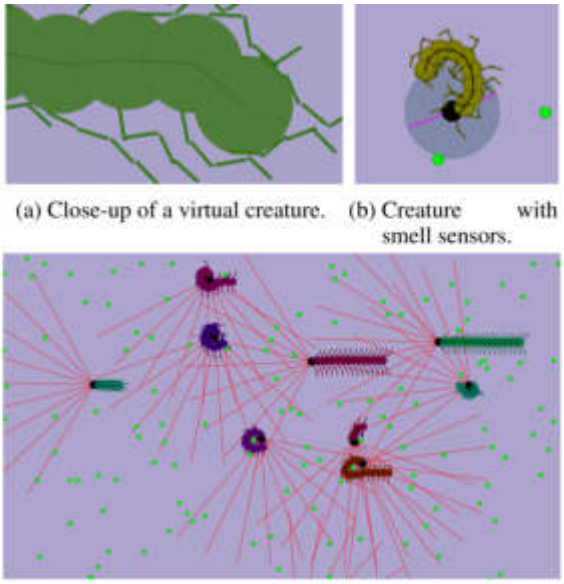
We implemented the project in the Java programming language, building on top of the dyn4j[2] physics engine, for which we set the time step to $\Delta t = \frac{1}{60}$ s, no gravity, and a linear speed damping coefficient that makes creatures movement appear like happening in a fluid. We made the project publicly available at `https://gitlab.com/step.lumumba/worm_simulator`.

We represent each creature as a *genotype*, that we map to a *phenotype*, i.e., the body and the brain of the creature.

Creature body. The creatures have a worm-like body consists of a variable number of *segments* chained together, starting from a "head" segment. Each segment is a circular mass of weight 10 kg and radius 1.5 m and is connected to up to two other segments with a joint that allows for some rotation: as a result, the body can bend and appears flexible. Two flagella, implemented as flexible strings of tiny rectangles, extrude from each segment (see Figure 2a for a close-up) and have a sensory function (detailed below) and an aesthetic function. The genetic encoding of a body is a numerical vector $\boldsymbol{g}_{\text{morph}} \in \mathbb{R}^5$, which encodes the following phenotypic traits: the number of segments, the number of rectangles per flagellum, the length of rectangles of the flagella, the body color, and the length of sensory memory. We include color as a trait because it is neutral with respect to selection and survival, and allows us to verify there is no bias dictated by the representation or the evolution. For the length of sensory memory, see next sub-section.

To ensure body traits lie in meaningful intervals (e.g., flagella do not disappear), we map each gene g_i to the corresponding phenotypic trait as $p_i = \frac{1+g_i'}{2}(p_i^{\max} - p_i^{\min}) + p_i^{\min}$,

[2] `https://dyn4j.org`



(a) Close-up of a virtual creature. (b) Creature with smell sensors.

(c) Creatures casting proximity sensor rays (depicted in red).

Figure 2: Details of creatures populating our artificial world.

where $g_i' = \min(\max(g_i, -1), 1)$: the first operand in the multiplication ensures the value lies in $[0, 1]$, and we then linearly rescale it to fit the interval $[p_i^{\min}, p_i^{\max}]$. After preliminary experiments, we set the intervals to be $\{5, 6, \ldots, 20\}$, $\{2, 3, \ldots, 40\}$, $[0.8, 1.8]$, and $\{0, 1, \ldots, 9\}$ for number of segments, number of rectangles per flagellum, length of rectangles of the flagella, and color, respectively. For color, integers in $\{0, 1, \ldots, 9\}$ correspond to 10 possible colors.

By virtue of such morphological representation, creatures are in effect "primitive" enough to dispense with unnecessary complexity and focus on the mutual influence of human life and ALife; indeed, Mahoor et al. (2017) reported that the more "intuitive" the morphology, the more engaged participants to crowd-sourced robotics experiments are. Moreover, creatures do indeed recall natural organisms, in particular invertebrates (e.g., annelids, whose body consists of multiple segments), some of the simplest, most common, and most widely studied animals on Earth (Stewart, 2005). Our artificial world thus satisfies the Realism objective.

Creature sensing. We equip every creature with proximity, smell, touch, energy, temperature, and human presence sensors. Proximity sensors, depicted in Figure 2c, cast 9 rays from the head circle and return the (normalized) distance from the closest object (either food or creature), clipping it to 1 if there is none. Two smell sensors perceive the number of food units (over the total) in the right and left semi-circumferences of radius 9 m centered on the head, as shown in Figure 2b. Three touch sensors per side perceive whether one of three objects among food, other creatures, and the creature itself touch any of the flagella for that side,

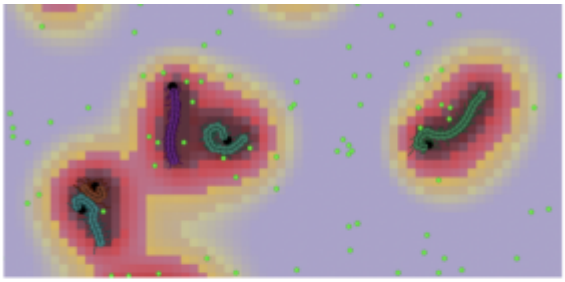


Figure 3: How creatures affect temperature, depicted as shades of red at every empty space (the darker, the warmer). Similarly, human observers condition temperature by moving their face across the screen.

and return 1 if yes, 0 if not. Energy sensor perceives the current energy of the creature, rescaled in $[0, 1]$. Temperature and human presence sensors, described below in detail, also return a value in $[0, 1]$ each. For every sensor reading, we also compute its trend over a window stretching T time steps into the past, where T is the fifth (and last) morphological gene and is thus subject to evolution. In this way, creatures can evolve some form of memory. Overall, creatures sense the world through $2(9+2+3\cdot2+1+1+1) = 40$ values in $[0, 1]$ at each time step.

Temperature and human presence sensors relate to the central piece of this study, and deserve an in-depth treatment. Temperature records the presence of "living" entities, either human or artificial, in the surrounding of a creature. For a simulation, we define the temperature matrix $\boldsymbol{H} \in \mathbb{R}^{+\,420\times240}$, where 420 and 240 are the side lengths of the artificial world. At each time step k of simulation, we increment every $h_{x,y} \in \boldsymbol{H}$ by τ if there is at least one body segment whose center of mass lies within it. Then, we diffuse temperature by averaging it over the nine neighboring cells and multiplying by a damping coefficient α: $h_{x,y}^{(k+1)} = \alpha \frac{\sum_{x',y' \in \{-1,0,1\}} h_{x+x',y+y'}^{(k)}}{9}$. Figure 3 is an instance of how creatures affect temperature during a simulation.

Moreover, humans (if present) do affect temperature. If a human observes the simulation on a computer screen, a computer vision system captures an image from the webcam placed over the screen and draws a bounding box around their face: we then increment h_{x_c,y_c} by the area of the bounding box, (x_c, y_c) being the coordinates of the center of the bounding box rescaled considering the world and captured image sizes. We remark that the area of the bounding box enclosing the observer's face, hence the temperature increase, depends on the proximity of the human to the computer screen. For a given creature, the temperature sensor perceives the average of the sub-matrix $\boldsymbol{H}_{\text{temp}} \in \mathbb{R}^{m\times m}$ centered on the head of the creature. After preliminary experiments, we set $\tau = 150$, $\alpha = 0.99$, and $m = 13$. We implemented face detection with the OpenCV library (Bradski, 2000), using Haar Cascades (Viola and Jones, 2001) as face detection algorithm.

Finally, the human presence sensor returns 1 if a human is observing the simulation, i.e., if the a face is detected by the webcam, and 0 otherwise.

By virtue of temperature and human presence sensors, creatures can sense the presence and location of humans, and thus affect one direction of the Interaction objective. Along the same direction (i.e., from humans to creatures), humans can influence the artificial world by placing food or killing creatures with a mouse click, as we shall see in Section "RQ2: human attitude towards ALife". In the other direction (i.e., from creatures to humans), the possible source of influence stays in the artificial world being depicted on the screen and hence being observable by humans.

Creature brain. We feed sensor readings and their trends to a feed-forward, fully-connected neural network with 40 input neurons (one for every sensor reading) and 3 output neurons, that correspond to the three possible actions for a creature: move ahead, to the right, or to the left. At every time step, we select the output having the highest absolute value and apply it as a force in that direction to the head circle. After preliminary experiments, we set one hidden layer with 10 neurons and $\tanh$ as activation function for all neurons. The genetic encoding for the controller is thus a numerical vector $\boldsymbol{g}_{\text{ctrl}} \in \mathbb{R}^{443}$ encoding the parameters of the neural network.

The genotype of a creature is then the concatenation $\boldsymbol{g} = [\boldsymbol{g}_{\text{morph}}\ \boldsymbol{g}_{\text{ctrl}}] \in \mathbb{R}^{5+443=448}$. Evolution operates on the representation $\boldsymbol{g}$, in a way that we detail in the next sub-section.

Simulation

Simulation takes place in discrete time and continuous space. At every time step, n_{agents} creatures and n_{food} food units populate the artificial world.

At the very beginning, we initialize n_{agents} creatures by sampling genotypes from $[-1, 1]^{448}$, i.e., each gene $g_i \sim U(-1, 1)$, mapping to the corresponding phenotypes, and giving birth to creatures at random positions, while making sure none of them overlap. At birth, we endow every creature with e_{init} units of energy and set its generation to 0.

Then, at every time step of the simulation loop proceeds as follows:

1. Each creature senses the environment and uses the brain for processing sensor readings and producing an action.

2. The physics engine steps by applying the forces corresponding to each creature's action.

3. For each creature, if its head overlaps with a food unit, its energy is incremented by e_{food} units. Upon the food consumption, the eaten food unit is removed from the world and a new one spawns at a random position.

4. For each creature, the energy is decreased by e_{step} units. If energy of a creature drops to 0, the creature dies and is removed from the world. As many food units as the number of its body segments spawn at the creature last position; to ensure a constant supply of food in the world, as many food units are randomly removed from the world.

5. For every creature just dead, if any, a new creature is born at a random position (making sure there is no overlapping), and its energy is set to e_{init}. With probability p, we randomly initialize its genotype by sampling $[-1, 1]^{448}$, and set the generation to 0; with probability $1 - p$ we perturb a parent genotype with Gaussian noise $\mathcal{N}(0, \sigma^2)$, to obtain a mutated copy of it, and set the generation to that of the parent plus one. In the latter case, we select a parent by performing roulette wheel selection (De Jong, 2016) on the age (i.e., number of time steps elapsed from birth) of the creatures. In this way, we use age as a proxy for fitness in our open-ended world, and ensure that the individuals most effective at surviving reproduce the most, while keeping some diversity in the population by choosing $p > 0$.

6. In the case of a human observer, they may interact with the creatures by performing "good" (placing food) or "bad" (eliminating a creature) actions, as we shall see in Section "RQ2: human attitude towards ALife".

By virtue of this procedure, our artificial world satisfies the basic conditions for evolution: selection of the fittest, variation of the offspring, and heredity (Darwin, 2004; Lewontin, 1970). Remarkably, evolution is a well-known example of an adaptation mechanism (Sipper et al., 1997): creatures must evolve to changes in their stimuli, including—in our case—human presence, leading us to satisfy the Adaptation objective. We remark that, as a consequence of the above procedure, both the number of creatures and of food units remain constant. In this way, we prevent the population from experiencing extinction before any interaction with humans and subsequent adaptation have taken place. After preliminary experiments, we set $e_{\text{init}} = 100$, $e_{\text{food}} = 20$, $e_{\text{food}} = 0.03$, $\sigma^2 = 0.35$, and $p = 0.1$.

Experiments and discussion

We are interested in characterizing the mutual influence of human and artificial life. To this end, we performed an experimental evaluation and a user study aimed at answering the following two research questions:

RQ1 Does an artificial world subjected to human interaction evolve differently than without human interaction? Have "good" and "bad" human actions a different impact on the evolution?

RQ2 What is the attitude of humans towards ALife? In other words, are they aware of their influence on artificial systems? If so, do they change their behavior accordingly?

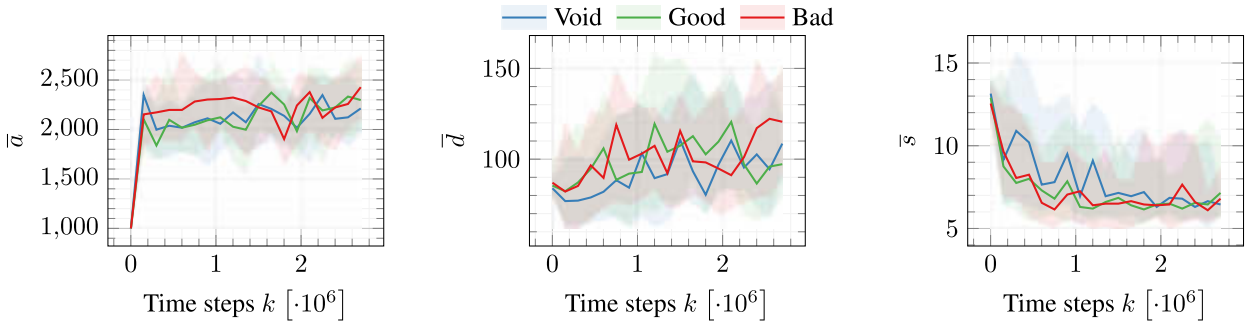
For addressing RQ1, we let our artificial world evolve under the influence of humans, i.e., with humans performing actions on it, and in the void, i.e., without human interactions. To evaluate the changes in the system, we took into consideration some indexes targeted at capturing variations in the artificial creatures. To make the human interaction long enough to impact on the evolution of our artificial creatures, we made use of simulated humans, displaying either good or bad behaviors.

Concerning RQ2, we designed a user study, with a pool of volunteer participants that interact with the simulator. In this case, we focused on appraising the attitude of humans towards our artificial world, by interviewing them and by examining the types of actions they conducted.

RQ1: ALife evolution under human influence

To answer to RQ1, we performed an experimental campaign comprising three types of simulations. First we considered an in-the-void simulation (Void), employed as a baseline, where the virtual creatures are not subject to any exogenous stimulus, i.e., there is no human interaction. For the other two types of simulations, instead, we focused on estimating the impact of humans on the evolution of the artificial world. To this extent, we simulated human interaction with the artificial world at regular intervals during its evolution, to assess if such interactions could steer the evolutionary path of the system. We performed two variants of human-influenced simulations, the first comprising only "good" simulated humans (Good), and the second involving only "bad" simulated humans (Bad). We outline both variants in more detail in the next paragraph. For each type of simulation, namely Void, Good, and Bad, we let the system open-endedly evolve for approximately $2.4 \cdot 10^6$ time steps (corresponding to approximately 3000 generations in the Void case). In all simulations, we set $n_{\text{agents}} = 10$ and $n_{\text{food}} = 35$. For every type of simulation, we performed 30 independent runs, i.e., based on different random seeds, for a total of $3 \cdot 30 = 90$ runs.

For the experiments involving human intervention on the artificial world, i.e., Good and Bad, we simulate humans by replicating the aspects that characterize their interaction with the system. As described in Section "The artificial world", the influence of humans is twofold, unraveling into a temperature increase and in the possibility to perform active actions on the artificial world. To accurately capture both aspects, we repeat the following cycle every 20 000 time steps: (1) we mimic a human approaching the artificial world by increasing the temperature at a randomly chosen point of the world by $\Delta_\tau = 50\,000$ for 1000 time steps (a new point being selected at each time step), and (2) we perform some actions, trying to counterfeit the behavior of a person interacting with the creatures. The actions we simulate are different for Good and Bad, in order to emulate the activity of



(a) Average age $\overline{a}$.

(b) Average traveled distance in the last 100 time steps $\overline{d}$.

(c) Average size $\overline{s}$.

Figure 4: Median and interquartile ranges of three indexes: age $\overline{a}$, traveled distance in the last 100 time steps $\overline{d}$, and size $\overline{s}$, averaged across the population. We use a different color for each type of simulation (Void, Good, or Bad).

a stereotypically "good" or "bad" human. In particular, we deem feeding creatures, i.e., placing food units near the head of a creature, as a positive action, hence associated to Good, while we consider killing a creature as a negative action, thus performed only in Bad. In both cases, we randomly select r_{agents} creatures to undergo the chosen action, i.e., the feeding or the killing, with r_{agents} randomly sampled from $\{1, 2, 3\}$ for every action.

In order to evaluate if evolution is actually taking place in the artificial world, we consider some indexes, which should capture the main features of the creatures. First, we consider the creatures age a, i.e., the amount of time steps since their birth, which should capture how well they adapted to survive in their environment. Then, to estimate movement, i.e., how lively creatures are, we examine the distance d traveled by a creature in the last 100 time steps, instead of the total distance as this indicator would be strongly polluted by how long a creature survives. Last, we take into account the number of body segments composing a creature, i.e., its size s, as a morphological indicator to enlighten us on which agents features are more favored by evolution. For each index, we computed its average value across the living creatures every 150 000 simulation time steps, yielding the averaged indexes $\overline{a}$, $\overline{d}$, and $\overline{s}$. We report the median and the interquartile range of the aforementioned indexes throughout the runs in Figure 4.

From observing the plots of Figure 4, the answer to RQ1 is that evolution indeed takes place in all three types of simulation, steering the system in a clear direction for the three indexes. Hence, we can affirm that the considered system is suitable for studying the mutual influence of artificial creatures and humans.

Focusing on each subplot, we can gain more insight into different aspects of what is happening in the system. First, concerning the average age $\overline{a}$ of the creatures, displayed in Figure 4a, it appears to be rising with the progress of the simulation. Thus, we can conclude that artificial creatures are adapting to the environment, improving their survival rate by becoming more skillful. However, it is unclear if the creatures live longer because they have developed the trait of hunting, i.e., moving to target food, or if they are just randomly roaming the artificial world, thus maximizing the likelihood of encountering a food unit. Figure 4b does not show any apparent trend in this sense to support any of the two hypotheses. Last, we can reason on the morphological traits favored by evolution, by looking at Figure 4c. From the plot it is not difficult to notice how creatures become smaller as evolution progresses, as it is likely easier for them to move, hence increasing the probability of coming across food units.

To deepen our analysis and estimate the impact of human actions on ALife, we can study Figure 4 comparing the trends corresponding to Void, Good, and Bad. Since none of the plots shows significant differences among the colored lines, we assume that none of the measured indexes is impacted by human actions. In addition, the analysis of other indicators, e.g., the length of flagella or the area covered by creatures, here omitted for brevity, gave similar results as the ones of Figure 4. However, we are cautious on declaring that human actions do not affect ALife. In fact, the absence of tangible outcomes could be caused by the too few simulated human interventions on the system or by the random selection of creatures to undergo the chosen actions.

RQ2: human attitude towards ALife

For providing an answer to the second research question, we moved our focus away from the system, to concentrate on the impact interacting with an artificial world has on humans. To this extent, we performed a user study involving of 36 unpaid volunteer participants, 12 females and 24 males, ranging from 18 to 57 years old, who were made to interact with the artificial world for a limited time span, and whose mindset and perceptions were registered by the means of two

Question	Answers
Do you think artificial life exists?	Yes (✓), I don't know (?), No (✗).
How will you behave towards the creatures in the simulation?	Positively (), I am still undecided (?), Negatively ().
Do you think artificial creatures can suffer?	Yes (✓), Maybe (?), No (✗).

Table 1: Pre-interaction questionnaire.

Question	Answers
How would you rate your perceived involvement?	1, 2, 3, 4, 5.
How would you define your behavior towards the creatures in the simulation?	Very Positive (), Fairly Positive (), Fairly Negative (), Very Negative ().
Do you think you have hurt these creatures?	Yes, definitely (✓✓), Yes (✓), I don't know (?), Not really (✗), Definitely not (✗✗).
If you have killed any creature, why have you?	-

Table 2: Post-interaction questionnaire.

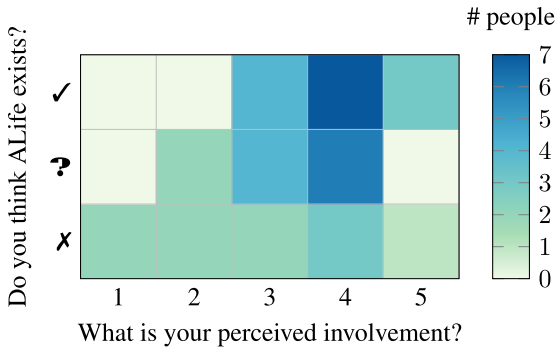


Figure 5: Relationship between participants views on the existence of ALife (pre-interaction "Do you think ALife exists?", on the y-axis) and their perceived involvement in the experiment (post-interaction "How would you rate your perceived involvement?", on the x-axis).

questionnaires.

Concerning the human-system interaction, we aimed at two goals: (a) arousing participants interest towards the artificial world, and (b) maintaining fairness and consistency across evaluations. For achieving the first goal, instead of employing a newly generated artificial world for each participant, we let the system evolve in-the-void for $100\,000$ time steps before interfacing it with humans, with the aim of having lively creatures displaying engaging traits. To tackle the consistency objective, we saved the state of the system (and of all the creatures populating it) after the preliminary in-the-void evolution, and we restored it upon each external interaction, to ensure every person was seeing the artificial world from the same starting point. In addition, each participant was given the same amount of time to interact with the system, which we set to $5\,\mathrm{min}$. We remark that we did not request participants to perform actions or even to pay attention for the entire duration of the experiment: if they were not interested anymore they could just sit idle and avoid active communication with the artificial world.

Since the ultimate goal of this experimentation was to assess the human perception of ALife and the attitude of humans towards artificial creatures, we gave great importance to registering the actions people performed in the simulation, together with their mindset. For the first, we simply recorded every action a person effected on the artificial world, taking note of the time, the location, and the type of action, i.e., placing food or killing a creature. Concerning the latter, we interviewed the participants before and after the experiments, asking them to fill out two short questionnaires. The pre-interaction questionnaire, reported in Table 1, aimed at evaluating the general approach towards ALife, together with the expected behavior towards creatures populating an artificial world. Similarly, we designed the post-interaction questionnaire, described in Table 2, to capture the feelings after interacting with ALife and to let participants self-assess their conduct.

We display aggregations of the collected data in Figures 5 to 7. First, we correlated the results gathered from the first question of both questionnaires, i.e., pre-interaction "Do you think artificial life exists?" and post-interaction "How would you rate your perceived involvement?", obtaining the heat map of Figure 5. This is the first noteworthy result of our study: participants who believe in the existence of ALife tend to feel more involved when interacting with artificial creatures. In particular, we speculate that such people perceive the importance of their role and the influence of their actions on the artificial world, thus feeling more concerned with it and more prone to actively interact with artificial creatures.

Moving on to Figure 6, we report the relationship between participants planned behavior (pre-interaction "How will you behave towards the creatures in the simulation?"), their self-assessed behavior (post-interaction "How would you define your behavior towards the creatures in the simulation?"), and the ratio of positive actions performed by each participant. The foremost observation we can make from such box plots, is that nobody decided to act negatively before interacting with the artificial world, which reveals a general tendency to avoid opting for negative actions in the first place. Such tendency is also confirmed by the overall ratio of good actions performed, which is always above 0.7. Focusing more on how participants rated their own conduct, we can notice that they are generally aware of the impact of

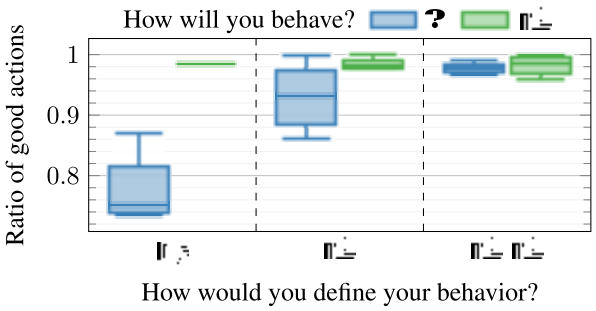


Figure 6: Relationship between participants planned behavior ("How will you behave towards the creatures in the simulation?", color), self-assessed behavior ("How would you define your behavior towards the creatures in the simulation?", x-axis), and the ratio of good actions performed (y-axis).

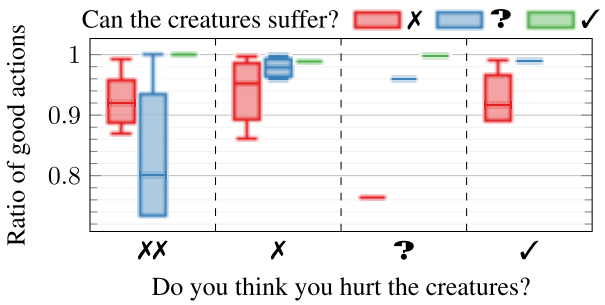


Figure 7: Relationship between participants feelings concerning the creatures ability to suffer ("Do you think artificial creatures can suffer?", color), the perception about hurting them ("Do you think you have hurt these creatures?", x-axis), and the ratio of good actions performed (y-axis).

their actions: those who described their behavior as fairly-negative ( ) show in general a lower ratio of good actions. Last, we can reason on participants coherence: in all cases those who decided to act positively towards the creatures in the artificial world ( ) show an averagely higher ratio of good actions performed than those who were undecided (?).

Another thought-provoking result is shown in Figure 7, where we report the results related to the participants perception of ALife suffering. From this figure, we note that the participants who perceive the creatures as alive, i.e., able to suffer, behave accordingly, trying to feed them and not to kill them. For the others, instead, the lower ratio of good actions performed reflects their perception of the creatures as a non-living artifact. At the same time, participants who performed more positive actions still believe they hurt the creatures far more than those who acted negatively. This, in our opinion, highlights how some people are extremely disconnected from ALife, and they consider it as a mere artifact. These results are in line with (Bongard and Anetsberger, 2016), which found that unpaid participants to a crowd-sourcing robotics experiment provided honest feedback.

Last, we examined the answers to the post-interaction question "If you have killed any creature, why have you?" to immerse ourselves in the motivations pushing participants to eliminate a creature from the artificial world. The collected responses were disparate, but they were mainly clustered into three categories: (a) curiosity, (b) mistakes, or (c) will to remove creatures displaying traits which people disliked. Among the listed categories, the first two ones were expected and, to us, reflect normal traits of human personalities. Conversely, the last answer is more disturbing.

Summing up, finding an answer to RQ2 is not straightforward. The experimental outcomes suggest that humans have mixed feelings with respect to ALife: some believe in its existence, feel involved when interacting with it, recognize their impact, and tend to act positively, while others perceive ALife as a pure fiction, not really worthy of attention and caution upon interaction.

Concluding remarks

We studied the mutual influence between human and Artificial Life (ALife) from two symmetrical perspectives. First, we aimed at assessing the impact of an external superior entity on an artificial world, measuring if the evolutionary path of the system could be steered by human actions on it, and if the creatures populating it would adapt to the external influence, be it positive or negative. Furthermore, we tried to characterize the mindset and the behavior of people interacting with an artificial world where they played the role of superior entities yielding power of life and death upon its creatures.

To this end, we designed an artificial world based on the pillars of interaction, adaptation, and realism, and we performed a twofold experimental evaluation including real and simulated humans, focusing on the system evolution and on the human attitude.

Our results show that our artificial world is capable of evolving in the presence of an external influence, yet it is hard to appraise the impact of people on artificial creatures. Moreover, we find both positively involved cooperative attitudes and fairly detached negative perceptions with respect to ALife among participants. We believe our work stresses how delicate and contradictory is the relationship between the human and the artificial, the living and the machine. In the future, we plan to scale our experiment and carry on deeper analyses by involving more participants and encompassing expanded psychological and philosophical validations.

Acknowledgments

The authors wish to thank the participants to the human experiment for their patience and dedication.

References

Bartoli, A., Catto, M., De Lorenzo, A., Medvet, E., and Talamini, J. (2020). Mechanisms of Social Learning in Evolved Artificial Life. In *ALIFE 2020: The 2020 Conference on Artificial Life*, pages 190–198. MIT Press.

Bongard, J. and Anetsberger, J. (2016). Robots can ground crowd-proposed symbols by forming theories of group mind. In *ALIFE 2016, the Fifteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 684–691. MIT Press.

Bongard, J. C., Cheney, N., Mahoor, Z., and Powers, J. P. (2018). The Role of Embodiment in Open-Ended Evolution. In *OOE3: The Third Workshop on Open-Ended Evolution*.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Clyde, A. (1998). Electronic pets. *Teacher Librarian*, 25(5):34.

Darwin, C. (2004). *On the origin of species, 1859*. Routledge.

De Jong, K. (2016). Evolutionary computation: a unified approach. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 185–199.

Dewdney, A. K. (1984). Core wars. *Sci. Amer.*

Grand, S. and Cliff, D. (1998). Creatures: Entertainment software agents with artificial life. *Autonomous Agents and Multi-Agent Systems*, 1(1):39–57.

Hobbes, T. (1651). *Leviathan or The Matter, Forme and Power of a Commonwealth Ecclesiasticall and Civil*.

Hunt, E. (2016). Tay, Microsoft's AI chatbot, gets a crash course in racism from Twitter. *The Guardian*, 24(3):2016.

Kasperkevic, J. (2015). Google says sorry for racist auto-tag in photo app. *The Guardian*, 1:2015.

Langton, C. G. (1997). Artificial life: An overview.

Lenski, R. E., Ofria, C., Pennock, R. T., and Adami, C. (2003). The evolutionary origin of complex features. *Nature*, 423(6936):139–144.

Lewontin, R. C. (1970). The units of selection. *Annual review of ecology and systematics*, 1(1):1–18.

Mahoor, Z., Felag, J., and Bongard, J. (2017). Morphology dictates a robot's ability to ground crowd-proposed language. *arXiv preprint arXiv:1712.05881*.

Milgram, S. (1963). Behavioral study of obedience. *The Journal of abnormal and social psychology*, 67(4):371.

Mori, M., MacDorman, K. F., and Kageki, N. (2012). The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, 19(2):98–100.

Ofria, C. and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Artificial life*, 10(2):191–229.

Pigozzi, F. (2022). Robots: the century past and the century ahead.

Ray, T. S. (1992). Evolution, ecology and optimization of digital organisms. *Santa Fe*.

Rousseau, J.-J. (1755). *Discourse on the Origin and Basis of Inequality Among Men*.

Sipper, M., Sanchez, E., Mange, D., Tomassini, M., Pérez-Uribe, A., and Stauffer, A. (1997). A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems. *IEEE Transactions on Evolutionary Computation*, 1(1):83–97.

Soros, L. and Stanley, K. (2014). Identifying necessary conditions for open-ended evolution through the artificial life world of chromaria. In *ALIFE 14: The Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 793–800. MIT Press.

Stewart, A. (2005). *The earth moved: on the remarkable achievements of earthworms*. Algonquin Books.

Suarez, J., Du, Y., Isola, P., and Mordatch, I. (2019). Neural mmo: A massively multiagent game environment for training and evaluating intelligent agents. *arXiv preprint arXiv:1903.00784*.

Terry, J. K., Black, B., Grammel, N., Jayakumar, M., Hari, A., Sullivan, R., Santos, L., Dieffendahl, C., Horsch, C., Perez-Vicente, R., et al. (2021). Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34.

Ventrella, J. (2005). Genepool: Exploring the interaction between natural selection and sexual selection. In *Artificial life models in software*, pages 81–96. Springer.

Vincent, J. (2019). That video of a robot getting beaten is fake, but feeling sorry for machines is no joke.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee.

Wachowski, A., Wachowski, L., Reeves, K., Fishburne, L., Moss, C.-A., Weaving, H., Foster, G., Pantoliano, J., and Staenberg, Z. (1999). *Matrix*. Warner Home Video Burbank, CA.

