University of Trento - Italy

Department of Information Engineering and Computer Science

Ph.D. Degree in
Information and Communication Technology

FINAL DISSERTATION

HUMAN-AWARE ROBOTICS: PREDICT, ASSIST, AND PLAN FOR SEAMLESS INTERACTION

Advisors
Prof. Luigi Palopoli
Prof. Daniele Fontanelli

Ph.D. Candidate
Placido Falqueto

Academic Year 2023–2024

Alla ricerca, che risponde alle mie domande, e ne genera altre mille di nuove.

Abstract

This thesis explores the intersection of human-centric design and robotics, focusing on enhancing human-robot interaction through predictive, assistive, and planning methodologies. The research is structured around three key objectives: predicting human motion in shared spaces using semantic maps and advanced neural architectures; designing adaptive shared control frameworks for assistive robotic devices like the FriWalk; and developing human-aware motion planning for collaborative robotic manipulators such as the UR5e.

Employing tools like Vision Transformers (ViTs) and Masked Autoencoders, the study achieves high-accuracy predictions of human trajectories and occupancy priors, which are essential for robots operating in dynamic environments. The shared control framework balances safety and user autonomy by dynamically adjusting robotic assistance based on behavioural analysis. For robotic manipulators, real-time human motion predictions integrate into trajectory planning algorithms, ensuring seamless and safe collaboration in mixed environments.

The findings advance the field of human-aware robotics, contributing to safer, more intuitive interactions between humans and robots. This work lays the groundwork for future assistive technology and collaborative robotics developments, aiming to enhance safety, efficiency, and user autonomy in diverse applications.

Keywords

Human-Robot Interaction (HRI), Predictive Human Motion, Adaptive Shared Control, Human-Aware Motion Planning, Assistive Robotics

Contents

Chapter 1

Introduction

The integration of robots into human environments represents a significant milestone in robotics, moving beyond traditional industrial applications to systems capable of close collaboration with humans. This transformation is driven by the need for robots to not only perform tasks efficiently but also interact intuitively and safely with people. Robots operating in dynamic, human-centric environments must adapt to the complexities of human behaviour, anticipate actions, and respond proactively to ensure seamless coexistence. These requirements are pivotal for applications such as navigation in crowded public spaces, assistive technologies for individuals with physical or cognitive impairments, and collaborative work environments in manufacturing or service industries.

A key capability for robots operating in such environments is the ability to anticipate human behaviour. Predicting where humans are likely to move, where they might pause, and their probable paths allow robots to plan their own trajectories accordingly. This capability is particularly useful for autonomous navigation, where robots must ensure their paths are efficient and non-invasive. To achieve this, semantic maps provide a powerful tool by contextualizing the environment into meaningful areas, such as roads, walkways, or obstacles, which indicate how humans are likely to interact with their surroundings. Building on these semantic maps, this thesis explores how robots can use advanced neural architectures, such as Vision Transformers (ViTs), to predict human occupancy priors and trajectories. These predictions enable robots to navigate seamlessly in shared spaces,

laying the groundwork for safer and more intuitive interactions in environments like public buildings and urban areas.

Understanding human behaviour not only aids in autonomous navigation but also opens opportunities for assistive applications, where robots directly support individuals in performing everyday activities. This thesis explores the case of the *FriWalk*, a robotic rollator designed to assist individuals with mild cognitive impairments. In such applications, modeling typical human motion patterns within specific environments becomes crucial. By observing how people usually move, turn, or pause in familiar spaces, the system generates behavioural maps that classify motion into recognizable patterns, such as left turns, right turns, or straight paths. These maps allow the robot to dynamically adjust its level of intervention, providing guidance only when the user deviates significantly from expected behaviours. Such an approach ensures that the user retains autonomy while receiving assistance when necessary, promoting confidence and independence. This paradigm of shared control (section 4.1.1) not only enhances safety but also empowers individuals, reducing the stress and fatigue associated with navigating complex environments.

While predicting behaviour and offering shared control are essential, robots operating in close proximity to humans must also plan their actions proactively to avoid interruptions or collisions. This becomes particularly important in human-robot collaborative settings, such as workplaces or homes, where robotic manipulators must adjust their trajectories based on human movements. Anticipating human skeleton motion, which includes predicting the position and orientation of limbs in real time, enables robots to make informed decisions about their paths. By integrating motion prediction techniques such as Gaussian Mixture Models (GMM), Dynamic Time Warping (DTW), and TransFusion, this thesis presents a framework that incorporates human motion predictions into a time-variant A* planning algorithm. This allows robots to proactively choose their trajectories, avoiding unnecessary stops or disruptions and ensuring smooth, uninterrupted collaboration. By equipping robots with these proactive planning capabilities, the work aims to transform them into reliable and intuitive partners in dynamic, shared environments.

In summary, this thesis addresses the challenge of enabling robots to coexist

with humans in shared spaces by combining motion prediction, adaptive assistance, and proactive planning. By predicting human behaviour through semantic maps, classifying motion patterns to aid assistive tasks, and integrating real-time motion predictions into planning algorithms, this work advances the field of human-aware robotics. The ultimate goal is to create robotic systems that not only operate safely and efficiently but also interact seamlessly and intuitively with the people around them.

1.1 Objectives of the Research

The primary objectives of this thesis are as follows:

- To develop a predictive model for human motion in shared environments using semantic maps and state-of-the-art neural architectures.
- To design a shared control framework for assistive robotic walkers that enhances user autonomy and comfort.
- To create a human-aware motion planning framework for robotic manipulators, integrating real-time human motion predictions with adaptive trajectory planning.

1.2 Contributions of the Thesis

This thesis makes the following contributions:

- A novel Vision Transformer-based model for predicting human motion from semantic maps, achieving high accuracy and real-time performance.
- A shared control framework for assistive robotic walkers, enabling dynamic adjustment of assistance based on behavioural maps and user actions.
- An integrated human-aware motion planning framework for robotic manipulators, combining advanced prediction techniques with adaptive trajectory planning to ensure safety and efficiency in collaborative settings.

1.3 Structure of the Thesis

Each chapter builds upon the foundational concepts and systematically develops methodologies and frameworks for safer and more intuitive human-robot interactions. The chapters are organized as follows:

- **Chapter 2: Background and Robotic Platforms.** This chapter provides an overview of the main research pillars addressed in the thesis: vision-based prior prediction, adaptive shared control, and human-aware motion planning. It presents the motivation for these areas and describes the two primary robotic platforms used in this work—the *FriWalk* robotic rollator and the UR5e collaborative manipulator— highlighting their features and suitability for human-robot interaction research. The chapter also outlines the role of the Robot Operating System (ROS) in integrating perception, control, and planning components. Detailed reviews of the state of the art for each research pillar are provided in the dedicated subsequent chapters.

- **Chapter 3: Semantic Map-Based Human Motion Prediction.** This chapter presents the methodology for predicting human motion based on semantic map representations, particularly focusing on how advanced neural architectures like Vision Transformers can leverage these maps. It discusses the theoretical foundation of semantic maps and their utility in encoding environmental context for prediction tasks. Experimental results demonstrate the effectiveness of the proposed approach in capturing and predicting human motion patterns in various environments.

- **Chapter 4: Shared Control for Robotic Walkers.** This chapter focuses on the development and validation of a shared control framework for assistive robotic walkers, specifically the *FriWalk*. It begins with a background on assistive robotics and shared control paradigms, followed by a detailed description of the framework's design, implementation, and the use of behavioural maps. Experimental results are analyzed to evaluate the framework's performance in improving user comfort and autonomy during assisted locomotion.

- **Chapter 5: Human-Aware Motion Planning for Robot Manipulators.** This chapter introduces a novel motion planning framework specifically for robotic manipulators operating in close proximity to humans. It incorporates human-awareness by integrating real-time human motion predictions (using techniques like GMM, DTW, and TransFusion) into a time-variant A* planning algorithm. The framework enables robots to plan motions that are safe, efficient, proactive, and predictable from the human perspective. Evaluations demonstrate its applicability in collaborative scenarios.

- **Chapter 6: Conclusions and Future Perspectives.** The final chapter synthesizes the findings of this thesis, reflecting on the contributions made to the field of human-aware robotics regarding prediction, shared control, and planning. It also discusses the limitations of the current work and outlines future research directions, emphasizing opportunities for advancing the integration of human-awareness in robotic systems for improved coexistence.

Through this structured approach, the thesis addresses critical challenges in human-aware robotics, proposing innovative solutions that enhance the safety, intuitiveness, and collaboration in human-robot interactions.

Chapter 2

Background and Robotic Platforms

Human-Robot Interaction (HRI) is increasingly central to robotics research, driven by the need for systems that can safely and effectively collaborate with or assist humans in complex, dynamic environments. This thesis investigates advanced techniques to improve HRI in both assistive and collaborative settings, focusing on three core research pillars:

- Vision-Based Prior Prediction
- Adaptive Shared Control
- Human-Aware Motion Planning

A common element enabling the integration and development on these platforms is the Robot Operating System (ROS) [Open Robotics, 2014]. ROS is a widely adopted, open-source middleware framework in robotics, providing a standardized set of libraries, tools, and conventions for building complex robot applications across diverse hardware. It facilitates crucial functionalities such as hardware abstraction, low-level device control, message-passing between processes (nodes), package management, and visualization, significantly accelerating development and promoting code reuse. Both the *FriWalk* and the UR5e leverage this framework to manage their software architectures and integrate the various per-

ceptual, control, and planning components investigated in this thesis, as detailed further in their respective chapters.

Addressing the challenges within these pillars requires robust robotic platforms capable of perception, interaction, and adaptation. This research utilizes two distinct systems: the *FriWalk* robotic rollator and the UR5e collaborative manipulator, described below.

Figure 2.1: The *FriWalk* robotic walker.

- **The *FriWalk* Robotic Rollator:** An assistive mobility platform designed to support individuals with impairments (Fig 2.1). The *FriWalk* robotic rollator was selected as the primary platform for investigating assistive mobility and adaptive shared control. Its development stems from the need for intelligent walkers that actively assist users with mobility impairments. Compared to passive walkers, *FriWalk* integrates active sensing (cameras, force sensors on handles) and computation, enabling environment perception and user intent inference via interaction forces. When contrasted with other research platforms like the GuideCane [Borenstein and Ulrich, 1997] (focused on obstacle avoidance using ultrasound), the Fraunhofer IPA Care-O-Bot/Rollator-Walker (often demonstrated in logistics or structured environments [Graf et al., 2009]), or the RoRo robotic rollator [Bieber et al.,

Figure 2.2: The *FriWalk* robot. The red circle highlights the DC motors mounted on the front wheels and their encoders.

2019], *FriWalk* emphasizes user-adaptive shared control informed by both direct force input and vision-based environmental understanding for proactive assistance in complex, everyday spaces. Its specific integration of handle force sensors for shared control input distinguishes it from platforms relying solely on reactive obstacle avoidance or simpler assistance models. Furthermore, its development within a modern ROS 2 framework facilitates the integration of the planning modules (Chapter 4) developed in this thesis. These specific capabilities make *FriWalk* particularly suitable for evaluating the proposed algorithms aimed at enhancing user safety, comfort, and autonomy in challenging navigation scenarios.

Hardware Overview

The *FriWalk* robotic rollator integrates advanced sensors, actuators, and computational modules to assist users effectively in navigation tasks. The system supports interaction paradigms that ensure user comfort and safety by balancing human and robotic contributions to shared tasks.

From a hardware perspective, the *FriWalk* employs front-mounted DC motors for steering control (Fig. 2.2), while the rear wheels are also equipped with motors. However, for the specific use case of shared control explored in Chapter 4, the rear wheels are used passively, with encoders installed to measure their angular motion [Farina et al., 2017]. The device incorporates advanced sensory hardware, including incremental and absolute encoders, along with a 2D camera system that uses visual markers such as ArUco tags for precise localization. This hardware architecture enables localization accuracy within 20 cm [Nazemzadeh et al., 2017], a critical feature for navigation in cluttered indoor environments.

The hardware architecture is managed by two primary computational platforms: an Intel NUC and a BeagleBone Black. The NUC serves as the high-performance computing unit, managing computationally intensive tasks such as real-time trajectory planning, learning-based algorithms, and integration with the ROS 2 middleware. The BeagleBone Black is designated for low-level hardware interfacing, utilizing ZeroMQ to expose services that control motors and encoders. ZeroMQ communicates with the CAN bus to monitor and manage the status of all connected devices. This dual-computer architecture enables efficient distribution of computational loads, ensuring real-time responsiveness and high operational reliability.

Software Architecture

The *FriWalk* utilizes a modular software architecture built upon the Robot Operating System (ROS), specifically ROS 2 running on the NUC, to integrate trajectory generation algorithms, shared control frameworks, and adaptive parameter tuning. As introduced earlier in this chapter, ROS provides standard interfaces and tools that facilitate seamless interaction between high-level cognitive tasks and low-level hardware management.

The BeagleBone Black's software stack manages the low-level hardware controls (motors, encoders) via the CAN bus. ZeroMQ serves as a lightweight messaging protocol to bridge these hardware-specific services with the ROS 2

ecosystem running on the NUC (which also manages the camera). The NUC uses ROS 2 interfaces (topics, services, actions) to manage communication and data flow, converting ZeroMQ messages as needed. This middleware-based approach allows researchers and developers to interact with the robotic platform through standard ROS 2 tools and languages.

Custom ROS 2 nodes were developed in both C++ and Python, enabling direct interfacing with the robotic platform's capabilities. Additionally, MATLAB and Simulink scripts were employed to connect to the ROS 2 middleware, supporting advanced algorithmic development and testing. This cross-platform compatibility via ROS highlights the system's flexibility and its potential for integration into diverse research and development workflows.

Localization System and Real-Time Obstacle Detection The localization system combines sensory data from encoders and cameras to achieve precise positioning. By leveraging visual markers such as ArUco tags, the *FriWalk* can navigate complex environments with high accuracy. This system is complemented by real-time obstacle detection capabilities. Sophisticated algorithms interpret user intentions and adapt the device's behaviour to ensure safety and autonomy [d'Addato et al., 2024].

- **The UR5e Collaborative Manipulator:** A six-degree-of-freedom robotic arm designed for tasks alongside humans (Fig 2.3). The UR5e collaborative robot was selected for its versatility, established safety features (detailed in the Hardware Overview), and extensive research community support for HRI tasks. When compared to traditional industrial robots lacking inherent safety mechanisms, and other prominent cobots (e.g., KUKA LBR iiwa [Bischoff et al., 2010], Franka Emika Panda [Gaz et al., 2019], FANUC CRX [FANUC Corporation, Accessed: 2024-07-04]), the UR5e strikes a balance in terms of sensitivity, payload, reach, programming ease, and research accessibility. It was chosen for its suitability for typical research assembly/interaction tasks, relative programming ease, and, critically, its wide adoption in the research community with robust ROS 1 and ROS 2 support [Universal Robots A/S and FZI Forschungszentrum Informatik, Accessed: 2024-07-04]. This exten-

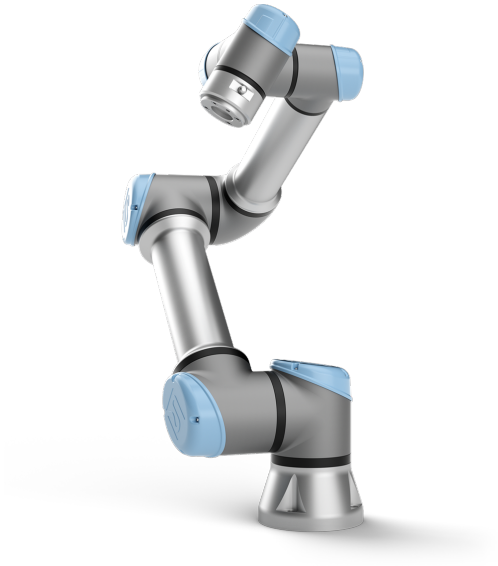


Figure 2.3: The UR5e collaborative manipulator.

sive support system facilitates the integration and testing of advanced algorithms, such as the human motion prediction and reactive planning methods explored in Chapter 5. Its reliable performance make it a suitable testbed for validating HRI algorithms requiring close proximity interaction.

Hardware Overview

The UR5e is a six-degree-of-freedom (6-DoF) collaborative robotic manipulator designed for flexible deployment in industrial and assistive tasks. Its lightweight and adaptable structure allows for seamless human-robot collaboration. The UR5e features a lightweight design with a payload capacity of 5 kg and a reach of 850 mm [Universal Robots Inc, 2018], making it suitable for tasks requiring high precision, such as assembly, pick-and-place operations, and the collaborative manufacturing scenarios investigated in Chapter 5.