Yaeger, L. et al. (1994). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or Poly World: Life in a new context. In *Santa Fe Institute Studies in the Science of Complexity*, volume 17.

On the Entanglement between Evolvability and Fitness: an Experimental Study on Voxel-based Soft Robots

Andrea Ferigo[1], L. B. Soros[2], Eric Medvet[3], and Giovanni Iacca[1]

[1]Department of Information Engineering and Computer Science, University of Trento, Italy
[2]Cross Labs, Cross Compass Ltd., Kyoto, Japan
[3]Department of Engineering and Architecture, University of Trieste, Italy
andrea.ferigo@unitn.it, lisa.soros@cross-compass.com, emedvet@units.it, giovanni.iacca@unitn.it

Abstract

The concept of evolvability, that is the capacity to produce heritable and adaptive phenotypic variation, is crucial in the current understanding of evolution. However, while its meaning is intuitive, there is no consensus on how to quantitatively measure it. As a consequence, in evolutionary robotics, it is hard to evaluate the interplay between evolvability and fitness and its dependency on key factors like the evolutionary algorithm (EA) or the representation of the individuals. Here, we propose to use MAP-Elites, a well-established Quality Diversity EA, as a support structure for measuring evolvability and for highlighting its interplay with fitness. We map the solutions generated during the evolutionary process to a MAP-Elites-like grid and then visualize their fitness and evolvability as maps. This procedures does not affect the EA execution and can hence be applied to any EA: it only requires to have two descriptors for the solutions that can be used to meaningfully characterize them. We apply this general methodology to the case of Voxel-based Soft Robots (VSR), a kind of modular robots with a body composed of uniform elements whose volume is individually varied by the robot brain. Namely, we optimize the robots for the task of locomotion using evolutionary computation. We consider four representations, i.e., ways of transforming a genotype into a robot, two for the brain only and two for both body and brain of the VSR, and two EAs (MAP-Elites and a simple evolutionary strategy) and examine the evolvability and fitness maps. The experiments suggest that our methodology permits us to discover interesting patterns in the maps: fitness maps appear to depend more on the representation of the solution, whereas evolvability maps appear to depend more on the EA. As an aside, we find that MAP-Elites is particularly effective in the simultaneous evolution of the body and the brain of Voxel-based Soft Robots.

1 Introduction

Evolution in nature has created a diversity of viable ways of living occupying vastly different niches (plants, mammals, etc.). Yet, despite the rich diversity observed in nature, much of this evolution depends on variation on common ancestors. For example, all breeds of domesticated dog (Canis familiaris), from poodles to Great Danes, are hypothesized to have descended from the grey wolf (Canis lupus) (Coppinger and Smith, 1983) or some other wild canid (Koler-Matznick, 2002)[1].

What is it about these common ancestors that makes them so likely to further evolve into high-quality descendants very distinct from themselves? Capturing this property, called *evolvability*, and replicating it inside an algorithmic process is a challenge for advancing our theoretical understanding of natural and artificial evolutionary systems. It is also likely that figuring out how to discover highly evolvable individuals will have positive practical results for the purposes of optimization by virtue of avoiding premature convergence (Squillero and Tonda, 2016).

This paper explores the concept of evolvability in the latter sense (as a tool for achieving high performance in an engineering context). In particular, we use two fundamentally different Evolutionary Algorithms (EAs): a simple form of Evolution Strategies (ES) (Beyer and Schwefel, 2002), and the Quality Diversity algorithm MAP-Elites (Cully et al., 2015). We apply them to discover highly evolvable morphologies and controllers for Voxel-based Soft Robots (VSRs). We choose these EAs as they represent two different ways to conduct evolutionary search: one mainly aimed at exploitation (ES), the other one mainly aimed at exploration (MAP-Elites). On the other hand, we choose VSRs for this study mainly due to their modularity, which makes them particular expressive and suitable for constituting an autonomous robotic ecosystem (e.g., for space exploration applications (Methenitis et al., 2015)) in which evolvability would be a key feature.

We review the concept of evolvability with a focus on its applications in artificial evolutionary systems. We introduce a method for keeping track of the evolvability and fitness of solutions generated during the execution of potentially any EA. We use a grid-like structure for storing the most relevant solutions in a way that enables a convenient visualization of their fitness and evolvability at the end of the

[1]Interestingly, when Darwin wrote about evolution, he concluded that the diversity of dogs necessarily must result from interbreeding of many kinds of wild dogs (Darwin, 1875). However, molecular dating techniques in recent years suggest that this conclusion is likely wrong (Wayne et al., 1991).

evolutionary process: this methodology only requires two descriptors for characterizing the solutions and does not interfere with the EA execution. Since fitness and evolvability of individual solutions are placed in a grid-like structure, the analysis of their interplay is facilitated. We apply this general methodology to the case of the optimization of VSRs (brain only and both body and brain) for the task of locomotion. We consider four different representations and combine them with the two EAs mentioned above and apply our methodology to discover insights about how the EA and the representation impact on the interplay between fitness and evolvability.

Results indicate that both the choices of the representation and the EA have a major impact not only on the quality of the solution, but also on how the search space is explored, which ultimately reflects in different observations of evolvability. More precisely, we observe that the fitness distribution in a phenotypic space determined by predefined descriptors (more on this below) is mostly determined by the adopted representation; on the contrary, the evolvability distribution over the same phenotypic space appears to be determined by the EA. Hence we conclude that fitness and evolvability are somehow "entangled" and that this entanglement is affected by multiple aspects of evolutionary systems.

The rest of the paper is structured as follows. In the next section, we introduce the background concepts and briefly summarize the related works. In Section 3, we describe the methods. Then, we present the results in Section 4. Finally, we give the conclusions in Section 5.

2 Background and related works

Evolvability is essentially a measure of potential for evolutionary innovation. However, measures of evolvability differ in what features for evolutionary innovation are salient (Pigliucci, 2008). Per one popular definition, "Evolvability is the ability of a biological system to produce phenotypic variation that is both heritable and adaptive" (Payne and Wagner, 2019; Nordmoen et al., 2021). It is important to note two distinct components of this definition: that there is variation (i.e., diversity) being passed from parent to offspring, and that this variation leads to positive effects on fitness. Interestingly and importantly, measures and studies from artificial life (a primary domain of interest for evolvability studies related to artificial evolution) regard evolvability purely as adaptation (Medvet et al., 2017; Veenstra et al., 2020; Liu et al., 2022; Tarapore and Mouret, 2015), or evolvability as diversification (Mengistu et al., 2016; Gajewski et al., 2019; Lehman and Stanley, 2011b, 2013; Lim et al., 2021; Carlo et al., 2021), but not both.

Searching directly for evolvability has become a recently popular trend. In Evolvability Search (Mengistu et al., 2016), the fitness function of a traditional EA rewards high evolvability (in this diversity-oriented interpretation, it is the number of distinct behaviors in the set of offspring generated by an individual) instead of rewarding maximizing a domain-specific objective. This algorithm is shown to outperform both greedy optimization and novelty search (Lehman and Stanley, 2011a). The subsequent Evolvability Evolution Strategy (E-ES) (Gajewski et al., 2019) introduces improvements in terms of computational expense to scale to deep neural networks. Quality Evolvability ES (QE-ES) (Katona et al., 2021) builds on both of these algorithms, simultaneously optimizing for both evolvability (as diversity) and fitness with non-dominated sorting, as in NSGA-II (Deb et al., 2002). Note that unlike Quality Diversity algorithms (Pugh et al., 2016), which seek to discover a diverse population of high-performing individuals, the goal in Quality Evolvability is to discover a single individual with diverse offspring. The work reported in this current paper will leverage the Quality Diversity algorithm MAP-Elites (see Section 3) to find diverse populations of highly evolvable individuals.

3 Methods

In the following, we present the main methods and tools we used in our study: the VSRs, the EAs used for evolving them, the descriptors adopted for characterizing evolved VSRs, and the evolvability metric.

Voxel-based Soft Robots

First presented by Hiller and Lipson (2011), VSRs are a type of modular soft robot composed of cubic elements that can vary their volume according to a control signal that is generated by a controller. In this work, we use the 2D (yet, physically plausible) VSR simulator presented by Medvet et al. (2020a). In fact, using a two-dimensional model allows to reduce the numerical complexity of the robot simulations, without losing the potential variety of robot shapes and behaviors.

From a conceptual level, a VSRs can be seen as a composition of two components: the *body*, i.e., a set of voxels arranged in a given shape, and the *brain*, i.e., a controller that produces the control signal for each voxel in the body. The details of these two components are reported below.

Body. The body of a VSR is defined by a number of deformable squares, called *voxels*, arranged (in our case) in a 2D grid (Figure 1). Pairs of adjacent voxels are glued together at their two common vertexes. During the simulation, each voxel changes its area as a result of the combined action of (a) external forces imposed by bodies in contact with the voxel, namely other voxels and the ground, and (b) an internal force that makes the voxel expand or contract. The internal force at time t is determined by a *control value* $c(t) \in [-1, 1]$, where -1 means maximum area expansion and 1 means maximum area contraction. The control value is itself determined, for each voxel of the body, by the brain of the VSR, described in the next section.

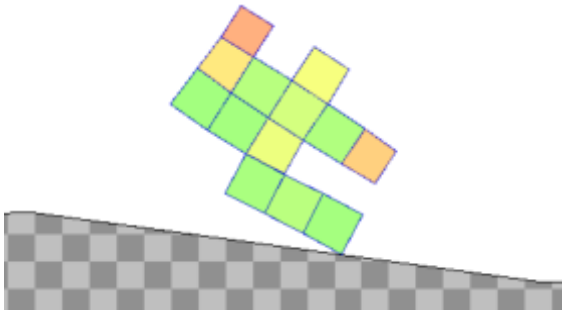


Figure 1: An example VSR with a body  enclosed in a 4×5 grid; namely, one of those obtained in our experiments (see Section 4 and Figure 5a). The color of each voxel depends on its current area: green for expanded, red for contracted, yellow for equal to the rest area.

We rely on 2D-VSR-Sim (Medvet et al., 2020a) for simulating the VSR body. In brief, in 2D-VSR-Sim voxels are modeled as an assembly of four masses (at the vertexes of the square) and multiple spring-damper systems connecting them. The internal force is modeled as an instantaneous change of the resting length of the spring-damper systems.

Clearly, the body of a VSR, together with its brain, plays a crucial role in making the robot more or less effective for a given task. It follows that the body, i.e., how to organize voxels in the 2D grid, can be optimized.

Brain. The brain generates the control signal for each voxel and hence acts as the controller for the robot. In this work, we use an open-loop controller where the control signal depends only on the current time t. Namely, we use a sinusoidal controller for which $c(t) = \sin(2\pi ft + \phi)$. Despite appearing trivial, this form of open-loop controller has been used for VSRs to achieve effective behaviors in different environments (Corucci et al., 2018) as well as for investigating complex adaptation dynamics (Kriegman et al., 2018).

The values of the frequency f and phase ϕ can be different among the voxels in the body. All together, they constitute the parameters of the controller and may be optimized for obtaining a desired behavior.

Evolutionary Algorithms

To optimize VSRs, we use two different EAs: a traditional fitness-driven EA, namely a simple form of Evolution Strategies (ES) (Beyer and Schwefel, 2002), and a Quality Diversity algorithm, namely MAP-Elites (ME) (Cully et al., 2015).

We decide to employ these two EAs because they support diversity, a substrate for evolvability, in radically different ways. While in ES the entire offspring is generated from a prototype individual deriving from a small subset of the best parents, in ME the offspring is generated by randomly sampling parents from the entire population, such that worse, but diverse parents can reproduce too.

Next, we briefly summarize the salient elements of the two algorithms. We denote by G the search space, i.e., the space where individuals are defined. In this study, we have $G = \mathbb{R}^p$, hence we denote individuals as numerical vectors $\boldsymbol{g} \in \mathbb{R}^p$; we remark that some of the components, namely, MAP-Elites, are defined for more general search spaces. We also assume that the fitness of an individual, i.e., its quality $f(\boldsymbol{g})$, can be evaluated as a single numerical value, i.e., with a fitness function $f : G \to \mathbb{R}$, for which it holds that the greater, the better.

Evolution Strategies (ES). In our simple form of ES, we iteratively evolve a fixed-size population of numerical vectors as individuals, i.e., $G = \mathbb{R}^p$, where p is the number of parameters to optimize.

Initially, we build the population by randomly generating n_{pop} individuals: namely, we build each individual $\boldsymbol{g}$ by sampling from the uniform distribution in $[-1, 1]$, i.e., $\boldsymbol{g} = (g_1, \dots, g_p)$ and $g_i \sim U(-1, 1)$. Then, we iterate the following steps (generations) until we have done n_{eval} fitness evaluations. First, we select the best $\frac{n_{\text{pop}}}{4}$ individuals as parents, i.e., those with the greatest fitness in the current population, and compute their element-wise mean $\boldsymbol{\mu} \in \mathbb{R}^p$. Then, we build $n_{\text{pop}} - 1$ offspring individuals, each one by adding to each element of $\boldsymbol{\mu}$ a Gaussian noise $\sim N(0, \sigma^2)$. Finally, we build the next population by taking the offspring and the best parent, i.e., we employ a form of elitism.

At the end of the evolution, ES outputs a single best solution being the individual with the largest fitness in the population at the last iteration of the algorithm.

MAP-Elites (ME). Multidimensional Archive of Phenotypic Elites (commonly known as MAP-Elites, here further abbreviated as ME), was originally introduced by Cully et al. (2015) for evolving robust behaviors in robots.

A key requirement in ME is the availability of some numerical *descriptors* of the solutions: the descriptors should be good at characterizing the solutions with respect to the problem being tackled, but, ideally, they should be orthogonal with respect to the fitness. Formally, we denote by $d : G \to \mathbb{R}^m$ the function for computing the descriptors $d(\boldsymbol{g}) = (d_1(\boldsymbol{g}), \dots, d_m(\boldsymbol{g}))$ of an individual $\boldsymbol{g}$. We assume that each descriptor $d_i(\boldsymbol{g})$ is defined in a bounded interval $D_i = [d_{i,\text{min}}, d_{i,\text{max}}]$. Given a number n_{bin} of bins, each individual can be mapped to the cell of an m-dimensional grid by considering, for each descriptor d_i, the index of the equal width bin of D_i in which the descriptor value falls. That is, we define the function $\boldsymbol{c} : G \to \mathbb{N}^m$ as $\boldsymbol{c}(\boldsymbol{g}) = (c_1(\boldsymbol{g}), \dots, c_m(\boldsymbol{g}))$ where $c_i(\boldsymbol{g}) = k \in \mathbb{N}$ such that $d_{i,\text{min}} + k\frac{|D_i|}{n_{\text{bin}}} \leq d_i(\boldsymbol{g}) < d_{i,\text{min}} + (k+1)\frac{|D_i|}{n_{\text{bin}}}$ with $|D_i| = d_{i,\text{max}} - d_{i,\text{min}}$. We say that $\boldsymbol{c}(\boldsymbol{g})$ are the coordinates of $\boldsymbol{g}$ in the descriptor grid.

Differently from ES, ME does not evolve a fixed-size population of individuals: the population, called here *archive*, can increase in size during the evolution, up to $m^{n_{\text{bin}}}$ individuals, yet never decreases. At the beginning of the evolution, we populate the initially empty archive A by repeating the following steps n_{init} times: first, we randomly generate a new individual $\boldsymbol{g}$, then, we add $\boldsymbol{g}$ to A if no other individual $\boldsymbol{g}'$ exists in A at the same coordinates, i.e., such that $\boldsymbol{c}(\boldsymbol{g}') = \boldsymbol{c}(\boldsymbol{g})$, or, otherwise, if such $\boldsymbol{g}'$ exists and $f(\boldsymbol{g}) \geq f(\boldsymbol{g}')$. In the latter case, we remove $\boldsymbol{g}'$ from A.

After the initialization, we iterate the following steps (generations) until we have done n_{eval} fitness evaluations. We select n_{parent} individuals from A with uniform probability as parents. For each parent $\boldsymbol{g}$, we apply a genetic operator (mutation) $o : G \to G$ and obtain a child $\boldsymbol{g}' = o(\boldsymbol{g})$. Then we add $\boldsymbol{g}'$ to A as in the initialization procedure, i.e., if no other individual $\boldsymbol{g}''$ exists in A such that $\boldsymbol{c}(\boldsymbol{g}'') = \boldsymbol{c}(\boldsymbol{g}')$ or, otherwise, if such $\boldsymbol{g}''$ exists and $f(\boldsymbol{g}') \geq f(\boldsymbol{g}'')$. In the latter case, we remove $\boldsymbol{g}''$ from A.

At the end of the process, the algorithm does not return a single best individual, but the entire archive A (from which, in principle, one may choose as single best solution the one with the highest fitness value).

Since in this study we work with numerical vectors as individuals, also in ME we use as mutation the Gaussian mutation that we adopt in ES, i.e., $o(\boldsymbol{g}) = \boldsymbol{g} + \boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)$ and $\alpha_i \sim N(0, \sigma^2)$. Accordingly, we build each of the n_{init} initial individuals by sampling the uniform distribution in $[-1, 1]$, as in ES.

Evolving VSRs with ES and ME

We want to evolve VSRs for the task of locomotion, i.e., moving along a surface as fast as possible, using the EAs described above. For this purpose, we need to define the solution representation, i.e., how to map a numerical vector $\boldsymbol{g} \in \mathbb{R}^p$ to a VSR, and the fitness function that quantifies the degree to which a VSR is doing locomotion. Moreover, for ME we also need to define the descriptors, i.e., some quantitative measures suitable for characterizing VSRs doing locomotion. In the following, we describe the choices adopted in this study for the fitness function, the representation, and the descriptors.

Fitness function for locomotion. Given a VSR, we perform a simulation lasting 60 s (simulated time) where the VSR is initially placed right above an terrain. We take as fitness of the VSR its average velocity v_x along the x-axis measured by considering its center of mass position at $t = 0$ s and at $t = 60$ s.

To make the task slightly more challenging, we consider a hilly terrain, instead of a flat, even terrain. The height of the terrain varies randomly along the x-axis: we use a single randomly generated terrain for all the experiments.

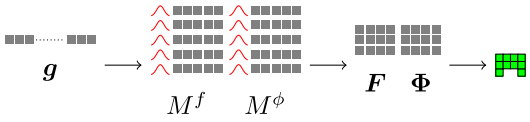
VSR representations. We consider four different ways of mapping numerical vectors to VSRs. They result from the combination of two options for two axes: direct vs. indirect representation; body and brain vs. brain only.

The direct representation for brain only optimization (DB) works as follows. Given a body consisting of n voxels, we map a vector $\boldsymbol{g} \in \mathbb{R}^p$, with $p = 2n$, to a VSR with the given body and equipped with the sinusoidal controller where, for each i-th voxel, the frequency is the i-th element of the first half ($\boldsymbol{f}$) of $\boldsymbol{g}$ and the phase is the i-th element of the second half ($\boldsymbol{\phi}$) of $\boldsymbol{g}$, with $\boldsymbol{g} = [\boldsymbol{f}\ \boldsymbol{\phi}]$. Since we use, for this representation, a 10-voxel body, that we call "biped", with two voxels as "legs" and a "trunk" of 4×2 voxels  , it follows that for this representation we have $p = 20$. We schematize this representation in Figure 2a.

The indirect representation for brain only optimization (IB) is based on the concept of Gaussian Mixture Model (GMM) (Lindsay, 1995) and has already been used for 2D VSRs by Medvet et al. (2020b): it works as follows. Let n_{GMM} be the number of bi-variate Gaussian models in the mixture and let $w \times h$ the size of a 2D grid enclosing the VSR body—i.e., in the case of the biped, $w = 4$, $h = 3$. Each bi-variate Gaussian is described by five parameters: μ_x, μ_y, σ_x, σ_y, and β. We first map an individual $\boldsymbol{g} \in \mathbb{R}^p$, with $p = 2 \cdot 5 n_{\text{GMM}}$ to two sets of n_{GMM} bi-variate Gaussian models, one for the frequency, $M^f = \left\{\left(\mu_x^{f,i}, \mu_y^{f,i}, \sigma_x^{f,i}, \sigma_y^{f,i}, \beta^{f,i}\right)\right\}_i$, and one for the phase, $M^\phi = \left\{\left(\mu_x^{\phi,i}, \mu_y^{\phi,i}, \sigma_x^{\phi,i}, \sigma_y^{\phi,i}, \beta^{\phi,i}\right)\right\}_i$, of a sinusoidal controller; when mapping to σ_x and σ_y, we take the absolute value of the corresponding elements of $\boldsymbol{g}$. Then, we build a VSR with the given body and with a sinusoidal controller where frequencies and phases are determined as follows. For a voxel at position x, y (in the $w \times h$ 2D grid), we set the frequency to $f = \boldsymbol{F}_{x,y} = \text{mix}(x', y'; M^f) = \sum_{i=1}^{n_{\text{GMM}}} \frac{\alpha^{f,i}}{2\pi\sigma_x^{f,i}\sigma_y^{f,i}} \exp\left(-\frac{1}{2}\left(\frac{(x'-\mu_x^{f,i})^2}{\sigma_x^{f,i}} + \frac{(y'-\mu_y^{f,i})^2}{\sigma_y^{f,i}}\right)\right)$, where $x' = \frac{x}{w}$ and $y' = \frac{y}{h}$. Similarly, we set the phase to $\phi = \boldsymbol{\Phi}_{x,y} = \text{mix}(x', y'; M^\phi)$. In our experiments, we set $n_{\text{GMM}} = 5$ and apply this representation to the biped, hence $p = 50$. We schematize this representation in Figure 2b.

The direct representation for body and brain (DB^2) works as follows. Let n_{size} be the side of a square enclosing the largest representable body, i.e., a square of $n_{\text{size}} \times n_{\text{size}}$ voxels. We first take the vector $\boldsymbol{g} \in \mathbb{R}^p$, with $p = 3n_{\text{size}}^2$ and reshape it to three matrices $\boldsymbol{B}, \boldsymbol{F}, \boldsymbol{\Phi}$, each defined in $\mathbb{R}^{n_{\text{size}} \times n_{\text{size}}}$. We transform $\boldsymbol{B}$ to a Boolean matrix $\boldsymbol{B'} = \{\text{T}, \text{F}\}^{n_{\text{size}} \times n_{\text{size}}}$ where $B'_{x,y}$ is set to true if and only if $b_{x,y}$ is greater or equal than the median value of $\boldsymbol{B}$. Then, we build the body by considering the largest connected component of $\boldsymbol{B'}$ elements set to true and putting a voxel in the square at the coordinates of each element of such set. Finally, we build a sinusoidal controller for the body where the frequency and phase for each voxel at coordinates x, y are taken from the corresponding elements of $\boldsymbol{F}$ and $\boldsymbol{\Phi}$. In

(a) Direct brain only (DB)



(b) Indirect brain only (IB)

Figure 2: Processing steps of the DB and IB representations for the brain only, starting from the numerical vector $\boldsymbol{g}$ to the final VSR. Grids of gray squares represent vectors (e.g., $\in \mathbb{R}^3$) or matrices (e.g., $\in \mathbb{R}^{3\times 2}$). $\wedge$ represents a bi-variate Gaussian model with its 5 parameters.

our experiments we set $n_{\text{size}} = 10$, hence $p = 300$. We schematize this representation in Figure 3a.

Finally, the indirect representation for body and brain (IB^2), based on GMM too, works as follows. Let n_{size} be the side of a square enclosing the largest representable body and let n_{GMM} be the number of bi-variate Gaussian models in the mixture. We first map a $\boldsymbol{g} \in \mathbb{R}^p$, with $p = 3 \cdot 5 n_{\text{GMM}}$ to three sets of n_{GMM} bi-variate Gaussian models, M^b, M^f, and M^ϕ, as for the IB case. We build a matrix $\boldsymbol{B} \in \mathbb{R}^{n_{\text{size}} \times n_{\text{size}}}$ by setting each element $B_{x,y} = \text{mix}(x', y'; M^B)$, with $x' = \frac{x}{n_{\text{size}}}$ and $y' = \frac{y}{n_{\text{size}}}$. Then, we build a body from $\boldsymbol{B}$ through an intermediate Boolean matrix $\boldsymbol{B}'$, as in the DB^2 case, but with 0.5 as threshold instead of the median of $\boldsymbol{B}$. Finally, we build a sinusoidal controller for the body where the frequency and phase for each voxel at coordinates x, y are set to $f = \text{mix}(x', y'; M^f)$ and $\phi = \text{mix}(x', y'; M^\phi)$. In our experiments we set $n_{\text{size}} = 10$ and $n_{\text{GMM}} = 5$, thus $p = 75$. We schematize this representation in Figure 3b.

VSR descriptors. We consider two sets of two descriptors: one characterizes the VSR body only, and the other characterizes the behavior, hence implicitly both together characterize the body and the brain. We use the former when using ME with the DB^2 and IB^2 representations, hence when evolving body and brain, and the latter when using ME with the DB and IB representations, hence when evolving the brain only.

As body descriptors, we simply consider the width and height of the smallest 2D grid enclosing the VSR body. Note that in our experiments, both descriptors are defined in $[1, 10] \in \mathbb{N}$, since $n_{\text{size}} = 10$ for DB^2, IB^2 and bodies are at most 10×10 large. We denote these descriptors by w and h respectively.

For the behavior descriptors, we consider the temporal pattern according to which the VSR touches the ground when doing locomotion. In detail, given a simulation of t_f seconds of the VSR, we proceed as follows. First, we build two binary signals $\tau_{\text{back}}, \tau_{\text{front}} : [0, t_f] \to \{0, 1\}$. At each t, $\tau_{\text{back}}(t)$ is 1 if at least one point of the leftmost half of the VSR was in contact, at t, with the ground, and 0 otherwise; with point of the leftmost half we mean a point whose x-coordinate is lower than the x-coordinate of the center of mass of the VSR. Similarly, $\tau_{\text{forward}}(t)$ considers the rightmost half of the VSR. Then, we compute the Fast-Fourier Transform for both signals and, for each one, we compute the amount of energy in the band 0 Hz–2 Hz and in the band 0 Hz–5 Hz. Finally, we define the descriptors ρ_{back} and ρ_{front} as the rate between the 0 Hz–2 Hz and 0 Hz–5 Hz energies for the τ_{back} and τ_{front} signals; both descriptors are defined in $[0, 1]$. Intuitively, the lower the gait pace, the greater the value of the descriptors.

Measuring and visualizing evolvability

As discussed earlier, evolvability is a characteristic of an evolutionary system that describes how much it is able to generate different and better performing individuals.

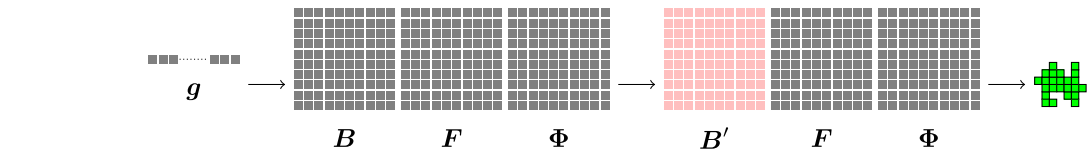
In this study, we define evolvability as a measure of an individual at a given iteration during the execution of an iterative EA. Formally, we define the evolvability $e(\boldsymbol{g}, i)$ of an individual $\boldsymbol{g}$ at iteration i as:

$$e(\boldsymbol{g}, i) = \frac{1}{|C_{\boldsymbol{g},i}|} \sum_{\boldsymbol{g}' \in C_{\boldsymbol{g},i}} f(\boldsymbol{g}') - f(\boldsymbol{g}), \tag{1}$$

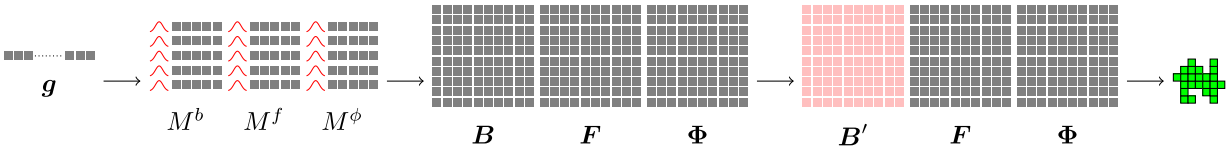
where $C_{\boldsymbol{g},i}$ is the multiset of all the individuals generated from $\boldsymbol{g}$, i.e., its children, up to iteration i and $f(\boldsymbol{g})$ is the fitness of the individual $\boldsymbol{g}$.

While for ME the notion of children of an individual is trivial, since each (non-initial) individual has exactly one parent, in ES we assume that all the $n_{\text{pop}} - 1$ individuals generated at a given iteration are children of all the $\frac{n_{\text{pop}}}{4}$ parents chosen at that iteration.

For providing an aggregate view of the evolvability of an entire EA execution based on the individual measure of Equation (1), and with the aim of balancing the trade-off between detail and compactness of that view, we proceed as follows. During the execution of the EA, given some descriptors $d_1, \ldots, d_m$, each defined as $d_i : G \to [d_{i,\min}, d_{i,\max}]$, and a number n_{bin} of bins, we maintain an initially empty archive A' with the same update policy of ME: whenever a new individual $\boldsymbol{g}$ is generated in the EA, it is added to A' if no other individuals exist in A at the same coordinates of $\boldsymbol{g}$ and it replaces, if any, the existing individual at those coordinates. At the end of the evolution, we analyze the individuals in A' by looking at their fitness and evolvability. Namely, if we use two descriptors, we can plot the values of $f(\boldsymbol{g})$ and $e(\boldsymbol{g}, i_{\text{last}})$ of each $\boldsymbol{g} \in A'$ in the form of two color maps, i_{last} being the last iteration in the EA execution. We refer to these plots as *fitness and evolvability maps*, respectively.



(a) Direct body and brain (DB^2)



(b) Indirect body and brain (IB^2)

Figure 3: Processing steps of the DB^2 and IB^2 representations for body and brain, starting from the numerical vector $\boldsymbol{g}$ to the final VSR. The visual syntax is the same of Figure 2; moreover, grids of pink squares represent Boolean matrices (e.g., ■ $\in \{T, F\}^{2\times 2}$).

Note that this way of tracking the evolvability of an EA execution does not interfere with the EA execution. In particular, in the case of ME, A' and A might be different, if based on different descriptors, domains, or numbers of bins. Nevertheless, we use for A', while executing both ME and ES, with the same parameters used for A in ME.

4 Experiments and discussion

We performed several evolutionary runs with the aims of (a) understanding if and how different EAs and representations produce different fitness and evolvability maps and (b) discovering relationships between fitness and evolvability maps.

For each of the eight combinations of representation (DB, IB, DB^2, IB^2) and EA (ES, ME), we performed 10 evolutionary runs (with different random seeds) and the following parameters.

For both ES and ME, we set $\sigma^2 = 0.35$ and $n_{\text{eval}} = 25\,000$. For ES, we set $n_{\text{pop}} = 20$. For ME, we set $n_{\text{parent}} = 20$, $n_{\text{bin}} = 10$, and, as descriptors, w, h with DB^2 and IB^2 and $\rho_{\text{back}}, \rho_{\text{front}}$ with DB and IB.

For DB and IB, we used the biped body ■. For IB and IB^2, we set $n_{\text{GMM}} = 5$. For DB^2 and IB^2, we set $n_{\text{size}} = 10$. As a result, the dimension of the search space $\mathbb{R}^p$ was 20, 50, 300, and 75, respectively for DB, IB, DB^2, and IB^2.

Finally, for computing the fitness and evolvability maps out of A' for each EA execution, we used the same descriptors and n_{bin} value used in ES, i.e., $n_{\text{bin}} = 10$ and w, h with DB^2 and IB^2 and $\rho_{\text{back}}, \rho_{\text{front}}$ with DB and IB.

The code for the experiments is publicly available at `https://github.com/ndr09/VSRevo`.

Overview: VSRs fitness and search efficiency

As initial point, we discuss the outcome of the evolutionary runs in terms of the effectiveness of the evolved VSRs in the task of locomotion. Figure 4 shows the trend of the fitness: at each iteration of the EA for each of the 8 combinations, the figure shows the median ± the standard deviation, computed across the 10 executions, of the fitness v_x of the best individual at that iteration.

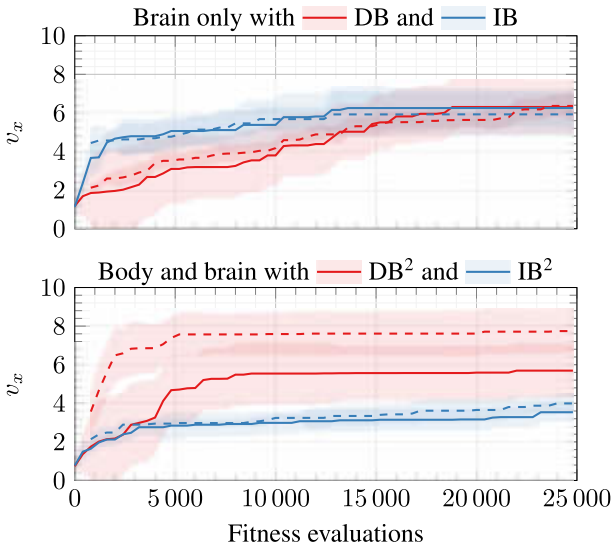


Figure 4: Fitness v_x (median ± standard deviation across the 10 runs) of the best individual in the during the evolution for the two EAs (—— ES and - - - ME) and the 4 representations, grouped in brain only (above) and body and brain (below).

First of all, we observe that all the approaches are able to produce effective solutions, achieving final best fitness values between 3, with the IB^2, and 8 with the DB^2+ME Values are in $\mathrm{m\,s^{-1}}$; as a reference, the side of each voxel is

	ES+DB^2	ME+DB^2	ES+IB^2	ME+IB^2
ES+DB^2	-	1.00	**0.0009**	**0.0030**
ME+DB^2	**0.016**	-	**0.0002**	**0.0002**
ES+IB^2	1.00	1.00	-	1.00
ME+IB^2	1.00	1.00	0.1818	-

Table 1: Table of p-values obtained with the Mann-Whitney U test with Bonferroni correction on the final fitness reached in the brain and body evolution. The alternative hypothesis was set to check if the distribution in the row was greater than the one on the column. The entries indicated in bold rejects the null hypothesis with the corrected significance level of $\alpha = 0.003$.

3 m long. In the other 5 cases, the final best fitness is ≈ 6.

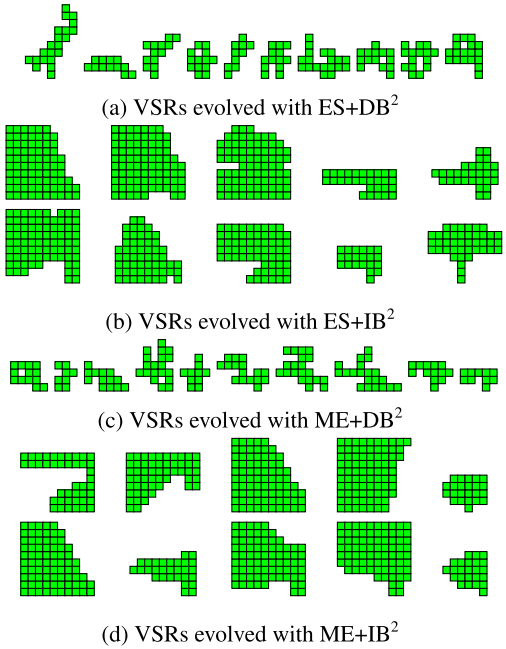
Concerning the efficiency of the evolutionary optimization, we observe that the chosen value for n_{eval} appears to be large enough to let every combination converge to good solutions. Nevertheless, there are some differences among the combinations. The convergence happens earlier in the body and brain case, taking only ≈ 2000 and ≈ 7000 fitness evaluations for IB^2 and DB^2, respectively, vs. the larger values of the brain only case, 13 000 and 19 000 for IB and DB, respectively. In the brain only case, the indirect representation seems to enable a faster convergence than the direct one, despite the larger search space ($p = 50$ vs. 20).

While in the brain only case there are no apparent differences in the final best fitness, i.e., in the effectiveness of the evolutionary search, among the four cases, in the body and brain case the direct representation appears to be more effective than the indirect one, the former obtaining v_x values that are 2 to 2.5 times larger than the latter. The EA does not appear to play a role when coupled with IB^2, as confirmed by the Mann-Whitney U test (Table 1).

For explaining this performance gap, we observed the evolved VSRs for the body and brain case—see Figure 6 for an example of a robot behavior. We show all the $10 \cdot 4$ bodies evolved in the body and brain case in Figure 5. The most apparent difference between VSRs evolved with DB^2 and IB^2 is in the size, i.e., the number of voxels consituting the body. IB^2 are in general much larger: this is the combined effect of the different threshold being applied while building $\boldsymbol{B}'$ and the fact that IB^2, based on GMM, favors regular shape, and hence larger connected components, by design. Despite the fact that big and regular shapes have potentially more power to control the movement, Talamini et al. (2021) show that irregularity in the shape is an enabling factor for faster and more robust robots.

Fitness and evolvability maps

We here discuss the evolutionary runs in terms of the interplay between evolvability and fitness. Figures 7 and 8 show the fitness (top row, greenish colors) and evolvability (bottom row, red-violet colors) maps, one pair of maps for each combination of EA and representation. Each single map is obtained from 10 evolutionary runs: since individual runs can cover different portions of the descriptor grid, each cell in the overall map results from up to 10 corresponding cells of the individual maps. In particular, we use the median value of the cell value in the overall map.



(a) VSRs evolved with ES+DB^2

(b) VSRs evolved with ES+IB^2

(c) VSRs evolved with ME+DB^2

(d) VSRs evolved with ME+IB^2

Figure 5: Body of the best robots at the end of the evolution for the DB^2 and IB^2 representations and the two EAs.

The most apparent finding resulting from Figures 7 and 8 is that the maps for the evolvability are similar for the same EA regardless of the representation. On the other hand, the fitness maps differ for representation and are similar for the EA—the latter being more evident for the brain only case.

As regards the brain only case shown in Figure 7, we recall that the map is based on the behavioral descriptors ρ_{back} and ρ_{front} descriptors, i.e., they indicate how the robots move. We can see that faster robots have low ρ_{back} and ρ_{front} values: their energy is spent more in the 2 Hz to 5 Hz band than in the 0 Hz to 2 Hz band, i.e., they move they limbs at higher frequencies. Concerning the evolvability, the maps show that the evolvability is, generally, smaller than 0. This indicates that the initial heritable individuals are replaced in the map, as fitter individuals are generated. Moreover, we see a good match between the evolvability and fitness map: good fitness is related to low evolvability, which is expected, as it becomes harder, while the evolution progress, that high performing individuals produce better offspring.

In the body and brain optimization shown in Figure 8, maps are based on body descriptors w and h. The maps shows that almost all the cells are covered, meaning that all

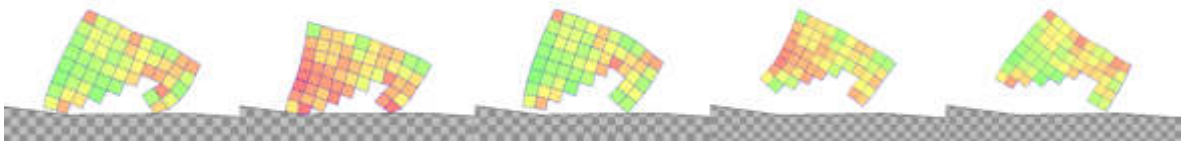


Figure 6: Frames of the simulation of one of the evolved VSRs (run 2 of ES+IB^2, see Figure 5b) doing locomotion. More videos of the behavior of evolved VSRs are available at `https://youtu.be/nDUjq1VebsE`.

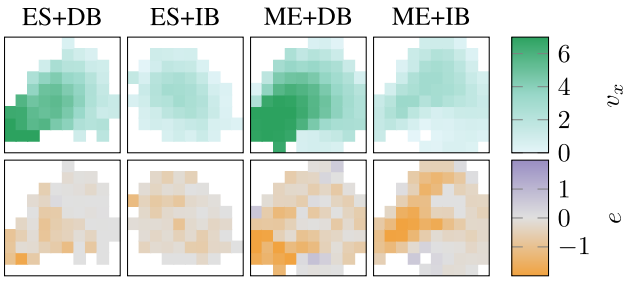


Figure 7: Fitness (top row) and evolvability (bottom row) maps for the DB and IB representations and the two EAs: ρ_{back} and ρ_{front} are used as descriptors.

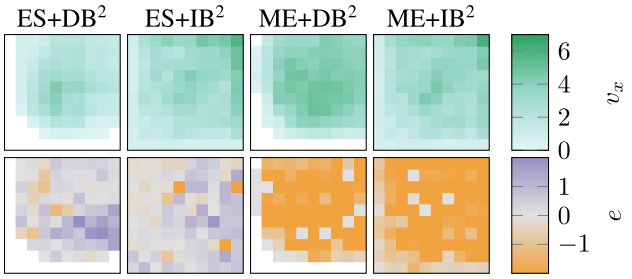


Figure 8: Fitness (top row) and evolvability (bottom row) maps for the DB^2 and IB^2 representations and the two EAs: w and h are used as descriptors.

the different sizes are found with the difference that DB^2 tends to avoid $1 \times n$ (and $n \times 1$) bodies. Differently from the previous case, here there are no regions in the grid with high fitness values. This is probably because it is harder to find good individual, while optimizing both the body and brain. For what concerns the evolvability map, we note that there are some grid portions with positive evolvability. This is likely because in those areas the individuals are selected just few times for reproduction, they give origin to high performing offspring, but are never replaced by better individuals.

Summarizing, the entanglement between the fitness and evolvability is related to the dynamics of the evolution. Negative values indicate that individuals in a cell generated less fit offspring, likely because they are themselves fairly fit. Positive values show that the individuals produced better children and were not replaced. Unfortunately, it is not possible to say if a cell in the map hosted an individual with good evolvability before being replaced by a fitter individual later in the evolution. This can be considered a limitation of this approach; the filling of the map is driven by the fitness: hence the historic information about the evolvability of individuals is lost when they are replaced. We plan to address this limitation as future work.

5 Conclusions and future works

We proposed the use of MAP-Elites as support structure for the calculation and visualization of evolvability. We tested this approach on a locomotion task carried out by Voxel-based Soft Robots, where we considered two different EAs (MAP-Elites and Evolution Strategies), with two genotypic representations (direct and indirect), applied to two optimization settings (robot brain only, and body and brain).

Our results show that, given a predefined phenotypic space (in the form of a descriptor grid, as in MAP-Elites), the distribution of evolvability over it (measured as the average difference in fitness between the offspring and their parents, considering only the offspring inserted in the grid during the evolutionary process), is mostly determined by the adopted EA. On the other hand, the distribution of fitness over the same phenotypic space depends, in our experiments, on the adopted representation. Overall, these findings suggest that evolvability is not an intrinsic property of the fitness landscape, or the genotypic representation, but rather it stems from multiple factors, including the EA being used and its capability to keep diversity and generate better solutions over time.

In future works, we will extend the analysis of evolvability to the case of sensor evolution and learning, which have been recently addressed in the context of VSRs in (Ferigo et al., 2021a, 2022) and (Ferigo et al., 2021b) respectively. Moreover, we will investigate how the proposed measure of evolvability correlate with some specific features of the fitness landscape, e.g., modality, as well as the behavior descriptors, which in turn may depend on the specific task (in the case of robots). Another possibility would be to embed a measure of evolvability into the evolutionary loop, for instance in the selection step. However, our preliminary results in this direction (not reported here for brevity) did not yield promising results. Finally, it would be interesting to test our proposed approach to calculate evolvability on other EAs, including e.g., novelty search (Lehman and Stanley, 2011a).

References

Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies–a comprehensive introduction. *Natural computing*, 1(1):3–52.

Carlo, M. D., Ferrante, E., Zeeuwe, D., Ellers, J., Meynen, G., and Eiben, A. E. (2021). Heritability in morphological robot evolution. *CoRR*, abs/2110.11187.

Coppinger, R. P. and Smith, C. K. (1983). The domestication of evolution. *Environmental Conservation*, 10(4):283–292.

Corucci, F., Cheney, N., Giorgio-Serchi, F., Bongard, J., and Laschi, C. (2018). Evolving soft locomotion in aquatic and terrestrial environments: effects of material properties and environmental transitions. *Soft robotics*, 5(4):475–495.

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521(7553):503–507.

Darwin, C. (1875). *The Variation of Animals and Plants under Domestication*. John Murray, London, UK.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Ferigo, A., Iacca, G., and Medvet, E. (2021a). Beyond body shape and brain: Evolving the sensory apparatus of voxel-based soft robots. In *EvoApplications 2021: Applications of Evolutionary Computation*, volume 12694, pages 210–226, Cham. Springer.

Ferigo, A., Iacca, G., Medvet, E., and Pigozzi, F. (2021b). Evolving hebbian learning rules in voxel-based soft robots. TechRxiv.

Ferigo, A., Medvet, E., and Iacca, G. (2022). Optimizing the sensory apparatus of voxel-based soft robots through evolution and babbling. *SN Computer Science*, 3(2):1–17.

Gajewski, A., Clune, J., Stanley, K. O., and Lehman, J. (2019). Evolvability es: scalable and direct optimization of evolvability. In *Genetic and Evolutionary Computation Conference*, pages 107–115, New York, NY, USA. ACM.

Hiller, J. and Lipson, H. (2011). Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466.

Katona, A., Franks, D. W., and Walker, J. A. (2021). Quality evolvability es: Evolving individuals with a distribution of well performing and diverse offspring. arXiv:2103.10790.

Koler-Matznick, J. (2002). The origin of the dog revisited. *Anthrozoös*, 15(2):98–118.

Kriegman, S., Cheney, N., and Bongard, J. (2018). How morphological development can guide evolution. *Scientific reports*, 8(1):1–10.

Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.

Lehman, J. and Stanley, K. O. (2011b). Improving evolvability through novelty search and self-adaptation. In *IEEE Congress of Evolutionary Computation*, pages 2693–2700, New York, NY, USA. IEEE.

Lehman, J. and Stanley, K. O. (2013). Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PloS one*, 8(4):e62186.

Lim, B., Grillotti, L., Bernasconi, L., and Cully, A. (2021). Dynamics-aware quality-diversity for efficient learning of skill repertoires. *CoRR*, abs/2109.08522.

Lindsay, B. G. (1995). Mixture models: theory, geometry and applications. In *NSF-CBMS regional conference series in probability and statistics*, pages i–163. JSTOR.

Liu, D., Virgolin, M., Alderliesten, T., and Bosman, P. A. N. (2022). Evolvability degeneration in multi-objective genetic programming for symbolic regression.

Medvet, E., Bartoli, A., De Lorenzo, A., and Seriani, S. (2020a). 2d-vsr-sim: A simulation tool for the optimization of 2-d voxel-based soft robots. *SoftwareX*, 12:100573.

Medvet, E., Bartoli, A., De Lorenzo, A., and Seriani, S. (2020b). Design, Validation, and Case Studies of 2D-VSR-Sim, an Optimization-friendly Simulator of 2-D Voxel-based Soft Robots. *arXiv*, pages arXiv–2001.

Medvet, E., Daolio, F., and Tagliapietra, D. (2017). Evolvability in grammatical evolution. In *Genetic and Evolutionary Computation Conference*, pages 977–984, New York, NY, USA. ACM.

Mengistu, H., Lehman, J., and Clune, J. (2016). Evolvability search: directly selecting for evolvability in order to study and produce it. In *Genetic and Evolutionary Computation Conference 2016*, pages 141–148, New York, NY, USA. ACM.

Methenitis, G., Hennes, D., Izzo, D., and Visser, A. (2015). Novelty search for soft robotic space exploration. In *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*, pages 193–200.

Nordmoen, J., Veenstra, F., Ellefsen, K. O., and Glette, K. (2021). Map-elites enables powerful stepping stones and diversity for modular robotics. *Frontiers in Robotics and AI*, 8:1–17.

Payne, J. L. and Wagner, A. (2019). The causes of evolvability and their evolution. *Nature Reviews Genetics*, 20(1):24–38.

Pigliucci, M. (2008). Is evolvability evolvable? *Nature Reviews Genetics*, 9(1):75–82.

Pugh, J. K., Soros, L. B., and Stanley, K. O. (2016). Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40.

Squillero, G. and Tonda, A. (2016). Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences*, 329:782–799.

Talamini, J., Medvet, E., and Nichele, S. (2021). Criticality-driven evolution of adaptable morphologies of voxel-based soft-robots. *Frontiers in Robotics and AI*, 8:172.

Tarapore, D. and Mouret, J.-B. (2015). Evolvability signatures of generative encodings: Beyond standard performance benchmarks. *Information Sciences*, 313:43–61.

Veenstra, F., de Prado Salas, P. G., Stoy, K., Bongard, J., and Risi, S. (2020). Death and progress: How evolvability is influenced by intrinsic mortality. *Artificial life*, 26(1):90–111.

Wayne, R., Van Valkenburgh, B., and O'Brien, S. J. (1991). Molecular distance and divergence time in carnivores and primates. *Molecular Biology and Evolution*, 8(3):297–319.

Lineage Selection in Mixed Populations for Genetic Improvement

Penny Faulkner Rainford and Barry Porter
School of Computing and Communications, Lancaster University
faulknerrainford@gmail.com; b.f.porter@lancaster.ac.uk

Abstract

Emergent Software Systems take a large pool of potential building blocks, for a given system such as a web server, and learn at runtime how best to compose selected blocks from that pool in order to maximise some utility function in each set of deployment conditions that is encountered. To support this approach, at least some building blocks in the available pool must have *implementation variants* – alternatives which have the same functionality but achieve it using a different approach (such as different sorting algorithms or different cache eviction policies). We can automatically derive new building block variants for our pool of potential behaviour by using genetic improvement (GI), which has long proven effective for optimisation and repair of source code. When a novel deployment environment is detected, however, it is unclear which existing building block variant(s) should be used as starting points for new a GI process to tailor a new block for that environment; in this situation it would be necessary to try one GI process from every possible existing building block variant as a starting point, a process which could be extremely expensive. In this paper we present a mixed-population approach to examine whether GI can simultaneously offer both *lineage selection* and *optimisation* to find the ideal source code for a new building block variant tailored to a given environment. Using a lowest-common-ancestor approach to producing evolvable individuals, our results demonstrate strong evidence that combined lineage selection and optimisation is viable in multiple scenarios, offering far reduced compute time to locate a good individual for a novel environment.

Introduction

Genetic algorithms have long been used across many different computing applications, from finding the ideal parameters of systems with large parameter search-spaces (Bäck and Schwefel, 1993), to a useful meta-heuristic approach to fixing bugs in code (Haraldsson et al., 2017; Forrest et al., 2009). In this paper we focus on genetic code improvement (GI), which aims to derive new versions of existing computation logic that are optimised towards a utility function such as calls-per-second. GI has been approached in a wide variety of ways, from modifying bytecode to working on syntax trees or with grammar models of the target programming language (Petke et al., 2018). We use an approach based on the compiler-derived syntax tree of a piece of source code, augmented with grammar rules to guide valid mutations. We particularly target our GI approach towards emergent software systems, which are composed of many small pieces of interchangeable building blocks such as a sorting algorithm or a hash table (Porter et al., 2016). At runtime, emergent software systems monitor their environment and learn which variants of each building block are best suited to each set of deployment conditions encountered. Because each building block in these systems is relatively small (100-200 lines of code), our particular GI approach mixes new code synthesis with more traditional mutation types so that sufficient new genetic material is available (Rainford and Porter, 2021).

When an emergent software system detects a novel environment, it will learn the best composition of building blocks from among those which currently exist; it is also useful, however, to seek to generate new variants of building blocks that are tailored to that novel environment using GI. Using a traditional GI approach, when we have several existing variants of the same building block (such as a cache eviction policy), it is unclear which variant we might use as the starting point for a new GI run to derive a tailored variant for the novel environment. A naive approach may use a set of distinct GI runs, each starting from one existing variant and running for e.g. 200 generations, to find the best tailored variant considering each of the existing variants. This approach requires significant computation power, however, which scales poorly as more variants are added to the pool.

In this paper we examine whether a GI process, operating on source code, can perform both *lineage selection* and *optimisation* at the same time in a single run for a novel environment: i.e., can we add each existing variant to a common starting population, and run a single GI process from that population for 200 generations, to see an equivalent fitness individual emerging for our novel environment – when compared to a set of per-variant GI processes each running for an equal number of generations. Demonstrating that this is possible offers a far reduced volume of GI computation for each novel environment that is encountered.

Our use of lineage selection takes inspiration from the biological analogue: when different genes provide for the

same functionality, selection can be seen at both an individual and a lineage level. Selection at a lineage level means that, because of inheritance, a favoured individual's offspring will share the same advantages and so will their children, so the whole lineage (family tree) will be selected for (Akçay and Van Cleve, 2016). In GI this would mean that if we create a population with members from different algorithms (lineages), and evolve over the whole population, the lineage with the better algorithm for the given environment will be favoured by selection and dominate the population.

We specifically propose the use of *lowest common ancestors* (LCAs) of previous runs to support lineage selection. This takes an individual from a previous run that was the ancestor of all the individuals remaining in the final population. We take these LCAs from two different runs which were specialised for two different deployment environments A and B. We then test to see if a starting population containing copies of both LCA individuals, run against each deployment environment A and B, can correctly select the appropriate lineage for each problem and then optimise that individual to the same level as the best fit individual of the LCA's originating GI run. This approach has the potential to automatically select and optimize the ideal base method for a genetic improver's target function from a set of possibilities. We also examine the form of LCA that offers highest utility as a starting point for future GI runs: the 'full' LCA, as it appeared in its original GI run, or a 'reduced' LCA which has had all extraneous source code removed.

We show that the reduced LCA has increased evolvabilty, versus the full LCA, as a start point for genetic improvement, due to the increased chance of mutations editing code which effects the fitness; this gives more variation in end-results of a new GI run than the full LCAs, which remain mostly homogeneous. We show that reduced LCAs are not more evolvable than final individuals due to early specialisation. Using these reduced LCAs we then show that a mixed population of these LCAs can successfully reach the same level of training as the original best-fit individual from the full GI run of the corresponding LCA – and that this mixed population does indeed perform both lineage selection and optimisation within 200 generations. We provide a replication package, with detailed instructions, with which all of our results can be repeated (Rainford and Porter, 2022).

Related Work

Non-homogeneous starting populations are not new in genetic algorithms. It is a common for genetic algorithms used for parameter optimization to be seeded with individuals from across the possible parameter settings to better cover the fitness landscape (Bäck and Schwefel, 1993). This allows the system to use selection to focus its evolution on areas of the landscape with high potential fitness. GI for code optimization has, by nature of the wish to optimize a particular function, started from a single start point (Forrest et al., 2009; Haraldsson et al., 2017). This is either the non-functional code to be repaired or a functional but non-optimal code to be improved. In the case of repair it is sensible to start with the code to be repaired rather than anything else. However in the case of optimization, multi-start points become an advantage in possibly providing a broader start to searching the fitness landscape allowing the system to focus in on the area of the landscape with higher potential.

In biology this selection of a particular area over another is called *lineage selection*. It has been explored in a digital context previously in relation to different artificial organisms and evolutionary systems (Dolson et al., 2020; Virgo et al., 2017). It is considered as part of analysis for evolvability (Kirschner and Gerhart, 1998). This is a natural connection. The existence of multiple lineages in an evolutionary system should produce higher robustness and variety in the system than a single lineage with a single start point. In most GI systems however, the focus is less on variety and robustness, and instead on finding the best optimisation to the immediate environment. In this work we test if multiple lineages detract from finding the best optimisation, or if they provide the same optimisation with greater evolutionary potential.

Algorithm

Our overall GI framework is illustrated in Fig. 1. An emergent software system is assembled from a large collection of small building blocks, such as stream processors, memory cache implementations, hash tables, and so on, and is deployed into a real environment. Once that system has learned the best composition of blocks for a given environment, it will select one of those building blocks and capture a short trace of the method calls that are issued to that block within the present environment. This trace is then sent to a GI system, along with the source code of the building block from which it was captured, so that the GI system can attempt to generate an improved variation of that building block which has higher performance for the given input data sample. If the GI system is successful, the improved building block is pushed back to the emergent software system which uses real-time learning to determine if the proposed improvement really does yield higher performance for the intended environment conditions in deployment.

For the purposes of this study we examine only the GI element of this overall concept. We focus on one particular building block throughout our experiments (a hash table), and we assume that short traces of function calls to this block have already been captured by the emergent system. We also assume that the GI process is able to identify the *hash function* of the hash table implementation as the specific area in which to focus when generating improved variations; in practice this focus area could be determined using function call frequency or CPU intensity analysis.

The overall core of our system is then based on a typical genetic improvement process, Algorithm 1, using mutation,

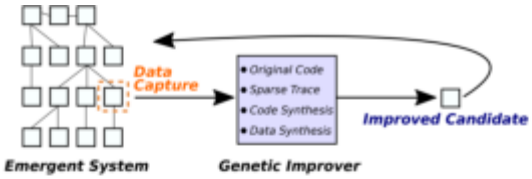


Figure 1: Our approach, which captures input data from a running emergent software system, and requests improved variants of particular building blocks from a GI system by replaying that captured data. The GI process uses mixed synthesis of code with traditional GI operators, to counter the small code sizes of each building block.

Algorithm 1 Genetic Improvement Algorithm

```
for i = 0 to generations do
  if i == 0 then
    create initial population of clones
  end if
  mutate a % of the population
  check fitness of all population members
  if i%5 == 0 then
    check performance on unseen data of all population
    members
  end if
  select new population (roulette wheel)
  crossover a % of the population
end for
```

fitness, selection, and crossover. We include crossover in this work to make use of existing code in expanding the code base of members of our population. Because our volume of starting genetic material is very small, we also include *new code synthesis* in our set of available mutations, combined with more typical mutation types.

Our selection process, where we choose which individuals to select from one generation for inclusion in the next generation (with associated crossover/mutation), uses a rank-weighted roulette wheel approach. Each population is ordered by fitness and then ranked. This rank is then used such that the fittest individuals have the highest probability of selection for the next generation. Selection is done with replacement, so that some individuals will appear multiple times in the following generation, and some will be completely absent, with a (small) possibility of even the worst individual being selected for the next generation.

A hash function (e.g. Listing 1) takes a set of key/value pairs, where a key is a string of characters, and uses a mathematical transformation of the key string to yield an integer result (such as adding together the binary representations of each string characeter). The integer result is used as an index value to place the key into a particular hash bucket, where the list of hash buckets is usually represented as a fixed-

```
1 int hash(char key[]) {
2 int result = 1
3 for (int i = 0; i < key.arrayLength; i++)
4 {
5 result = result * key[i]
6 }
7 return result % HT_LEN
8 }
```

Listing 1: Original Hash Function

length array. Each array cell has a linked list of all keys that have been placed into that array cell / bucket. When retrieving a key from a hash table, the hash function is applied to the key to derive the correct bucket, and the linked list of keys in that bucket is scanned iteratively to locate the matching key. The general objective of a hash function is to divide the set of keys it is given evenly between each hash bucket, so that the linked list within each hash bucket is the same length, thereby minimising average lookup time for a given key (compared to a situation, for example, in which every key mapped to hash bucket index 0, potentially necessitating a scan over every single key in the hash table).

Throughout this study we use two different environments for our hash function: one set of keys derived from English words, and one set of keys derived from Polish words. These two key sets are sufficiently different that they suggest very different hash functions to gain an even distribution of keys across buckets. In the final part of our study we use a third, novel environment, with keys based on French words.

Lowest Common Ancestor

When considering the results of an existing GI run, and the choice of an individual from that run to inject into a new GI run training to a novel environment, we hypothesise that using the best-fit individual from the final generation of an existing run is likely to be a poor choice. This individual is likely to be highly specialised to its own environment (e.g. the set of English keys) and may take longer to evolve to a novel environment. Instead we look further up the phylogenetic tree, shown in Figure 2, for an earlier less-specialised but still somewhat optimised individual: the lowest common ancestor (LCA) of a genetic improver's final population. The lower trees in Figure 2, for example, show the LCAs up to the final population for GI runs on English and Polish environments. The LCA is the most recent ancestor which is common to every member of the final population. The LCA may not have been the fittest individual of its generation, and indeed may not be a particularly fit individual at all, but evidently was an individual with high evolutionary potential shown by its ability to produce the offspring that lead to the final (and fittest) population. We assume that an LCA has higher evolvability to novel environments, when placed into a new GI process, than a final best-fit individual.

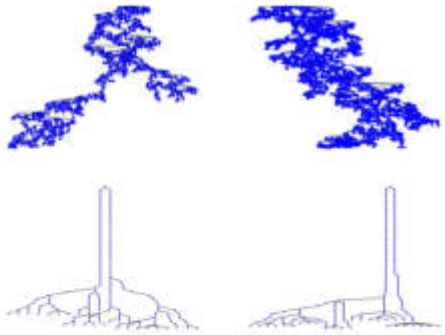


Figure 2: Phylogenetic trees for two runs of the genetic improver trained on different data: English (left) and Polish (right). The full tree (top) is busy so we also show the tree below the lower common ancestor (bottom).

```
int hash(char key[]) {
int result = 1
for (int i = 0; i < key.arrayLength; i++)
{
result = result * key[i]
result = result - key.arrayLength
result = result * key[i]
}
return result % HT_LEN
}
```

Listing 2: Reduced English LCA

```
int hash(char key[]) {
int result = 1
dec a = 0.8653064475373070635
for (int i = 0; i < key.arrayLength; i++)
{
result = result * key[i]
a = 0.8653064475373070635
result = key.arrayLength + key.arrayLength
}
return result % HT_LEN
}
```

Listing 3: Reduced Polish LCA

While this assumption seems reasonable, there are two remaining sources of uncertainty in this approach. First, if we start a new GI process from a population of LCA copies, do we observe within that process a return to equivalent fitness compared to the best-fit final-generation individual from the LCA's original GI run. In other words, having removed all the additional genetic material that was available from the other members of the LCA's original population, is there sufficient material left to yield an equivalent fitness result.

Second, the precise form of LCA to use is uncertain. Here we have two options: we can use the 'full' LCA source code, exactly as it appeared in its original GI run. This has a significant amount of 'redundant' genetic material which does not contribute to the functional logic of the LCA. Alternatively, we can use a 'reduced' LCA which has all of its redundant genetic material removed. The trade off we expect here is that the full LCA has more genetic material to offer and potentially more diversity, but that mutations are less likely to affect the fitness of an individual (since each mutation has a higher chance of affecting redundant code); while the reduced LCA has less diversity of genetic material to offer, but mutations are more likely to affect the fitness of individuals since all source code contributes to functionality.

Experiments

We begin with two control runs of our GI system trained on each key set: the set of English keys, and the set of Polish keys. Both control runs start with a population made entirely of a single piece of source code (a hand-crafted hash function that was optimised for general speed of execution rather than key-distribution effectiveness, shown in Listing 1).

Using these two control runs, we extract their LCAs, and derive both the full LCA and reduced LCA (English reduced LCA, Listing 2, and Polish reduced LCA, Listing 3). We then perform new GI runs that start from populations comprised entirely of copies of the full or reduced LCA, yielding 4 GI runs in total – two for English LCAs training on the English key set, and two for Polish LCAs on the Polish key set.