One of the defining hardware features of the UR5e is its integrated force/-torque sensor, embedded at the wrist, which provides real-time feedback. This capability enables the robot to handle delicate tasks and detect external forces, enhancing its ability to adapt to dynamic environments and

ensure safe interaction with humans.

The UR5e's emphasis on safety is further reflected in its compliance with ISO 10218 and ISO/TS 15066 standards for collaborative robots [International Organization for Standardization, 2016]. These certifications, along with features like configurable safety planes, ensure the robot's operation alongside humans without the need for physical barriers, making it particularly suitable for close-proximity human-robot collaboration. These features make the UR5e an optimal choice for applications involving shared workspaces, such as collaborative assembly processes.

Software Architecture

The UR5e offers a highly programmable interface supported by Universal Robots' Polyscope software and the UR+ ecosystem, streamlining development. Crucially for research and integration, the UR5e also supports the Robot Operating System (ROS).

As introduced in this chapter, ROS provides standardized tools for hardware abstraction, device control, message passing, and package management. Leveraging ROS allows researchers to focus on developing innovative solutions (like the perception, prediction, and planning algorithms in Chapter 5) while benefiting from a robust and scalable software infrastructure. The extensive ROS ecosystem facilitates rapid prototyping and experimentation with advanced motion planning frameworks (e.g., PRM) relevant to Chapter 5.

While this chapter provides a high-level motivation for the chosen research areas and platforms, a comprehensive review of the State-of-the-Art (SotA) for each pillar is crucial for contextualizing the contributions of this thesis. Due to the distinct nature and depth required for each topic, the detailed literature reviews, discussion of existing methods, identification of specific research gaps, and justification for the algorithms developed or employed in this thesis are presented within the dedicated chapters that follow.

Chapter 3

Semantic Map-Based Human Motion Prediction

3.1 Introduction and Problem Formulation

Understanding and predicting human motion is a critical requirement for mobile robots operating in human-populated environments. This task involves evaluating human occupancy in different areas and forecasting their likely motion directions, which are essential for efficient and safe navigation. While humans instinctively use environmental cues to anticipate motion patterns, robots require sophisticated algorithms to achieve similar results.

This chapter tackles the challenge of enabling robots to anticipate human behaviour by leveraging rich environmental context, moving beyond purely reactive strategies. Central to providing this context is the concept of a semantic map. A semantic map represents an environment not only in terms of its physical geometry (such as an occupancy grid map, which primarily indicates free versus occupied space), but also by associating meaningful, human-understandable labels or categories with different spatial regions or objects. For instance, areas might be labelled as 'sidewalk', 'road', 'crosswalk', 'building', 'vegetation', 'doorway', or 'stairs'. This layer of semantic information allows a robot to reason about the environment in a way more akin to human understanding, inferring affordances (e.g., 'sidewalk' is walkable, 'road' is drivable but potentially dangerous for pedestrians)

and context (e.g., 'doorway' implies potential entry/exit points). Semantic maps are typically constructed by processing raw sensor data. A common approach in modern robotics is to apply deep learning-based semantic segmentation algorithms to data streams from sensors like cameras (providing visual information) or LiDAR (providing 3D point clouds). These algorithms classify each pixel or point in the sensor data into predefined semantic categories. The classified data is then often projected or fused onto a 2D or 3D map structure to create the final semantic map. Alternatively, especially for smaller or well-structured environments, semantic maps can also be created or refined through manual annotation, where a human operator labels different parts of a pre-existing geometric map. By providing this rich contextual understanding, semantic maps serve as crucial input for tasks such as context-aware navigation, human behaviour prediction (as explored in this chapter), and more complex human-robot interaction scenarios.

3.1.1 Problem Formulation

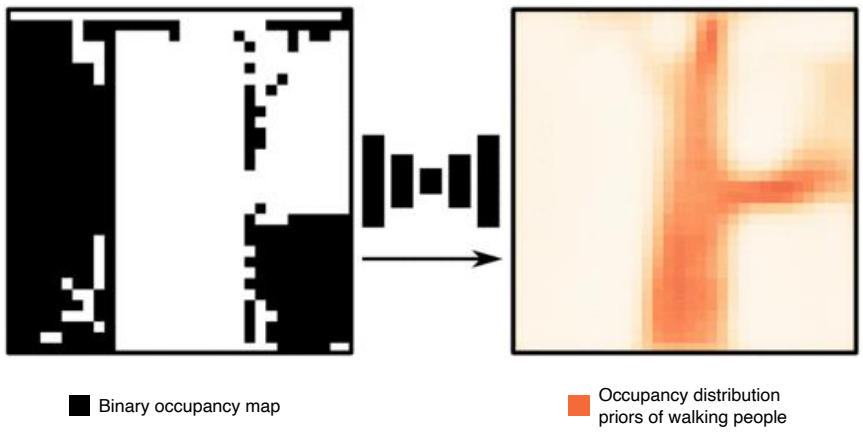


Figure 3.1: Example of a predicted prior map showing occupancy likelihood based on the environment's static features. Image from Doellinger et al. [2018].

Given these informative static semantic maps, the specific problem addressed

in this chapter is the inference of *prior distributions* of human behaviour—such as typical occupancy patterns (Fig. 3.1), common stopping locations, and average velocities—directly from the map data itself. Accurate prediction of these priors is crucial for mobile robots navigating human-populated spaces, as it informs safer and more efficient path planning and interaction strategies. While humans intuitively perform such reasoning, enabling robots to achieve similar predictive capabilities purely from the static map presents a significant challenge, especially as many existing approaches focus on predicting individual trajectories and lack the ability to effectively leverage this rich semantic context.

3.1.2 Contributions and Proposed Approach Overview

This chapter introduces `semapp2`, a novel approach based on Vision Transformers (ViTs) specifically designed to predict multiple human behaviour priors (occupancy, stops, velocities) directly from static semantic maps. The key contributions are:

- A ViT-based architecture (`semapp2` and its MAE variant `MAE-semapp2`) that effectively leverages global context via self-attention for improved prior prediction compared to CNN-based methods.
- Extension of prior prediction beyond occupancy to include stop locations and average velocities, providing richer behavioural insights.
- Comprehensive experimental validation on the Stanford Drone Dataset, including ablation studies and comparison against the state-of-the-art `semapp` baseline, demonstrating the benefits of the proposed approach.

We detail the proposed architecture in Section 3.3, and present the experimental results in Section 3.4.

The prediction of human behaviour priors from static semantic maps, as explored in this chapter, connects to and complements other research pillars of this thesis, contributing to the overarching goal of enhancing human-robot interaction across different contexts and platforms.

- **Connection to Adaptive Shared Control (Chapter 4):** The semantic priors predicted in this chapter have potential applications in adaptive shared control for assistive robotics, such as the *FriWalk* platform. A detailed discussion of how these priors can enhance user guidance and arbitration of control authority is provided in Chapter 4.

- **Overall Thesis Contribution:** The integration of environment-driven priors and motion-driven forecasting is discussed in the Conclusions (Chapter 6).

3.2 Challenges and Related Work

Human-aware navigation has been extensively studied, with recent surveys highlighting advances and ongoing challenges in modeling and predicting human behaviour for social robot navigation [Singamaneni et al., 2024].

Predicting human behaviour priors solely from static environmental maps presents several challenges. These include capturing the complex interplay between environmental structure and human movement patterns, understanding the long-range spatial dependencies that influence navigation choices (e.g., how the layout of distant intersections affects flow on a path), and handling the inherent variability and stochasticity in human behaviour. Furthermore, traditional geometric maps often lack the rich contextual information needed to infer likely behaviours accurately.

Anticipating human motion intentions represents a longstanding challenge, demanding a nuanced comprehension of social dynamics [Mavrogiannis et al., 2023]. As described in the survey [Rudenko et al., 2020], the modelling of human motion trajectories can be categorized through the representation of the underlying causes, including physics-based, pattern-based, and planning-based methods. While trajectory prediction focuses on individual paths, predicting the prior distribution of occupancy offers a different perspective, analyzing the environment itself to understand typical behaviours.

Predicting prior occupancy distribution, rather than individual trajectories, proves valuable in extrapolating contextual information and enriching our understanding of a location [Kaleci et al., 2020]. This involves analyzing the environment

itself (e.g., via semantic maps), offering insights into typical human behaviours within that context. This differentiation enhances our ability to anticipate future events and make informed decisions based primarily on environmental information. Despite the evident importance of this approach, there is a distinct gap in existing literature dedicated to the direct prediction of priors (like occupancy, stops, velocities) purely from static maps, which this chapter aims to address.

Semantic maps are essential for real-time navigation and understanding human-populated environments. These maps provide a contextual framework for robots, enabling adaptive and context-aware navigation in complex, dynamic spaces. By identifying key features such as obstacles, walkways, seating areas, and intersections, semantic mapping enriches the robot's understanding of its environment, providing valuable input for prior prediction models.

In the following, we review several vision-based approaches from the literature that have been proposed for interpreting environmental data to predict human behaviour.

3.2.1 Convolutional Neural Networks (CNNs)

CNNs have been pivotal in advancing computer vision [Krizhevsky et al., 2012], and their success in image classification and semantic segmentation makes them relevant for interpreting map data. However, segmentation itself remains challenging due to factors like intra-class variations and occlusions [Minaee et al., 2021]. These challenges extend to inferring map priors. Doellinger et al. [2018] used a Fully Convolutional Network architecture, based on [Jégou et al., 2017] with dense blocks [Huang et al., 2018], to predict average occupancy maps of walking humans directly from static grid maps. Rudenko et al. [2021] (whose work, 'semapp', serves as a baseline in this chapter) extended this by using semantic maps as input instead of plain grid maps, improving predictions by leveraging richer contextual information. Specifically, `semapp` is defined as follows:

- **Input:** A static semantic map of the environment, represented as a multi-channel grid where each channel corresponds to a semantic class (e.g., walkway, road, building). The original work utilized 9 distinct semantic classes.

- **Output:** A predicted prior occupancy distribution map, usually visualized as a heatmap, indicating the likelihood of human presence across different areas of the environment based on the semantic context.
- **Architecture:** `semapp` employs a Fully Convolutional Network (FCN) architecture. FCNs utilize stacked convolutional layers to process the spatial information present in the input semantic map, making them effective at capturing local features and textures associated with occupancy patterns.

While effective at capturing local features, CNNs like `semapp` can sometimes struggle to model the long-range spatial dependencies inherent in navigation tasks, motivating the exploration of architectures with global receptive fields, such as the one proposed in this chapter.

3.2.2 Vision Transformers (ViTs)

Vision Transformers (ViTs) [Dosovitskiy et al., 2021] are gaining traction in computer vision, leveraging self-attention mechanisms to capture global contextual information effectively, potentially overcoming local biases seen in CNNs. Self-attention allows the model to weigh the importance of different parts of the input map when making predictions for a specific location, effectively modelling long-range dependencies. For tasks like segmentation or prior map generation, which require dense, pixel-level outputs, encoder-decoder architectures are common. Strudel et al. [2021] reframed segmentation as a sequence-to-sequence problem using a transformer, outperforming CNNs by better exploiting global context [Vaswani et al., 2023]. This chapter explores the potential of ViTs for inferring multiple priors (occupancy, stops, velocities) from semantic maps.

3.2.3 Masked Autoencoders (MAEs)

Inspired by masked language modeling (e.g., BERT [Devlin et al., 2019]), Masked Autoencoders (MAEs) [He et al., 2021] offer a self-supervised approach for learning visual representations. By masking a significant portion of input image patches and training the model to reconstruct the missing pixels, MAEs learn rich, semantically meaningful latent representations without relying heavily on labeled

data. This self-supervised pre-training has shown excellent generalization capabilities, potentially enabling models to learn robust representations of environmental structure relevant to human behaviour even with limited explicit trajectory data. We investigate a variation of our ViT approach using an MAE architecture to investigate whether this pre-training strategy enhances the prediction of occupancy priors from semantic maps [Falqueto, Sanfeliu, et al., 2024].

In this context, the pre-training phase involves training the MAE on the generic task of reconstructing masked image patches from large amounts of unlabeled data. This enables the model to acquire general-purpose visual features, which can subsequently be fine-tuned for the specific task of predicting occupancy priors from semantic maps. By leveraging this two-stage training process, we aim to determine whether MAE-based pre-training leads to more robust and transferable representations for our downstream application.

3.2.4 Inference and Applications of Priors

The prediction of priors involves constructing probabilistic maps (e.g., heatmaps) that encapsulate the likelihood of human presence, stops, or typical velocities across specific regions of an environment [Doellinger et al., 2018; Rudenko et al., 2021]. Such maps are instrumental in enabling robots to navigate effectively within human-populated environments.

Prior maps leverage various inputs, primarily semantic information about the environment as explored in this chapter, but can also incorporate historical trajectory data when available. Techniques range from direct prediction using neural networks (CNNs, ViTs as discussed) to synthesizing predictions based on environmental geometry (e.g., using clothoid paths to generate likely routes [Arechavaleta et al., 2008a; Bevilacqua et al., 2020]).

These priors are crucial in applications like robot-assisted navigation, particularly for systems like the *FriWalk* (discussed in Chapter 4), which use them to guide users safely and comfortably [Falqueto, Antonucci, et al., 2024; Nazemzadeh et al., 2017; Palopoli et al., 2015]. By predicting likely user paths or areas of high congestion/stops based on the map, the robot can proactively adjust guidance or plan safer routes, integrating these priors with real-time observations, enabling

robust autonomy.

3.3 Proposed Method: The Semapp2 Architecture

To address the challenge of predicting multiple human motion priors (occupancy, stops, velocities) directly from static semantic maps, and to potentially overcome limitations of purely local feature extraction, this chapter presents `semapp2`, a novel approach based on Vision Transformers (ViTs) [Falqueto, Sanfeliu, et al., 2024]. Formally, the proposed method has the following characteristics:

- **Input:** `semapp2` processes fixed-size crops extracted from a multi-channel static semantic map. In our final configuration, we utilize maps with 13 distinct semantic classes to provide rich environmental context.
- **Output:** The model predicts corresponding crops for various prior distributions relevant to human behaviour. This includes not only occupancy likelihood but also stop probability distributions and average velocity magnitude maps.
- **Architecture:** We employ a ViT-based autoencoder architecture, illustrated conceptually in Figure 3.2. A more detailed schematic showing the flow from input semantic crop processing (13 classes) through the encoder and decoder Transformer blocks to the predicted prior map output (occupancy, stops, or velocity) is provided in Figure 3.3. The encoder leverages self-attention for global context to compress the input into a latent representation, and a lightweight decoder reconstructs the target prior crop. We also investigate a Masked Autoencoder (MAE) variant, `MAE-semapp2`. The high-level concept is shown in Figure 3.4, while Figure 3.5 provides a detailed schematic, highlighting the masking step and the use of visible segments plus mask tokens as input to the decoder for reconstruction from partially masked semantic inputs.

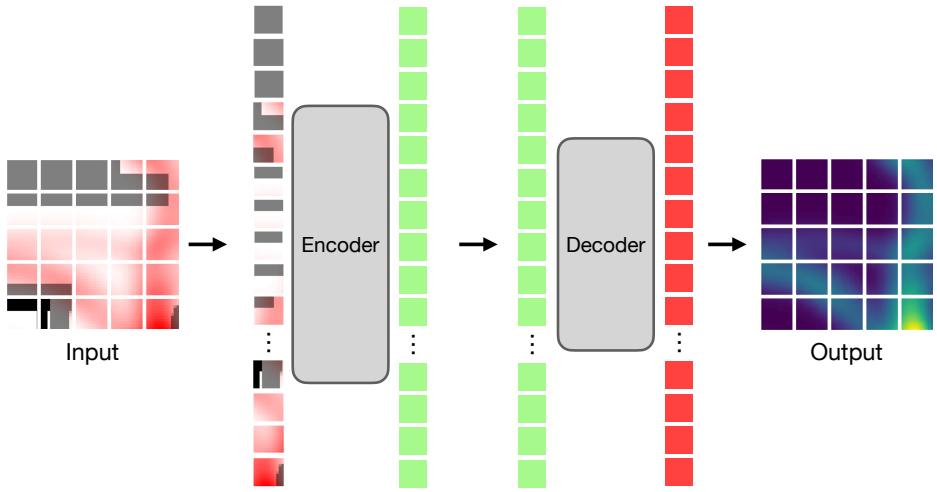


Figure 3.2: The proposed `semapp2` architecture: A Vision Transformer (ViT)-based autoencoder. The encoder processes crops of the input semantic map, and the decoder generates the corresponding prior distribution map (e.g., occupancy).