We perform 30 GI runs for each case, and compare with the original control runs using the original hand-crafted start code. These runs are analysed in terms of relative improvement when compared with the control runs. The relative improvement is important to understanding the ability of the system to still train and specialise from the LCAs.

Results

In general, when using the LCAs, we do not expect much training to occur as the LCAs are relatively well optimised. In Figure 3 we see the fitness relative to the original source code for both English language data (left column) and Polish language data (right column). All of our GI graphs show the average fitness of the best individual of each generation, and the standard deviation, for 30 repeated runs.

For the English environment we can see there is no training in the full LCA run, but the reduced LCA run does become varied including up to 5% improvement after 100 generations. The reduced LCA might be more likely to make changes to the critical path than the full LCA. In the context of the original code, these results show that the LCA selected was from a successful run, as both LCA experiments start with better fitness than the average of the control run, and that the reduced LCA run actually attains slightly better fitness than the control run. This offers confirmation that using reduced LCAs is the better choice, and that using these LCAs can yield equivalent fitness compared to the original

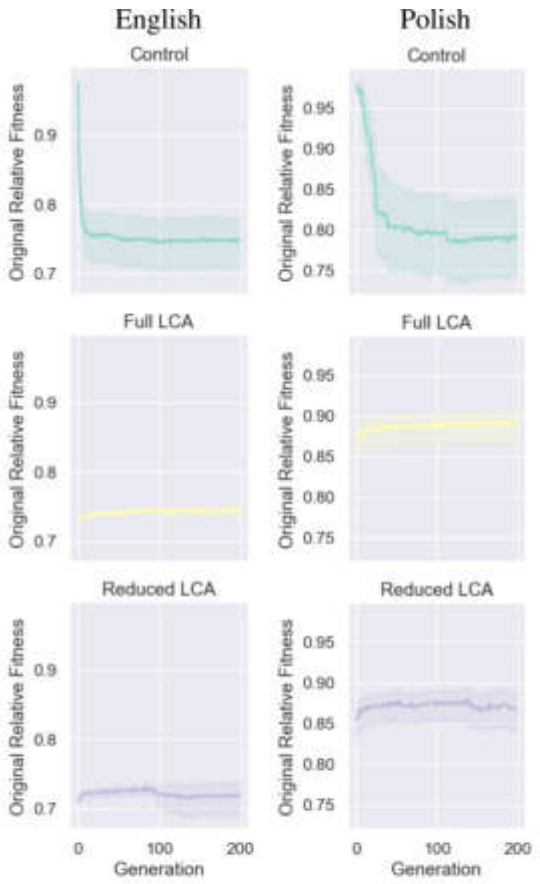


Figure 3: Relative fitness of experiments with English (left column) and Polish (right) data, showing: Control (Top), Full LCA (Middle) and Reduced LCA (Bottom).

run despite having far less genetic material overall.

Examining the Polish data runs, shown in the right column of Figure 3, we see similar trends. There is very little variation or training in the full LCA experiment, but clear evidence of training in the reduced LCA run towards generation 150. The LCA-derived training is generally much weaker in the Polish environment than the English; this is possibly explained when we consider that the LCA run started with relatively poor fitness, suggesting that the LCA in the control run was more reliant on other genetic material in its population compared to the English environment.

Overall the data here clearly indicates that reduced LCAs have higher utility, and the use of LCAs in general is promising as start points for new GI runs for novel environments.

Meta-populations

Having established the utility of LCAs as evolvable starting points, in this section we examine the ability of a GI process to simultaneously perform *lineage selection* and *optimisation* within 200 generations. This demonstrates the major saving in computation cost when deriving a new variant for a novel environment, versus running a set of individual GI processes on each starting point.

In these experiments, instead of cloning a single piece of code to form our initial population, we divide our population equally into clones of each of our LCA candidates (in this case 15 clones of our English LCA, and 15 of our Polish LCA). These are assigned to lineages based on which piece of code their first ancestor was cloned from. We do not otherwise separate the populations or make them known to the GI process. The biological analogy here would be if one took equal numbers of individuals from meta-populations and put them together to form a single new population.

To explore this we use the end-result of our two existing single-lineage GI runs – on English and Polish key sets, which start from their respective reduced LCAs – as our ground truth. When our mixed-lineage LCA-based starting point run is provided with our English key training set, it should ideally be able to both select the English lineage to become dominant in the population, and simultaneously optimise that population from its LCA starting point to a similar level to that seen in the final generation of our single-lineage English key training set. Demonstrating this would potentially allow us to train over many LCA-based start points with very little increase in resources used, versus training independently from a set of single start points.

Validation Experiment

We execute our mixed-population LCA experiment, in which half of the starting population is from our English LCA and half is from our Polish LCA, separately against the English training data and the Polish training data. Each run has 200 generations, with a population size of 30, and we repeat each run 30 times to record an average.

We compare the results against the single-lineage LCA runs for both English and Polish training data, which again used 200 generations and a population size of 30. We examine the results of the first two training set-ups first to ensure they match the better of the two single-lineage runs trained of the same data, and second to study the distribution of lineages in the population over time to see if one lineage dominates the other and if so how long it takes for this to occur.

Validation Results

Figure 4 shows the results when training against our English key set. The left column of graphs shows fitness against the training data, while the right column of graphs shows performance against unseen data (which is drawn from the same distribution as the training data). Fitness indicates how well the algorithm is specialising, while performance shows how well it is generalising to the *class* of data on which it is being trained. The top row of graphs in Figure 4 are when we start with the single-lineage English LCA, the bottom two graphs show a start point of the single-lineage Polish LCA, and the middle two show our mixed-lineage experiment.

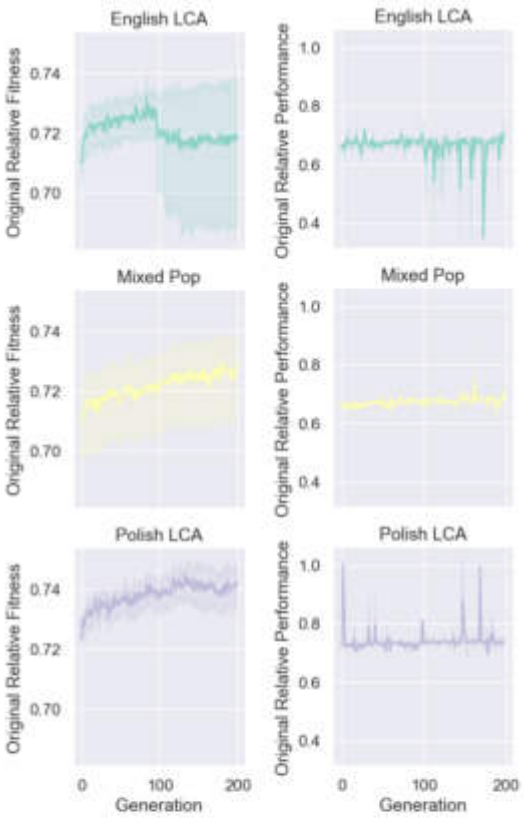


Figure 4: Fitness (left) and performance (right) for systems trained on English data: Control (top) using English-trained reduced-LCA, Mixed population (mid) using both reduced LCAs, and the Polish-trained reduced LCA (bottom).

Starting with fitness, the single-lineage English-trained LCA (top) shows the expected improvement as discussed earlier in the paper. The single-lineage Polish-trained LCA (bottom), when trained against English data, likewise shows an expected poor fitness training towards the alternative data set. When we examine our mixed-lineage experiment (middle), we observe a fitness score in the final generation which comes very close to the final fitness of the single-lineage English LCA comparison. This suggests that the mixed-lineage GI run experiences both lineage selection and training within the same number of generations. When we examine relative performance against unseen data of the best individuals, in the right column of graphs, we see that our mixed-lineage experiment performs as well as than the single-lineage English comparison with improvements of 30.2% and 32.6% respectively and no statistically significant difference in the final populations (comparison of best individuals with signed-rand Wilcoxon test, 5% significance); this may be as the benefit of the diversity in genetic material at the beginning of the run outweighs the smaller number of individuals of the correct lineage in the starting population.

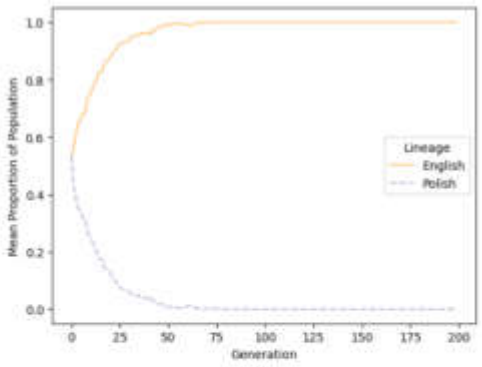


Figure 5: Proportion of population at each generation of each lineage for all runs of the Mixed Population against English training data.

We also examine lineage selection in isolation, with the results shown in Figure 5. This confirms our above data, showing that the English LCA lineage on average occupies more than 80% of the population after 14 generations, with the Polish LCA linage extinct after 77 generations. We can therefore say that the first 77 generations is spent doing a mixture of lineage selection and training (including potential exchange of genetic material between the lineage members), with the remaining generations exclusively doing training.

We next examine the same set of starting populations (English, Polish, and mixed-lineage) when trained on our Polish hash key data set. The Polish LCA was a weaker starting candidate as discussed in the previous section, however in Figure 6 we see that it still performs far better than the English LCA when trained on Polish data. On fitness we again see that our mixed-lineage experiment more closely tracks the single-lineage Polish comparison, though it diverges more than in our English target experiment. Examining performance, in the column of graphs on the right, the single-lineage Polish LCA has an average improvement in performance on unseen data of 13.4%, while the English LCA only has a mean improvement of 1.8%. Here our mixed-lineage experiment does closely match the control performance (Polish LCA) with an average performance of 12.2% (with our tests indicating no statistical difference in either performance or training fitness in the later generations between the Polish LCA and the mixed population).

Turning again to lineage selection, shown in Figure 7, here we see an even more pronounced effect; the Polish LCA lineage represents more than 80% the population after just 15 generations and drives the English LCA lineage into extinction after only 60 generations.

New Environment Experiment

Having confirmed that lineage selection is possible and correct when we know which lineage will perform better, we

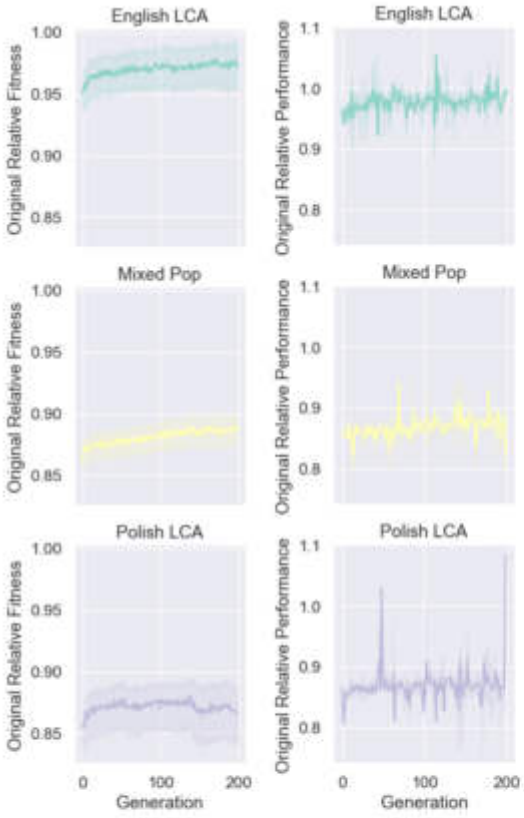


Figure 6: Fitness (left) and performance (right) for systems trained on Polish data: the English-trained reduced LCA (top), Mixed population (mid) using both reduced LCAs, and the Polish-trained reduced LCA (bottom).

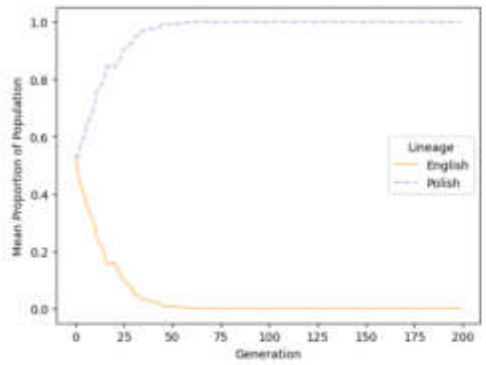


Figure 7: Proportion of population at each generation of each lineage for all runs of the Mixed Population against Polish training data.

now examine whether we see equivalent results in a novel environment. To do this we execute our mixed-population LCA experiment, in which half of the starting population is from our English LCA and half is from our Polish LCA, against a set of hash keys based on French words. Each run has 200 generations, with a population size of 30, and we repeat each run 30 times to record an average.

We also in this case test the use of the best final individual for each run that produced the reduced LCAs used in previous experiments; we do this to re-verify that the use of LCAs has value in yielding individuals with higher evolvability.

We compare the best individual, reduced LCA, and mixed-lineage runs on fitness and performance for unseen (French) data. To meet our requirements the mixed-lineage run should have equivalent final population performance and fitness to the best of the other runs; we will also examine whether, and how quickly, lineage selection occurred.

New Environment Results

Figure 8 shows the results of these experiments. Addressing trained fitness (on the left) first we can see that the English trained specialist (both finalist and LCA, top 2 graphs) have less than 20% improvement on the original code, while the mixed-lineage and Polish specialists all have greater than 25% improvement. The mean improvement in the final generations for the mixed-lineage (27.3%), Polish Finalist (27.3%) and Polish LCA (26.8%) are very similar and there is no statistical difference in their final generation.

Examining performance on unseen data, in the column of graphs on the right, shows closer results but still demonstrates that Polish and multi-lineage populations still perform statistically-significantly better than the English populations. The English Finalist and English LCA achieve performance improvements of 18.9% and 19.3% respectively, while the mixed-lineage, Polish Finalist and Polish LCA achieve 20.5%, 20.9%, and 20.5% improvements with no statistical difference again in their final generation.

The implication that lineage selection has successfully chosen and optimised to the Polish lineage is confirmed in Figure 9 which shows that the Polish lineage represents more than 80% of the population after just 12 generations and the English lineage has been driven to extinction after only 50 generations. Overall these results confirm that both lineage selection and optimisation is possible in a single GI run, and can save on significant computation time when presented with a novel environment for which a new source code variant is required.

Discussion

Our set of experiments demonstrate that the use of a lowest-common-ancestor has good utility as an evolvable starting point (and one which may not yet have been over-specialised). However, the lack of very significant improvements in training from the mixed-lineage runs suggests that

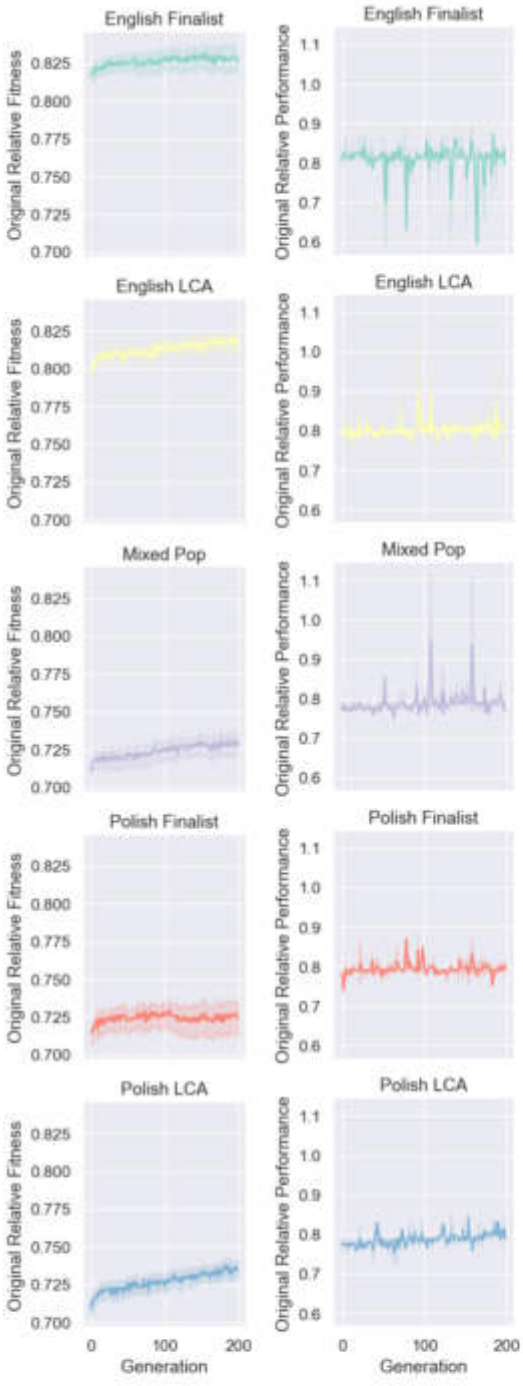


Figure 8: Fitness (left) and performance on unseen data (right) for systems trained on French data: the best English trained individual (top), the reduced English LCA (top mid) , Mixed population (mid) using both reduced LCAs, the best Polish trained individual (bottom mid) and the reduced Polish trained LCA (bottom) for comparison.

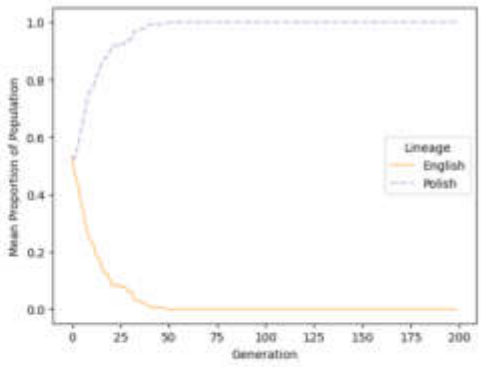


Figure 9: Proportion of population at each generation of each lineage for all runs of the Mixed Population against French training data.

our LCAs are still relatively well specialised – and that taking earlier ancestors may be beneficial.

Our lineage selection experiments show a very clear result, that a single GI run starting from a mixed population quickly selects the most ideal lineage for its training data, and that it maintains potential to perform further training towards new environments at the same time. While we have used two lineages in this study, an interesting question for future work is how many different lineages can be used in a starting population – without changing the population size – while maintaining clear lineage selection and training, and potentially gaining the benefit of higher genetic diversity.

Conclusions

We have shown here that lineage selection using lowest common ancestors can be used to select for the correct variant code for deployment conditions. The use of lowest common ancestors has successfully provided two specialist variants as starting points for training towards different deployment conditions. In future we will investigate the use of higher common ancestors for increased training potential and evolvability. We have shown that in the same population size and run length (and so the same computational resources) our genetic improvement algorithm can quickly select the correct lineage for the deployment conditions while also making use of the increased genetic material to improve the selected lineage resulting in the same overall improvements as the single source code alternatives. Further investigations into more divergent code in multiple populations and the use of Meta-populations with more separation in crossover and selection could yield further improvements.

Acknowledgements

This work was partly supported by the Leverhulme Trust Research Grant 'The Emergent Data Centre', RPG-2017-166.

References

Akçay, E. and Van Cleve, J. (2016). There is no fitness but fitness, and the lineage is its bearer. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 371(1687):20150085.

Bäck, T. and Schwefel, H.-P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evol. Comput.*, 1(1):1–23.

Dolson, E., Lalejini, A., Jorgensen, S., and Ofria, C. (2020). Interpreting the tape of life: Ancestry-based analyses provide insights and intuition about evolutionary dynamics. *Artif. Life*, 26(1):58–79.

Forrest, S., Nguyen, T., Weimer, W., and Le Goues, C. (2009). A genetic programming approach to automated software repair. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, GECCO '09, pages 947–954, New York, NY, USA. Association for Computing Machinery.

Haraldsson, S. O., Woodward, J. R., Brownlee, A. E. I., and Siggeirsdottir, K. (2017). Fixing bugs in your sleep: how genetic improvement became an overnight success. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, pages 1513–1520, New York, NY, USA. Association for Computing Machinery.

Kirschner, M. and Gerhart, J. (1998). Evolvability. *Proc. Natl. Acad. Sci. U. S. A.*, 95(15):8420–8427.

Petke, J., Haraldsson, S. O., Harman, M., Langdon, W. B., White, D. R., and Woodward, J. R. (2018). Genetic improvement of software: A comprehensive survey. *IEEE Trans. Evol. Comput.*, 22(3):415–432.

Porter, B., Grieves, M., Rodrigues Filho, R., and Leslie, D. (2016). {REX}: A development platform and online learning approach for runtime emergent software systems. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 333–348. usenix.org.

Rainford, F. P. and Porter, B. (2022). ALife 2022 replication package: http://www.projectdana.com/research/alife2022rainford.

Rainford, P. F. and Porter, B. (2021). Open challenges in genetic improvement for emergent software systems. *geneticimprovementofsoftware.com*.

Virgo, N., Agmon, E., and Fernando, C. (2017). Lineage selection leads to evolvability at large population sizes. In *Artificial Life Conference Proceedings 14*, pages 420–427. MIT Press.

The Evolution of Fractal Protein Modules in Multicellular Development

Harry Booth[1] and Peter J Bentley[1]

[1] Department of Computer Science, University College London
Gower Street, London WC1E 6BT, UK
harry.booth.19@ucl.ac.uk, p.bentley@cs.ucl.ac.uk

Abstract

Regional specification, or pattern formation, is the process by which developing cells in different regions are switched into different developmental pathways. We investigate this process through an ALife model of multicellular development using fractal proteins, where genes are expressed into proteins comprised of subsets of the Mandelbrot Set. The resulting network of gene and protein interactions can be designed by evolution to produce specific patterns, that in turn can be used to solve problems. Here fractal gene regulatory networks are incorporated into a multicellular model of development, and tested on the morphological problem of regional specification, using Map-Elites to explore the space of solutions. The results indicate the ability of this system to learn regularities in solutions and automatically create and use developmental modules, illustrating how an artificial system can replicate some of the fundamental processes of development.

Introduction

The phenotypes of organisms throughout the biological world are incredibly complex. As an example, a fully formed adult is made up of approximately 10^{13} cells, which form an all manner of specialised tissues and organs. However, the complete instruction set for the development of a human being is found in every cell, and contains a relatively small amount of information compared to the complexity of the phenotype previously described. There are approximately 3^9 base pairs within the human genome (Brown, 2002), and since there are 4 types of nucleotide bases (adenine, cytosine, guanine and thymine) the identity of each base pair can be specified using only 2 bits. This means the entire genome contains around 750 megabytes of data required for development - for reference, this is slightly larger than an audio CD. The crucial components in this journey from genotype to phenotype are *development processes*, which are coordinated through the expression of genes in specific temporal and spatial patterns (Levine and Davidson, 2005). There is growing evidence that many of these patterns are highly preserved and recombined during evolution as developmental modules (Lacquaniti et al., 2013).

The inclusion of development processes into artificial evolutionary systems has many advantages (Eggenberger et al., 1997). This work describes an artificial model of development, in which fractal proteins (Bentley, 2003b) are used for the first time to both regulate gene expression and determine other developmental parameters within a spatially extended multicellular environment. The system is tested on a fundamental morphological problem known as regional specification, with the aim of understanding how evolution is able to coordinate temporal and spatial gene expression in a multicellular assembly, paying specific attention to the evolution of developmental modules in the gene regulatory networks evolved by Map-Elites.

Background

The mapping from genotype to phenotype in natural evolution is a development growth process (Bowers, 2005). The study of such artificial mappings is sometimes referred to as computational embryology (CE) (Kumar and Bentley, 2003). Cell chemistry approaches to CE attempt to mimic how physical structures emerge in biology (Stanley and Miikkulainen, 2003). Here, interacting systems inspired by nature generate complex and indirect development programs which result in an emergent phenotype (Bentley et al., 1999). Often these systems consist of multicellular assemblies, where the regulation and subsequent expression of certain genes within each cell can trigger events such as mitosis, cell differentiation and apoptosis (Eggenberger et al., 1997). A fundamental process in the controlled formation and development of plant and animal phenotypes is regional specification, also known as pattern formation (Kumar and Bentley, 2003). For a multicellular assembly, the problem is one of self-organization - cells must differentiate in the correct spatial locations, with no explicit knowledge of their position within the final assembly (Friston et al., 2015). The essence of spatial patterning has been distilled into a metaphor which was first described by Wolpert (1969) and which subsequently became known as the French Flag (FF) Problem. Due to its self-contained nature and clear problem statement, the FF Problem has become a test bed for artificial development systems.

Some of the earliest work in testing an artificial develop-

ment system on the FF Problem was by Miller (2004), who used Cartesian Genetic Programming to evolve a program which controlled the actions of cells on a grid. This organism was able to successfully grow into a FF pattern, and in addition the development program was robust to damage applied to the developing phenotype. Chavoya and Duthen (2008) used a gene regulatory network (GRN) based on a bit-string representation (Banzhaf, 2003) within a cellular automata (CA) framework. Specific regulatory proteins determined the activation of structural genes, which in turn led to the adoption of an associated CA look-up table for cell reproduction. Knabe et al. (2008) integrated a GRN into the Cellular Potts computational model of cells and tissues. Protein concentrations determined individual cell parameters such as size, shape, adhesion, morphogen secretion and orientation. Evolved organisms were able to autonomously set up an asymmetric morphogen gradient and ultimately organize into a close match of the FF pattern. Joachimczak and Wróbel (2009) extended the the FF Problem into 3D by designing a development system consisting of spherical cells within a simulated physics environment.

This work introduces a system of artificial development which incorporates fractal proteins. Fractal proteins were first introduced by Bentley (2003b), and are an example of an artificial chemistry. Fundamental to the interaction of fractal proteins with any system is their shape (Bentley, 2009). This shape is genetically specified by a triple (x, y, w), which points to a finite region of the Mandelbrot set. As well as affording a compact representation, the geometry of the Mandelbrot set makes the 'fractal genetic space' highly evolvable, due to its continuity and self-similarity. For tasks such as gene regulation - which involves the interaction of fractal proteins and the genome - the shape of fractal proteins have been exploited very successfully to produce desirable concentration dynamics (Bentley, 2004b). In addition, gene regulation with fractal proteins has displayed desirable properties such as graceful degradation (Bentley, 2004a) and the emergence of modules - sub-routines which are reused and used with minor modifications to build more complex solutions (Bentley, 2003a). Modularity is a key feature of biological systems, and is seen as a desirable feature of evolutionary design. Consequently, there have been several studies examining modularity in simulated evolution. Garibay et al. (2003) introduced the modular genetic algorithm (MGA), which was explicitly designed to exploit regularity within the problem space using modularity. The MGA was found to both outperform a standard genetic algorithm and scale better for increasing problem complexity. Pollack and Lowell (2016) investigated the concept of hierarchical modularity. They found that an evolved gene regulatory network (GRN) displayed greater modularity than a neural network, due to the use of developmental encodings.

Methodology

This section describes an artificial development system which combines fractal proteins with a Cellular Potts model of cells. This system is referred to as Fractal-Potts. As in other works connecting gene regulatory networks with multicellular models of development, cellular behaviours are derived from regulated concentration dynamics to construct a development program. However, Fractal-Potts additionally exploits the underlying spatial nature of fractal proteins to define other interactions within the multicellular environment, thereby increasing the number of evolvable developmental parameters.

Fractal proteins and gene regulation

Fractal proteins have the following attributes, which determine their interactions and behaviour within the context of various systems.

- *Shape* - The shape of a fractal protein is represented by a subset of the Mandelbrot set
- *Environment concentration* - Fractal proteins exist in varying concentrations within an environment, such as a cell cytoplasm or an extra-cellular medium
- *Functional role(s)* - Fractal proteins can: regulate gene expression (F_r-proteins); be used within the cell for behavioural and structural purposes (F_b-proteins); move outside of the cell (F_e-proteins); behave as receptors for extracellular signals (F_s-proteins)
- *Chemical meta-properties* - Protein degradation rates specify the rate at which their concentration decreases, in absence of their production within an environment

The regulation of gene expression within a cell is achieved by the production of appropriate transcription factors. Transcription factors are formed by 'merging' the fractal proteins found within a cells cytoplasm at each developmental time step. This creates a single fractal protein through a kind of 'fractal chemistry', which considers both the shape and concentration of each reactant protein in determining the final product. These transcription factors then interact with the genome, activating specific genes. Each gene within the genome has the following attributes, which determine their interaction with transcription factors and the subsequent regulatory dynamics:

- *Promoter region* - This specifies a subset of the Mandelbrot set.
- *Affinity threshold* - This along with the promoter region creates a precondition for gene expression, by specifying a minimum shape similarity that must exist between the promoter region and transcription factor for gene activation

- *Coding region* - This specifies the shape of the fractal protein produced given gene activation

- *Gene type* - This specifies the functional role(s) of the fractal protein associated with the coding region

- *Transcription meta-properties* - Gene activation results in the concentration of the coding protein increasing. The amount produced is dependent on the concentration of the transcription factor, a concentration threshold specific to each gene, and a number of global parameters (i.e. constant across all genes).

For full details of the regulatory dynamics, interested readers should consult Bentley (2004a).

This work introduces an additional fractal protein attribute. It is a real valued parameter, and is encoded within each gene for the associated coding protein. It has different purposes, depending on the proteins functional role:

- *Regulatory (F_r) and environmental (F_e) proteins* - Specifies a minimum concentration, below which the protein is not considered part of the 'merge' operation for the determination of transcription factors. Whilst Bentley (2003b) does specify a minimum concentration below which a fractal protein is no longer considered present in the cell, it is not specific to each protein and furthermore is not evolved. This is likely to make the transcription factors seen through development increasingly varied and dynamic.

- *Behavioural (F_b) proteins* - Allows the re-interpretation of continuous protein concentrations as Boolean values, for all-or-nothing structural changes or decision-like behaviours. This is referred to as a 'switching threshold'.

This additional attribute can be thought of as a function (and therefore context) dependent expansion of the chemical meta-properties associated with general fractal proteins.