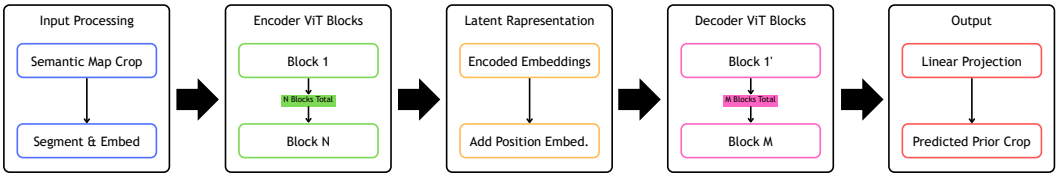


Figure 3.3: Detailed schematic diagram of the `semapp2` (ViT Autoencoder) architecture, illustrating the flow from input semantic crop processing to the predicted prior map output through the encoder and decoder Transformer blocks.

The core idea involves dividing the environment map into manageable, fixed-size crops, allowing for efficient processing. To capture the essential global context and spatial relationships (affordances) that might be lost through this parcelling, we employ ViTs. ViTs, known for their self-attention mechanisms, are well-suited to understanding the relationships between different parts of the map, even across distant patches. The autoencoder structure forces the network to learn salient

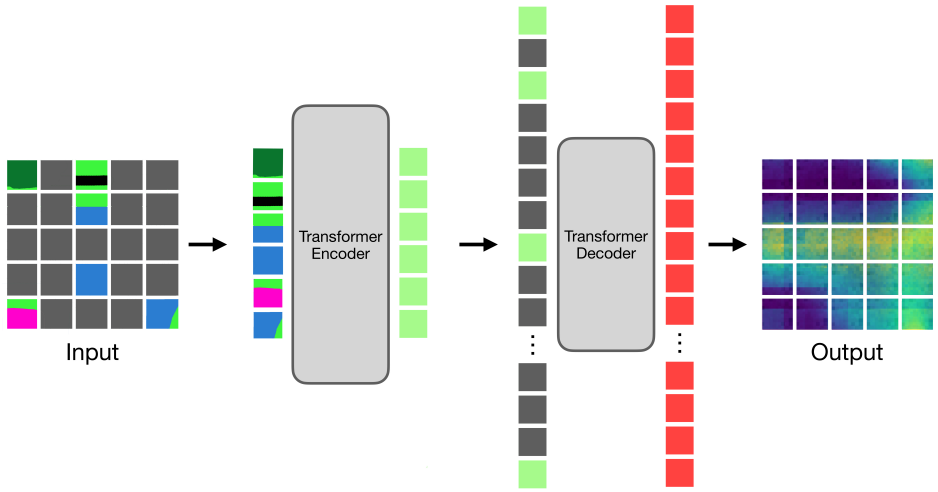


Figure 3.4: The `MAE-semapp2` architecture: A variation of `semapp2` using a Masked Autoencoder (MAE). A significant portion (e.g., 75%) of input patches are masked, and the model learns to predict priors from the visible patches.

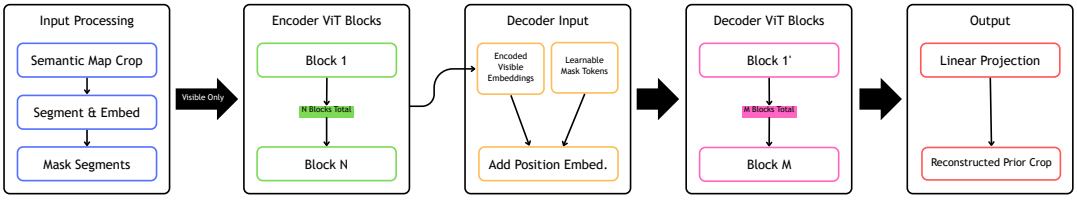


Figure 3.5: Detailed schematic diagram of the `MAE-semapp2` (Masked Autoencoder) architecture. Note the masking step after input processing and the use of visible segments plus mask tokens as input to the decoder.

features in the latent space that are predictive of the target priors.

Furthermore, we investigate a variation using Masked Autoencoders (MAEs), termed `MAE-semapp2` (Figure 3.4), exploring the potential benefits of self-supervised pre-training for this prediction problem. In this variation, a portion of the input crops are masked (hidden) before being fed to the encoder, and the model is trained to reconstruct the target priors even with this incomplete input. The hypothe-

sis is that these transformer-based architectures, by effectively utilizing semantic information and global context captured through self-attention and potentially enhanced by self-supervised pre-training, will outperform traditional CNN-based baselines in prediction accuracy, while remaining suitable for real-time applications due to their computational efficiency.

Note that the two architectures are conceptually similar: if we set a masking ratio of 0% on the `MAE-semapp2`, we obtain the same behaviour as the ViT-based `semapp2`. For simplicity, we will refer to a `semapp2` variant with a non-zero masking ratio (specifically 75% in our experiments) as `MAE-semapp2`.

3.3.1 Encoder

The encoder employs the standard ViT architecture, adapted for processing multi-channel semantic maps. The input semantic map patch undergoes a linear projection into patch embeddings, with added positional embeddings to retain spatial information. Subsequently, the resulting sequence of tokens (patch embeddings) is processed through a series of Transformer blocks, each containing multi-head self-attention and feed-forward layers. In the MAE variation (`MAE-semapp2`), the encoder architecture is identical, but it only processes the sequence corresponding to the **unmasked** patches of the semantic map input.

3.3.2 Decoder

The decoder takes the latent representation produced by the encoder (consisting of encoded visible patches and, in the MAE case, special mask tokens representing the missing patches) and reconstructs the target prior distribution map patch. Positional embeddings are added to all tokens fed into the decoder to provide spatial context for reconstruction. The decoder also consists of a series of Transformer blocks, and a final linear layer projects the decoder's output tokens back into the pixel space to form the predicted prior map patch. Notably, the decoder architecture can be designed independently of the encoder (e.g., it can be shallower or wider), providing flexibility in balancing model capacity and computational cost.

3.4 Experimental Evaluation

This section details the experimental setup, evaluation metrics, datasets, training procedures, ablation studies, and results used to validate the proposed `semapp2` and `MAE-semapp2` models. Our primary objective is predicting prior occupancy distribution based on semantic information, encompassing stop distribution and velocity heatmap prediction. We compare our ViT-based frameworks against Rudenko et al.'s CNN-based `semapp` [Rudenko et al., 2021].

3.4.1 Metrics

Before presenting the results, we define the metrics used for evaluation. To quantitatively evaluate the difference between the predicted prior distributions (Q_{pred}) and the ground truth distributions (P_{GT}), we employ several metrics commonly used for comparing probability distributions.

Kullback-Leibler Divergence (KL-div)

The Kullback-Leibler (KL) divergence [Kullback and Leibler, 1951] measures how one probability distribution diverges from a second, expected probability distribution:

$$\mathrm{KL}(P_{GT}||Q_{pred}) = \sum_i P_{GT}(i) \log\left(\frac{P_{GT}(i)}{Q_{pred}(i)}\right),$$

where the sum is over all discrete states (pixels) i. It quantifies the information lost when Q_{pred} is used to approximate P_{GT}. Lower values indicate better agreement. Standard KL divergence is asymmetric, meaning that swapping the two distributions yields different results. $\mathrm{KL}(P_{GT}||Q_{pred}) \neq \mathrm{KL}(Q_{pred}||P_{GT})$.

Reverse Kullback-Leibler Divergence (rKL-div)

We also compute the reverse KL divergence:

$$\mathrm{rKL}(Q_{pred}||P_{GT}) = \sum_i Q_{pred}(i) \log\left(\frac{Q_{pred}(i)}{P_{GT}(i)}\right).$$

While standard KL penalizes predictions that underestimate non-zero ground truth probabilities, reverse KL strongly penalizes predictions that assign high probability to regions where the ground truth probability is near zero. Calculating both provides a more comprehensive evaluation.

Earth Mover's Distance (EMD)

The Earth Mover's Distance (EMD) [Rubner et al., 2000], also known as the first Wasserstein distance, measures the minimum "cost" or "work" required to transform one distribution into the other. It is defined as:

$$\text{EMD}(P_{GT}, Q_{pred}) = \min_{\gamma \in \Gamma(P_{GT}, Q_{pred})} \sum_{(x,y) \in \text{supp}(\gamma)} \gamma(x,y) \cdot d(x,y),$$

where $\Gamma(P_{GT}, Q_{pred})$ is the set of all joint distributions (transport plans) $\gamma(x, y)$ whose marginals are P_{GT} and Q_{pred}, and $d(x, y)$ is the "ground distance" between locations x and y (typically the Euclidean distance). EMD is a true metric (symmetric and satisfies the triangle inequality) and is particularly effective at capturing perceptual similarity between spatial distributions, as it considers the cost of moving probability mass across space.

3.4.2 Datasets

Our study utilizes the Stanford Drone Dataset (SDD) [Robicquet et al., 2016]. This extensive dataset captures images and videos featuring diverse agents like pedestrians, bicyclists, skateboarders, cars, buses, and golf carts navigating a real-world outdoor university campus environment. It provides ground truth trajectories and corresponding video frames, suitable for extracting both semantic maps and ground truth behaviour priors (occupancy, stops, velocity).

We utilized a subset of 20 scenes (spanning 8 different locations) from the SDD for training and evaluation: 'bookstore' (5 scenes), 'coupa' (4 scenes), 'deathCircle' (5 scenes), 'gates' (2 scenes), 'hyang' (2 scenes), 'little' (1 scene), 'nexus' (2 scenes), and 'quad' (1 scene). Due to the limited number of distinct map locations, we employed a leave-one-scene-out cross-validation strategy. For each fold, one scene was held out for testing, while the remaining scenes were used for training and

Table 3.1: Quantitative Evaluation using 9 Semantic Classes on SDD (Mean ± Std Dev)

Method	KL-div	rKL-div	EMD
`semapp` [Rudenko et al., 2021]	0.66 ± 0.15	2.50 ± 1.51	40.18 ± 26.55
`semapp2` (ours)	0.49 ± 0.15	2.15 ± 1.20	34.24 ± 26.47

Table 3.2: Quantitative Evaluation using 13 Semantic Classes on SDD (Mean ± Std Dev)

Method	KL-div	rKL-div	EMD
`semapp` [Rudenko et al., 2021]	0.58 ± 0.14	2.43 ± 1.24	41.16 ± 26.98
`semapp2` (ours)	0.46 ± 0.16	2.19 ± 1.50	27.65 ± 19.89

validation (an 80/20 split of the remaining scenes). This approach ensures a robust evaluation of the model's generalization capability across different environments and observation periods.

During preprocessing, the SDD scenes were manually segmented into refined semantic classes, and ground truth priors were computed from the trajectory data. All scenes were uniformly scaled to a resolution of 0.4 meters per pixel. Since our network operates on map crops of a fixed size (determined via ablation study, see Section 3.4.4), we decomposed the larger input semantic maps and ground truth prior maps into overlapping crops. Following [Rudenko et al., 2021], we generated multiple random crops from each scene. During inference, the final prior distribution for the entire map was reconstructed by averaging the predicted prior values for each pixel across all crops containing that pixel. This averaging helps mitigate potential boundary artifacts between crops. Data augmentation, including rotations and mirroring, was applied to the crops during training to improve robustness.

To enhance prediction accuracy, we extended the semantic classes beyond the 9 used by Rudenko et al. [Rudenko et al., 2021] (pedestrian area, vehicle road, bicycle road, grass, tree foliage, building, entrance, obstacle, parking). We added 4 classes relevant to pedestrian movement: sitting area, stairs, shaded area, and intersection zone, for a total of 13 semantic classes. We hypothesized that these additional

classes provide finer-grained context that influences human motion. Tables 3.1 and 3.2 compare the performance of `semapp` and our `semapp2` using 9 versus 13 classes, supporting this hypothesis, particularly for `semapp2`.

Notably, `semapp2` outperforms `semapp` even with only 9 classes, and the improvement is more significant with 13 classes, particularly in terms of EMD, suggesting a better capture of the spatial distribution.

3.4.3 Training Setup

The models were trained for up to 100 epochs using the AdamW optimizer [Loshchilov and Hutter, 2019]. We employed a Mean Squared Error (MSE) loss between the predicted prior map crop and the ground truth prior map crop. Training was stopped early if the validation loss did not improve for 15 consecutive epochs. A cosine learning rate schedule with a 20-epoch warmup period was used. The base learning rate was set to 1×10^{-4}, adjusted proportionally to the total batch size according to the formula $absolute\_lr = base\_lr \times (total\_batch\_size/256)$. A weight decay of 0.3 was applied. Training was distributed across two NVIDIA RTX A5000 GPUs using PyTorch's Distributed Data Parallel (DDP) framework.

3.4.4 Ablation Study and Hyperparameter Analysis

To methodically examine the effects of various architectural choices and hyperparameters in the proposed `semapp2` model, we conducted a series of experiments including both ablation studies (removing or altering core components like the backbone size or the masking mechanism) and hyperparameter tuning (optimizing configuration settings such as input crop/patch sizes and masking ratio). The goal was to understand the contribution of key components and identify the optimal configuration. We used the quantitative metrics (KL-div, rKL-div, EMD) on a validation subset to measure the impact of each variation. Unless otherwise specified, these analyses were performed using the `MAE-semapp2` variant.

Architectural Components

We systematically tweaked key elements within the ViT-based architecture.

Backbone Size: We compared standard ViT backbones: ViT-Base, ViT-Large, and ViT-Huge [Dosovitskiy et al., 2021]. Experiments were run with a fixed mask ratio (75%), crop size (64x64 pixels), and patch size (8x8 pixels). Results are shown in Table 3.3. ViT-Huge achieved the lowest error metrics. However, considering

Table 3.3: Impact of ViT Backbone Size on `MAE-semapp2` Performance (Mean ± Std Dev)

Backbone	KL-div	rKL-div	EMD
ViT-Base	0.42 ± 0.13	2.52 ± 1.88	54.53 ± 30.80
ViT-Large	$\mathbf{0.34 \pm 0.21}$	$\mathbf{2.19 \pm 1.84}$	$\mathbf{45.77 \pm 30.74}$
ViT-Huge	0.31 ± 0.15	1.69 ± 1.11	39.64 ± 30.16

Note: ViT-Large was chosen as the best trade-off between performance and training time.

the significantly longer training times and relatively small performance gain over ViT-Large, we selected ViT-Large as the backbone for subsequent experiments. Furthermore, inspired by [He et al., 2021], we used a lightweight decoder (depth 1) with the ViT-Large encoder to accelerate training without substantial performance loss.

Crop Size: We varied the size of the square input crops fed to the network. Results (Table 3.4) indicated that a crop size of 64x64 pixels yielded the best performance.

Table 3.4: Impact of Crop Size on `MAE-semapp2` Performance (ViT-Large, 8x8 Patch, 75% Masking)

Crop Size (pixels)	KL-div	rKL-div	EMD
32x32	0.62 ± 0.18	3.74 ± 1.13	54.56 ± 29.84
64x64	$\mathbf{0.34 \pm 0.21}$	$\mathbf{2.19 \pm 1.84}$	$\mathbf{45.77 \pm 30.74}$
100x100	0.56 ± 0.19	3.60 ± 1.77	117.38 ± 93.33

Patch Size: Within a fixed crop size (64x64), we tested different patch sizes for the ViT's linear projection. A patch size of 8x8 pixels provided the optimal

balance (Table 3.5). Smaller patches increase sequence length and computational cost, while larger patches might lose fine-grained detail.

Table 3.5: Impact of ViT Patch Size on `MAE-semapp2` Performance (ViT-Large, 64x64 Crop, 75% Masking)

Patch Size (pixels)	KL-div	rKL-div	EMD
8x8	$\mathbf{0.34 \pm 0.21}$	$\mathbf{2.19 \pm 1.84}$	$\mathbf{45.77 \pm 30.74}$
16x16	0.52 ± 0.10	2.43 ± 1.02	53.02 ± 34.39
32x32	0.60 ± 0.20	4.57 ± 1.79	46.69 ± 30.03

MAE Masking Percentage

For the `MAE-semapp2` variant, we investigated the impact of the masking ratio used during pre-training. A higher ratio forces the model to rely more heavily on context from visible patches. Results are shown in Table 3.6. Interestingly,

Table 3.6: Impact of MAE Masking Percentage on `MAE-semapp2` Performance (ViT-Large, 64x64 Crop, 8x8 Patch)

Masking Ratio	KL-div	rKL-div	EMD
0% (`semapp2`)	0.46 ± 0.16	2.19 ± 1.50	27.65 ± 19.89
25%	0.45 ± 0.17	2.32 ± 1.66	38.78 ± 31.72
50%	0.41 ± 0.11	2.36 ± 1.44	49.30 ± 29.92
75%	$\mathbf{0.34 \pm 0.21}$	$\mathbf{2.19 \pm 1.84}$	$\mathbf{45.77 \pm 30.74}$

Note: 75% masking yields the lowest KL-div among the MAE variants, but high EMD.

while a high masking ratio (75%) achieved the best KL divergence, suggesting good reconstruction of high-probability areas, it resulted in a higher EMD compared to the non-masked (0%) version (`semapp2`). This suggests that while MAE pre-training helps learn strong representations, the standard `semapp2` (ViT autoencoder without masking) achieved a better overall spatial distribution match according to EMD in our final configuration using 13 semantic classes. We select 75% masking for the `MAE-semapp2` model presented in the results for comparison, acknowledging this trade-off.