Multicellular fractal proteins

The Cellular Potts (CP) model (also known as the Glazier-Graner-Hogeweg model) is a computational model of cells and tissues. Introduced by Graner and Glazier (1992), it was used to simulate the sorting of a mixture of two types of biological cells. The fundamental components of the basic CP model are a lattice $L \subset \mathbb{Z}^2$ representing a spatial environment and two functions, the cell identity function $\sigma : L \to \{1, \cdots, N_{\text{cells}}\}$ and the cell type function $\tau : \{1, \cdots, N_{\text{cells}}\} \to \{1, \cdots, N_{\text{types}}\}$. In addition, there is a matrix containing the differential surface energies between cell types, $J \in \mathbb{R}^{N_{\text{types}} \times N_{\text{types}}}$. A hamiltonian $H(t) : L \to \mathbb{R}$ describes the energy of a given configuration of cells on the lattice:

$$H(t) = \sum_{\substack{(i,j),(i',j') \\ \text{neighbors}}} J\Big[\tau(\sigma(i,j)), \tau(\sigma(i',j'))\Big]\Big[1 - \delta_{\sigma(i,j),\sigma(i',j')}\Big] + \lambda \sum_{\substack{\text{cells} \\ k}} \Big[a_k(t) - A_k(t)\Big]^2 \mathbb{1}(A_k(t) > 0) \tag{1}$$

Here $a_k(t)$ is the size of cell k, and $A_k(t)$ is its target size at time t. λ is a constant which determines the elasticity of the cell wall. The dynamics and evolution of the system can be simulated through a metropolis-style update (Graner and Glazier, 1992). Fractal proteins are introduced quite naturally to this model - at each point $(i,j) \in L$ in the environment and at a given time t there is a concentration of each fractal protein (which can of course be zero). The concentration of fractal protein F_a at site $(i,j) \in L$ at a time t is denoted $C_{i,j}(F_a, t)$. At each development time step, each of the N_{cells} completes its own gene regulation cycle independently based on the concentrations of proteins found within its interior. The total concentration of protein F_a found within cell $k \in \{1, \cdots, N_{\text{cells}}\}$ at time t is

$$C_k(F_a, t) = \sum_{(i,j) \in \text{cyto}_k} C_{i,j}(F_a, t) \tag{2}$$

where $\text{cyto}_k = \{(i,j) \in L \text{ s.t. } \sigma(i,j) = k\}$. These aggregated concentrations determine the set of fractal proteins which react to form the transcription factor at each development time step. If a gene whose coding region specifies F_a is activated, resultant concentration updates $\Delta C_k(F_a, t)$ are distributed across the cells cytoplasm uniformly - for each $(i,j) \in \text{cyto}_k \subset L$,

$$C_{i,j}(F_a, t) = C_{i,j}(F_a, t) + \frac{\Delta C_k(F_a, t)}{|\text{cyto}_k|} \tag{3}$$

Environmental proteins have the ability to move freely throughout the multicellular environment. This is achieved by introducing a diffusion process on the lattice L. Given an environmental protein F_e a diffusion term is added to the usual update rule, which then gives the following: For each $(i,j) \in \text{cyto}_k \subset L$,

$$C_{i,j}(F_e, t) = C_{i,j}(F_e, t) + \frac{\Delta C_k(F_e, t)}{|\text{cyto}_k|} + \alpha_e \nabla_L^2 C_{i,j}(F_e, t) \tag{4}$$

where ∇_L^2 is the appropriate lattice Laplacian and $\alpha_e \in \mathbb{R}$ is a diffusion constant specific to each environmental protein. The complete dynamics of environmental proteins can therefore be considered as a reaction-diffusion system.

Cellular behaviours

In this work, both protein concentrations and their fractal shape determine their role within development mech-

anisms. A key tool is the fractal protein similarity measure, which is defined as follows for two $n \times n$ proteins $F_1, F_2 \in [0, \cdots, 255]^2$:

$$S(F_1, F_2) = 1 - \frac{1}{255 \times n \times n} \sum_{i,j \in n \times n} |F_1[i,j] - F_2[i,j]| \quad (5)$$

Note that $0 \leq S(F_1, F_2) \leq 1$ for any two fractal proteins.

Inter-cellular communication Environmental proteins can be absorbed into one cells cytoplasm after being produced within the cytoplasm of a neighbour through diffusion, establishing a form of inter-cellular communication. In this work, diffusion speed is derived from the shape of the diffusing protein itself. We model a 'cellular substrate' as the diffusing medium. This medium contains a specific fractal protein F_{sub} in which all entries are zero, i.e. $F_{\text{sub}}[i,j] = 0$. The speed α_e at which an environmental protein F_e diffuses through the cellular substrate is then assumed to be its similarity in shape to F_{sub}:

$$\alpha_e = \beta_{\text{diff}} \cdot S(F_e, F_{\text{sub}}) \quad (6)$$

where β_{diff} is a user set parameter that ensures the numerical stability of Eq. 4.

Mitosis and apoptosis Cell size scales with the concentration of a specific behavioural protein F_g within the cell. Computationally this is achieved by linking the target cell size in Eq. 1 with this concentration: $A_k(t) = \beta_{\text{size}} \cdot C_k(F_g, t)$, where β_{size} is a user-set parameter based on lattice size. Mitosis occurs when $a_k > 0.25 * \beta_{\text{size}}$ - the cell is split into two, perpendicular to the longest axis associated with its 2D grid shape. After such an event, the concentration of all proteins are split randomly between the two cells. In this work, a cell can only undergo mitosis once. Apoptosis occurs when the concentration $C_k(F_g, t)$ falls below a user set threshold β_{death} - all proteins within the cell are destroyed, except from environmental proteins which remain.

Cell type differentiation Cell type is represented by one of a finite number of pre-determined colours - red, green, blue or white. Differentiation is interpreted to be an all-or-nothing event. The approach taken is to re-interpret continuous protein concentrations as Boolean values using the switching threshold values specified in the genome for each coding protein. Using a binary mapping, the cellular concentrations of two behavioural proteins can specify each of the four types, as shown in Table 1.

Inter-cellular adhesion The importance of the interplay between cell type and inter-cellular adhesion in producing complex morphologies has been demonstrated by Hogeweg (2000). In the CP model, cellular adhesion is determined energetically - in Eq. 1, the matrix $J \in \mathbb{R}^{N_{\text{types}} \times N_{\text{types}}}$ specifies the surface energy between each possible pairing of the

State	$C(F_{b1}, t) > T_{b1}$	$C(F_{b1}, t) \leq T_{b1}$
$C(F_{b2}, t) > T_{b2}$	red	green
$C(F_{b2}, t) \leq T_{b2}$	blue	white

Table 1: Specification of cell type through behavioural protein concentrations

N_{types} cell types via an appropriate matrix entry. In addition, an energy is specified between each cell type and the extra-cellular medium, which in this work is made equal to 1. Similar cell types are expected to adhere strongly to each other, and hence cells of the same type are not energetically penalised (the diagonal of J is zero). In this work the approach is to specify the surface energies between different cell types through a cellular fractal chemistry - the matrix J is constructed by associating a new fractal protein with each distinct cell type. The proteins associated with the four cell types are derived from the two behavioral proteins responsible for cell type determination. This is done by considering four possible reactions between the proteins, as shown in Table 2.

State	$C(F_{b1}, t) > T_{b1}$	$C(F_{b1}, t) \leq T_{b1}$
$C(F_{b2}, t) > T_{b2}$	$R_{\hat{F}} = \{F_{b1}, F_{b2}\}$	$R_{\hat{F}} = \{F_{b2}\}$
$C(F_{b2}, t) \leq T_{b2}$	$R_{\hat{F}} = \{F_{b1}\}$	$R_{\hat{F}} = \emptyset$

Table 2: Protein metabolism products associated with type-defining behavioural proteins

These four combinations of reactants $R_{\hat{F}}$ define four protein products through the same fractal chemistry used to determine transcription factors within the cell. In this work, the empty set of reactants is associated with the substrate protein F_{sub}. The entries of the matrix J are then calculated as one-minus the similarity measure between the different pairs of products. For example, given the products P_{red} and P_{blue}

$$J[1,3] = J[\text{red}, \text{blue}] = 1 - S(P_{\text{red}}, P_{\text{blue}}) \quad (7)$$

Chemotaxis Chemotaxis can be incorporated into the CP model by introducing a chemokine which diffuses across the lattice, and increasing the likelihood that a lattice site $(i,j) \in L$ will be changed to its neighbour $(i',j') \in L$ if the chemokine concentration is higher at (i,j). In our model an environmental protein F_c acts as a chemokine. Following Savill and Hogeweg (1997), the modification to the energy in the metropolis update is therefore

$$\Delta H' = \Delta H - \mu\Big(C_{i,j}(F_c, t) - C_{i',j'}(F_c, t)\Big) \quad (8)$$

Receptor clustering is a metabolic process that results in grouping of a set of receptors at a cellular location, often

to amplify the sensitivity of a signaling response. In some cellular systems, clusters form dynamically in response to activation by an extracellular ligand (Duke and Graham, 2009). In this model, clustering is imagined to occur when the chemokine binds to a specific receptor protein. The similarity between these proteins hence determines the degree of clustering, and as a result the baseline chemotaxis sensitivity μ:

$$\mu = \beta_{\text{chemo}} * S(R_{\text{prom}}, F_c) \tag{9}$$

Here β_{chemo} is a user-specified parameter which determines the maximum possible sensitivity - it is chosen in consideration with the other CP meta-parameters such as T and λ.

Developmental modules

In this work a module is a sub-routine of the development process, and is identified by a characteristic pattern of gene expression occurring at a number of points throughout development. Within the Fractal-Potts model, the activation of a gene at a specific point in development occurs when a suitable transcription factor is created within the cell. In order to be activated, the fractal similarity between the genes promoter region and the transcription factor must reach a threshold. The full set of unique transcription factors created through the cells history therefore characterises the full set of gene activation patterns used throughout development. In the case of the multicellular model, these unique transcription factors can be collected for each cell and combined into a super-set, which then characterises the full set of gene activation patterns for the complete development of the organism.

Evolutionary strategies

In this work the MAP-Elites algorithm (Mouret and Clune, 2015) is used for evolutionary search. The MAP-Elites algorithm is designed to deliver a large set of diverse, high-performing individuals, embedded in an archive that describes where they are located in the phenotype space. This archive is a collection of 'bins', a discretization of the phenotype space using some dimension of variation which is of interest. In this work, the selected dimension of variation is final colour composition. In the Fractal-Potts model there are four possible cell colours - red, green, blue and white. This therefore defines an archive of size $2^4 - 1 = 15$ (minus one since we do not search for organisms with no colour - i.e. dead). In order to create offspring from fit individuals the genetic operators described by Bentley (2003b) are used. Crossover allows for variable sized genomes, with appropriate mutation operators responsible for enlarging or shrinking the genomes within a population. Individuals develop for T_{dev} time steps. To improve robustness, development is run 20 times with different random seeds. From this set of trials the most common phenotype classification is identified, which then becomes the archive bin to which it is assigned. From the set of trials corresponding to this assignment, the average fitness is taken. Details of the fitness function are given in the next section.

Experiments

The aim of the experiments completed in this work were twofold. The first aim was to investigate the use of fractal proteins in coordinating temporal and spatial gene expression within a multicellular development system, and determine whether the system inherited any desirable properties such as stability and robustness. The second aim was to identify evidence of module formation in the regulatory networks evolved for the development task.

To this end, Fractal-Potts was tested on an expanded formulation of the FF Problem, referred to in this work as the General Flag (GF) Problem. The GF Problem arises naturally from the novelty search algorithm and involves the search for organisms which can develop flag-like morphologies. This abstraction of the development target creates a related set of problems, and therefore it might be expected that evolution preserves specific developmental modules. This was investigated in the second part of the experiment, where a cross-sectional study across evolved organisms was completed to search for evidence of module formation.

Fitness evaluation In the GF Problem, an n-colour flag-like morphology has n equally sized cellular regions of homogeneous cell type (colour) which are organised sequentially from left to right. The MAP-Elites archive bins only consider which colours exist in the final phenotype, and not the order in which they are sequenced across the grid. The fitness function is therefore designed to be flexible to alternative orderings - the pattern B-W-R (blue,white,red) would be assigned the same fitness as the pattern R-B-W within the [W,B,R] archive bin. This is achieved by assessing the number of pixel matches given each possible colour ordering, and selecting the one which provides the maximum fitness. Hence for n-colour phenotypes there are $n!$ assessments. Fitness is ultimately measured as the percentage of correct pixel matches with what would be considered the perfect flag.

Measuring stability and robustness Evolved organisms are assessed for stability by running the development time for twice that used in evolution. Organisms are assessed for robustness by randomly perturbing the internal concentration levels of a subset of cells at a random point through development. Each cellular perturbation is applied as a random increase or decrease in its protein concentrations, the magnitude of which can be up to 10% of the original. In each trial, 30% of cells are randomly targeted. In both stability and robustness, results are reported as a percentage of the original fitness.

Cross-sectional study of transcription factors The cross-sectional study of unique transcription factors is completed across all of the evolved organisms. The fittest organism from each of the 15 flag species is selected and developed. Throughout development the unique set of transcription factors is then collected, as well as information relating to each transcription factor's time of appearance and its duration within the organism.

Experimental setup Evolution ran for 500 generations. Evolutionary runs were seeded with a population of 10,000 genomes, each of which were initialized with 35 genes - 1 receptor, 3 behavioural, 6 regulatory and 1 environmental, with the remaining assigned random functional roles. Organisms were allowed to develop for 800 time steps, starting from a single cell with concentrations set at zero. Protein size was 15x15. For determining transcription dynamics the following parameters were used: $C_t = 0.2$, $C_s = 0.1$, $C_w = -0.1$, $C_p = 20$, $C_i = 1.5$. For more details see Bentley (2003b). The CP lattice size was 30x30, with temperature $T = 1$ and cell elasticity $\lambda = 10000$. For the cellular behaviours, $\beta_{\text{diff}} = 0.25$, $\beta_{\text{size}} = 100$, $\beta_{\text{death}} = 0.05$, $\beta_{\text{chemo}} = 5000$.

Results

In the experiments all of the possible colour combinations were discovered in a single evolutionary run, resulting in single, bi, tri and quad-colour flags. Organisms of high fitness were obtained, with excellent degrees of regional colour specification. In addition some phenotypes demonstrated stability when allowed to develop for a longer amount of time, with the average phenotype maintaining 80% of its original fitness. The fitness of some however did deteriorate quite significantly. Some organisms additionally demonstrated robustness, being able to recover from random chemical perturbations applied through development and still obtain high fitness solutions. The full set of quantitative assessments is shown in Table 3, along with examples of developed phenotypes in Fig. 1.

In the cross-sectional study of transcription factors, a total of 199 unique transcription factors were collected through development across the 15 species. Interestingly but perhaps expected, the distribution of this total was not uniform across the different flag species. Single colour organisms had an average of 9 unique transcription factors created through development. Bi-colour and Tri-colour organisms on average had 19 and 24 respectively, with the quad-colour organism demonstrating 21. This is suggestive that the number of transcription factors observed is related to the complexity of the development task. The full history of unique transcription factors for each species is shown in Fig. 2. The colour-bar indicates the time at which the transcription factor first appears in the organism. A number of the transcription factors (36) were found to be present in multiple organisms at some point through their development. This set of common transcription factors is shown in Fig. 3. It was found that every organism had at least one transcription factor which was identifiable in another organism at some point.

Species	Average fitness (50 trials)	Fitness standard deviation	Average stability (50 trials)	Average robustness (50 trials)
d-R	100%	0%	100%	99.7%
d-B	100%	2.6%	99.9%	99.6%
d-G	100%	0%	100%	99.3%
d-W	100%	0.7%	98.3%	99.3%
d-BR	86.6%	1.8%	98.3%	83.3%
d-RG	92.5%	6.2%	56.6%	71.5%
d-GB	67%	3.6%	74.2%	82.1%
d-WR	75%	6.1%	66.8%	78.2%
d-BW	92.1%	10.8%	72.5%	61.6%
d-GW	92.2%	5.6%	99.9%	91%
d-RGB	72.4%	2.5%	61%	71.8%
d-BWR	77.3%	7.4%	62.2%	68.4%
d-RGW	60.3%	7.4%	59.2%	76.2%
d-GWB	76.7%	2.4%	78.3%	91.4%
d-RGWB	46.9%	3.7%	92.3%	83.3%

Table 3: Fitness, Stability and Robustness of fittest organism per species. The species notation indicates the final colours and orderings, e.g. d-RGB develops red-green-blue, patterned left to right.

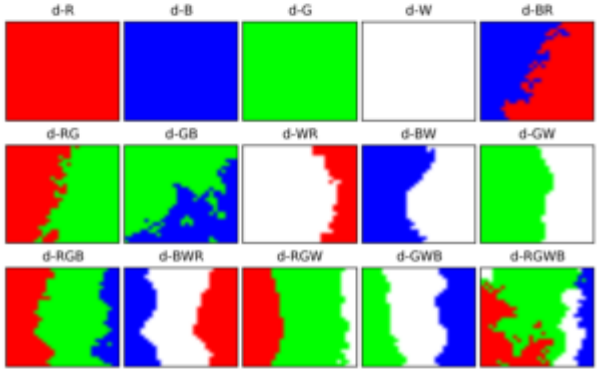


Figure 1: Example development of the fittest evolved organisms

Discussion

The first part of the results shows that Fractal-Potts is well specified, and can achieve success on the classic problem of flag development. In addition some evolved organisms demonstrate desirable properties, such as robustness to environmental perturbations and the ability to maintain stable phenotypes. MAP-Elites was beneficial in the search for flag-like organisms. In particular no penalization terms were added to the fitness function in order to encourage organisms early on in evolution to use all colours - the novelty search naturally resulted in this behaviour.

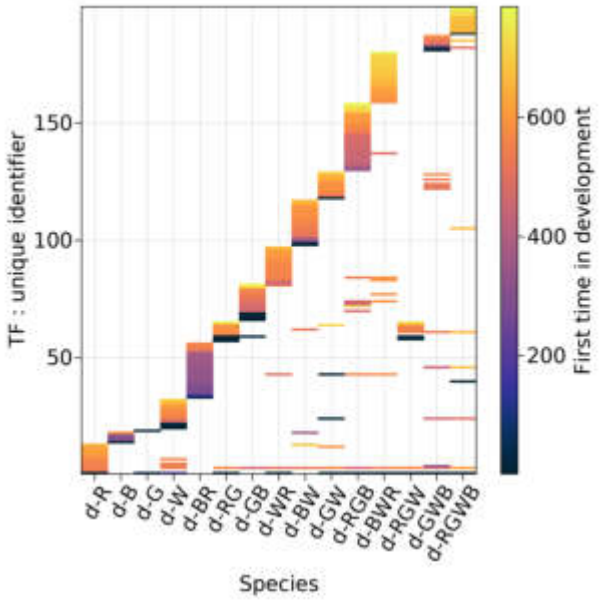


Figure 2: Transcription factors (TF) seen across organisms through development

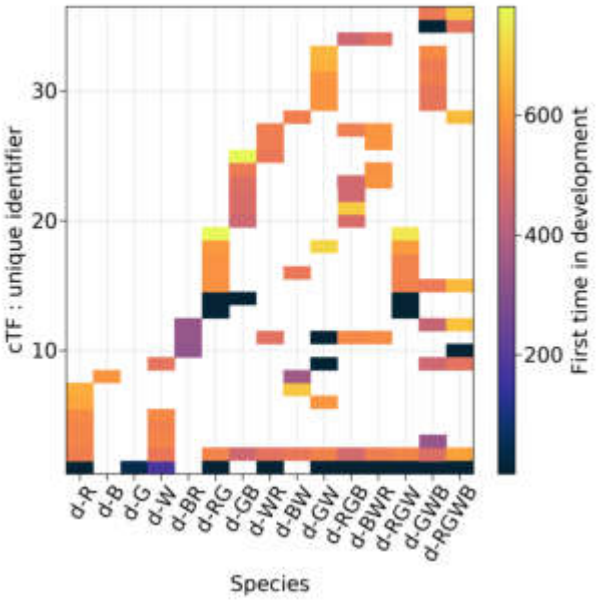


Figure 3: Common transcription factors (cTF) seen across organisms through development

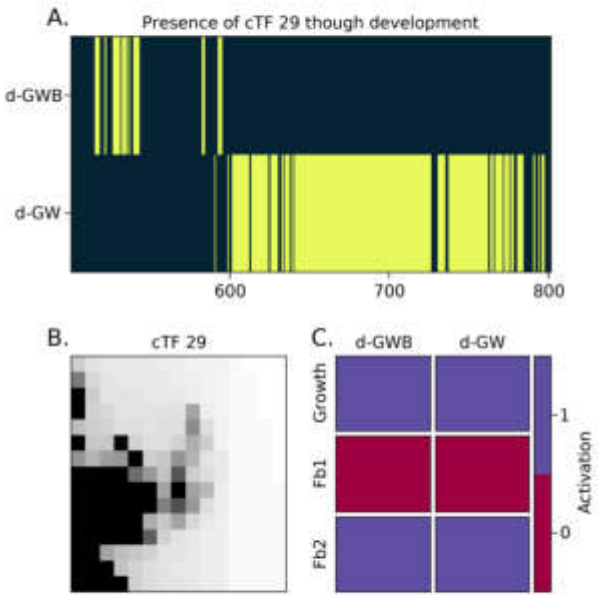


Figure 4: Presence of a common transcription factor (cTF-29) and the corresponding behavioural gene activation pattern in two different species

The existence of common transcription factors across different organisms suggests that evolution has discovered and re-used patterns of gene activation, both within an organism's development (module re-use) and across different organisms (module inheritance) despite the selective pressure of MAP-Elites to find novel phenotypes. Evidence to support this hypothesis can be found by analysing the effects of individual transcription factors. For example, Fig. 4 illustrates one common transcription factor cTF 29. In Fig. 4.B the fractal shape of cTF 29 is displayed, and in Fig. 4.C the corresponding activation pattern of the 3 behavioral genes responsible for cell growth and colour differentiation is shown. It can be seen that the activation pattern is identical for the two organisms d-GWB and d-GW. Examination of Table 1 indicates that increased levels of the F_{b2} protein has a positive logical association with the transition to red or green cell types, depending on the concentration of F_{b1}. By observing the development process associated with both d-GW and d-GWB, it was confirmed that the occurrence of cTF 29 corresponds with the proliferation of green cells, in agreement with the activation pattern. In Fig. 4.A the presence (yellow bars) of cTF 29 is shown throughout development. The intermittency of the protein within both organisms indicates that the gene activation pattern occurs during different times through development, rather than continuously. Out of the 36 common transcription factors it was found that 17 displayed these characteristics, producing a common pattern of activation across the three behavioural genes in all organisms, but at different times. Of the 3! = 6 logical combinations of gene activation, 3 were present.

This is strong evidence that evolution preserved and re-used developmental modules - common patterns of be-

haviour gene expression triggered by specific transcription factors - both across organisms and throughout the development process. It was also found that a number of the common transcription factors activated equivalent patterns amongst the set of regulatory genes shared across organisms. Whilst a full analysis is outside of the scope of this paper, this is an indication that evolution has additionally preserved important regulatory sub-routines.

Regional specification remains a relatively unexplored and rarely modeled process in ALife. Here we have shown that a combination of fractal proteins, Cellular Potts and MAP-Elites enables us to explore the space of developmental solutions and furthermore observe the emergence of biologically plausible phenomena, such as the learning of regularities in solutions and the automatic creation and reuse of modules. We anticipate that these methods will enable further insights into artificial developmental processes in the future.

References

Banzhaf, W. (2003). On the dynamics of an artificial regulatory network. In *European Conference on Artificial Life*, pages 217–227. Springer.

Bentley, P. J. (2003a). Evolving fractal gene regulatory networks for robot control. In *European Conference on Artificial Life*, pages 753–762. Springer.

Bentley, P. J. (2003b). Evolving fractal proteins. In *International Conference on Evolvable Systems*, pages 81–92. Springer.

Bentley, P. J. (2004a). Evolving beyond perfection: An investigation of the effects of long-term evolution on fractal gene regulatory networks. *Biosystems*, 76(1-3):291–301.

Bentley, P. J. (2004b). Fractal proteins. *Genetic Programming and Evolvable Machines*, 5(1):71–101.

Bentley, P. J. (2009). Methods for improving simulations of biological systems: systemic computation and fractal proteins. *Journal of The Royal Society Interface*, 6(suppl_4):S451–S466.

Bentley, P. J., Kumar, S., et al. (1999). Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In *GECCO*, volume 99, pages 35–43.

Bowers, C. P. (2005). Formation of modules in a computational model of embryogeny. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 537–542. IEEE.

Brown, T. A. (2002). The human genome. In *Genomes. 2nd edition*. Wiley-Liss.

Chavoya, A. and Duthen, Y. (2008). A cell pattern generation model based on an extended artificial regulatory network. *Biosystems*, 94(1-2):95–101.

Duke, T. and Graham, I. (2009). Equilibrium mechanisms of receptor clustering. *Progress in biophysics and molecular biology*, 100(1-3):18–24.

Eggenberger, P. et al. (1997). Evolving morphologies of simulated 3d organisms based on differential gene expression. In *Proceedings of the fourth european conference on Artificial Life*, pages 205–213. Citeseer.

Friston, K., Levin, M., Sengupta, B., and Pezzulo, G. (2015). Knowing one's place: a free-energy approach to pattern regulation. *Journal of the Royal Society Interface*, 12(105):20141383.

Garibay, O. O., Garibay, I. I., and Wu, A. S. (2003). The modular genetic algorithm: Exploiting regularities in the problem space. In *International Symposium on Computer and Information Sciences*, pages 584–591. Springer.

Graner, F. and Glazier, J. A. (1992). Simulation of biological cell sorting using a two-dimensional extended potts model. *Physical review letters*, 69(13):2013.

Hogeweg, P. (2000). Evolving mechanisms of morphogenesis: on the interplay between differential adhesion and cell differentiation. *Journal of theoretical biology*, 203(4):317–333.

Joachimczak, M. and Wróbel, B. (2009). Evolution of the morphology and patterning of artificial embryos: scaling the tricolour problem to the third dimension. In *European Conference on Artificial Life*, pages 35–43. Springer.

Knabe, J., Schilstra, M., and Nehaniv, C. L. (2008). Evolution and morphogenesis of differentiated multicellular organisms: autonomously generated diffusion gradients for positional information. *Artificial Life XI*.

Kumar, S. and Bentley, P. J. (2003). Computational embryology: past, present and future. In *Advances in evolutionary computing*, pages 461–477. Springer.

Lacquaniti, F., Ivanenko, Y. P., d'Avella, A., Zelik, K., and Zago, M. (2013). Evolutionary and developmental modules. *Frontiers in Computational Neuroscience*, 7:61.

Levine, M. and Davidson, E. H. (2005). Gene regulatory networks for development. *Proceedings of the National Academy of Sciences*, 102(14):4936–4942.

Miller, J. F. (2004). Evolving a self-repairing, self-regulating, french flag organism. In *Genetic and Evolutionary Computation Conference*, pages 129–139. Springer.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Pollack, J. and Lowell, J. (2016). Developmental encodings promote the emergence of hierarchical modularity. In *ALIFE 2016, the Fifteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 344–351. MIT Press.

Savill, N. J. and Hogeweg, P. (1997). Modelling morphogenesis: from single cells to crawling slugs. *Journal of theoretical biology*, 184(3):229–235.

Stanley, K. O. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial life*, 9(2):93–130.

Wolpert, L. (1969). Positional information and the spatial pattern of cellular differentiation. *Journal of theoretical biology*, 25(1):1–47.

Empathic Active Inference: Active Inference with Empathy Mechanism for Socially Behaved Artificial Agent

Tadayuki Matsumura[1], Kanako Esaki[1] and Hiroyuki Mizuno[1]

[1]Research & Development Group, Hitachi, Ltd., 1-280, Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan.

tadayuki.matsumura.bh@hitachi.com

Abstract

This paper proposes a method for an artificial agent to behave socially by controlling it by active inference with an empathy mechanism. Active inference is a Bayesian hypothesis for understanding the mechanism of a biological agent's cognitive activities and is basically defined for single-agent cases. We extended active inference to the case of an agent surrounded by other agents. These other agents are not only objects of recognition but also sources of social perceptions and actions. An agent controlled with the proposed method infers the others' expectations toward itself on the basis of an empathy mechanism and tries to act in response to the expectations. Although defining proper sociality for a given situation is difficult since it differs by situation, we define sociality as an agent behaving as others expect. Accordingly, the others surrounding the agent are teachers for the agent to learn proper sociality; thus, an agent controlled with the proposed method can learn proper sociality in a variety of situations in a unified manner. We evaluated the proposed method regarding the controlling of autonomous mobile robots (AMRs) and evaluated sociality from the trajectory of the AMRs. From the evaluation results, an agent controlled with the proposed method could behave more socially than an agent controlled by standard active inference. In two agents case, the agent controlled with the proposed method behaved in a social way that decreased travel distance of another by 13.7% and increased margin between the agents by 25.8%, even if it increased travel distance of the agent by 8.2%. They also indicate that an agent controlled with the proposed method behaves more socially when it is surrounded by altruistic others but less socially when surrounded by selfish others.

Introduction

We humans are social animals. We form a community and establish explicit or non-explicit rules in the community. It is essential for the normal functioning of a society that each individual follows these rules. Therefore, artificial agents operating in our daily spaces are also required to follow such rules. In other words, an artificial agent is required to behave socially. Having sociality helps artificial agents operate in our complicated real environment. Our social abilities are acquired through a long evolutionary process and fundamental for our cognitive systems. We do not live alone, and there are always others around us. We can make appropriate cognition and take proper actions due to the help of the presence of others. Others surrounding us are not only objects of recognition but also sources of social recognition and social action.

On the basis of these ideas, we propose a method for an artificial agent to behave in a socially desirable manner with the help of others surrounding it. The proposed method is based on active inference (Friston et al., 2011; Adams et al., 2013; Friston et al., 2016; Friston et al., 2017;). Active inference was proposed under the context of the free energy principle (FEP), which is a hypothesis for understanding the mechanism of a biological agent's cognitive activities (Friston et al., 2006; Friston, 2010a). In FEP, the brain is viewed as a device performing variational Bayes inference. The human brain is explained as always predicting the future and works to decrease the uncertainty of predictions. Similar ideas were widely studied with certain contexts such as Bayesian brain hypothesis (Knill and Pouget, 2004), predictive coding (Rao and Ballard, 1999), and Helmholtz machine (Dayan et al., 1995). The unique point of active inference is that it explains actions as well as perception with only one principle: minimization of variational free energy. It is assumed that there is an internal model for predicting external environments in the human brain. The process of perceptions is explained as the process of minimizing free energy of the internal model by updating the parameters of that model. The process of taking actions is also explained as the process of minimizing expected free energy for the future under a certain action. Active inference is studied in a variety of environments (Pio-Lopez et al., 2016; Parr and Friston, 2017; Friston et al., 2018). It has been combined with deep learning for applying it to more complicated environments such as robot control (Ueltzhöffer, 2018; Millidge, 2020; Fountas, 2020; Tschantz, 2020; Catal, 2020; Catal, 2021). Although active inference is basically defined for single-agent cases, some works recently extended it to multi-agent cases (Friedman, 2021; Kaufmann, 2021; Albarracin, 2022). We also extended active inference to include free energy of others surrounding an agent. More specifically, the action of the agent is determined on the basis of two type of uncertainty, i.e., (1) the agent's uncertainty of others and (2) others' uncertainty of the agent. By assuming the second type of uncertainty, an agent controlled with the proposed method (hereafter, empathetic agent) attempts to act based on the other's expectations, which makes it behave in a socially desirable manner. This is achieved by estimating others' predictions regarding the agent on the basis of the idea of an empathy mechanism called a mirror system (Rizzolatti and Craighero, 2004; Cattaneo and Rizzolatti, 2009). With this mirror system, the empathetic agent estimates the prediction of others using the same model used to

predict the future of its external environment. When the model is used to predict others' predictions, the input data of the model is changed from the observation of the agent to the observation of the others, which is also estimated by the agent.

We evaluated the proposed method in a situation of controlling autonomous mobile robots (AMRs). AMRs should not only avoid collisions but should also have sociality when they are operating in human spaces. For example, the sociality for AMRs can be defined as the margin of distance from others during their movement. Defining the proper sociality is difficult since it varies from situation to situation. Therefore, it is difficult to learn the proper sociality in standard reinforcement learning in which the proper margin of distance from others has to be manually encoded as the reward in the training for each situation. In this paper, this problem is solved by defining sociality as behaving as others expect. Namely, proper sociality in a certain environment is expressed as the behavior of others in the environment. Others are not obstacles but help to generate social behavior in our agent for controlling the AMRs. The sociality of the empathetic agent is acquired through the learning of the internal models for predicting the external environment including others' behaviors. In the evaluations, the empathetic agent learned different sociality according to the difference in the surrounding environments. When it learned in the environment in which others were selfish, it tended to behave selfishly. On the other hand, when it learned in the environment in which others were altruistic, it tended to behave altruistically.

Empathic Active Inference

Active Inference

As mentioned above, active inference is a hypothesis for understanding the mechanism of actions of biological agents (Friston et al., 2011; Adams et al., 2013; Friston et al., 2016; Friston et al., 2017;) and was proposed under the context of the FEP (Friston et al., 2006; Friston, 2010a). The FEP is widely studied and is applied to explain many cognitive abilities/phenomena such as behavior (Friston et al., 2010b), planning (Kaplan and Friston, 2018), autism (Quattrocki and Friston, 2014), and attention (Feldman and Friston, 2010). The most basic cognitive mechanisms, perceptions, and actions are also explained as the process for minimizing variational free energy. It seems natural to explain perception as inference minimizing the uncertainty of the internal model for the inference. More interestingly, an action is also explained with the same principle. In the FEP, an action is explained as the process of inference in which biological agents actively act to decrease uncertainty, and the best action is that expected decrease uncertainty the most. This process for performing actions is called active inference.

We now mathematically describe the FEP and active inference. As illustrated in Figure 1, there is an agent that receives observations (o_t) from an environment at time t. There are hidden states (s_t) behind the process of generating the observation. The agent takes an action (a_t) each t then receives the next observation (o_{t+1}) from the environment. The agent always infers the hidden state and action at the current state from the current observation as the following posterior,

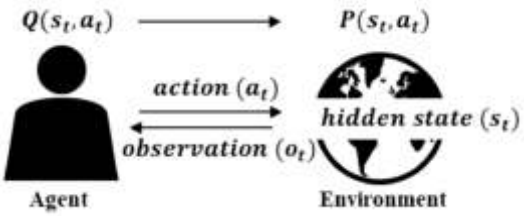


Figure. 1: Overview of free energy principle

$$P(s_t, a_t|o_t). \quad (1)$$

A variational density, $Q(s_t, a_t)$, is assumed to infer the posterior by variation methods. Under this context, variational free energy (F) is expressed as

$$F = -logP(o_t) + KL[Q(s_t, a_t)||P(s_t, a_t|o_t)], \quad (2)$$

where KL is a Kullback–Leibler divergence. Variational free energy is the same as evidence lower bound (ELBO) in machine learning (Blei et al., 2017). In the FEP, it is important idea that the action (a_t) is also inferred as like as the hidden state (o_t). On the basis of this idea, the process of minimizing free energy can be two types of processes. The first type of minimization process is related to the inference for hidden state. This is related to perception, i.e., it can decrease by refining the internal model for inferring the probability of the hidden state and its transition. More interestingly, an agent can decrease the free energy by refining the internal model for inferring the probability of the action, and this is the second type of process of minimizing free energy. From this viewpoint, it can be said that our action-making process is also the process of inference, as with perception. This process is called active inference. Agents based on the FEP perceive and act by minimizing variational free energy with these two types of minimization process.

In active inference, the desired distribution of actions at a given hidden state, $P(a_t|s_t)$, is expected to minimize free energy for a future state when an action is taken from the distribution. In previous works (Millidge, 2020), $P(a_t|s_t)$ was defined by

$$P(a_t|s_t) = \sigma\big(-\gamma G(s_t, a_t)\big). \quad (3)$$

where σ is a softmax function, γ is a precision weight, and $G(s_t, a_t)$ is the expected free energy. Expected free energy is the estimated free energy for future t. Eq. (3) means that the agent estimates free energy for a certain action, i.e., the agent makes a planning. If the agent is assumed to plan several steps ahead, the expected free energy is estimated for the sequence of actions, $\pi = \{a_t, a_{t+1}, \dots a_{t+T}\}$. In active inference literature, π is called a policy. From Eq. (2), expected free energy for a single time-step is expressed by

$$G(s_t, a_t) = -logP(o_t) + KL[Q(s_t)||P(s_t|o_t)]. \quad (4)$$

A neural network model is used to estimate expected free energy in (Millidge, 2020), and a Monte-Carlo tree search is used in (Fountas, 2020). In active inference literature, the first term $(-logP(o_t))$ is treated as preference of the agent to the observation, and the intention of the agent is encoded into this term as a reward signal (r),

$$G(s_t, a_t) = -r(o_t) + KL[Q(s_t)||Q(s_t|o_t)], \quad (5)$$

where $P(s_t|o_t)$ is approximated by $Q(s_t|o_t)$. The second term in Eq. (5) is called intrinsic value and expresses curiosity

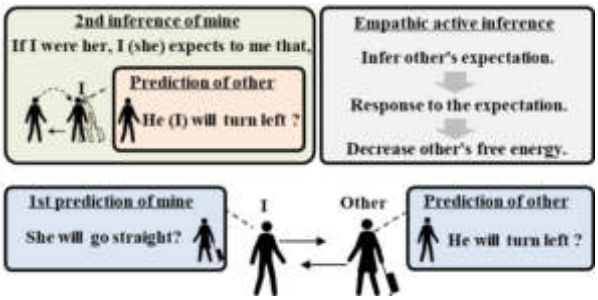


Figure. 2: Overview of empathic active inference

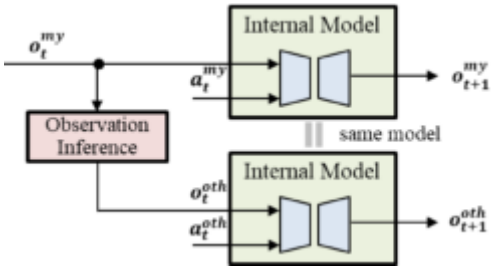


Figure 3: Processes of an inference of another's inference

to explore the environment. From Eq. (5), the agent acting on the basis of active inference takes into account both the intentional behavior expressed by the reward term and curiosity to explore the environment.

Simulation Theory and Mirror System

We extended active inference to generate social behavior. For this purpose, we embed the human capacity of empathy into active inference. There are mainly two types of human activity regarding empathy; (1) cognitive empathy and (2) emotional (or affective) empathy (Davis, 1983). Cognitive empathy is the ability of inferring another's mental state. Emotional empathy is the ability to feel what others are feeling as if it were your own feelings. In any types of empathy, understanding or sharing the experiences or feelings of others is thought to be related to our sociality (Eisenberg and Miller, 1987). Theory of mind and simulation theory are theoretical frameworks for understanding the mechanisms of these emotion. According to simulation theory, we can infer another's mental state by simulating what we would infer or feel if we were in the same situation as to other. Although there is still much room for discussion, it is suggested that mirror neurons and mirror systems are involved in such empathy (Keen, 2006; Gazzola et al., 2006). A Mirror neuron is a neuron that fires both when an animal acts and when the animal observes the same action performed by another. Mirror neurons were first observed in other primate species, and similar brain activities were suggested in humans. Neural systems related to mirror neurons are called mirror systems. Our proposed method enables an agent to empathize with others and is inspired by mirror system and simulation theory to virtually experience the experiences of others using its internal models as like simulating others' mental states with its own body.

Active Inference with Empathy Mechanism

Although there have been studies on applying active inference to control artificial agents, the application mainly focused on a case in which only a single controllable agent is assumed. We extend the situation to the multiple-agent case, especially, when multiple agents are us humans. In Figure 2, there are two types of agents, *I* and *others*, and both acting with active inference. The empathetic agent *I* always infers the future external environment and attempts to decrease the uncertainty of the inference. It is important that the *others* surrounding *I* also always infer the future external environment and attempt to decrease the uncertainty of inference. *I* is an uncertain factor for the *others*. Given these situations, there is another way to decrease free energy in addition to the ways described in the previous section about active inference. The way is to act as others' uncertainty decrease. Although it will not decrease *I's* free energy, it will decrease the total amount of free energy in the group of agents. Similarly, the free energy of *I* can decrease by the actions of the *others* if the *others* act to decrease the free energy of *I*. We can manipulate free energy not only through our actions but also through the actions of others by thinking of free energy collectively rather than individually. Actions based on this way can be said to be for others, i.e., social action. If each individual in a group behaves in this manner, it means that they behave as the others expect them to behave. We assume this state is a preferable social state for a group of agents, and we developed our method on the basis of this idea.

An empathetic agent first needs to infer another's inference to it. For this purpose, the empathetic agent uses an idea inspired by mirror systems and simulation theory. The core idea for inferring another's inference is that the other's inference is inferred using the same internal model used for inferring the empathetic agent's external environment. The process of inference of another's inference consists of two processes, as illustrated in Figure 3.

The first process is to estimate the observation of others (o_t^{oth}) from the observation of the empathic agent (o_t^{my}). If the observation is given by an image of vision, then the agent infers the observed image of the others. In this case, the method for novel view synthesis, such as Generative Query Network (GQN) (Eslami et al., 2018), can be used. If the observation is the set of positions of the others extracted from object detectors such as infrared sensors, the observation of others can be estimated simply by coordinate transformation. However, even in such a simple case, due to partial observability, it is not always possible to completely estimate the observation of others. For example, if there is an object that is not visible to the empathic agent due to occlusions or other factors, information about that object will be missing, even if the others see it. The second process generates prediction for the external environment of others (o_{t+1}^{oth}) by giving the estimated observation of others (o_t^{oth}) to the internal model that the empathic agent uses to predict its external environment. This method inspires a mirror system and simulation theory to infer the inferences of others by simulating the situation of if 'I were in the other's situation'. The action of the other (a_t^{oth}) is also required to infer the inference of the other. Simple methods can be applicable to generate the action of others such as randomly

selected action. We assumed no-operation (NOP) action as the action of others. This is the idea of predicting 'when I stop, the other will act first'.

As explained in the previous section, an agent acting in accordance with active inference will take the action that will decrease the expected free energy the most. Intentional behavior is achieved by encoding the reward information corresponding to the behavioral intention as a preferable observation. We similarly extend Eq. (5) to a form that encodes intentional response to others' expectations to the agent as a reward,

$$\begin{aligned} G(s_t, a_t) &= -r(o_t) + KL[Q(s_t)||Q(s_t|o_t)] \\ &= -\{r_{my}(o_t) + \sum_i r_{oth}(o_t, e_t^i)\} \\ &\quad + KL[Q(s_t)||Q(s_t|o_t)]. \end{aligned} \tag{6}$$

The first term is the reward for the agent's goal (r_{my}), which is determined by the observed information, and is the same as the reward in Eq. (5). In Eq. (6), a reward term for the expectations of others (r_{oth}) are added, and they are determined by the observation (o_t) and expectation of others to the agent (e_t^i) which is estimated by the agent. The expectation from others is derived from the inference of others (o_t^{oth}). For example, if the positions of the others surrounding the others are estimated as the inference of others, the positions of the empathic agent inferred by the others are the expectations from the others. Similar to the flexibility in encoding rewards in active inference (Millidge, 2020), expectations from others can be encoded flexibly, such as by using probability distributions. For multiple others, the reward for each of the other (i) is summed. It is important that an inference of others to the empathic agent is interpreted as an expectation to the empathic agent from others. The proposed agent takes the action that will decrease the expected free energy described in Eq. 6 the most.

Evaluation

Simulation Setup

To evaluate the sociality of an empathetic agent, we ran multi-agent simulations in which multiple agents, including the empathetic agent, are walking from their initial positions to their destinations and need to avoid colliding with each other. The empathetic agent was controlled with the proposed method, and the others were controlled by the social force model (SFM) (Helbing and Molnar, 1995), which is a model for controlling pedestrians in a social space. Although there are a variety of extensions, the SFM basically models the motion of agents by the combination of a driving and repulsive forces. The driving force describes the motivation of agents to move toward the given goal at a certain desired velocity. The repulsive force represents the motivation of agents to avoid colliding with others or with obstacles such as walls.

To evaluate the sociality for a variety of scenarios, two types of conditions were changed depending on the scenario. The first condition was the situation of the scenario. Three types of situations were assumed for the simulation, as illustrated in Figure 4. In this section, the standard/empathetic agent is called the "player", and the other agents controlled with the SFM are simply called the "others" or "other". The player is the red dot in Figure 4. Situation (a) is the simplest in which two agents,

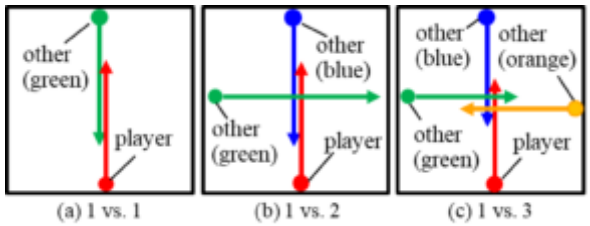


Figure 4: Types of simulation situations

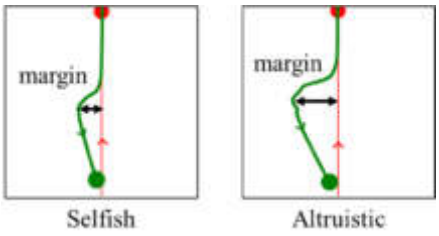


Figure 5: Two types of other

the player and one other, walk from their initial points to their destinations. Because their initial points and destinations are opposite each other, the player and other must take the non-shortest path (straight line between initial point and destination) to avoid colliding. Situation (b) is denser than (a) in which two others plus the player are walking and will cross each other at the center of the field. Situation (c) is the densest in which three others plus the player walk in a crossroad and will pass each other at the center of the field. There are no obstacles (i.e., walls) in all situations, and agents can walk in any area of the field.

The second condition is the type (characteristics) of other, i.e., selfish or altruistic. The type of other is controlled by the weight of the driving and repulsive forces in the SFM. When others are selfish, the weight of driving force toward the destination is set higher than that of the repulsive force with others. When others are altruistic, the weight of the repulsive force is set higher than that of the driving force toward the destination. Example trajectories of each type are illustrated in Figure 5. The player moves in a straight line toward the destination, and the other takes the non-shortest path to avoid colliding with the player in both cases. The difference between the types of other appears in the difference of the margin to avoid collision. When the other is altruistic, it walks with a larger margin with the player than that when it is selfish.

The observation of the player is the position of agents relative to the current position of the player. The observation is constructed for the two simulation steps, i.e., $[o_{t-1},\ o_t]$. As an ideal case of observation assumed in this simulation, there is no lack of data caused by occlusions, and there is no noise in the observation. The action space is defined as a discrete space. The player moves a constant distance in five directions $(-60°, -30°, 0°, 30°, 60°)$ toward the current direction. The constant distance of the player's movement is almost the same as that of others. In addition to these actions, the player can select NOP action, i.e., stop at the current position. A total of six actions is the action space for the player.

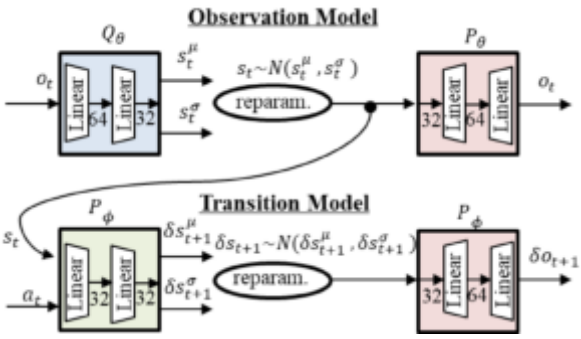


Figure 6: Models for empathic active inference

Model and Training Setup

Two densities are assumed in the evaluation, and they are modeled using simple neural networks. The models are illustrated in Figure 6. The first model is the observation model for $Q_\theta(s_t|o_t)$ and $P_\theta(o_t|s_t)$ parameterized by θ. The second model is the transition model for $P_\phi(\delta s_{t+1}|s_t, a_t)$ and $P_\phi(\delta o_{t+1}|\delta s_{t+1})$ parameterized by ϕ. Both of the models consist of two fully connected layers (Liner), and the dimension of the latent space is set to 32. The observation and transition models use a re-parameterization trick to express probabilistic distribution with neural networks as like variational auto-encoder (VAE) (Kingma and Welling, 2013). The latent vectors in both the observation and transition models are assumed to be distributed in normal distribution. The distributions of the latent vectors of the observation model (Q_θ) and transition model (P_ϕ) are learned to get closer to each other. Here, the transition model and its learning are modified from the pure FEP in this paper. The transition model is constructed to model the difference in the observation (i.e., difference in positions) between the present and future, not the absolute position in the future. This is because there is continuity in this environment, namely, the position of agents continuously changes, and agents do not jump to far away places in a time. This feature can help to learn transition model. For this modification, each of the distributions of the latent vectors of the observation model (Q_θ) and the transition model (P_ϕ) are learned to become closer to the standard normal distribution, $N(0, I)$, respectively, and simultaneously, the distribution of the observation model (Q_θ) for s_{t+1} is learned to become closer to the sum of the normal distribution of the observation model (Q_θ) for s_t and the transition model (P_ϕ) for s_{t+1}. Namely, the KL divergence described in Eq. (7) also decreases in the learning process,

$$KL[N(s^\mu_{t+1}, s^\sigma_{t+1})||N(s^\mu_t + \delta s^\mu_{t+1_t} s^\sigma_t + \delta s^\sigma_{t+1}). \quad (7)$$

The action is encoded into one-hot vectors.

In this evaluation, we do not assume the density of $Q(a_t|s_t)$ while it is assumed as a policy model in the previous works (Millidge, 2020; Fountas, 2020) because the action of the player is determined by the action probability $P(a_t|s_t)$ described in Eq. (3) with Eq. (6). The training of policy model is out of the scope in this study. The neural networks were trained in an offline manner since the aim was evaluating sociality of the empathetic agent's behavior, not validation of the feasibility of online training. Therefore, the training data for each scenario, i.e., three situations and two others' characteristics, are generated for a randomly acting player. During training-data generation, the player and one of the others have interchanged each other at random. This is for giving the player experiences of others' viewpoints, and it is necessary because the player must infer the other's inference with the other's viewpoint. When the player has no experience for the others' viewpoint, it cannot infer the inference of others. 1M samples were generated for each scenario. Adam was used for optimizing the neural network's parameters (Kingma and Ba, 2014), and its learning rate was constantly set to 1e-4. Training proceeded for 10,000 epochs. KL vanishment is a known difficulty in the learning of the VAE model, and KL annealing was proposed to tackle this problem (Bowman et al., 2016). In the learning process, the weight of KLD loss is doubled from 1e-4 at every 1000 epochs as KLD annealing.

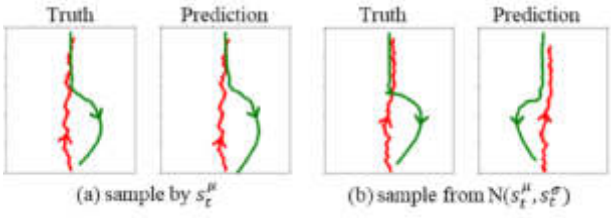


Figure 7: Examples of prediction results

Results

Examples of prediction from the trained observation and the transition models are shown in Figure 7 for situation (a) with the other's being altruistic. Predictions were cumulatively generated for a time horizon until reaching the destination. The player predicted the future for several steps by using only the first observation (o_1) and its action sequence ($a_1, a_2, \ldots a_T$). The action sequence is randomly generated. The player predicted future observation (p_1) by using (o_1, a_1) then further predicted future observation (p_2) by using (p_1, a_1), and so on until a certain time. This procedure for predicting future by accumulating its prediction is necessary when the player decides the action on the basis of the expected free energy for several time steps ahead. In Figure 7, prediction for both (a) sampling by the center of distribution (s^μ_t) and (b) sampling from $N(s^\mu_t, s^\sigma_t)$ are shown. The prediction result matches the truth trajectory for (a). However, the prediction result does not match the truth trajectory for (b). This is because the player predicts with uncertainty, and the uncertainty is accumulated. This uncertainty intentionally fluctuates the prediction, and it makes the predictions diverse. The player decides the action that decreases expected free energy the most on the basis of these diverse predictions.

Next, we evaluated the sociality of the player. In this evaluation, sociality is discussed on the basis of the trajectory of the agents. If the player has less sociality, it goes straightly to the destination regardless of the others. On the other hand, the player actively takes a circuitous trajectory to avoid colliding with others when it has more sociality. The difference in sociality appears in the shape of a trajectory like that

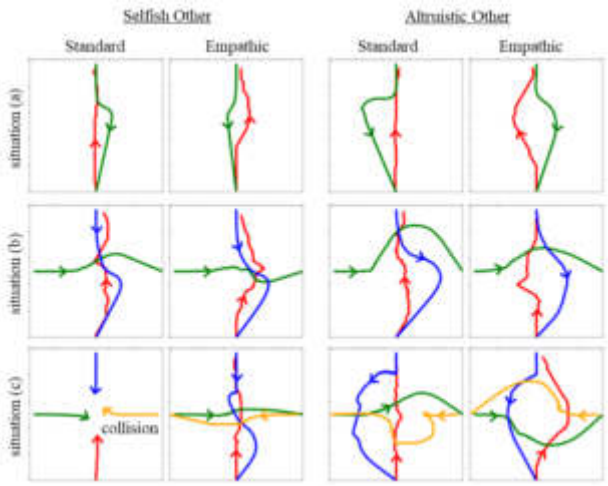


Figure 8: Trajectories of agents for each scenario

illustrated in Figure 5. We show the trajectory of the agent to intuitively understand its sociality of agent. Moreover, the total travel and minimum distances between the player and others during the simulation are also shown to quantitatively understand the sociality. The behaviors of the agents are shown in Figure 8. In this evaluation, the player estimated the expected free energy described in Eq. (6) by Monte-Carlo tree search (MCTS) with the learned models (Coulom, 2006; Fountas et al., 2020). The learned models were used for predicting the future state and value of each action (i.e., expected free energy) in MCTS. The maximum depth of the tree was set to 3 (i.e., three time steps are maximally estimated) and the search was run 3000 iterations for each time step. The action of the player is determined by Eq. (3) with the estimated expected free energy for each action. The precision weight (γ) in Eq. (3) is set to 1. The behavior of an agent controlled with standard active inference (hereafter, standard agent) was also evaluated for comparison. Although the standard agent is also based on the prediction about a future environment using internal models, it does not take into account the expectation from others. The reward (r_{my}) in Eq. (6) is defined by how close the agent is to the destination. If the agent moves closer to the goal, then positive reward is returned, and the expected free energy will be smaller than if it moves away from the destination. Moreover, the empathic player (i.e., empathetic agent) estimates the reward of its future state for others' expectations, r_{oth} in Eq. (6). This reward is defined by the distance between the position of the player and others' expected positions.

From the result for situation (a), the standard agent almost walked straight to the destination. This is because it predicted that it is the highest reward (i.e., lowest free energy) when it walks straight to the destination and simultaneously predicted that another will pass without colliding with it even if it goes straight. On the other hand, the empathetic agent, however, moved in a more circuitous trajectory. From Table 1, the total travel distance of the empathetic player is larger than that of the standard player in situation (a) with regardless of the type of others (selfish or altruistic). On the other hand, the total travel distance of the others is smaller and minimum distance between

Table 1: Quantitative evaluation of behaviors

		Method	Travel distance (player)	Travel distance (Ave. of others)	Minimum distance
situation (a)	selfish	Standard	4.68	5.04	0.51
		Empathic	4.77	4.80	0.81
	alt.	Standard	4.65	5.93	1.20
		Empathic	5.03	5.12	1.51
situation (b)	selfish	Standard	5.35	5.17	0.49
		Empathic	5.45	5.38	0.42
	alt.	Standard	5.35	6.33	1.07
		Empathic	5.30	5.35	1.45
situation (c)	selfish	Standard	-	-	-
		Empathic	4.85	5.12	0.61
	alt.	Standard	4.85	6.45	0.87
		Empathic	5.40	5.74	1.39

the player and others is larger when the player is the empathic agent than those when the player is the standard agent. Comparing the empathic agent and the standard agent when the other is altruistic for situation (a), the travel distance of the empathic agent increased by 8.2% (4.65→5.03), while the travel distance of the other decreased by 13.7% (5.93→5.12) and the minimum distance between the player and the other increased by 25.8% (1.20→1.51). From these quantitative and qualitative results, the empathetic player behaved more socially than the standard player. In other words, the empathetic player behaved in a way that benefited the others, even if it was to its detriment. Because the difference between the empathetic and standard players was only the term of reward for the others (r_{oth}), the difference in behavior stems from this term. Motivation to respond to others' expectations changes the behavior of the player, in other words, the others surrounding the player leads to the player being social. The evaluation also showed that sociality of the player changes in according with the others around it. The total travel distance of the player when others were altruistic was larger by 5.5% (4.77→5.03) than that when others were selfish for situation (a). The difference in others reflected to the player's behavior.

The difference in others also changed the player's behavior for situations (b) and (c). Figure 9 shows the transitions in movement when others were selfish in situation (b). At first, the empathetic player (red) and blue-other started to turn to avoid colliding with the green-other (t1). The player and blue-other continued to turn (t2) and avoided each other by a small margin and passed each other (t3). Finally, the green-other passed the goal (t4). Meanwhile, Figure 10 shows the different transitions in movement when others were altruistic in situation (b). At first, the empathetic player (red) and blue-other started to turn to avoid colliding as similar to the case when others were selfish (t1). The empathic player turned to the left to avoid the blue-other unlike when the others are selfish, and the green-other also started to turn (t2). The all agents moved like in one

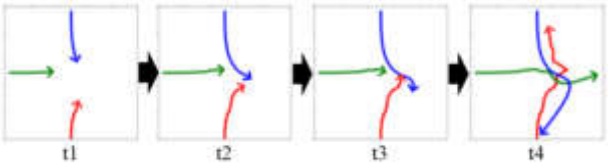


Figure 9: Transition of movements when others were selfish in situation (b).

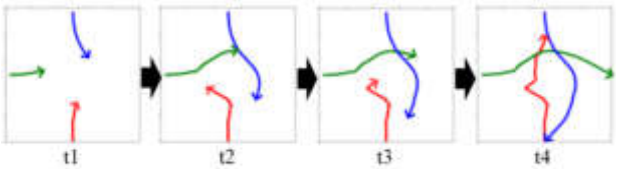


Figure 10: Transition of movements when others were altruistic in situation (b).

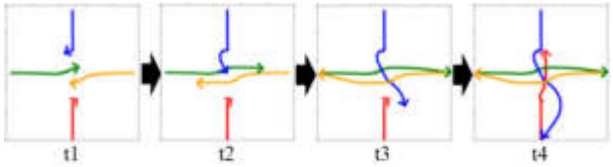


Figure 11: Transition of movements when others were selfish in situation (c).

circle, all together (t3). Finally, the all agents reached to the goals. The similar behaviors were also shown when the others are altruistic in situation (c). The all agents moved like in one circle as shown in Figure 8. On the other hand, the behaviors were more complex when the others are selfish in situation (c). In situation (c), when the player was controlled by the standard active inference and the others were selfish, the player and others collided. Meanwhile, Figure 11 shows the transitions in movement when the others were selfish and player was controlled by the empathic active inference in situation (c). In this situation, the green-other and orange-other first pass each other, while the blue-other and player waited (t1). The blue-other then passed the center point, while the player kept waiting (t2, t3). Finally, the player passed the center point (t4). The behavior of the player is similar to that of the blue-other.

From these results, the behavior of the empathetic agent was influenced by the surrounding others, i.e., it behaved in a similar manner to the surrounding others. This is because the empathetic agent responds to the expectations of others, and the expectation of others is predicted as 'how would I predict the observed other if I were in his situation by using the internal model learned by the other's behavior. Therefore, the empathetic agent behaves like others surrounding it. The empathetic agent can change its sociality for a given situation without manually changing the reward for the situation. Sociality is automatically adjusted to every scene from the behavior of others.

Conclusion and Future Work

Currently, artificial intelligence (AI)/robot ethics is recognized an important issue for integrating AI/robot technology into our society (Jobin et al., 2019). Although many concrete problems such as expandability, transparency, and safety, are being actively studied, one of the most important is to define what an ethical AI/robot is. The proposed method for socially behaved agent can be a solution for this problem in the ethics of AI /robots. From the viewpoint of this paper, an ethical AI/robot is considered to behave on the basis of not only its goal but also others' expectations of it and behaves like the others living around it in society. The mechanism behind the proposed method is empathy towards others. An agent controlled with the proposed method attempts to decrease not only its free energy but also that of others. It is as if human brains and agents are supposed to be shared. These sharing mechanisms generate social behavior. The others around artificial agents are its teachers for sociality. Namely, our behavior is reflected in the ethics of the artificial agent in the proposed method.

For future work, a more feasible training process is evaluated such as training of the policy network ($Q(a_t|s_t)$) and online-training. When the policy network is trained, the action of the empathic agent is directly determined by it. As another future work, we will improve the empathy mechanism. Currently, the empathy mechanism of the proposed method is mainly inspired by affective empathy and is evaluated for control tasks that require action decisions in a short time. However, there is another type of empathy, cognitive empathy, which is related to the ability to infer others' high-order mental states. This type of empathy is important for applying AI/robots to tasks that require more careful consideration such as planning strategies.

Related Work

Active inference has been evaluated for simple control systems (Pio-Lopez et al., 2016; Parr and Friston, 2017; Friston et al., 2018) and for more complex control systems by leveraging advances in deep learning (Ueltzhöffer, 2018; Millidge, 2020; Fountas, 2020; Tschantz, 2020; Catal, 2020; Catal, 2021). These studies mainly assumed a situation in which there is a single control target and no other agents around. Active inference is also discussed for two agents, and the synchronization process of the two agents has been examined (Friston and Frith, 2015). Recently, active inference is also applied to multi-agent cases for discussing emergences of collective intelligences from autonomous behaviors of individuals (Friedman, 2021; Kaufmann, 2021). Moreover, although it is not active inference, inference of others' mental states and sociality of agents for a simple discrete environment has been discussed (Yoshida et al., 2008).

In the field of multi-agent systems, there have been many studies on the emergence of cooperative behavior (Hernandez-Leal et al., 2019). In these multi-agent systems, it is mainly assumed that other agents are also machines, and the agents can explicitly communicate observations, model parameters, and prediction with each other (Foerster et al., 2016; Tampuu et al., 2017; Gupta et al., 2017). In this study, we assumed that the other agents are humans, and there is no explicit communication between an artificial agent and human agents. Similar to the situation discussed in this paper, social robotics

assumes that a robot functioning around humans has sociality and required to infer mental states of humans from their behavior. For example, ProxEmo generates a trajectory for moving based on estimations of others' emotions from their gait behavior (Narayanan et al, 2020). Most of these studies modeled others as different agents from the agent itself, on the other hand, the proposed method models others as being the same as the agent, following the empathy model of the mirror system.

References

Adams, R. A., Shipp, S., & Friston, K. J. (2013). Predictions not commands: active inference in the motor system. Brain Structure and Function, 218(3), 611−643.

Albarracin M, Demekas D, Ramstead MJD, Heins C. (2022). Epistemic communities under active inference. Entropy.; 24(4), 476.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. Journal of the American statistical Association, 112(518), 859-877.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2016). Generating sentences from a continuous space. In International Conference on Computational Natural Language Learning (pp. 10−21). Association for Computational Linguistics.

Çatal, O., Verbelen, T., Nauta, J., De Boom, C., & Dhoedt, B. (2020). Learning perception and planning with deep active inference. In 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 3952−3956). IEEE.

Çatal, O., Verbelen, T., Van de Maele, T., Dhoedt, B., & Safron, A. (2021). Robot navigation as hierarchical active inference. Neural Networks, 142, 192−204.

Cattaneo, L., & Rizzolatti, G. (2009). The mirror neuron system. Archives of neurology, 66(5), 557−560.

Coulom, R. (2006). Efficient selectivity and backup operators in Monte-Carlo tree search. In International Conference on computers and games (pp. 72−83). Springer.

Davis, M. H. (1983). Measuring individual differences in empathy: evidence for a multidimensional approach. Journal of personality and social psychology, 44(1), 113.

Dayan, P., Hinton, G. E., Neal, R. M., & Zemel, R. S. (1995). The helmholtz machine. Neural computation, 7(5), 889−904.