3.4.5 Results and Discussion

Based on the ablation studies, we compare the final optimized models: `semapp` (baseline CNN), `semapp2` (our ViT-Autoencoder, ViT-Large, 64x64 crop, 8x8 patch, 0% mask, 13 classes), and `MAE-semapp2` (our MAE, ViT-Large, 64x64 crop, 8x8 patch, 75% mask, 13 classes).

Quantitative Evaluation

Table 3.7 summarizes the overall performance on the SDD test scenes using the leave-one-scene-out cross-validation protocol. Our proposed `semapp2` model demonstrates competitive performance, achieving the lowest (best) EMD, indicating a superior match to the overall spatial distribution compared to the baseline `semapp`. It also shows improvements in KL-div and rKL-div over `semapp`. The `MAE-semapp2` variant achieves the best KL-div, suggesting very accurate predictions in high-occupancy areas, but performs worse on EMD, potentially due to the reconstruction process from heavily masked inputs affecting the broader spatial structure. This highlights the effectiveness of ViTs in capturing the necessary context for predicting occupancy priors based on semantic information, outperforming the CNN baseline, particularly regarding the spatial layout (EMD).

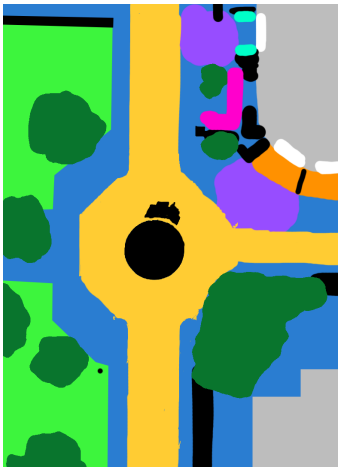
Table 3.7: Final Quantitative Evaluation on Stanford Drone Dataset (Mean ± Std Dev)

Method	Average KL-Div	Average rKL-Div	Average EMD
`semapp` [Rudenko et al., 2021]	0.58 ± 0.14	2.43 ± 1.24	41.16 ± 26.98
`semapp2` (ours)	0.46 ± 0.16	2.19 ± 1.50	$\mathbf{27.65 \pm 19.89}$
`MAE-semapp2` (ours)	$\mathbf{0.34 \pm 0.21}$	2.19 ± 1.84	45.77 ± 30.74

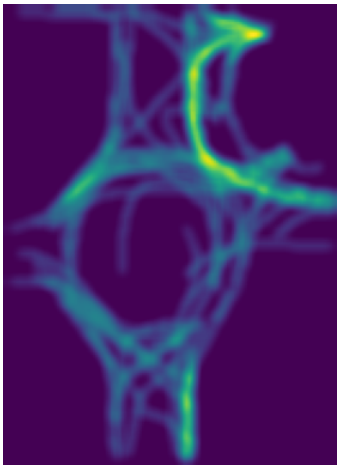
Note: Using 13 semantic classes for all models shown here. `semapp2` shows the best EMD, while `MAE-semapp2` achieves the lowest KL-div.

Qualitative Evaluation

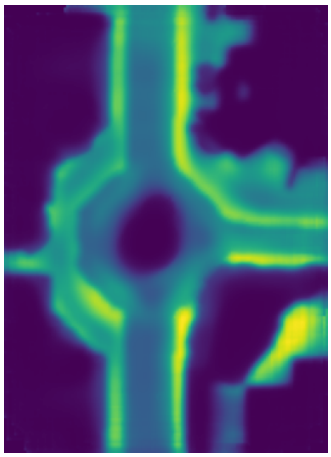
Figure 3.6 provides a visual comparison of the predictions from `semapp`, `semapp2`, and `MAE-semapp2` for a sample scene.



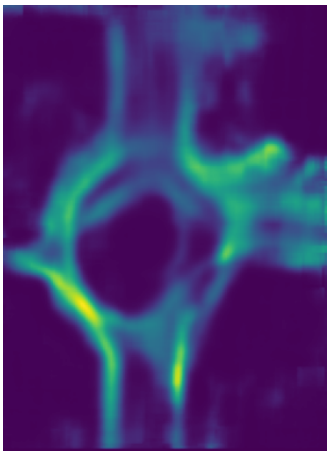
(a) Semantic Map



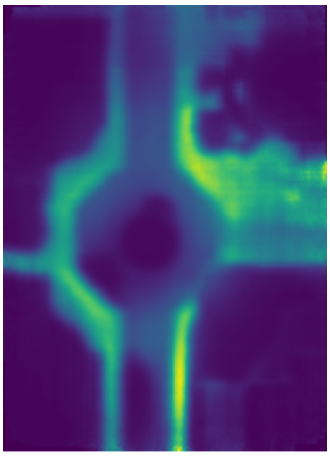
(b) Ground Truth Occupancy



(c) `semapp` Prediction



(d) `semapp2` Prediction

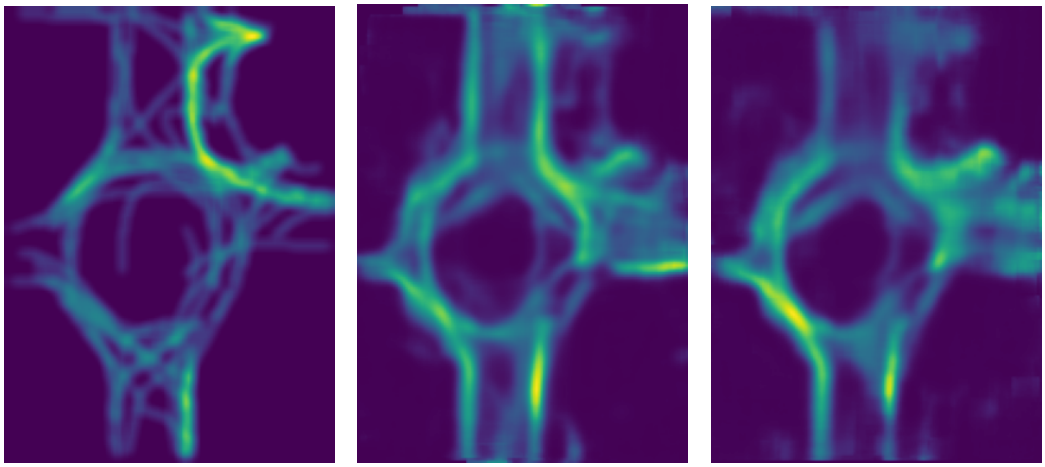


(e) `MAE-semapp2` Prediction

Figure 3.6: Qualitative comparison of occupancy prior predictions on a sample scene from the Stanford Drone Dataset (gates, video 1). Our ViT-based models (`semapp2`, `MAE-semapp2`) capture the main occupancy patterns.

All models capture the general high-occupancy areas corresponding to walkways. `semapp2` often produces smoother and potentially more spatially accurate predictions compared to `semapp`. `MAE-semapp2` shows strong prediction in core areas but might exhibit slightly different diffusion patterns due to the reconstruction from masked inputs.

Figure 3.7 visually compares `semapp2` predictions using 9 versus 13 semantic



(a) Ground Truth Occupancy

(b) `semapp2` Prediction (9 Classes)

(c) `semapp2` Prediction (13 Classes)

Figure 3.7: Qualitative comparison of `semapp2` predictions using 9 versus 13 semantic input classes on the ‘gates1‘ scene. The richer semantic input leads to quantitatively better results.

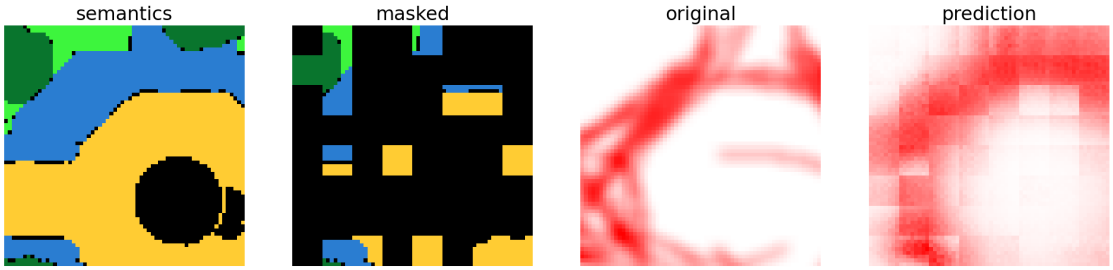


Figure 3.8: Example of the prediction process using `MAE-semapp2`. From left to right: Input semantic map crop, masked input fed to the encoder, ground truth occupancy prior, and the model's prediction.

classes. While the visual difference is subtle, the quantitative results (Tables 3.1 and 3.2) confirm a slight improvement with the richer 13-class representation.

Figure 3.8 illustrates the MAE process, showing the masked input and the corresponding reconstruction alongside the ground truth.

Figure 3.9 provides further qualitative comparisons highlighting cases where `MAE-semapp2` appears to capture fine-grained details well, despite its higher overall EMD score in the quantitative evaluation. This suggests MAE's potential strength

in learning underlying patterns, even if the global spatial match isn't always optimal according to EMD. The discrepancy between strong qualitative examples and the average EMD score might stem from the limited duration of trajectories in some SDD videos, meaning the ground truth itself might not represent a fully converged long-term prior. The generalization ability of MAE warrants further investigation, potentially with datasets covering longer observation periods.

Predicting Stops and Velocities

A key contribution of this work is extending prior prediction beyond occupancy to include stop locations and average velocities. Figure 3.10 shows qualitative results for these priors on the 'coupa3' scene, demonstrating the model's ability to learn these related behaviours (e.g., predicting stops near entrances/crossings and higher velocities on clear paths). Table 3.8 provides the quantitative evaluation for these additional priors using the `semapp2` model. The performance is comparable to the occupancy prediction task, indicating the model successfully learns these related distributions from the semantic map context. Predicting these nuanced behaviours is valuable for applications requiring more detailed motion understanding, such as proactive robot navigation and human-robot interaction planning.

Table 3.8: Quantitative Evaluation of Velocity and Stop Prior Prediction using `semapp2`

Prior Type	KL-div	rKL-div	EMD
Velocities	0.47 ± 0.15	2.50 ± 1.51	40.18 ± 26.55
Stops	0.63 ± 0.15	2.15 ± 1.20	52.88 ± 27.94

In summary, the experimental results demonstrate that the proposed ViT-based approach, `semapp2`, effectively leverages semantic map information to predict human behaviour priors, outperforming a CNN baseline and showing promise for predicting not just occupancy but also stops and velocities.

Real-World Application Considerations

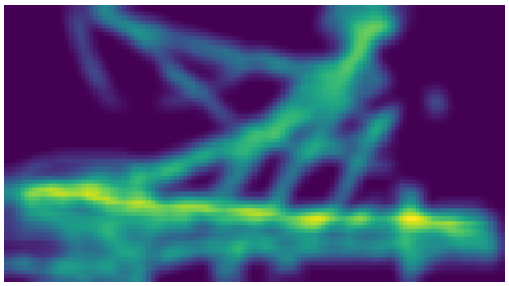
While the validation on the SDD dataset demonstrates the potential of `semapp2`, claiming it is well-suited for real-world application warrants further discussion

(a) Semantics (bookstore0) (b) Ground Truth (c) MAE-semapp2 (d) semapp2

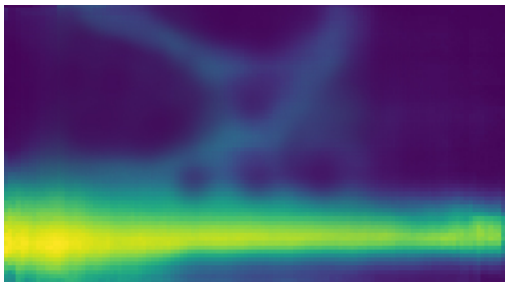
(e) Semantics (coupa3) (f) Ground Truth (g) MAE-semapp2 (h) semapp2

Figure 3.9: Qualitative comparison between MAE-semapp2 and semapp2 predictions for two different scenes (bookstore0 and coupa3). While semapp2 has better average EMD, MAE-semapp2 can capture fine details effectively in specific cases.

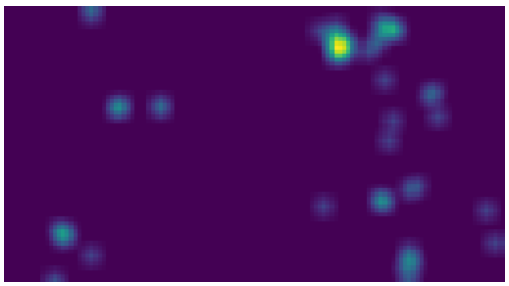
(a) Semantic Map (coupa3)



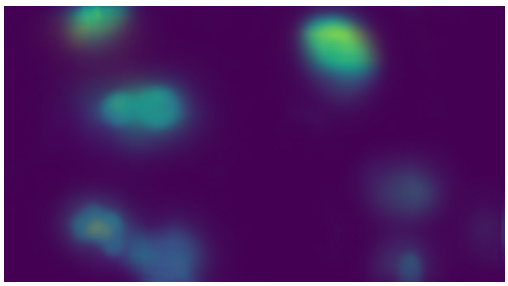
(b) Ground Truth Velocity



(c) Predicted Velocity



(d) Ground Truth Stops



(e) Predicted Stops

Figure 3.10: Qualitative results for predicting velocity magnitude priors (top row) and stop location priors (bottom row) using `semapp2` on the 'coupa3' scene.

regarding deployment on robotic platforms.

Deployment Path: In a practical robotic system, `semapp2` would operate downstream from a semantic mapping module. The robot's perception system (e.g., using cameras, LiDAR) would first generate or update a semantic map of the current environment. This map, potentially processed into patches, would serve as the input to `semapp2`. The output prior maps (occupancy, stops, velocity) would then be consumed by higher-level modules, such as navigation planners or risk assessment

systems. For instance, a path planner could use the predicted occupancy prior to favour routes through less typically congested areas or use the velocity prior to anticipate speeds along potential paths (as conceptually utilized in systems like the *FriWalk* mentioned in Section 3.2).

Advantages: The primary advantage for real-world robots lies in proactively leveraging static environmental context. Even before observing any humans, the robot can anticipate likely movement patterns, stop locations, and speeds based purely on the environment's structure (e.g., sidewalks encourage walking, intersections are potential stopping/slowing points). This allows for more informed initial planning and potentially safer and more efficient navigation compared to purely reactive approaches that only consider currently perceived agents. The ability to predict multiple priors (stops, velocities) offers richer context than occupancy alone.

Limitations and Challenges: Several factors must be considered for successful real-world deployment.

- **Semantic Map Quality:** The performance of `semapp2` is inherently tied to the accuracy and richness of the input semantic map. Errors or noise in the semantic classes in the upstream mapping module will directly impact the quality of the predicted priors.
- **Static Assumption vs. Dynamic Reality:** The model learns priors from aggregated historical data reflected in static maps. It does not inherently capture real-time deviations from typical behaviour, such as temporary blockages, unusual crowd formations, or emergency situations. Therefore, these priors must be integrated with real-time perception and prediction systems for robust and safe operation. The priors provide a baseline expectation, which needs to be updated based on live observations.
- **Computational Cost:** While ViT architectures are powerful, models like ViT-Large can be computationally intensive. Deployment on resource-constrained robotic hardware might require model optimization techniques such as prun-

ing, quantization, or knowledge distillation, or the selection of smaller backbone architectures (like ViT-Base, albeit with a potential trade-off in accuracy as seen in Table 3.3). Inference time analysis on target hardware would be crucial.

- **Generalization:** Although cross-validation was used on SDD, generalization to environments significantly different in structure, scale, or semantic complexity requires further investigation. Training or fine-tuning on data representative of the target deployment domain might be necessary.

Addressing these limitations, particularly the integration with dynamic perception and optimizing for on-board computation, are key steps towards robust real-world deployment.

Chapter 4

Shared Control for Robotic Walkers

This chapter explores shared control for robotic walkers, focusing on a use case involving the *FriWalk* robotic rollator. The system supports individuals with mild cognitive deficits by enabling safe navigation while allowing the human to retain primary control. The robot intervenes only when the user's actions deviate significantly from expected human behaviours.

Our proposed framework [Falqueto, Antonucci, et al., 2024] leverages learning-based methods to classify expected human motion patterns, enabling the robot to assess deviations and dynamically adjust the shared authority. By employing a visco-elastic control mechanism regulated by classification confidence, the walker achieves a balance between autonomy and user agency, promoting both safety and usability.

4.1 State-of-the-Art in Human-Robot Collaboration

Human-robot interaction (HRI) has emerged as a crucial research field given the increasing integration of robotic systems in various settings, illustrated by reviews covering industrial collaboration trends in Industry 4.0 [Baratta et al., 2023], the systematic application of social robots within hospitals [], and evolving applica-

tion trends in domestic and home service robotics [Zachiotis et al., 2018]. Unlike traditional robots, collaborative robots (cobots) work closely with humans, necessitating nuanced interaction paradigms to balance safety, comfort, and performance. As Goodrich et al. [2007] emphasize, human-robot interaction spans diverse application fields such as search and rescue, assistive robotics, and education, showcasing the wide applicability and transformative potential of shared control systems.

In collaborative robot applications, there is the need to understand and support the human decisions, so that the human-robot interaction can be comfortable for the user, maintaining the accuracy characteristic of a robotic system. When the collaborative robot assists the human in performing a task, the machine should provide help without being perceived as a hindrance. When the robot acts to deviate from the user's intent, usually to avoid safety issues, it should do it in the less noticeable way to maintain a high comfort level for the human.

Research in HRI focuses on enabling robots to effectively interpret and respond to human intentions and actions, facilitating seamless collaboration. Flemisch et al. [2019] define human-machine cooperation as a process in which both agents actively contribute to achieving shared goals, particularly in dynamic and unpredictable environments. Unlike human-computer or human-human interactions, HRI uniquely involves physical embodiment, real-time decision-making, and bidirectional communication [Dautenhahn, 2007]. This section explores foundational theories and current advancements in shared control, adaptive shared control, and reinforcement learning (RL) applied to HRI, particularly relevant to assistive robotic walkers like the *FriWalk*.

4.1.1 Shared Control

Shared control is a paradigm that balances human and robot autonomy, ensuring safety, comfort, and efficiency in HRI. This section differentiates traditional shared control from adaptive frameworks that employ RL for dynamic authority arbitration.

Early work in passive robotics by Goswami et al. [1990] explored control laws implemented through mechanical devices with unpowered hydraulic components

and variable stiffness. These systems highlighted the potential for mechanical design to influence control strategies, laying the groundwork for modern shared and active robotics frameworks. Shared control represents a collaborative paradigm where human and robotic agents simultaneously contribute to a task. Sheridan and Verplank [1978] initially introduced this concept, emphasizing the importance of balancing human autonomy with robotic assistance.

Abbink et al. [2018] refined this concept by emphasizing the cyclic and dynamic nature of shared control interactions. Abbink's definition of shared control is

> *In shared control, human(s) and robot(s) are interacting congruently in a perception-action cycle to perform a dynamic task that either the human or the robot could execute individually under ideal circumstances.*

In their study, Abbink's group point out how full automation is not always possible since the environments are complex and unpredictable or there is a high level of risk involved, such as in surgical procedures. The human is therefore needed to achieve adequate performance. They state how Shared Control is essential to overcome the "limits of the human sensorimotor system, which are often approached due to the scale of some procedures" and that "can be enhanced to help the surgeon operate within the environment constraints (e.g., small anatomical structures, delicate tissues, low interaction forces)" or how "incorporating different sources of information (e.g., patient-specific anatomy from different imaging modalities) in a seamless manner can aid the surgeon in efficiently determining and executing a desired plan".

Similarly, Yanco and Drury [2004] propose a taxonomy for Human-Robot Interaction (HRI), defining it as a subset of human-computer interaction and computer-supported cooperative work. They describe an important property of HRI: the *autonomy level* and *amount of intervention*. Specifically, a high autonomy level implies a low level of human intervention, allowing the robot to execute tasks autonomously, whereas a low autonomy level results in significant user control over the system.

Traditionally, the HRI autonomy level has been hard-wired and set to a fixed point, blending both human and robotic commands. This traditional type of control is defined as "Shared Control." Yanco and Drury highlight the promise of

research into "adjustable autonomy," also referred to as "sliding scale autonomy" or "mixed initiative," where the authority ratio dynamically shifts between the human and robot. This dynamic control paradigm, known as *Adaptive Shared Control*, offers significant potential to enhance interaction flexibility, facilitating seamless transitions in authority between the human and robotic agents.

The lack of a standardized definition within the research community has led to some misconceptions, as pointed out by Abbink et al. [2018]. For example, sometimes control paradigms such as partitioning and traded control are confused with related to shared control, when actually they fit better in the cooperative control field. In fact, Abbink's group state that partitioning control consists in dividing a task into subtasks that are executed by individual agents (i.e., human and robot), while in traded control human and robot work on the same task but at different times.

Marcano et al. [2020] further review shared control approaches, emphasizing the benefits of haptic feedback and its applicability across various robotic systems, including assistive technologies. Their findings underline the importance of sensory feedback in enhancing user experience and fostering intuitive human-robot collaboration.

Two main modalities of shared control are commonly discussed in the literature [Abbink et al., 2018; Marcano et al., 2020]:

- **Haptic Shared Control:** The human user receives force feedback from the robotic system through a mechanically coupled interface, enhancing mutual awareness of conflicts and enabling immediate resolution.

- **Input-Mixing Shared Control:** The robot combines human input with corrective signals to achieve task goals. This modality is particularly useful in systems where direct physical coupling is absent, such as UAV teleoperation or brain-machine interfaces. The final authority is assigned to the robot and the activity of the automation is not continuously communicated to the user.

Many studies have explored haptic shared control [Abbink et al., 2018; Andreetto et al., 2019b; Argall, 2018; Dani et al., 2020; Eraslan et al., 2020; Fei Shi et al., 2010; Li et al., 2020; Losey et al., 2018; Marcano et al., 2020; Mohebbi,

2020; Patoglu et al., 2009; D. Zhang et al., 2021], showing its advantages over input mixing shared control. However, haptic shared control is not always applicable due to the typology of the task. For example, in brain-machine interfaces, the interaction occurs through the user's neural signals rather than a mechanical system capable of providing rich, bidirectional force feedback, as is possible with a joystick or robotic arm. In such cases, input-mixing shared control is the preferred solution. The *FriWalk*, which is physically coupled to the user through the handlebars, transmits the robot's authority directly to the user: when the wheel motors apply corrective actions, the user perceives these interventions through the mechanical connection. This places the *FriWalk* firmly within the haptic shared control category.

Traditional shared control approaches rely on methods like artificial potential fields and virtual fixtures. In potential field methods [Aigner and McCarragher, 1997; Crandall and Goodrich, 2002; Gerdes and Rossetter, 1999], human commands are combined with attractive or repulsive force fields, typically at a fixed mixing ratio. Virtual fixture-based methods [Marayong et al., 2002; Rosenberg, 1993] constrain user motion in undesired directions while providing assistance along desired paths. These methods, however, lack predictive models of user intent. Yu et al. [2005] integrate artificial potential fields and virtual fixtures with Hidden Markov Models for intent recognition, illustrating the importance of combining traditional methods with advanced predictive algorithms.

Shared control has been widely applied across domains. In rehabilitation, systems translate user-applied forces into robotic motions or predict user intent to provide adaptive support [Kyrarini et al., 2021; Tang and Cao, 2012]. Industrial logistics benefit from cobots capable of learning user preferences to enhance warehouse operations [L. Zhang et al., 2016]. Moreover, supernumerary robotic limbs offer additional functionality, such as assisting in multitasking or improving ergonomics in industrial settings [Tran et al., 2018].

4.1.2 Adaptive Shared Control

Also known as "adjustable autonomy", "sliding scale autonomy" or "mixed initiative". The use of static arbitration [Fujioka et al., 1999] in shared control systems

poses several challenges that can significantly impact both the user experience and the effectiveness of the system. Specifically, static arbitration is detrimental to:

- **User acceptance:** A rigid control framework can lead to the user perceiving the robotic system as overly dominant, which might evoke feelings of being constrained or enslaved by the robot. This lack of perceived agency can reduce the overall trust and willingness to engage with the system.

- **Motor learning:** Static arbitration alters the task dynamics experienced by the user, potentially impeding the natural learning process. Studies, such as those conducted by Patoglu et al. [2009], demonstrate that motor learning relies heavily on the ability to adapt to variable task conditions, which static systems fail to provide.

In contrast, *adaptive shared control* frameworks dynamically adjust the level of autonomy in response to several factors, including task complexity, user behaviour, and environmental context. This dynamic adjustment ensures a more tailored interaction between the human user and the robotic system, leading to improved overall outcomes.

Unlike static shared control systems, adaptive frameworks leverage real-time data streams to continuously assess and optimize the interaction. These systems incorporate various metrics, such as user input patterns, physiological signals, and environmental variables, to inform the arbitration strategy. By doing so, adaptive shared control provides the following key benefits:

- **Enhanced user comfort and acceptance:** By tailoring the control level to the user's capabilities and preferences, adaptive systems foster a sense of partnership between the user and the robot, thereby improving trust and satisfaction [Patoglu et al., 2009].

- **Improved task performance:** Dynamic adaptation enables the system to support the user more effectively during challenging tasks while allowing greater autonomy during simpler tasks. This balance optimizes task efficiency and success rates.

- **Support for motor learning:** Adaptive shared control systems preserve natural task dynamics, providing opportunities for users to gradually improve their motor skills through practice and variability.

- **Robustness to environmental changes:** Real-time adjustments ensure that the system can seamlessly adapt to unexpected changes or uncertainties in the environment, maintaining performance and safety.

In neurorehabilitation, in contrast with what happens in static arbitration, the level of assistance is usually decreased over time as motor skills are relearned.

For example, Dragan and Srinivasa [2013] show that users prefer fully autonomous modes for planning tasks but manual modes for object manipulation. Similarly, Carlson et al. [2012] propose an online adaptive shared control system that adjusts authority dynamically based on the user's behaviour and environmental context, ensuring seamless collaboration.

Oh et al. [2020] propose a natural gradient approach for adaptive shared control, addressing challenges like conflicting goals and task uncertainty. Zurek et al. [2021] introduce confidence-aware control to mitigate "noisy" operator inputs. They emphasize that ensuring the system correctly identifies user intent is critical in dynamic environments, where prediction errors can lead to reduced performance or safety concerns.

Adaptive Shared Control for Assistive Robots

The *FriWalk* employs visco-elastic control mechanisms to regulate steering torque, balancing user autonomy with robotic intervention. This dynamic mechanism ensures that corrective guidance is applied only when deviations from the expected behaviour are detected, enabling a seamless and intuitive user experience [Andreetto et al., 2018; Bevilacqua et al., 2020]. Figure 4.1, shows the *FriWalk* robotic rollator using adaptive shared control paradigms with an elderly person, where the autonomy level dynamically adjusts based on user intent and contextual demands.

The *FriWalk* robotic rollator exemplifies adaptive shared control paradigms, where the autonomy level dynamically adjusts based on user intent and contextual demands. Shared control involves blending user commands and robotic autonomy

Figure 4.1: The *FriWalk* robotic rollator using adaptive shared control paradigms with an elderly person.

to optimize both safety and user independence. User input, such as forces applied to the handlebars, is continuously monitored and processed to infer motion intentions.

Other Applications in Collaborative Robotics

Adaptive shared control has transformative potential across multiple domains:

- **Rehabilitation:** Algorithms predict user intent and assist during physical therapy [Kyrarini et al., 2021; Santhanaraj et al., 2021].
- **Industrial Logistics:** Cobots enhance efficiency by adapting to user-specific patterns [L. Zhang et al., 2016].
- **Wearable Robotics:** Adaptive shared control is used in wearable robots to assist users in challenging tasks [Tong and Liu, 2021].

Despite these advancements, challenges remain in intent inference and arbitration strategies. For instance, Tong and Liu [2021] discuss control strategies such as electromyography (EMG) and brain-machine interfaces (BMI) that are gaining traction in wearable robotics.

Adaptive Shared Control has also been applied in smart vehicles [Backman et al., 2021], drone teleoperation [Reddy et al., 2018], and assistive robots [Zurek et al., 2021], demonstrating its flexibility across domains.

4.1.3 Behavioural Analysis from Trajectories

Several studies have explored methods to infer abstract characteristics of human behaviour from trajectories, predominantly leveraging machine learning techniques. For example, Support Vector Machines (SVMs) have been employed to classify diverse walking styles and behaviours, including straight walking, left-turns, right-turns, U-turns, and stationary states [Kanda et al., 2009]. The features provided as input to the SVM include normalized coordinates, orientation, velocity, and the bounding boxes of trajectories within a defined observation window.

Autoencoders (AEs), on the other hand, offer the capability to reconstruct input data while simultaneously learning lower-dimensional representations, commonly referred to as the *latent space*. A dual approach combining clustering with linear autoencoders was introduced by Murray and Perera [2020] to forecast vessel trajectories. The state space of the vessels in this approach encompasses pose and linear velocities. Similarly, Rakos et al. [2020] utilized a convolutional Variational Autoencoder (VAE) to derive latent representations of real-world vehicle trajectories represented as time-series of 2D coordinates. Their approach reportedly reduced the dimensionality of the latent space from 10 to 2, achieving significant data reconstruction efficiency without compromising classification accuracy.

Although modern machine learning methods aim to discern meaningful features from simple time-series data, we posit that incorporating prior geometric information as input to neural models can streamline their design and enhance performance. For instance, Lu et al. [2020] enriched the input of a Convolutional Autoencoder (CAE) with spatio-temporal features such as velocity, acceleration, and heading change rate. In this work, we adopt an autoencoder to extract geometric parameters from trajectories, which are subsequently classified into behaviour categories using a secondary neural network. Additionally, during the control phase, we utilize Bayesian inference to quantify the confidence in the human be-

haviour classification.

4.1.4 Reinforcement Learning in Shared Control

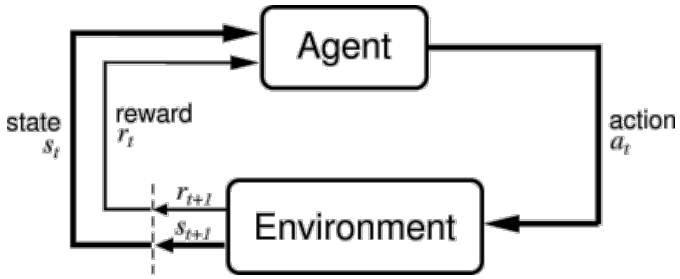


Figure 4.2: Essential parts in Reinforcement Learning. The agent observes the states and rewards obtained after a performed action, and decides the optimal next action.

Reinforcement Learning (RL) is a machine learning online training method based on reward functions. Fig. 4.2 shows the essential parts in Reinforcement Learning.

RL is becoming popular in robotics as it enables the robot to learn an optimal behaviour by training to perform the tasks it is designed for in its environment, exploring new possible states and exploiting the discerned optimal policy.

In the robotics field, using Reinforcement Learning can be complex since we often deal with high-dimensional continuous states and actions. Moreover the states are often only partially observable and noisy. Even if real-world experience is costly, hard to reproduce and tedious, it cannot be completely replaced by learning in simulation.

RL has gained traction in robotics due to its ability to optimize behaviour through iterative learning. Model-free methods like Q-Learning [Watkins, 1989] and its deep-learning variants [Reddy et al., 2018] have been successfully applied to shared control problems. However, these methods face challenges in real-world applications with continuous state spaces and noisy observations [Kober et al., 2013; Polydoros and Nalpantidis, 2017].

RL algorithms have been increasingly adopted to dynamically balance robot autonomy and user intervention, ensuring that robotic assistance occurs only when

necessary [F. O. Flemisch et al., 2003]. In assistive technologies, Wei et al. [2017] analyze RL applications focused on patient-centric rehabilitation and adaptive robotic support. Song et al. [2017] highlight the importance of incorporating cooperative weights based on user velocity and obstacle proximity to enhance safety and user experience, especially in dynamic environments requiring precise guidance and fall prevention. Birku and Agrawal [2018] provide a comprehensive survey of RL's potential in rehabilitation, particularly for fall detection and patient intention estimation.