Eisenberg, N., & Miller, P. A. (1987). The relation of empathy to prosocial and related behaviors. Psychological bulletin, 101(1), 91.

Eslami, S. A., Jimenez Rezende, D., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., ... & Hassabis, D. (2018). Neural scene representation and rendering. Science, 360(6394), 1204-1210.

Feldman, H., & Friston, K. (2010). Attention, uncertainty, and free-energy. Frontiers in human neuroscience, 4, 215.

Friedman, D. A., Tschantz, A. D. D., Ramstead, M. J. D., Friston, K., & Constant, A. (2021). Active inferants: The basis for an active inference framework for ant colony behavior. Frontiers in Behavioral Neuroscience, 15, 126.

Friston, K., Kilner, J., & Harrison, L. (2006). A free energy principle for the brain. Journal of physiology-Paris, 100(1−3), 70−87.

Friston, K. (2010a). The free-energy principle: a unified brain theory?. Nature reviews neuroscience, 11(2), 127−138.

Friston, K. J., Daunizeau, J., Kilner, J., & Kiebel, S. J. (2010b). Action and behavior: a free-energy formulation. Biological cybernetics, 102(3), 227−260.

Friston, K., Mattout, J., & Kilner, J. (2011). Action understanding and active inference. Biological cybernetics, 104(1), 137−160.

Friston, K., & Frith, C. (2015). A duet for one. Consciousness and cognition, 36, 390−405.

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., & Pezzulo, G. (2016). Active inference and learning. Neuroscience & Biobehavioral Reviews, 68, 862−879.

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., & Pezzulo, G. (2017). Active inference: a process theory. Neural computation, 29(1), 1−49.

Friston, K. J., Rosch, R., Parr, T., Price, C., & Bowman, H. (2018). Deep temporal models and active inference. Neuroscience & Biobehavioral Reviews, 90, 486−501.

Foerster, J., Assael, I. A., De Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. Advances in neural information processing systems, 29.

Fountas, Z., Sajid, N., Mediano, P., & Friston, K. (2020). Deep active inference agents using Monte-Carlo methods. Advances in neural information processing systems, 33, 11662−11675.

Gazzola, V., Aziz-Zadeh, L., & Keysers, C. (2006). Empathy and the somatotopic auditory mirror system in humans. Current biology, 16(18), 1824−1829.

Gupta, J. K., Egorov, M., & Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. In International conference on autonomous agents and multiagent systems (pp. 66−83). Springer.

Helbing, D., & Molnar, P. (1995). Social force model for pedestrian dynamics. Physical review E, 51(5), 4282.

Hernandez-Leal, P., Kartal, B., & Taylor, M. E. (2019). A survey and critique of multiagent deep reinforcement learning. Autonomous Agents and Multi-Agent Systems, 33(6), 750−797.

Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. Nature Machine Intelligence, 1(9), 389-399.

Kaplan, R., & Friston, K. J. (2018). Planning and navigation as active inference. Biological cybernetics, 112(4), 323−343.

Kaufmann, R., Gupta, P., & Taylor, J. (2021). An active inference model of collective intelligence. *Entropy*, *23*(7), 830.

Keen, S. (2006). A theory of narrative empathy. Narrative, 14(3), 207−236.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Knill, D. C., & Pouget, A. (2004). The Bayesian brain: the role of uncertainty in neural coding and computation. TRENDS in Neurosciences, 27(12), 712−719.

Millidge, B. (2020). Deep active inference as variational policy gradients. Journal of Mathematical Psychology, 96, 102348.

Narayanan, V., Manoghar, B. M., Dorbala, V. S., Manocha, D., & Bera, A. (2020). Proxemo: Gait-based emotion learning and multi-view proxemic fusion for socially-aware robot navigation. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 8200−8207). IEEE.

Parr, T., & Friston, K. J. (2017). Uncertainty, epistemics and active inference. Journal of the Royal Society Interface, 14(136), 20170376.

Pio-Lopez, L., Nizard, A., Friston, K., & Pezzulo, G. (2016). Active inference and robot control: a case study. Journal of The Royal Society Interface, 13(122), 20160616.

Quattrocki, E., & Friston, K. (2014). Autism, oxytocin and interoception. Neuroscience & Biobehavioral Reviews, 47, 410−430.

Rao, R. P., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature neuroscience, 2(1), 79−87.

Rizzolatti, G., & Craighero, L. (2004). The mirror-neuron system. Annu. Rev. Neurosci., 27, 169−192.

Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., ... & Vicente, R. (2017). Multiagent cooperation and competition with deep reinforcement learning. PloS one, 12(4).

Tschantz, A., Baltieri, M., Seth, A. K., & Buckley, C. L. (2020). Scaling active inference. In 2020 International joint Conference on neural networks (pp. 1−8). IEEE.

Ueltzhöffer, K. (2018). Deep active inference. Biological cybernetics, 112(6), 547−573.

Yoshida, W., Dolan, R. J., & Friston, K. J. (2008). Game theory of mind. PLoS computational biology, 4(12).

Q-learning for real time control of heterogeneous microagent collectives

Ana Rubio Denniss[1] , **Laia Freixas Mateu**[1] , **Thomas Gorochowski**[2] and **Sabine Hauert**[1]
[1]Department of Engineering Mathematics, University of Bristol, UK
[2]School of Biological Sciences, University of Bristol, UK
am.rubiodenniss@bristol.ac.uk

Abstract

The effective control of microscopic collectives has many promising applications, from environmental remediation to targeted drug delivery. A key challenge is understanding how to control these agents given their limited programmability, and in many cases heterogeneous dynamics. The ability to learn control strategies in real time could allow for the application of robotics solutions to drive the behaviour of microscopic collectives towards desired outcomes. Here, we demonstrate Q-learning on the closed-loop Dynamic Optical Micro-Environment (DOME) platform to control the motion of light-responsive *Volvox* agents. The results show that Q-learning is efficient in autonomously learning how to reduce the speed of agents on an individual basis.

1 Introduction

The ability to control the behaviour of agents at the microscale or smaller such has implications across fields such as nanomedicine ([Hauert and Bhatia, 2014]) and environmental remediation ([Wang *et al.*, 2019]), with possible agent types including micromotors, nanoparticles and bacterial cells. Exerting control at this scale remains challenging however, due in large part to the simplicity and limited programmability of typical microagents. In this work, an external optical control scheme is used control the microagents, here *Volvox* algae. Machine learning allows for the fine-tuning of the control to each individual *Volvox* in real-time. Through this, individual models can be learnt that enable optimal motion control, in this case learning how to alternate illumination and relaxation periods to stop the motion of individual *Volvox*.

Light is a powerful tool at the microscale, capable of forming and breaking bonds [[Chen *et al.*, 2018]], powering micromotors ([Palagi *et al.*, 2019]), and interacting with light sensitive organisms ([Jékely *et al.*, 2008]). Furthermore, the use of spatially structured light offers interaction with agents independently and in parallel ([Palagi *et al.*, 2019]), making it particularly well suited to the control of collective systems ([Mukherjee *et al.*, 2018; Izquierdo *et al.*, 2018; Schmidt *et al.*, 2019; Deng *et al.*, 2018]). The dynamic nature of light also means that it can be combined with Q-learning to produce rapid and effective and closed-loop control outcomes. This was demonstrated by Muiños-Landin et al., with the use of tabular Q-learning on self-thermophoretic microswimmers to achieve navigation in a noisy, grid-like environment ([Muiños-Landin *et al.*, 2021]). The work presented here similarly uses tabular Q-learning to influence the dynamics of motile microscale agents using optical interactions, however in this case, each agent performs the learning independently, with significant heterogeneity present among the collective of agents owing to their biological nature. Furthermore, the learning and closed-loop optical control were here implemented on a low-cost, open source platform, demonstrating the power of this learning process even in instances with limited computational resources.

Optical control is enacted using the open source DOME platform, a light-weight device which combines digital light projection with microscopy to image a microsystem in real time and provide closed-loop localised light patterning. Given the limited computing power of the DOME, which operates on a Raspberry Pi computer, this work provides an exploration of the potential for the application of Q-learning algorithms in low computational resource environments.

Controlling a complex biological system requires some assumptions to be made about how light affects the *Volvox*. Using this, a finite set of states is defined, where each state measures the amount and frequency of light received by each organism. Although this is not fully representative of the unpredictability of an algae colony, it is good enough for the Q-learning algorithm to run.

Results show that tabular Q-learning allows us to learn how light may be projected onto *Volvox* algae in order to maximally reduce their velocity. The state and action space of a complex biological system is simplified so as to run tabular Q-learning experiment, and the learnt values for individual agents used to achieve herding behaviour in living algae.

2 Methodology

This section introduces the experimental setup for light-based control of *Volvox*, the simulation environment, and Q-learning methodology applied both in simulation and reality.

2.1 Optically controlling *Volvox*

Volvox are a type of green microscopic algae that exhibit phototactic behaviour. They are multicellular organisms, with so-

matic cells that have flagella for locomotion and an eyespot for light perception. These cells allow the *Volvox* to move towards a light source ([Ueki *et al.*, 2010]). This phototactic response is adaptive, meaning that when a *Volvox* comes into contact with light its speed is typically reduced for around 2s before adapting to the new light environment and recovering previous velocity ([Drescher *et al.*, 2010]).

In this work, the light response exhibited by *Volvox* is used as a means to regulate the velocity of individual agents by providing spatially localised illumination. To overcome the adaptive nature of the response, illumination must be provided intermittently rather than as a continuous stimuli. Q-learning is therefore applied as a means to determine the optimum cycle length of illumination and relaxation for each agent that results in the largest velocity reduction. The *Volvox* used here were acquired from Blades Biological UK and are of the species *Volvox aureus*. This algae type is non-harmful and easy to visualise, which alongside the light-responsive property makes it a useful control agent. For all experiments described here, *Volvox* agents were maintained at room temperature in a liquid medium containing water and algae grow solution.

2.2 The DOME

The experimental part of this work was performed using the DOME (Figure 1), an open source platform for the study and engineering of microagent collectives through spatiotemporal illumination ([Denniss *et al.*, 2020]). In this device, a closed-loop control scheme is established by linking a digital light processing unit to real time imaging and image analysis, enabling the optical micro-environment to be shaped around the evolving system dynamics. The DOME has a maximum projection resolution of $30\times30\,\mu$m, and is thus well suited to illumination of individual *Volvox* agents, which are around 350–500 µm in diameter.

2.3 Q-Learning for *Volvox* control

Due to inherent variability of living algae, in order to have an adaptable method of control of the *Volvox*, a reinforcement learning algorithm was required. Because the algae would be controlled using the DOME system, the learning algorithm could not be computationally expensive. Although this could have been circumnavigated by running the algorithm an external computer in communication with the DOME, this work aimed to explore the potential for implementing reinforcement learning in limited resource environments. Additionally, maintaining a self-contained computational set up allows for the possibility of operating the system in enclosed conditions, such as within an incubator for live cell study.

For this reason, tabular Q-learning was chosen, instead of more flexible alternatives such as deep Q-learning.

Tabular Q-learning has the restriction of needing a discrete action space and state space, but biological systems are inherently continuous. Due to this restriction, the action and state space were defined in a discrete way: The action space consisted on two actions, either to illuminate the *Volvox*, or not. The state space needed to represent the amount of light that a *Volvox* had received.

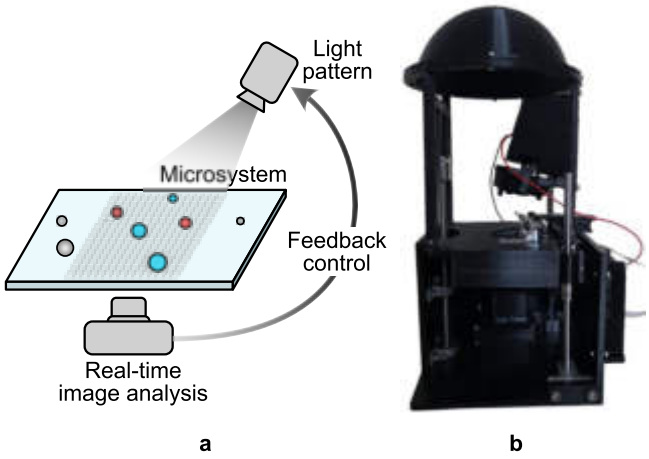


Figure 1: The DOME platform, shown schematically **(a)** and pictorially **(b)**. A digital light processing projection module is used to a controllable pixel grid of light. Real-time image analysis provided by standard light microscopy allow the light patterns to be shaped around the evolving dynamics of the microagent system to achieve closed-loop control.

The *Volvox*'s speed is affected by the amount of light and darkness received. If the state space could be continuous, it would be defined by the amount of time (in milliseconds) that the agent had been illuminated and non-illuminated. Instead of measuring milliseconds, the measurement was discretised using the amount of frames. Since the number of states had to be finite, the number of frames of light or darkness could not grow infinitely. However, observation showed that after 10 frames of either illumination or darkness, the agent's behaviour did not change anymore. Because of this, if an agent hasn't had a change in illumination for over 10 frames, it will be in the same state as if it had had the same illumination for 10 frames.

A state was therefore defined by the number of frames for which the agent was subjected to light, the number of frames for which it was not subjected to light, and the present light value. The present light value was necessary to distinguish between a state that had light on, then light off, and a state that had light off, then light on. Using this method, the total number of states was 242, which was the combination of possible frames on (f_{on}, between 0 and 10) and frames off (f_{off}, between 0 and 10) and the light value (l, either ON or OFF). Note that testing all combinations would not be possible in real time, as each trial is a real-world experiment involving a *Volvox* reaction. Table 1 shows some examples of states with their descriptions. The state index $S(f_{on}, off, l)$ was calculated as follows

$$S(f_{on}, f_{off}, l) = f_{on} + 11 * f_{off} + 121 * l. \qquad (1)$$

The reward for each state was calculated based on the agent's velocity (v) and acceleration (a) at that state. Since the goal was to minimize the magnitude of the velocity, rewards were given for agents with their velocity below a threshold, while accelerating agents were penalized. The direction of

Light description	f_{on}	f_{off}	l	Index
15 frames on, then 3 off	10	3	OFF	43
7 frames on, then 4 off	7	4	OFF	51
4 frames off, then 7 on	7	4	ON	172
7 frames off, then 4 on	4	7	ON	202

Table 1: Examples of states and their indices. The parameters f_{on} and f_{off} represent the number of frames for which an agent had light on and off respectively, while l describes the present light value at a given point.

acceleration and velocity were not considered. Furthermore, because we wanted to minimize the number of light transitions (from on to off and vice-versa), states that had more frames on and off would have higher rewards. The reward function $R(v,a,f_{on},f_{off})$ was defined as

$$R(v,a,f_{on},f_{off}) = \left\{ \begin{array}{ll} f_{on}+f_{off} & |v| < 0.05 \\ -5 & |v| \geq 0.05 \text{ and } a > 0 \\ -1 & |v| \geq 0.05 \end{array} \right\}. \quad (2)$$

At each step, the possible actions that could be performed were to turn the light either on or off for each agent, giving an action space of size 2. Given this, the Q-table was initialized as an empty matrix with $NumStates = 242$ and $NumActions = 2$, where each cell encodes the quality of choosing that action for that state. The Q-table was updated at each step of the learning as the agent explored the environment and different possible states. The Q-learning algorithm stored the previously chosen action ($action$), and the previous state (s), so as to update the reward at the next iteration.

Volvox simulator

The proposed learning methodology was refined in simulation before use in reality. To this end, an agent-based *Volvox* simulator was built in Python to perform rapid iterations on the control algorithms. This simulator replicates the way in which *Volvox* behave in response to light, and was designed such that all code developed was also suitable to run on the DOME platform. *Volvox* agents were modelled based on three assumptions from observation and literature:

- Agent velocity is reduced for period of time when coming into contact with light.
- After a period of time in contact with light, agent velocity recovers.
- The duration of the aforementioned two time periods vary from agent to agent.

The simulated agents follow a straight line, with a probability of them changing direction. This is to replicate the randomness of the *Volvox* movement. In both the simulator and real world experiments, the passage of time was broken up using the number of elapsed camera frames, allowing an otherwise continuous measurement to be discretised. Since each *Volvox* reacts to light in a different way, there exists are a pair of values for the number frames with light on f_{on}, and number of frames with light off f_{off} that if repeated continuously, will keep the agent at its minimum speed. To model this light responsive behaviour of the *Volvox*, a light accumulator model was developed. The emulated agent had two local variables, in which the amount of light (a_L) and darkness (a_D) were stored. These variables were bounded between the values of 1 and 20, and increased exponentially with every new frame of light or darkness. The choice of having an exponential increase was to reflect the fact that at each frame of light that an agent received, the agent would have more capacity to absorb the light, because it would adapt to its new illumination environment. If any of these variables went above the maximum value, it meant that the agent would no longer react to that impulse. In the following equation, t_L indicates the number of consecutive frames of light, and t_D indicates the number of consecutive frames without light.

$$a_L(t_L) = e^{\lambda_L * t_L}$$
$$a_D(t_D) = e^{\lambda_D * t_D}$$

The parameters λ_D and λ_L indicate the rate at which an agent stops reacting to darkness or light, respectively. These values were calculated based on the number of required on and off frames for the agent to stop.

$$\lambda_L = \ln(20) * \frac{1}{f_{on}}$$
$$\lambda_D = \ln(20) * \frac{1}{f_{off}}$$

The code used for this simulator is publicly available online at bitbucket.org/hauertlab.

3 Results

3.1 Simulation

Initially, the *Volvox* simulator described in Section 2.3 was used to develop and test a learning algorithm for reducing agent velocity.

Single agent control

The Q-learning algorithm was initially run on the simulator for a single agent, attempting to reduce agent speed as much as possible for the longest amount of time. The total duration of the experiment was of 10 minutes, representing a typical trial in reality. The simulated agents were programmed to stop when they had received 4 frames of light, and then 3 frames without light. The goal of the Q-learning algorithm was to learn this sequence of actions in order to stop it. The sum of the Q-table over time (Figure 2a) shows that initially, many of the rewards were negative because the agent was penalized. After 5 minutes however, the table values stagnate as the agent has discovered which states will yield the largest rewards. Accordingly, Figure 2b similarly shows that the agent has a variable speed until 5 minutes, at which point the speed stabilizes at low values. The learnt Q-table was analyzed to understand the best actions for the agent, as well as the actions that were chosen most frequently in the last 1000 actions.

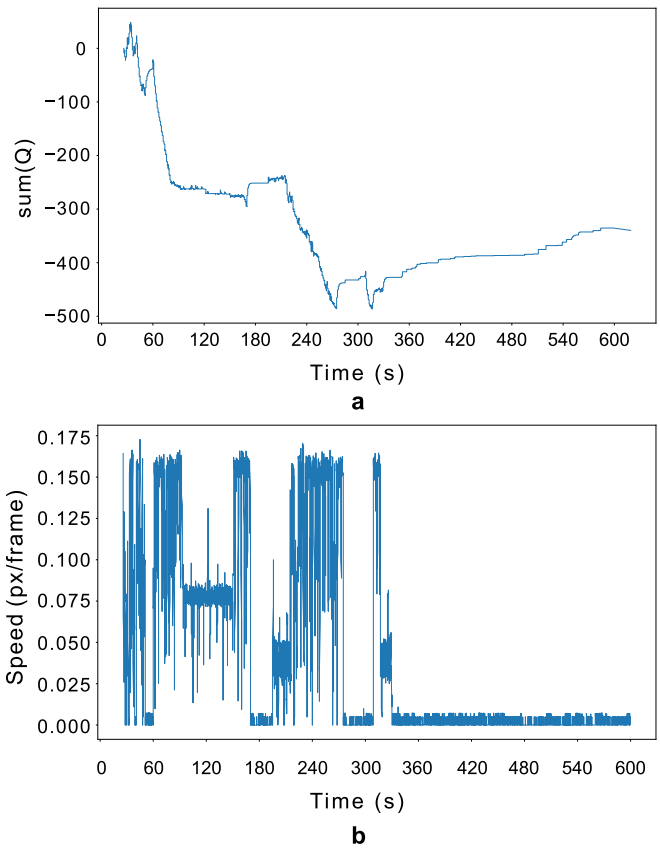


Figure 2: **(a)** Q-table sum converging after 10 minutes for one simulated *Volvox*. **(b)** Speed decreasing over time for one simulated *Volvox*.

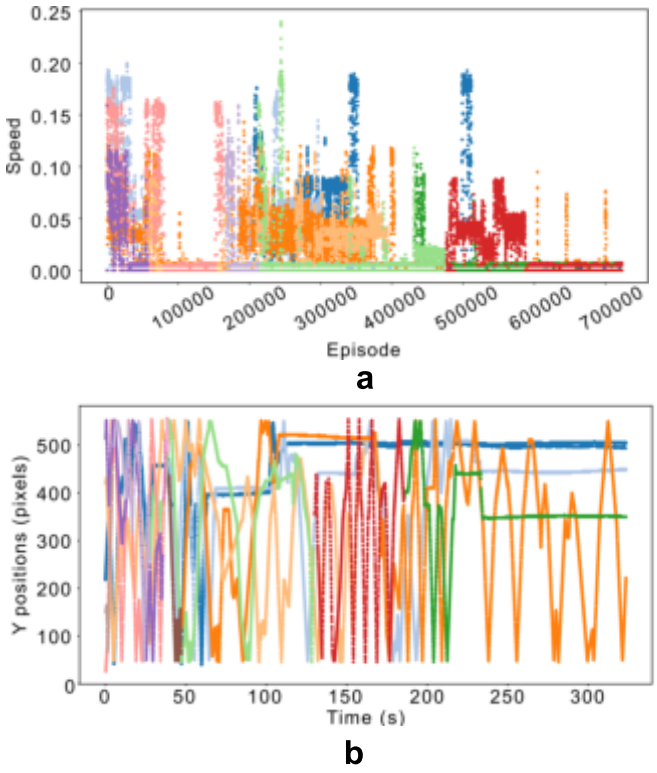


Figure 3: **(a)** Speeds of multiple agents over time, as Q-learning attempts to stop them. Speeds are reduced over time because the learning algorithm discovers optimal parameters for each agent **(b)** Y positions over time of emulated agents during herding, showing that 3 agents stop moving in positions where $Y \geq 328$. For both plots, each colour represents a different agent. It can be seen that one agent (orange) has not yet learned the illumination pattern required to stop moving.

Multi-agent control

The same learning process was repeated for a system of multiple emulated *Volvox* agents, replicating a typical experiment. The plot in Figure 3a shows how the speeds of all the detected agents is reduced over time. Consistently across agents, velocity is ultimately reduced after a period of variation during which the learning process occurs.

The velocity control demonstrated above was then used to explore the possibility of 'herding' the *Volvox* by attempting to gather agents at the lower part of the screen where $Y \geq 328$. The agents positioned in this lower half were controlled using the learning algorithm in an attempt to prevent them moving out of the area, while the other ones were not illuminated, allowing them to move freely. Figure 3b shows that over time, 3 of the agents move to the parts of the screen with high Y coordinate. For the time point shown, the learning algorithm had not yet learnt how to control the agent plotted in orange, hence the position moves up and down the screen continuously while other agents are controlled and kept at the correct position.

3.2 Experimental validation

The algorithms developed in simulation were then implemented on the DOME for experimental validation with real *Volvox* agents, with the aim of showing velocity reduction and, if possible, herding. First, three runs with no Q-learning were performed to provide a comparison point. The movement of *Volvox* was initially observed under no illumination, then under continuous localised illumination. Following this, a blinking illumination experiment was run in which light was provided intermittently at $f_{on} = 1$ and $f_{off} = 1$ to reduce the degree to which *Volvox* were able to adapt to the light without using a complex learning algorithm. For the continuous and blinking illumination, localised light was provided to the *Volvox* agents positioned in the lower half of the sample, or in terms of image coordinate system, where $Y \geq 328$. This aimed to recreate the conditions that led to a herding outcome in simulation. Following the previous experiments, the Q-learning algorithm was run on the *Volvox*, a video of which can be found at youtu.be/Uep5J6RIGHM.

The speed of the *Volvox* was compared across the three conditions to understand how effectively each method achieved velocity reduction. The box plot in Figure 4 was created by averaging the speed of each of the detected agents over each of the experiments. Except in the case of no illumination, only *Volvox* speeds from the illuminated portion of the

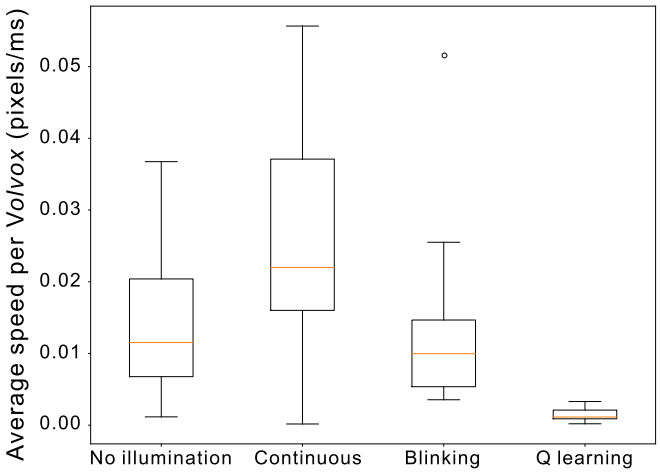


Figure 4: Average speeds in each of the experimental conditions, showing that using Q-learning, there is a lower average speed and less variance in the Y position. For the blinking light condition, an outlier can be seen, represented by a circular point.

sample (continuous or blinking) were considered. The plot shows that the Q-learning algorithm maintains the *Volvox* at a lower speed than the other strategies with significantly less variance. Despite blinking light having a low average speed, there is an outlier that moves at more than 0.05 px/ms, meaning that the algorithm is not good enough to stop all agents. A t-test was performed comparing the Q-learning speed values with each of the other conditions, all three of them showing that the difference was significant ($p < 0.0003$). Similarly, a t-test was done comparing the speeds of the continuous illumination condition with the others, all of which showing that the difference was significant ($p < 0.03$). In all of these conditions, it was possible for agents to stop moving, despite not having changes in illumination, due to the randomness of biological systems.

To better understand the qualitative behaviour of learned strategies, two agents that had been tracked over a long time period were analysed. For both agents, the Q-tables (Figures 5a and b) converged, with the sum of the table stagnating after some time, and agent speeds were kept under 0.05 px/ms during the whole detection. The first part of the corresponding Q-table for agent A (Figure 5c) shows that the reward is maximum when the light was on for 3 frames, and then off for 4 frames. For agent B however, the Q-table outcomes (Figure 5d) differ from that of agent A, suggesting that each *Volvox* reacts to light in a different manner.

During the learning phase, the algorithm tried different combinations of actions depending on the state of the Q-table at that moment. In Figure 5e, eight consecutive frames from the camera are shown. In all of them, two agents are detected, but each is illuminated at a different rate to keep the speed as low as possible, with continuous analysis of the speed to update the rewards of each state. The agent at the top left starts illuminated ($t = 0s$), then light is turned off for 1 frame (until $t = 0.3s$), then back on for 2 frames (until $t = 0.9s$), then off for 2 frames (until $t = 1.5s$), on for 1 frame, and off again.

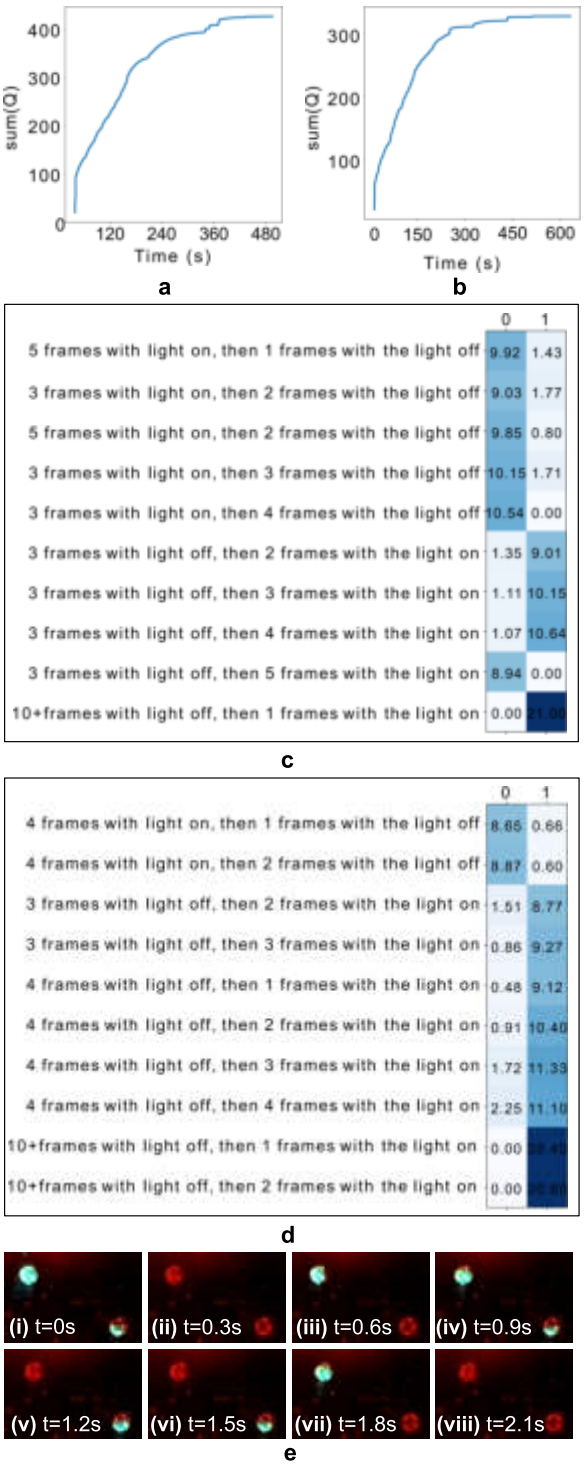


Figure 5: **(a)** Evolution of the sum of the Q-table for agent A, showing stagnation. **(b)** Evolution of the sum of the Q-table for agent B, showing stagnation. **(C)** Partial Q-table values for agent A, showing that after 3 frames off, then 5 on, the best action is to turn the light off again. **(d)** Partial Q-table values for agent B, showing that after 4 frames with light off, then 4 frameswith light on, the best action is to keep the light on. **(e)** Two *Volvox* illuminated at different rates using Q-learning.

The agent at the bottom right also starts illuminated ($t = 0s$), then light is turned off for two frames (until $t = 0.6s$), then is turned on for three frames (until $t = 1.5s$), then back off for two more frames (until $t = 2.1s$).

4 Discussion

Q-learning is suited in an unknown environment, and can be used to understand the best way to control microscopic agents in a way that is independent from the agent type and its characteristics. This means that, although *Volox* algae are employed here as a model microagent, the tools developed could be adapted to suit other stimuli responsive agents. As with most organic systems, and many inorganic, large degree of heterogeneity exists in the stimuli-responses of *Volox* agents even within the same population. For this reason, learning individually tuned parameters is powerful in achieving precise control.

In both simulation and experiment, Q-learning proved a successful strategy for reducing the velocity of emulated and real *Volvox* agents respectively. The results of the Q-tables differ between simulation (Figure 2a) and experiment (Figures 5c and d) in that values are larger in real-world experiments than in simulation. This is possibly because the simulator assumed that stopping the *Volvox* would be more difficult than it turned out to be, meaning that agents were less likely to be rewarded. Despite this, both simulated and real results found that agents do not only have one combination of light on and off that is valid. Instead, there are many combinations that allow to keep a low speed, and Q-learning successfully learns these. In the experimental section, it was found that Q-learning provided the most efficient strategy for slowing the *Volvox* when compared to standard continuous or intermittent illumination patterns. This was demonstrated by the lower average speed, and smaller variance in speeds seen in Figure 4.

In addition to the ability to regulate velocity, herding of agents into a particular area was also explored. In simulation, this was found to work well, with the Y positions of all but one agent being inside the chosen half by the end of the control period (Figure 3b). In experiments, preliminary results suggest that the same may be possible. Figure 6 shows the average position for a collection of *Volvox* agents over time, where the illuminated section of the space was switched three times during the experiment. In all cases, the switch occurred when most or all agents had moved to the illuminated region. This experiment suggests that using Q-learning, light could be used to gather agents in an area of the sample despite not being able to directly control their direction. However, due to the small agent number (4-5) and the large variance in the natural movement of *Volvox*, even in the absence of light, further experiments are required to verify this outcome.

The potential to control an entire microagent collective in parallel could also allow for exploration of swarm behaviours and control strategies at the microscale. Broadly, a swarm system is one in which agents are able to collectively perform actions that are beyond the capabilities of an individual, typically facilitated through local interactions ([Brambilla *et al.*, 2013]). Unlike in macroscale swarm engineering, microagents cannot be straightforwardly programmed with interaction rules, rather agents must typically interact through physical means such as chemical signalling. This requires the design and fabrication of highly complex agents, something that can be costly and time-consuming. Given this, the development of generic microswarm control strategies could be crucial in informing the design of these intelligent micro/nanoagents, efficiently directing the production process to best suit a given application.

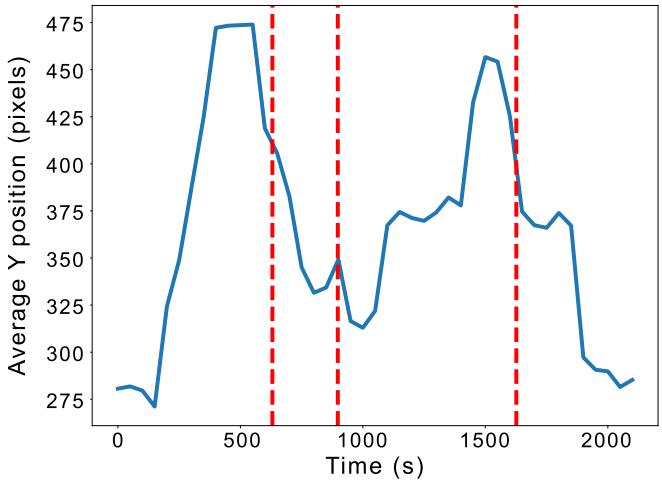


Figure 6: The average Y position of all *Volvox* agents over time the Q-learning algorithm alternates between illuminating the top and bottom of the screen. Red dashed lines indicate the time point at which the illumination half was switched.

Overall, this work suggests that tabular Q-learning is a useful tool to efficiently learn microagent control strategies in real-time on platforms with limited computational resources, as it is not as computationally expensive as other more complex learning algorithms, and is suited to an unknown environment. The use here of the DOME as a low-cost, open source platform is also significant in widening accessibility to similar control techniques.

5 Future work

The primary goal of this project was to demonstrate the potential of using Q-learning to control individual agents within a complex, living biological system in real time. In line with this, a control strategy developed using tabular Q-learning was found to be effective compared to non-learning baselines in regulating the motion of these biological agents. In future work, comparison with alternative learning-based algorithms would be informative in optimising the control process. Furthermore, due to the on-board, lightweight nature of the computational strategy developed here there is the potential to apply similar schemes to live mammalian cell environments through operation inside an incubator environment, or even as a miniaturised wearable medical device. This could allow exploration of learning based control for cell collectives such as tumours or healing wounds.

References

[Brambilla *et al.*, 2013] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[Chen *et al.*, 2018] Yihuang Chen, Zewei Wang, Yanjie He, Young Jun Yoon, Jaehan Jung, Guangzhao Zhang, and Zhiqun Lin. Light-enabled reversible self-assembly and tunable optical properties of stable hairy nanoparticles. *Proceedings of the National Academy of Sciences*, 115(7):E1391–E1400, 2018.

[Deng *et al.*, 2018] Zhuoyi Deng, Fangzhi Mou, Shaowen Tang, Leilei Xu, Ming Luo, and Jianguo Guan. Swarming and collective migration of micromotors under near infrared light. *Applied Materials Today*, 13:45–53, 2018.

[Denniss *et al.*, 2020] Ana Rubio Denniss, Thomas E Gorochowski, and Sabine Hauert. An open platform for high-resolution light-based control of microscopic collectives. *bioRxiv*, 2020.

[Drescher *et al.*, 2010] Knut Drescher, Raymond E Goldstein, and Idan Tuval. Fidelity of adaptive phototaxis. *Proceedings of the National Academy of Sciences*, 107(25):11171–11176, 2010.

[Hauert and Bhatia, 2014] Sabine Hauert and Sangeeta N Bhatia. Mechanisms of cooperation in cancer nanomedicine: towards systems nanotechnology. *Trends in Biotechnology*, 32(9):448–455, 2014.

[Izquierdo *et al.*, 2018] Emiliano Izquierdo, Theresa Quinkler, and Stefano De Renzis. Guided morphogenesis through optogenetic activation of rho signalling during early drosophila embryogenesis. *Nature communications*, 9(1):1–13, 2018.

[Jékely *et al.*, 2008] Gáspár Jékely, Julien Colombelli, Harald Hausen, Keren Guy, Ernst Stelzer, François Nédélec, and Detlev Arendt. Mechanism of phototaxis in marine zooplankton. *Nature*, 456(7220):395–399, 2008.

[Muiños-Landin *et al.*, 2021] S. Muiños-Landin, A. Fischer, V. Holubec, and F. Cichos. Reinforcement learning with artificial microswimmers. *Science Robotics*, 6(52), 2021.

[Mukherjee *et al.*, 2018] Manisha Mukherjee, Yidan Hu, Chuan Hao Tan, Scott A Rice, and Bin Cao. Engineering a light-responsive, quorum quenching biofilm to mitigate biofouling on water purification membranes. *Science advances*, 4(12):eaau1459, 2018.

[Palagi *et al.*, 2019] Stefano Palagi, Dhruv P Singh, and Peer Fischer. Light-controlled micromotors and soft microrobots. *Advanced Optical Materials*, 7(16):1900370, 2019.

[Schmidt *et al.*, 2019] Falko Schmidt, Benno Liebchen, Hartmut Löwen, and Giovanni Volpe. Light-controlled assembly of active colloidal molecules. *The Journal of chemical physics*, 150(9):094905, 2019.

[Ueki *et al.*, 2010] Noriko Ueki, Shigeru Matsunaga, Isao Inouye, and Armin Hallmann. How 5000 independent rowers coordinate their strokes in order to row into the sunlight: Phototaxis in the multicellular green alga volvox. *BMC biology*, 8(1):1–21, 2010.

[Wang *et al.*, 2019] Linlin Wang, Andrea Kaeppler, Dieter Fischer, and Juliane Simmchen. Photocatalytic tio2 micromotors for removal of microplastics and suspended matter. *ACS applied materials & interfaces*, 11(36):32937–32944, 2019.

Acknowledgments

This work was supported by an EPSRC DTP scholarship (A.R.D) and the EPSRC TAS pump priming fund (A.R.D, S.H., T.E.G.).

Multi-Objective Evolutionary Game Theory: A case study in cancer therapy

Lukas Bostelmann-Arp[1], Sanaz Mostaghim[1], Andreas Braun[2] and Thomas Tüting[2]

[1] Faculty of Computer Science, Otto-von-Guericke University Magdeburg
[2] Department of Dermatology, Otto-von-Guericke University Magdeburg
Lukas.Bostelmann-Arp@ovgu.de

Abstract

In this paper, we introduce an early concept of using multi-objective optimization to study various emerging strategies in evolutionary game theory and show its application in a case study. We aim to analyze the emergent behavior when changing the game's environment through optimization. The multi-objective approach allows looking at the results of each model evaluation from different points of view. For the realization, we suggest the use of a multi-agent model to compute the outcome of a game. Such a model allows modeling even complex interrelationships and can be used as input to multi-objective optimization algorithms. Finally, we demonstrate a use case by optimizing therapy plans for melanoma through the incorporation of medications into a multi-agent model of concurring cell populations in the tumor micro environment.

Introduction

In evolutionary game theory (EGT), regular game theory (GT) is used in an evolutionary context that represents a sequence of interactions among the participants. The focus lies on populations instead of individual players. Fitness plays a fundamental role in population survival through reproduction (Sandholm, 2020). The individuals who form the populations do not actively reason about their decisions, but inherit strategies through genetic operators. Therefore, the strategies or behaviors that emerge are of great interest. Analogous to the Nash equilibrium in GT, evolutionarily stable strategies (ESS) are especially important. Once adopted by a population in a specific environment, a set of ESS cannot be substituted by a novel set of strategies.

However, the emerging set of strategies depend on the game's rules. Therefore, altering those rules might allow exploring additional emerging sets of strategies. This raises the question of how the different populations adapt to the changed environment. Note that the environment is not changed during the interactions, but only for different runs of the model. Further, the properties of the rule changes and their effects on the emerging set of strategies can be evaluated based on various, sometimes conflicting, criteria.

Therefore, the goal of this paper is to propose the incorporation of a multi-objective optimization scheme into the EGT to allow analyzing several sets of strategies which are the results of an optimization problem.

For this purpose, we study agent-based models (ABMs) as a mode to simulate the evolutionary game. In such models, autonomous agents with basic rules interact with each other. The result of these interactions is usually an emerging behavior. An overview regarding agent-based modelling and its tools is provided by Abar et al. (2017). Agent-based models, or multi-agent systems in general, require the use of a robust optimization algorithm, especially in case of stochastically determined decisions of agents. There are several works in the literature that use Evolutionary Algorithms for this purpose, e.g., Moya et al. (2021) studied the use of different evolutionary multi-objective algorithms (EMOAs) for the automatic calibration of the model itself.

Moreover, there is another reason why agent-based models can be the connecting piece between EGT and evolutionary optimization algorithms: Adami et al. (2016) compared standard EGT methods to the results of ABMs and concluded that the latter can be beneficial in the prediction of more realistic scenarios, where current mathematical tools reach their limits. Nonetheless, they and others (Hilbe and Traulsen, 2016) make clear, that both have their purpose and that mathematical methods in particular can help keeping agent-based systems from becoming too arbitrary. Regarding the computational side, a framework has been proposed (Izquierdo et al., 2019) that allows the simulation of evolutionary game dynamics through agent-based systems. Based on this, there are already a few works applied to real scenarios, e.g., Coelho and Ralha (2022) studied the land use, respectively coverage, by simulating interacting human entities.

Multi-Objective EGT (MO-EGT)

In MO-EGT, we introduce two counterparts, namely ABMs based on EGT, and the multi-objective optimization algorithm (MOA). The ABM contains several populations which interact according to the rules defined by a payoff matrix defined in an EGT framework. Our goal, in using MOA, is to obtain several optimal values for the pay-off matrix by optimizing two conflicting functions. The first function is

meant to regulate the population size and the second function describes the cost in the change of environment or payoff matrix, respectively. These can be best described using our case study.

Here we consider the interaction of multiple cell populations in the tumor micro environment. This is modeled using an ABM in an EGT framework. Within this environment, many physical restrictions, but also inter cell conflicts, exert pressure on cancer cells. These influences can be seen as selective forces, and therefore justify the interpretation of the tumor progression as an evolutionary game (Wölfl et al., 2021). We define the interactions between the normal cells and tumor cells using a pay-off matrix. In the optimization process, the focus lies on therapeutic intervention, which influence the values in the pay-off matrix. For example, one value of the payoff matrix indicates the damage caused by an immune cell to a tumor cell, considering the current drug concentration at the site of interaction. A current research question deals with finding adequate sequences and intervals for various medications for therapeutic interventions. This is necessary to prevent the formation of resistances due to the cellular plasticity given by the evolutionary characteristics of cancer cells. The goal is then to keep a stable population of tumor cells, or even a declining one. Therefore, the number of surviving tumor cells is counted for the first objective function $f1$. The second objective function $f2$ for MOA concerns the cost of the therapy.

In our experiments, we take the NSGA-II (Deb et al., 2002) as the multi-objective optimization algorithm. Further, a population size of 52 with a termination criterion of 50 generations is used. Note, that the term population is used here in the context of the EMOA and is different from the population in an ABM. Regarding the variation operators, a simple one-point crossover is used together with a Gaussian mutation. Both operators are extended to handle a variable number of drug administrations, as this, together with the dosage, is part of the optimization problem.

The underlying ABM consists of five different cell populations. The most important one, the tumor cells, feature the ability to change their state. This behavior, called differentiation in a biological sense, represents the evolutionary aspect of this game.

Results

Regarding the experiment, the optimization algorithm was run five times to mitigate stochastic effects contained in the algorithm itself. Since a simulation also contains probabilistic decisions, each evaluation was repeated 16 times before computing the fitness values from the average.

The combined Pareto front is shown in Figure 1 for the two objectives $f1$ and $f2$ introduced earlier. Since the tumor started with 100 tumor cells, the results that end up with around 100 tumor cells can be considered stable regarding the size of the tumor. This is marked by the dashed vertical line in the plot. Nonetheless, the algorithm was able to produce a well distributed front with different compromises between the two objectives. Each of those solutions can be analyzed regarding the composition of the tumor and the cell's strategy or behavior, respectively. This information can help the physician better understand the tumor's response to the corresponding therapy. The next step, deals with the decision-making process. As an evolutionary multi-objective algorithm already presents a wide range of possible solutions, it is easier for the decision-maker to choose a final one, than designing one from scratch. This results in a higher confidence towards the decision. However, selecting among the solutions requires additional information about the patient which exceeds the tumor composition data required for the simulation initialization. That includes, for example, information regarding the lifestyle or psychological condition.

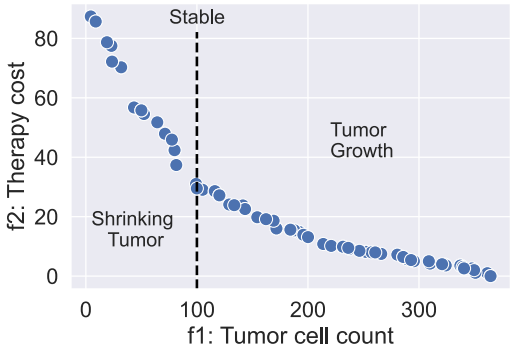


Figure 1: Combined Pareto front

Conclusion and Future Work

In this paper, we proposed a workflow and illustrated, how multi-objective optimization can be implemented into EGT. So far, EGT is being used to compute a certain optimal strategy, in most cases, the so called evolutionarily stable strategy. Here, we study several sets of strategies which are the results of a multi-objective optimization problem that altered the game's rules.

While the presented case study showed that our proposed method works, there are still some gaps to fill. This concerns above all the explainability. As the original EGT is based on mathematical tools, its correctness can be proofed. While it may be possible for simple agent-based models that apply the same rules used in EGT, it is hard for more complex ones and especially for the optimization algorithm itself. In addition, the possible outcomes heavily depend on the implementation of the variation operators as well as the individual of the EMOA. Nonetheless, the multi-objective approach allows analyzing and comparing different results and emerged strategies of an evolutionary game, which were created by optimizing rule changes.

References

Abar, S., Theodoropoulos, G. K., Lemarinier, P., and O'Hare, G. M. (2017). Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33.

Adami, C., Schossau, J., and Hintze, A. (2016). Evolutionary game theory using agent-based methods. *Physics of Life Reviews*, 19:1–26.

Coelho, C. G. C. and Ralha, C. G. (2022). Mase-egti: An agent-based simulator for environmental land change. *Environmental Modelling & Software*, 147:105252.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197.

Hilbe, C. and Traulsen, A. (2016). Only the combination of mathematics and agent-based simulations can leverage the full potential of evolutionary modeling. *Physics of Life Reviews*, 19:29–31.

Izquierdo, L. R., Izquierdo, S. S., and Sandholm, W. H. (2019). An introduction to abed: Agent-based simulation of evolutionary game dynamics. *Games and Economic Behavior*, 118:434–462.

Moya, I., Chica, M., and Cordon, O. (2021). Evolutionary multi-objective optimization for automatic agent-based model calibration: A comparative study. *IEEE Access*, 9:55284–55299. Interesting.

Sandholm, W. H. (2020). *Evolutionary Game Theory*, pages 573–608. Springer US.

Wölfl, B., te Rietmole, H., Salvioli, M., Kaznatcheev, A., Thuijsman, F., Brown, J. S., Burgering, B., and Staňková, K. (2021). The contribution of evolutionary game theory to understanding and treating cancer. *Dynamic Games and Applications*.

The evolution of adaptive phenotypic plasticity stabilizes populations against environmental fluctuations

Alexander Lalejini[1], Austin J. Ferguson[2], Nkrumah A. Grant[3], and Charles Ofria[2]

[1] University of Michigan, Ann Arbor, MI, USA
[2] Michigan State University, East Lansing, MI, USA
[3] University of Idaho, Moscow, ID, USA
lalejini@umich.edu

Introduction

Environmental fluctuations are ubiquitous in nature. Populations have evolved a wide range of strategies to cope with environmental change, including periodic migration (Winger et al., 2019), bet-hedging (Beaumont et al., 2009), adaptive tracking (Barrett and Schluter, 2008), and phenotypic plasticity (Ghalambor et al., 2007). The particular mechanisms that evolve in response to environmental fluctuations profoundly influence subsequent evolution.

Here, we summarize our recently published study using digital evolution experiments to investigate the evolutionary consequences of adaptive phenotypic plasticity (Lalejini et al., 2021). Phenotypic plasticity is the capacity for a single genotype to produce alternate phenotypes depending on environmental conditions (West-Eberhard, 2003). Such plasticity is controlled by genes whose expression is coupled to one or more environmental signals, which may be either biotic or abiotic.

Phenotypic plasticity's effect on evolutionary change has long interested evolutionary biologists because of its role in generating phenotypic variance (Gibert et al., 2019). However, the effects of plasticity on adaptive evolution have been disputed, as few studies have been able to observe both the *de novo* evolution of plasticity and subsequent evolutionary change in natural populations (Ghalambor et al., 2007; Wund, 2012; Forsman, 2015; Ghalambor et al., 2015; Hendry, 2016). Adaptive plasticity has been predicted to both promote and constrain evolutionary change depending on the genetic and environmental contexts (*e.g.*, Lalejini et al. 2021, Figure 1).

In (Lalejini et al., 2021), we used populations of self-replicating computer programs ("digital organisms") to investigate the evolutionary consequences of adaptive plasticity in a cyclically changing environment. We examined the evolutionary histories of both adaptively plastic and non-plastic populations of digital organisms in order to ask: (1) Does adaptive plasticity promote or constrain evolutionary change? (2) Are plastic populations better able to evolve and then maintain novel traits? And, (3) how does adaptive plasticity affect the potential for maladaptive alleles to accumulate in evolving genomes? Note that this study does not focus on *how* phenotypic plasticity evolves initially (see Clune et al., 2007; Lalejini and Ofria, 2016), but instead, we focus on how plasticity influences subsequent evolutionary dynamics after it evolves.

Experimental results

We conducted three evolution experiments using the Avida Digital Evolution Platform (Ofria et al., 2009) in order to examine the effects of adaptive plasticity on subsequent genomic and phenotypic change, the capacity to evolve and then maintain novel traits, and the accumulation of deleterious alleles. We divided each experiment into two phases. In the first phase, we preconditioned sets of founder organisms with differing plastic or non-plastic adaptations, and in phase two, we examined the subsequent evolution of populations founded with organisms from phase one (Lalejini et al., 2021, Figure 2). For each experiment, we compared the evolutionary outcomes of populations evolved under three treatments: (1) a PLASTIC treatment where the environment fluctuates, and digital organisms can sense the current environmental state; (2) a NON-PLASTIC treatment where the environment fluctuates, but organisms can not sense the current environment; and (3) a STATIC control where organisms evolve in a constant environment. See (Lalejini et al., 2021, Section 2) for complete methods.

Adaptive plasticity slows evolutionary change in fluctuating environments

In our first experiment, we tested whether the evolution of adaptive plasticity constrained or promoted subsequent evolution, comparing the number of selective sweeps as well as the frequency of both genotypic and phenotypic changes along lineages evolved under each treatment. We found strong evidence that adaptive plasticity slows evolutionary change in fluctuating environments (Lalejini et al., 2021, Figures 3, 4). PLASTIC populations where adaptive plasticity evolved underwent fewer total selective sweeps and fewer total genetic and phenotypic changes relative to NON-PLASTIC populations evolving under identical environmen-

tal conditions. NON-PLASTIC populations relied on *de novo* mutations to adapt to each environmental fluctuation, which repeatedly drive the fixation of mutations that align an organism's phenotype to the new conditions. PLASTIC populations, however, could use sensing mechanisms to dynamically align their phenotype with the environment.

Adaptive plasticity improves novel function retention in fluctuating environments

While adaptive plasticity constrains the rate of evolution in fluctuating environments, it is unclear how this dynamic influences the evolution of novel functions. Based on relative rates of evolutionary change, we might expect NON-PLASTIC populations to be able to evolve more novel functions than PLASTIC or STATIC populations. But how much of the evolutionary change in NON-PLASTIC populations is useful for exploring novel regions of the fitness landscape versus continuously revisiting the same regions?

In our second experiment, we compared the capacity for novel functions to evolve during phase two of each treatment (Figure 1). We found that organisms evolved under PLASTIC and STATIC conditions performed a greater number of novel functions than those evolved under the NON-PLASTIC treatment. This result, however, was not due to PLASTIC and STATIC populations *discovering* more novel functions. Instead, the evolutionary stability of PLASTIC and STATIC populations allowed for better retention of any evolved novel functions. Indeed, lineages evolved under NON-PLASTIC conditions exhibited a substantially greater number of loss-of-novel-function mutations than lineages evolved under PLASTIC or STATIC conditions.

Lineages with plasticity express fewer deleterious functions in fluctuating environments

Plasticity allows for genetic variation to accumulate in unexpressed genomic regions, which can lead to the fixation of deleterious alleles in PLASTIC populations. However, in our previous experiment, we observed higher rates of novel function loss in NON-PLASTIC lineages, indicating that they may be more susceptible to deleterious mutations. In our third experiment, we investigated whether the evolution of adaptive plasticity can increase the incidence of deleterious function performance. We found that the lineages of organisms evolved under the NON-PLASTIC treatment exhibited both greater totals and higher rates of deleterious function acquisition than that of PLASTIC lineages (Lalejini et al., 2021, Figure 8).

Conclusion

In general, we found that the evolution of adaptive phenotypic plasticity shifted evolutionary dynamics to be more similar to that of populations evolving in a static environment than to non-plastic populations evolving in an identical fluctuating environment. Our work lays the groundwork for how digital evolution experiments can be used to study the evolutionary consequences of phenotypic plasticity in a range of contexts. Future work will build on these experiments, investigating the evolutionary consequences of maladaptive and non-adaptive plasticity as well as expanding the types of environmental change studied.

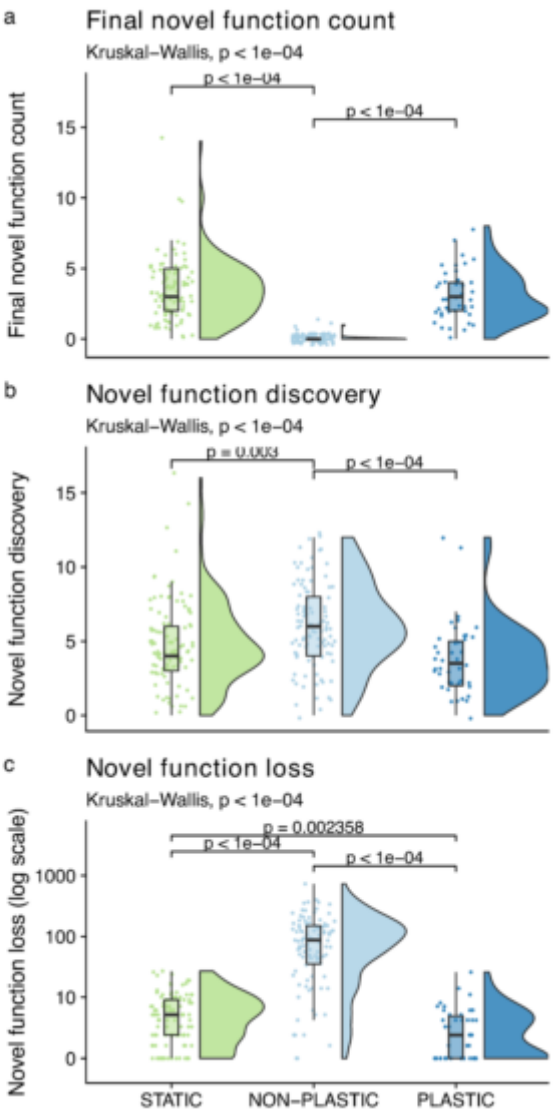


Figure 1: **Novel function evolution.** Raincloud plots of (a) final novel function count, (b) novel function discovery, and (c) novel function loss. See (Lalejini et al., 2021, Table 1) for further descriptions of each metric. Each plot is annotated with statistically significant comparisons (Bonferroni-corrected pairwise Wilcoxon rank-sum tests). Figure adapted from (Lalejini et al., 2021).

References

Barrett, R. and Schluter, D. (2008). Adaptation from standing genetic variation. *Trends in Ecology & Evolution*, 23(1):38–44.

Beaumont, H. J. E., Gallie, J., Kost, C., Ferguson, G. C., and Rainey, P. B. (2009). Experimental evolution of bet hedging. *Nature*, 462(7269):90–93.

Clune, J., Ofria, C., and Pennock, R. T. (2007). Investigating the Emergence of Phenotypic Plasticity in Evolving Digital Organisms. In Almeida e Costa, F., Rocha, L. M., Costa, E., Harvey, I., and Coutinho, A., editors, *Advances in Artificial Life*, volume 4648, pages 74–83. Springer Berlin Heidelberg, Berlin, Heidelberg.

Forsman, A. (2015). Rethinking phenotypic plasticity and its consequences for individuals, populations and species. *Heredity*, 115(4):276–284.

Ghalambor, C. K., Hoke, K. L., Ruell, E. W., Fischer, E. K., Reznick, D. N., and Hughes, K. A. (2015). Non-adaptive plasticity potentiates rapid adaptive evolution of gene expression in nature. *Nature*, 525(7569):372–375.

Ghalambor, C. K., McKay, J. K., Carroll, S. P., and Reznick, D. N. (2007). Adaptive versus non-adaptive phenotypic plasticity and the potential for contemporary adaptation in new environments. *Functional Ecology*, 21(3):394–407.

Gibert, P., Debat, V., and Ghalambor, C. K. (2019). Phenotypic plasticity, global change, and the speed of adaptive evolution. *Current Opinion in Insect Science*, 35:34–40.

Hendry, A. P. (2016). Key Questions on the Role of Phenotypic Plasticity in Eco-Evolutionary Dynamics. *Journal of Heredity*, 107(1):25–41.

Lalejini, A., Ferguson, A. J., Grant, N. A., and Ofria, C. (2021). Adaptive phenotypic plasticity stabilizes evolution in fluctuating environments. *Frontiers in Ecology and Evolution*, 9:550.

Lalejini, A. and Ofria, C. (2016). The Evolutionary Origins of Phenotypic Plasticity. In *Proceedings of the Artificial Life Conference 2016*, pages 372–379, Cancun, Mexico. MIT Press.

Ofria, C., Bryson, D. M., and Wilke, C. O. (2009). Avida: A Software Platform for Research in Computational Evolutionary Biology. In Komosinski, M. and Adamatzky, A., editors, *Artificial Life Models in Software*, pages 3–35. Springer London, London.

West-Eberhard, M. J. (2003). *Developmental Plasticity and Evolution.* Oxford University Press.

Winger, B. M., Auteri, G. G., Pegan, T. M., and Weeks, B. C. (2019). A long winter for the Red Queen: rethinking the evolution of seasonal migration. *Biological Reviews*, 94(3):737–752.

Wund, M. A. (2012). Assessing the Impacts of Phenotypic Plasticity on Evolution. *Integrative and Comparative Biology*, 52(1):5–15.

Emergence of Novelty in Evolutionary Algorithms

David Herel [1], Dominika Zogatova[1], Matej Kripner[2] and Tomas Mikolov[1]

[1] CIIRC, CTU in Prague, 160 00 Dejvice
[2] Charles University, 121 16 Nové Město
david.herel@seznam.cz

Abstract

One of the main problems of evolutionary algorithms is the convergence of the population to local minima. In this paper, we explore techniques that can avoid this problem by encouraging a diverse behavior of the agents through a shared reward system. The rewards are randomly distributed in the environment, and the agents are only rewarded for collecting them first. This leads to an emergence of a novel behavior of the agents. We introduce our approach to the maze problem and compare it to the previously proposed solution, denoted as Novelty Search (Lehman and Stanley, 2011a). We find that our solution leads to an improved performance while being significantly simpler. Building on that, we generalize the problem and apply our approach to a more advanced set of tasks, Atari Games, where we observe a similar performance quality with much less computational power needed.

Introduction

In nature, individuals of one population compete with each other for survival, and only the fittest ones can pass their genes on to the next generation. Similarly, in evolutionary algorithms, it is the agents who are rewarded for being the fittest. In order to realize the rewards, an objective function is needed to evaluate the individual solutions based on their proximity to the optimum (Goldberg and Holland, 1988) and to guide us through the search space to a valid solution. Even though evolutionary algorithms have been proven successful in numerous optimization tasks (Goldberg and Holland, 1988), the deceptiveness of the environment is rather problematic. In reality, increasing fitness does not always reveal the best path and can mislead us from finding the global optimum (Goldberg, 1987).

Aiming to solve this problem, many algorithms were developed (Hutter and Legg, 2006)(Basu and Bhatia, 2006) that significantly reduce the deceptiveness. But the underlying problem remains – the objective function may still misdirect the search toward a dead end. Actually, finding a solution to this problem requires thinking counter-intuitively and abandoning the objective completely, as Novelty Search does (Lehman and Stanley, 2011a). We can think of the Novelty Search as a divergent search technique applied to an evolutionary computation, which also introduces a new reward system. Instead of rewarding agents based on their proximity to the goal, agents are rewarded for being different from others in the population. In comparison with the traditional objective-based evolutionary processes, Novelty Search algorithms have demonstrated their efficiency in many highly deceptive problems (Lehman and Stanley, 2011a).

In this paper, we propose our own search method called Sugar Search, which aims to reproduce or surpass Novelty Search results without explicitly defining novelty. Thus, the behavioral novelty will emerge as a by-product of the environment and objective function definition. Building on the qualities of the Novelty Search (Lehman and Stanley, 2011a), we reproduce this method on the maze problem, which is identical to the mazes presented in (Lehman and Stanley, 2011a), and use it for comparison with our Sugar Search technique. Following the proof of concept and given the competitive results, we also conduct experiments with different reward densities and a combined fitness-based and Sugar Search approach in the maze environments.

In the second part of this paper, the main objective is to present a more complex problem that our search technique can be applied to. For this purpose, we present Sugar Search in the Atari games environment. We compare the results of our method to the fitness-based approaches, which have shown good results on some Atari games (Bellemare et al., 2012), as well as some reinforcement algorithms like DQN (Mnih et al., 2013) and AC3 (Mnih et al., 2016). Because of the challenges that arise with agent allocation in such a complex environment, we also generalize our approach to this problem.

Related work

There have been several approaches to solve the problem of local minima by encouraging novelty. One of the most well-known ones is the Novelty Search algorithm (Lehman and Stanley, 2011a). It ignores the objective and instead optimizes how unique the individual's behavior is. It