Recent studies, such as Papudesi et al. [2003], demonstrate the potential of leveraging human-controlled rewards in RL. This approach enables robots to adapt their behaviour more effectively by incorporating real-time user feedback, guiding their learning process in dynamic environments. Backman et al. [2021] use variational auto-encoders and policy-gradient RL for drone landing assistance, demonstrating robust user adaptation. Similarly, Reddy et al. [2018] employ user feedback to train assistive agents. These studies highlight the potential for RL to personalize assistive behaviours, adapting dynamically to individual user preferences and constraints.

For instance, Kartoun et al. [2010] highlight the integration of human guidance into reinforcement learning, enabling robots to improve their policies autonomously while intermittently relying on human advice for complex scenarios. This approach is particularly valuable in dynamic environments where the robot must make critical decisions in real-time, blending learned behaviours with human-provided insights.

Additionally, RL frameworks like Deep Q-Networks (DQN) [Mnih et al., 2015] and Actor-Critic methods [Haarnoja et al., 2018] have extended the applicability of RL to high-dimensional robotics tasks. Combining these methods with shared control paradigms remains an active area of research. Specifically, developing adaptive shared control strategies that dynamically arbitrate authority based on a learned understanding of context-specific, typical human behaviour, rather than relying solely on instantaneous deviations or predefined rules, presents a significant challenge addressed in this chapter.

It is important to note that while Reinforcement Learning provides a robust framework for adaptive control, the specific methodology detailed in this chapter

will not employ RL directly. Instead, our approach focuses on leveraging supervised learning techniques to classify human-like trajectory patterns. The confidence derived from this classification will then be used to dynamically arbitrate control authority within the shared control system, as elaborated in the subsequent sections.

4.2 The *FriWalk* Platform

The research on adaptive shared control presented in this chapter is experimentally validated using the *FriWalk* robotic rollator. The *FriWalk* is an assistive mobility platform designed to support individuals by enabling safe navigation while allowing the user to retain primary control, with the robot intervening based on learned human motion patterns. A detailed technical overview of the *FriWalk* platform, including its hardware and software architecture, was provided in Chapter 2.

A key aspect of the *FriWalk*'s functionality is its ability to generate and model motion effectively. This is crucial for both understanding human intent and guiding the user when necessary.

Trajectory Generation and Motion Modeling The trajectory generation process is significantly enhanced by mathematical models that approximate human motion. The unicycle model, presented formally in Section 4.3 (Eq. 4.1), forms the kinematic foundation for both approximating human locomotion and describing the *FriWalk*'s movement [Arechavaleta et al., 2008a; Farina et al., 2017]. It represents motion through smooth, nonholonomic trajectories. Clothoid curves are used in conjunction with this model to further refine trajectory planning, ensuring linear curvature and real-time adaptability [Bertolazzi and Frego, 2018]. Concatenated clothoid arcs are used for optimal path generation, enabling smooth transitions between increasing and decreasing curvature. This approach enhances the device's adaptability and provides a natural, user-friendly experience [Laumond et al., 2010].

Cognitive and Physical Assistance The *FriWalk* offers features tailored to users with cognitive impairments, such as route planning, reminders, and guid-

ance cues. For those with physical impairments, the device provides stability and ergonomic support, adapting its assistance to user needs through customizable settings. This human-centered design enhances both user well-being and independence [Bevilacqua et al., 2016; F. O. Flemisch et al., 2003].

Predictive modeling techniques, such as the Social Force Model (SFM) [Helbing and Molnar, 1995], are integrated with neural networks to anticipate user behaviour and environmental interactions. This allows the *FriWalk* to adjust its trajectory dynamically, ensuring safety and minimizing cognitive load for the user [Antonucci, Papini, et al., 2021; Antonucci, Bevilacqua, et al., 2021].

4.3 Problem Statement and Solution Overview

Addressing the need for assistance that is both effective and acceptable to the user (as highlighted by the drawbacks of static arbitration discussed in Section 4.1.2), the core problem is to enable the robot to intervene supportively, but only when necessary. The following unicycle kinematic equations are used to describe the *FriWalk* robot motion:

$$\begin{cases} x(t_{k+1}) = x(t_k) + \cos(\theta(t_k))\delta_t v(t_k), \\ y(t_{k+1}) = y(t_k) + \sin(\theta(t_k))\delta_t v(t_k), \\ \theta(t_{k+1}) = \theta(t_k) + \delta_t \omega(t_k), \end{cases} \tag{4.1}$$

where, $\boldsymbol{q}(t_k) = [x(t_k), y(t_k), \theta(t_k)]$ represents the state of the vehicle, with $v(t_k)$ and $\omega(t_k)$ denoting longitudinal and angular velocities, respectively, and δ_t as the sampling time.

For the particular problem at hand, $v(t_k)$ is imposed by the human (also for safety reasons [Andreetto et al., 2018], to avoid pulling/pushing the human), while $\omega(t_k)$ is the control output and it is shared between the human and the robot. The problem to solve is to control the vehicle from a starting position $p_0 = [x_0, y_0]^T$ to a desired position p_f in a known environment. The key requirement is to use the robot controller contribution to $\omega(t_k)$ only when the human behaviour deviates significantly from the expected behaviour. Figure 4.3 illustrates the reference frames involved and the concept of allowing compliant movements while correcting

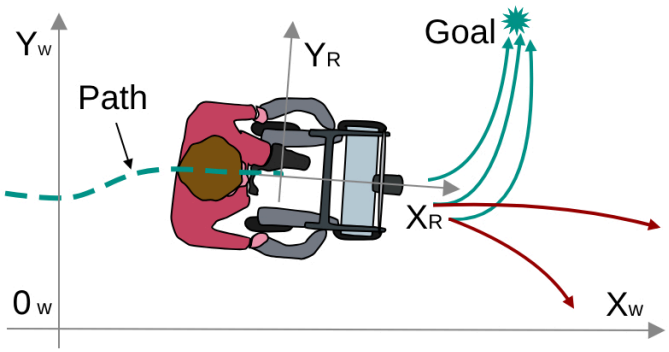


Figure 4.3: Representation of the Walker's environment interaction, showing the world ($\langle W \rangle$) and robot ($\langle R \rangle$) reference frames. The example illustrates the concept of distinguishing between compliant (green) and non-compliant (red) movements relative to an intended path or behaviour.

deviations.

To this end, we need first to abstract the path following problem, that is usually defined in the space $\boldsymbol{q}(t_k)$, into a high level representation that preserves the implicit features of the human trajectories. Therefore, let us denote by $\mathcal{H} \subset \mathbb{R}^2$ the path travelled by the human in $\langle W \rangle$, i.e., the sequence $(x(h_k), y(h_k))$ of coordinates expressed with respect to the curvilinear abscissa h sampled at times δ_t. Let $\mathcal{R} \subset \mathbb{R}^2$ be the reference path connecting p_0 to p_f. For both paths, we extract a set of features of dimensionality m, denoted as $\boldsymbol{z}_{k,\mathcal{H}}$, $\boldsymbol{z}_{k,\mathcal{R}} \in \mathbb{R}^m$, respectively, which are associated to a class of human-like behaviours {Left-turn, Right-turn, Straight}.

The overall framework of the proposed solution, illustrated in the block diagram in Fig. 4.4, comprises the following steps, divided into an offline map generation phase and an online control phase:

Offline Phase:

1. **Synthetic Trajectory Generation:** Given an a-priori map of the environment containing static obstacles (walls, furniture, etc.), the **"PRM with clothoid paths"** block generates a large number of collision-free trajectories mimicking human-like behaviour [Laumond et al., 2010]. This provides a dense sampling of plausible paths within the environment.

2. **Behavioural Map Creation:** The environment is partitioned into cells (e.g.,

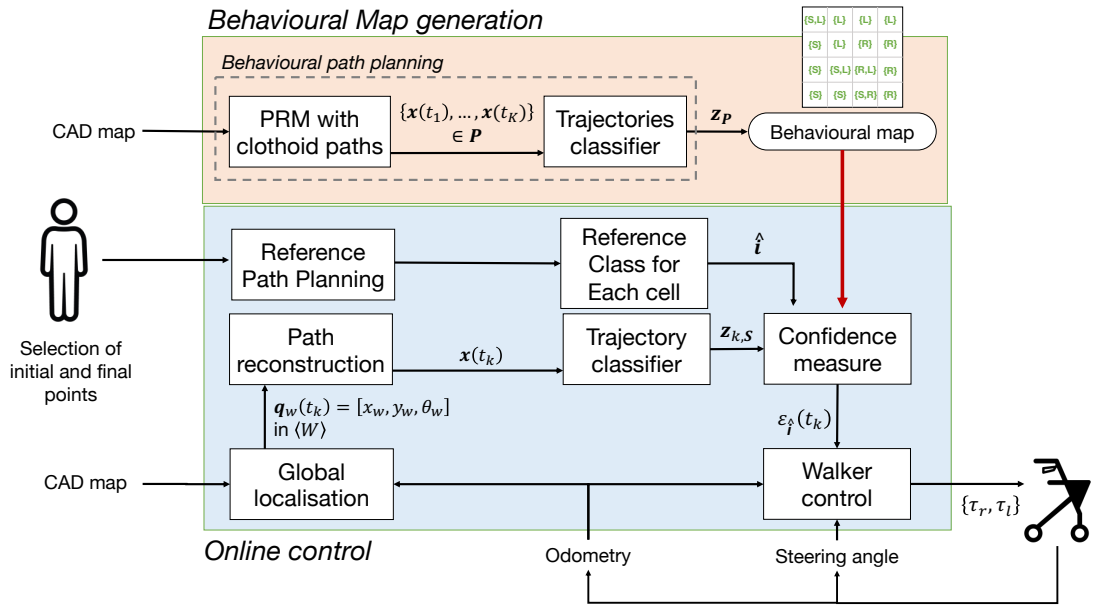


Figure 4.4: Overall scheme of the algorithm, depicting the offline generation of the behavioural map and its online use for adaptive shared control.

square grid). The synthetic trajectories generated in Step 1 are segmented within each cell. The **"Trajectories classifier"** block (see Figure 4.4), which comprises two neural networks detailed in Section 4.4.1 – Net1, a convolutional autoencoder that takes trajectory geometry as input and outputs compressed latent features, and Net2, a simple fully-connected classifier that takes these features as input and outputs behaviour class probabilities – processes each trajectory segment to classify it into one of the predefined behavioural classes {Left-turn, Right-turn, Straight}. The output of this classification, along with the orientation of the segments, populates the **"Behavioural map"** block. This map essentially stores the expected or typical human-like manoeuvre(s) for traversing each cell in the environment.

Online Phase:

3. **Adaptive Shared Control:** a) Given the user's current position p_0 and selected destination p_f, a reference path is planned (e.g., using PRM and clothoids) to filter the relevant behaviour from the behavioural map to reach the target.

b) As the user moves, their current trajectory segment $\mathcal{H}$ is processed in real-time by the *same* **"Trajectories classifier"** block used offline. This extracts the segment's latent features $\boldsymbol{z}_{k,\mathcal{H}}$ and determines its current behavioural class.
c) The **"Behavioural map"** is queried using the user's current location (cell j) to retrieve the reference behaviour $(c_{\hat{i}}^{(j)}, \theta_{\hat{i}}^{(j)})$ expected for that area.
d) The **"Confidence measure"** block uses the output of the trajectory classifier (*Net2*, Section 4.4.1) for the user's current segment (from step 3b). Specifically, it identifies the confidence score $\varepsilon_{\hat{i}}(t_k)$ corresponding to the reference class $c_{\hat{i}}^{(j)}$ (retrieved in step 3c). This score is one of the outputs of the classifier's final softmax activation function, which represents the estimated probability that the current segment belongs to the reference class. As a result of the softmax function, this confidence score $\varepsilon_{\hat{i}}(t_k)$ is formally computed to be within the range $[0, 1]$, where a value close to 1 indicates a high match between the user's action and the expected manoeuvre, and a value close to 0 indicates a significant deviation.
e) This confidence score $\varepsilon_{\hat{i}}(t_k)$ is fed into the **"Walker control"** block. This block implements the visco-elastic shared control mechanism (detailed in Section 4.4.2), dynamically adjusting the control parameters (e.g., stiffness a, damping b, and arbitration factor λ via Eq. 4.7) based on the confidence score. High confidence leads to minimal robot intervention (user has control), while low confidence triggers assistive or corrective torques from the robot.

4.4 Model generation and behaviour–based control

The main pillars of our approach are an offline analysis of the environment that generates the behavioural map (i.e., the map of admissible behaviours for every area of the environment) and the online control module that adapts the shared authority controller to the degree of compliance of the user. The two modules are described next.

4.4.1 Behavioural map generation

Given the environment map, the behavioural map associates each area of the space with the class of trajectories (straight, right turn, left turn) possibly followed by humans when they behave "correctly". This information is generated in different steps.

In the first step, we generate a Probabilistic Road Map (PRM) [Kavraki et al., 1996] covering the entire space. The PRM provides collision free geometric paths connecting any pair of locations in the space. The PRM is generated ensuring an average density of 4 nodes per squared meter, which is a good trade-off between fine distribution of nodes and elaboration time of the paths (e.g. in a 5x5 meters room we have an average of 100 nodes).

In the second step, we consider pairs of random starting positions and ending positions, find the shortest path connecting them through the PRM, and interpolate the different nodes by clothoids. A *clothoid* is a line with curvature proportional to the arc-length described by the equation $X(s) = x_0 + \int_0^s \cos\left(\kappa' \frac{\tau^2}{2} + \kappa_0 \tau + \theta_0\right) \mathrm{d}\tau$, $Y(s) = y_0 + \int_0^s \sin\left(\kappa' \frac{\tau^2}{2} + \kappa_0 \tau + \theta_0\right) \mathrm{d}\tau$, where s is the curvilinear abscissa, (x_0, y_0) is the Cartesian coordinate of the initial point, θ_0 is the initial bearing, κ_0 is the initial curvature and κ' is the change rate of the curvature. The interpolation is done minimising the derivative of the squared curvature [Bertolazzi and Frego, 2018]. We can argue that the trajectories constructed in this way are a reasonable approximation of human-like trajectories, as supported by numerous results in the literature. The most important are in the work of Laumond et al. [2010], in which clothoids are explicitly addressed as a good approximation of human trajectories, and in the work of Arechevaleta et al. [2008b], who have shown that humans tend to minimise the derivative of the squared curvature when they move.

In the third step, the environment is discretised in a grid map using 1x1 meters cells. For each trajectory i intersecting a cell j, we identify a class $c_i^{(j)}$ in the finite set $\boldsymbol{c}$: $c_i^{(j)} \in \boldsymbol{c}$. For instance, one class could be "left turn" (L) or "move straight" (S). This operation is performed by the *Trajectory Classifier*, which allows us to partition each trajectory into a sequence of elementary moves (straight, left/right turn) and determine the class that identify each of them in every cell. To account for the different direction of motion of the i-th trajectory within the j-th cell,

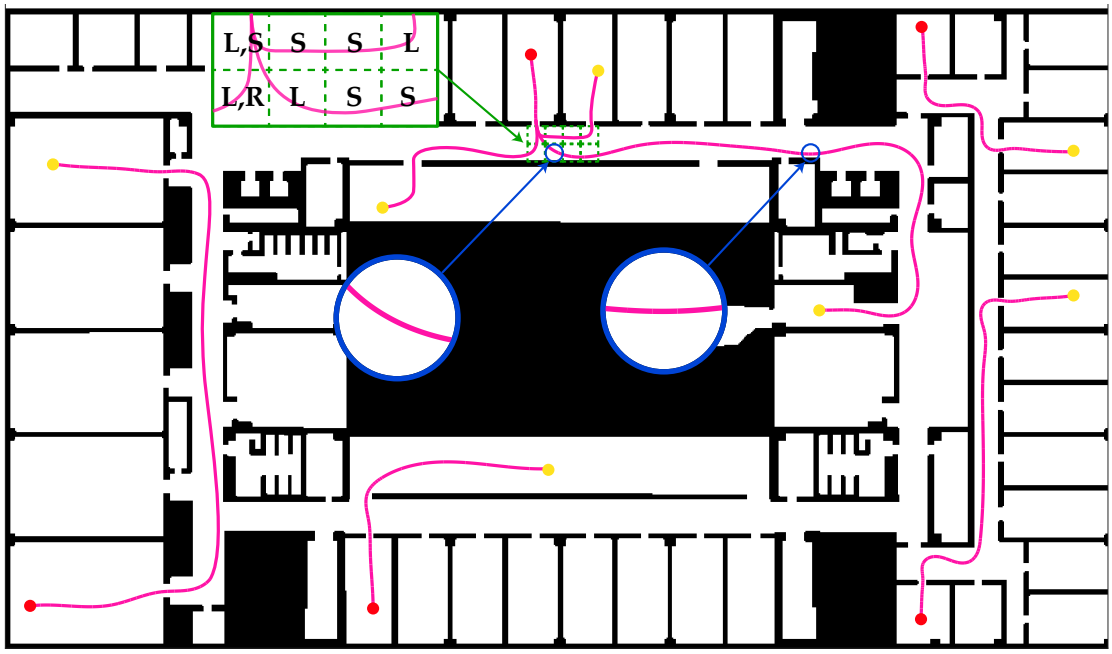


Figure 4.5: Synthetic trajectories generated with PRM and clothoids.

we associate the tangential direction $\theta_i^{(j)}$, which is the mean direction of travel of the vehicle in the j-th cell w.r.t the map reference frame, with the class $c_i^{(j)}$. For instance if the user is moving with the walker straight from west to east the tangential direction $\theta_i^{(j)}$ will be 0. The set of all the pairs $\left(c_i^{(j)}, \theta_i^{(j)}\right)$ form the behavioural map. In Figure 4.5 some of the synthetic trajectories are shown in magenta in a map of the Department of Engineering and Computer Science of the University of Trento, the grid map is shown in green and the corresponding sub-trajectories that will be fed to the classifier are highlighted with the blue circles. The starting points of the magenta trajectories are depicted with the red circles, while the ending points with the yellow circles. The fundamental motion primitives forming the behaviour map are shown in black capital letters in Figure 4.5.

The trajectory classifier. The trajectory features are extracted with an encoder neural network from the path geometry. More precisely, the k-th abscissa s_k of the path $\mathcal{R}$, sampled such that $s_k - s_{k-1} = \delta_s$ is constant, is used to define

the vector of geometric parameters

$$\boldsymbol{p}(s_k) = \begin{bmatrix} x_p(s_k) \\ y_p(s_k) \\ \cos(\theta_p(s_k)) \\ \sin(\theta_p(s_k)) \\ \kappa_p(s_k) \end{bmatrix}, \tag{4.2}$$

where $x_p(s_k)$ and $y_p(s_k)$ are the Cartesian coordinates, while $\theta_p(s_k)$ and $\kappa_p(s_k) = d\theta_p(s_k)/ds_k$ are the tangential axis and the curvature of $\mathcal{R}$ in $(x_p(s_k), y_p(s_k))$, respectively. To account for the path characteristics, n consecutive parameters are collected on the sampled abscissa coordinates $s_{k-(n-1)}$ to s_k, so as to build the matrix comprising $\boldsymbol{p}(s_{k-(n-1)})$ to $\boldsymbol{p}(s_k)$, which is then normalised to avoid spatial biases, i.e.

$$\boldsymbol{x}_p(t_k) = \begin{bmatrix} \left[x_p^1, \ldots, x_p^n\right] - x_p^1 \mathbf{1}^T \\ \left[y_p^1, \ldots, y_p^n\right] - y_p^1 \mathbf{1}^T \\ \cos\left(\theta_p^1\right), \ldots, \cos\left(\theta_p^n\right) \\ \sin\left(\theta_p^1\right), \ldots, \sin\left(\theta_p^n\right) \\ \kappa_p^1, \ldots, \kappa_p^n \end{bmatrix}, \tag{4.3}$$

where $\mathbf{1}$ is an 1-dimensional column vector with all ones, used for the normalisation of the position vectors, and we adopt the compact notation $x_p^i = x_p(s_{k-(n-i)})$. In order to avoid the problem of angular periodicity, we used both $\cos\big(\theta_p(s_{k-(n-i)})\big)$ and $\sin\big(\theta_p(s_{k-(n-i)})\big)$ instead of θ_p^i.

In the training process of the encoder, $\boldsymbol{x}_p(t_k) \in \mathbb{R}^{5\times n}$ is used as input. The weights of the encoder are learned by training an autoencoder and minimising the reconstruction error between $\boldsymbol{x}_p(s_k)$ and the reconstructed output $\tilde{\boldsymbol{x}}_p(s_k)$. The encoder and the decoder sub-networks of the autoencoder, have a symmetrical structure: the input $\boldsymbol{x}_p(t_k)$ passes through 3 convolutions and 3 fully-connected layers, resulting in a final latent space of $m = 5$ neurons. The decoder, then, has the same structure, but takes as input the latent space $\boldsymbol{z}_{k,\mathcal{R}} \in \mathbb{R}^5$. It is important to note that this 5-dimensional feature vector $\boldsymbol{z}_{k,\mathcal{R}}$ represents a compressed, abstract encoding of the trajectory segment's geometry learned by the network. The individual components do not have direct physical interpretations; their meaning

is derived from how the subsequent classifier (*Net2*) uses them to differentiate behaviours. While classifying trajectories into broad categories like 'Left-turn', 'Right-turn', and 'Straight' might seem achievable using direct geometric rules (e.g., based on average curvature or heading change), such approaches can be brittle. They often struggle with the inherent variability in human-like motion, where turns can have different radii, speeds, and subtle deviations. Hand-crafting rules robust to this variability is challenging and may require extensive tuning. The autoencoder addresses this by learning a compressed, latent representation ($\boldsymbol{z}_{k,\mathcal{R}}$) that captures the salient geometric features of the trajectory segment in a data-driven manner. This learned representation is potentially more robust to noise and minor variations than raw geometric parameters alone. Crucially, this learned, compressed representation also enables the same neural network architecture to effectively classify the user's current movement path in real-time, even from a limited number of initial samples. This facilitates early prediction of the user's manoeuvre, which is vital for responsive shared control. Furthermore, the softmax output of the final layer of the neural network can be exploited to scale the control amplitude based on the classification probabilities.

After learning the autoencoder weights, the decoder sub-network is discarded as we will use the latent space of the autoencoder as a compressed representation of the human behaviour (*Net1*). A second neural network (*Net2*) classifies such compressed representation $\boldsymbol{z}_{k,\mathcal{R}}$ into the behavioural classes in the set $\boldsymbol{c}$. Using a classifier (*Net2*) trained on these learned features allows for a flexible mapping from the nuanced latent representation to the discrete behavioural classes, potentially offering better generalization compared to thresholding simple geometric measures. More precisely, the behaviour is identified in the minimalistic set {Left-turn, Right-turn, Straight} and encoded by the numeric label $\boldsymbol{c} = \{1, 2, 3\}$. Hence, during the learning phase we define the transformation

$$\tilde{\boldsymbol{c}} = h_{\psi}\left(\boldsymbol{z}_{k,\mathcal{R}}\right), \tag{4.4}$$

where ψ is a set of parameters obtained by minimising the cross–entropy between the predicted $\tilde{\boldsymbol{c}}$ and the actual $\boldsymbol{c}$ class. In the architecture of the classifier network the input latent feature $\boldsymbol{z}_{k,\mathcal{R}}$ of 5 neurons passes through a single fully-connected

layer with just 3 neurons. Therefore, the combination of the neural networks maps the geometric characteristics of the path $\boldsymbol{x}_p(s_k)$ into the trajectory classes encoded in $\boldsymbol{c}$. This method of classifying expected behaviour based on learned geometric features provides the contextual reference needed for the adaptive control described later, offering an alternative to methods relying solely on instantaneous intent inference [Yu et al., 2005] or predefined rules. As a final step, the standard softmax$(\cdot)$ activation function is applied to the raw output scores (logits) o_1, o_2, o_3 corresponding to the three classes from the final layer of *Net2*. This function converts the logits into probabilities by taking the exponential of each score and normalizing by the sum of the exponentials: $\varepsilon_i = \text{softmax}(o_i) = \frac{e^{o_i}}{\sum_{j=1}^{3} e^{o_j}}$. The resulting values $\varepsilon_i(s_k)$ (or equivalently $\varepsilon_i(t_k)$) represent the confidence (probability) that the input segment belongs to class $c_i \in \boldsymbol{c}$. By definition, these confidences are in the range $[0, 1]$ and sum to 1: $\sum_{i=1}^{3} \varepsilon_i(s_k) = 1$. Notice that this same network is adopted to classify the synthetic generated paths and the current user behaviour, as will be explained in the next Section 4.4.2.

Geometry of the Grid In the discussion above we suggest a decomposition into a grid made of square cells. This choice is not mandatory. For other types of environments with the presence of static obstacles with non-rectangular shape, it could be more convenient to use a different type of cell decomposition (e.g., maximum clearance maps, maps resulting from plane sweep, etc.) [LaValle, 2006]. The technique proposed in this chapter would not be significantly affected by the choices of a different polygonal geometry.

4.4.2 Online Control

As a first step in the online control, the user selects their destination (p_f) starting from the initial point p_0 where the device is currently located. It is prudent to impose limits on the maximum traveled distance when dealing with individuals who may have limited mobility or other frailties. Equally important is the optimisation of routes connecting subgoals based on specific metrics [Bevilacqua et al., 2020]. In our current research, we assume that all pertinent decisions regarding these constraints and optimisation criteria have been predetermined before executing

our algorithm.

Following the same steps as for the behaviour map generation, the system connects the two points *via* the PRM and interpolates the intermediate points by using a G2 spline that minimises the derivative of the squared curvature. This allows us to determine for the current cell j the reference class and its orientation $(c_{\hat{i}}^{(j)}, \theta_{\hat{i}}^{(j)})$ among those found in the offline phase. Roughly speaking, this pair encodes the most sensible behaviour that a human would follow if they want to reach p_f from the cell j, and will be used to measure the degree of compliance of the human.

A custom path reconstruction module [Antonucci, Bevilacqua, et al., 2021] processes the odometry information received by the *FriWalk* and produces in real–time the user's current path segment $\mathcal{H}$ and its corresponding input representation $\boldsymbol{x}_s(s_k)$. This input is fed through the trained trajectory classifier (Net1+Net2, Section 4.4.1). As mentioned at the end of Section 4.4.1, the final layer of Net2 uses a softmax activation function. This function outputs three values, $\varepsilon_1(s_k), \varepsilon_2(s_k), \varepsilon_3(s_k)$, representing the estimated probabilities (confidences) that the current segment $\boldsymbol{x}_s(s_k)$ belongs to the classes {Left-turn, Right-turn, Straight}, respectively. By definition of the softmax function, these confidence values are in the range $[0, 1]$ and sum to 1 ($\sum_{i=1}^{3} \varepsilon_i(s_k) = 1$). Simultaneously, the reference class index $\hat{i}$ (corresponding to the expected behaviour $c_{\hat{i}}^{(j)}$) is retrieved from the *Behavioural map* for the current cell j along the planned path. The specific confidence value used for control, denoted $\varepsilon_{\hat{i}}(s_k)$ and computed by the *Confidence measure* block in Figure 4.4, is precisely the softmax output probability corresponding to this reference class $\hat{i}$. A high $\varepsilon_{\hat{i}}(s_k)$ (close to 1) signifies that the user's current motion strongly matches the expected behaviour for that location, while a low value (close to 0) indicates a deviation. Hence, this confidence $\varepsilon_{\hat{i}}(s_k)$ is used as the key hyper-parameter modulating the shared control intervention in the Walker.

The control module is designed synthesising a visco–elastic torque that is applied to the steering angle of the front wheels of the robot. The core control mechanism builds upon the concept of visco-elastic control [Andreetto et al., 2019a]. In that previous approach, the robot applied a corrective torque based on the deviation from a single, pre-planned reference path. This torque, resembling a

spring-damper system, was defined as $\tau = -a(\theta - \theta^\star) - b(\dot{\theta} - \dot{\theta}^\star)$, where $(\theta, \dot{\theta})$ is the current robot state (orientation and angular velocity) and $(\theta^\star, \dot{\theta}^\star)$ is the desired state derived from the reference path. The key aspect of that method was adapting the stiffness (a) and damping (b) parameters based purely on the geometric deviation from this single reference path – the further the deviation, the stiffer the controller became. While practically stable, this adaptation lacked context about why the user might be deviating or what behaviour is typical for a given area.

The central contribution presented in this chapter extends this concept significantly by introducing a behaviour-based, context-aware adaptation. Instead of relying solely on geometric deviation from one path, the system leverages the behavioural map (Section 4.4.1) to understand the expected manoeuvre ($c_{\hat{i}}^{(j)}$) for the current region to reach a given target. The core novelty lies in using the confidence $\varepsilon_{\hat{i}}(t_k)$ – the classified likelihood that the user's current action matches this expected behaviour – to dynamically modulate not only the visco-elastic parameters (a and b) but also the primary arbitration parameter λ itself, as detailed in Eq. 4.7. This allows the controller's intervention level and characteristics (stiffness/damping) to adapt based on the user's compliance with learned, contextually appropriate behaviours, rather than just their distance from an arbitrary planned reference line.

To implement this behaviour-based visco-elastic control, we first define the actual right (left) wheel steering angle relative to the robot chassis as α_r (α_l), which can be measured by an absolute encoder (we dropped the reference to the time t_k for ease of notation). The states are expressed in the robot reference frame $\langle R \rangle = \{X_r, Y_r, Z_r\}$, with X_R oriented along the longitudinal direction and Z_R pointing upwards. The desired relative wheel angles $\alpha_r^\star$ and $\alpha_l^\star$ (relative to the robot chassis), instead, can be obtained by the desired angular velocity $\omega^\star$ (derived from the planned reference path's curvature), the actual measured longitudinal velocity v (obtained by the encoders on the rear wheels) and the Ackermann steering geometry. Hence, the wheels orientation errors

$$e_{\alpha_r} = \alpha_r^\star - \alpha_r \text{ and } e_{\alpha_l} = \alpha_l^\star - \alpha_l,$$

can be immediately obtained. The component of the visco-elastic controller that

controls the torque to apply to the right wheel based on this relative angle error is determined as

$$\tau_{\alpha_r} = ae_{\alpha_r} + b\dot{e}_{\alpha_r}, \tag{4.5}$$

and the same for the left wheel to obtain τ_{α_l}.

While this controller component (τ_α) accounts for the local turning rate relative to the robot frame $\langle R \rangle$ (derived from the reference path's curvature $\omega^\star$), we also introduce a term to guide the robot towards the desired absolute orientation in the world frame $\langle W \rangle$. This desired absolute orientation, $\theta_{\hat{i}}^{(j)}$, is retrieved from the behavioural map for the current cell j and the reference class $\hat{i}$. We then compute the error between the desired absolute wheel direction and the current absolute wheel direction. For the right wheel, this error is:

$$e_{\beta_r} = (\theta^\star + \alpha_r^\star) - (\theta + \alpha_r).$$

The same logic applies to the left wheel error e_{β_l}. We then compute the corresponding torque components τ_{β_r} and τ_{β_l} using the same visco-elastic controller structure as in (4.5) (with gains a and b from Eq. 4.7). The final control laws combine the relative turning torque and this absolute direction torque:

$$\tau_r = \lambda\tau_{\alpha_r} + \tau_{\beta_r} \text{ and } \tau_l = \lambda\tau_{\alpha_l} + \tau_{\beta_l}. \tag{4.6}$$

It is worth noting the interaction between these torque components, particularly when confidence $\varepsilon_{\hat{i}}$ approaches zero, causing λ to approach one. In this scenario ($\lambda = 1$), the final torque becomes $\tau = \tau_\alpha + \tau_\beta$. The relative wheel angle error $e_\alpha = \alpha^\star - \alpha$ is effectively incorporated twice: directly through τ_α (which corrects the steering relative to the robot's frame based on the planned path's curvature $\omega^\star$) and indirectly through τ_β (which corrects the absolute wheel orientation $(\theta + \alpha)$ towards the desired absolute direction $(\theta^\star + \alpha^\star)$ derived from the behavioural map's $\theta_{\hat{i}}^{(j)}$). This summation signifies the robot exerting maximum corrective effort when user deviation is high. It simultaneously attempts to align the robot with the planned path's local curvature (τ_α) and steer it towards the globally expected direction for that region (τ_β), providing strong guidance back towards compliant behaviour. The visco-elastic gains a and b (whose adaptation is

detailed in Eq. 4.7 and the subsequent discussion on parameter tuning) are carefully chosen to ensure that this combined corrective influence remains moderate and user-friendly, avoiding overly aggressive interventions even in this maximum effort scenario.

The key contribution of this adaptive shared control approach lies in how the parameters λ, a and b are dynamically adjusted based on the user's compliance with expected behaviour, directly addressing the limitations of static arbitration discussed in Section 4.1.2. Specifically, these parameters are functions of the confidence $\varepsilon_{\hat{i}}(t_k)$ associated with the reference class $c_{\hat{i}}^{(j)}$ by means of

$$\begin{aligned} \lambda &= 1 - \varepsilon_{\hat{i}}(t_k), \\ a &= a_0 + a_1\lambda, \\ b &= b_0 + b_1\lambda. \end{aligned} \tag{4.7}$$

Here, λ acts as the primary arbitration parameter, blending the human's input (low λ, high confidence ε) with the robot's guidance (high λ, low confidence). Crucially, the visco-elastic parameters a (elasticity) and b (viscosity) are also modulated by this confidence-derived λ. This ensures that not only the amount but also the nature (stiffness/damping) of the robotic intervention adapts based on how closely the user's current trajectory segment matches the learned, contextually appropriate behaviour for that area ($c_{\hat{i}}^{(j)}$). This contrasts with simpler adaptive methods that might only adjust a blending ratio or use less nuanced triggers for adaptation. Specifically, a_0 and b_0 are the minimum coefficients used when the controller does not intervene (when the confidence is high). The a_1 and b_1 are modulated by the hyperparameter epsilon (the confidence). This means that increasing values of higher a_0 and b_0 will result in more intervention from the control, even when the human is performing expected movements. Similarly, the values of a_1 and b_1 widen or shrink the range of the applied control signal between when the system intervenes and when it does not. These parameters have to be fine-tuned by trial-and-error sessions on the specific application.

To summarise, we first compute the reference class $c_{\hat{i}}^{(j)}$, then from the actual state $\boldsymbol{x}_s(t_k)$ we compute $\varepsilon_{\hat{i}}(t_k)$ and then, by means of (4.7), the desired torques are computed with (4.6). The term b_0 is needed to avoid oscillatory behaviours while

a_0 is needed to generate the correct control signal that forces the wheel angle α_r (α_l) to the desired value. In this way, when the confidence is high (i.e., λ is low), the applied torque is predominantly imposed by the user and the computed torques τ_r and τ_l tend to zero. The system, instead, becomes increasingly authoritative (i.e., torques τ_r and τ_l imposed by the system) when λ gets closer to 1.

Management of obstacles and of exceptions The approach outlined above hinges on the definition of a reference behaviour for each cell (represented by the reference class and orientation $(c_{\hat{i}}^{(j)}, \theta_{\hat{i}}^{(j)})$ determined during the offline map generation, as described in Section 4.4.1) and on the application of visco-elastic control to make sure that the user does not deviate too much from this expected behaviour. Two type of exceptions can occur:
1. An unexpected dynamic obstacle (e.g., another human) materialises,
2. The user strongly opposes the suggestion and forces her/his way.
The first case is handled by using the so called reactive planning [Bevilacqua et al., 2018]: the system replans a new clothoidal trajectory that travels around the obstacle and joins into the reference trajectory as soon as the obstacle is overcome. This change has no significant impact on the framework: we can either use the visco-elastic control modulated by the likelihood ε or opt for a stiffer behaviour until the anomaly is over. For the second exception, we interpret the strong opposition of the user as her/his better understanding of the scenario. Therefore, we disengage the guidance system for a reconfigurable time. This choice does not apply if the user is travelling across areas that we deem dangerous (e.g., a stairway).

4.5 Experimental Validation and Results

Generation of the behavioural map. The experimental validation of the approach has been carried out in our Department premises. The first step of the approach was the construction of the behaviour map generating the described human-like synthetic trajectories (some examples are shown in Figure 4.5). For the training of *Net1* and *Net2* we focused on an area consisting of two intersecting corridors (conventional cross–intersection). We simulated 1800 paths selecting

Table 4.1: RMSE of *Net1* (Autoencoder) Reconstruction on the Validation Set of Synthetic Trajectories.

Metric Component	x (m)	y (m)	$\cos(\theta)$	$\sin(\theta)$	κ
RMSE	0.0076	0.0118	0.0293	0.0449	0.0241

Table 4.2: Classification Accuracy of *Net2* on the Validation Set of Synthetic Trajectories.

Class	Left	Right	Straight	Average
Accuracy	88.4%	88.3%	76.8%	84.3%

randomly pairs of waypoint positions p_0 and p_f. The simulations were equally partitioned in the Left-turn, Right-turn and Straight classes. We select $n = 12$ samples for the inputs $\mathbf{x}$, as shown in Eq. (4.3), this implies that the encoder will have 60 input features. A step size of $\delta_s = 0.1$ m has been chosen to sample the trajectory. A fraction of 80% of the dataset was used as training set, while the remaining samples were randomly selected for the validation. Both *Net1* and *Net2* were implemented in Keras and trained with the Adam optimiser with a learning rate of 0.001, batch size 64, and number of epochs 300 using a 2.7 GHz Intel Core i7 processor. *Net1* was trained using the set of $\mathbf{x}$ as both inputs and outputs of the network. Then, we transferred the learned weights of the encoder in the *Net2*, and performed a supervised training by comparing its estimates with the one-hot encoded labels of classes {Left-turn, Right-turn, Straight}.

In Table 4.1, we report the inference accuracy of the network *Net1* (the autoencoder) on the validation set, in terms of Root Mean Squared Error (RMSE) for trajectory reconstruction. The results show that the autoencoder was correctly trained on the dataset, and the even distribution of the error over the different components of the input indicates that no bias was produced in favour of a particular component.

Table 4.2 and Fig. 4.6 present the performance of *Net2* (the classifier), highlighting its classification accuracy across trajectory categories (left, right, and straight), with detailed insights into prediction accuracy and misclassifications. It can be noticed that the Left and Right classes obtained a higher percentage with respect

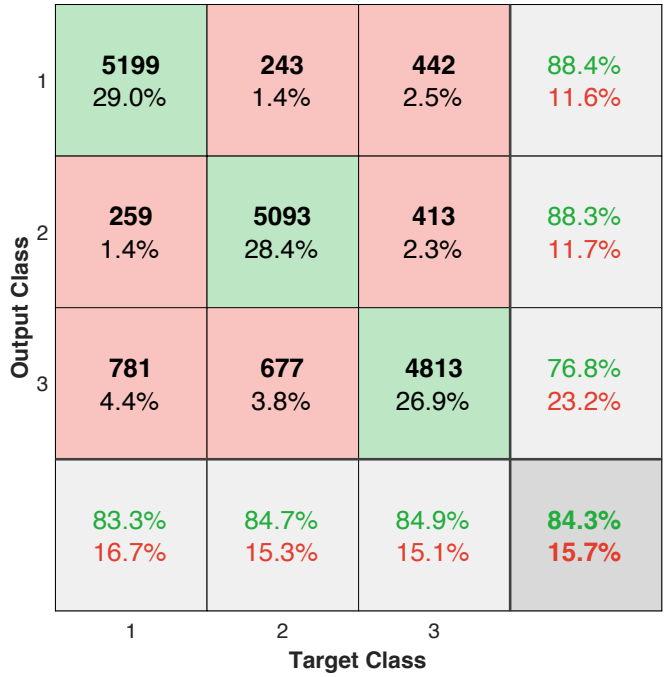
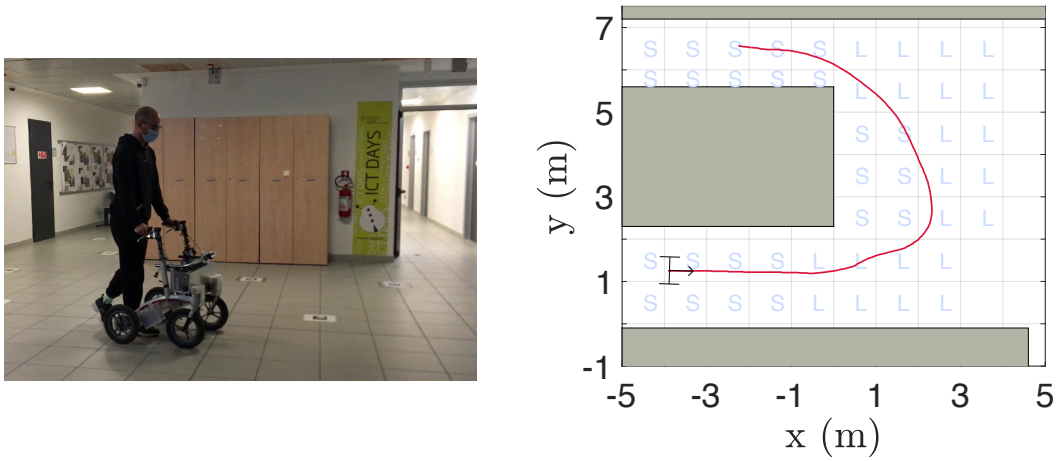


Figure 4.6: Confusion matrix illustrating the classification performance of *Net2* on the validation set. Each cell (i, j) represents the percentage of instances of true class j (column) predicted as class i (row). The diagonal elements show the percentage of total samples correctly classified for that class. The bottom gray row displays the Recall (True Positive Rate or per-class accuracy) for each true class. The rightmost gray column shows the Precision (Positive Predictive Value) for each predicted class, corresponding to the accuracy values reported in Table 4.2. The bottom-right gray cell indicates the overall average accuracy.

to the Straight class: the reason behind this behaviour is that the trajectories of the Straight class include features in common with the ones of the other classes (e.g., when the human slightly bends along an almost straight path). This is noticeable in Figure 4.5, where the sub-trajectories not always are distinguishable between turns and straight sectors.

4.5.1 Experiments with the *FriWalk*

The experimental evaluation of the approach presented in Section 4.3 and Section 4.4 was conducted on the real *FriWalk* in an indoor hallway at the University of Trento. A collection of ArUco markers was placed in the testing area, which has



(a) (b)

Figure 4.7: (a) Photo of the experimental area and (b) the associated behavioural map.

a dimension of roughly 7×7 m, allowing the walker to localise itself with sufficient accuracy (error below 20 cm). A ROS interface was used to send the control to the actuators and to receive the localisation data, including the odometry-based estimates of $\boldsymbol{q}(t_k)$ in $\langle W \rangle$ and the angular position of the wheels α_r and α_l in $\langle R \rangle$.

The parameters a and b for the visco-elastic controller, as explained in Section 4.4.2 and Eq. (4.7), are functions of the confidence $\varepsilon_{\hat{i}}(t_k)$. Based on experience with the system, the specific values used in these experiments were set as follows: for the component related to relative angular velocity error (e_α), $a_0 = 25$ N, $a_1 = 15$ N, $b_0 = 15$ Ns, and $b_1 = 10$ Ns; for the component related to absolute direction error (e_β), $a_0 = 25$ N and $b_0 = 25$ Ns (with a_1 and b_1 implicitly zero for this component as per Eq. 4.6). Changing these base parameters (a_0, b_0) and modulation gains (a_1, b_1) modifies the baseline stiffness/damping and the sensitivity of the robot's intervention to the confidence measure.

We fixed the initial and final waypoint areas for the tests and executed offline the behavioural path planning described in Section 4.4.1, obtaining the behavioural map. We then executed several trials of the same mission, varying the general behaviour of the human experimenter between three macro categories: following diligently the predefined mission, following the mission roughly and deviating from

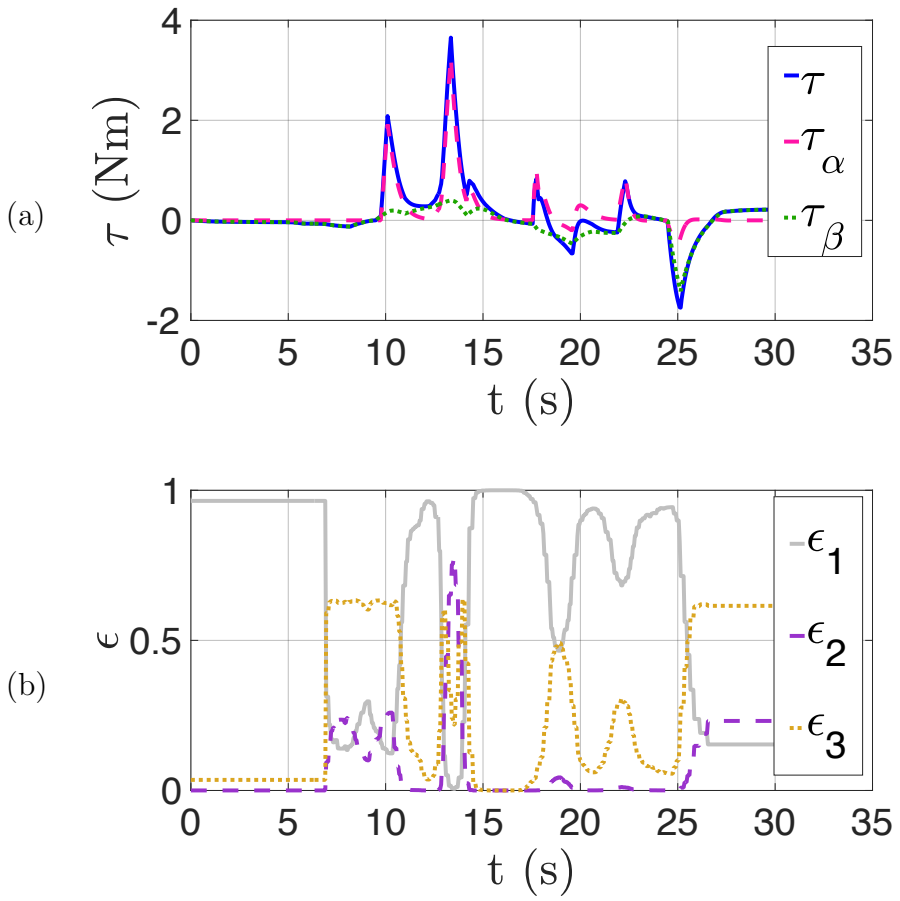


Figure 4.8: (a) Torque controls for e_{α_r} and e_{α_l} (magenta-dashed line) and for e_{β_r} and e_{β_l} (green-dotted line) applied to the walker front wheels while performing the trajectory in (Fig 4.7-b). (b) confidence for Left-turn (grey-solid line), Right-turn (purple-dashed line) and Straight (yellow-dotted line).

the mission. Figure 4.8-a shows the control action of the robot while the human moved for the leftmost corridor towards an exit on the upper part of the map (see Figure 4.7-b).

After 10 seconds from the beginning of the experiment, the user kept walking straight an area where the Left-turn class was instead foreseen: the low likelihood on the Left behaviour (grey line in Figure 4.8-b) triggered a compensating action on the control signal τ_{α_r} and τ_{α_l} in (4.6) (dashed magenta curve in Figure 4.8-a), thus causing a compensation in the trajectory. This intervention results into the

increasing likelihood of the Left-turn behaviour class (grey line in Figure 4.8-b). Similarly, as the human tried to steer right after 13 seconds, the corresponding Right-turn behaviour was caught (purple dashed line in Fig. 4.8-b) and the authority was again transferred to the robot, i.e., the human was progressively pushed towards the correct turning behaviour. Notice that when the compensation action occurs, the human user corrects the erratic behaviour in a few instants, indulging the robot action and indirectly lowering the control action. Hence, the robot action is perceived as a brief suggestion that vanishes immediately if the user follows the change of the route, otherwise the control action will persistently assist the manoeuvre towards the correct direction.

In Figure 4.9, we depict the performance of the control for three different user's behaviours.

When the person is compliant with the planning (blue trajectory), the control does not intervene, so the person is fully in charge and does not feel any opposing action from the robot. When, instead, the user purposely acts against the planned path, the control actions are extremely evident (orange trajectory). Finally, in the most typical case, the control acts loosely without excessively forcing the path correction (green trajectory), keeping the motion in the appropriate direction.

4.5.2 User evaluation

Since this work hinges for a large part on human-robot interaction, a qualitative evaluation was needed to validate user acceptability. All participants were informed about the ethical approval and their right to withdraw at any time, and provided consent before performing the tasks with the FriWalk.

The user evaluation was conducted with 16 adult participants (11 males, 5 females, ages 21–50, all from the University of Trento, none of whom use walking aids in daily life). Each participant completed a single experimental session lasting up to 15 minutes. During the session, they tested two navigation techniques with the *FriWalk*: the proposed behaviour-based control and a classic visco-elastic control [Andreetto et al., 2019a]. The order in which the techniques were presented was randomized to avoid bias, and the methods were labeled as "A" and "B" so that participants were unaware which was the new approach.

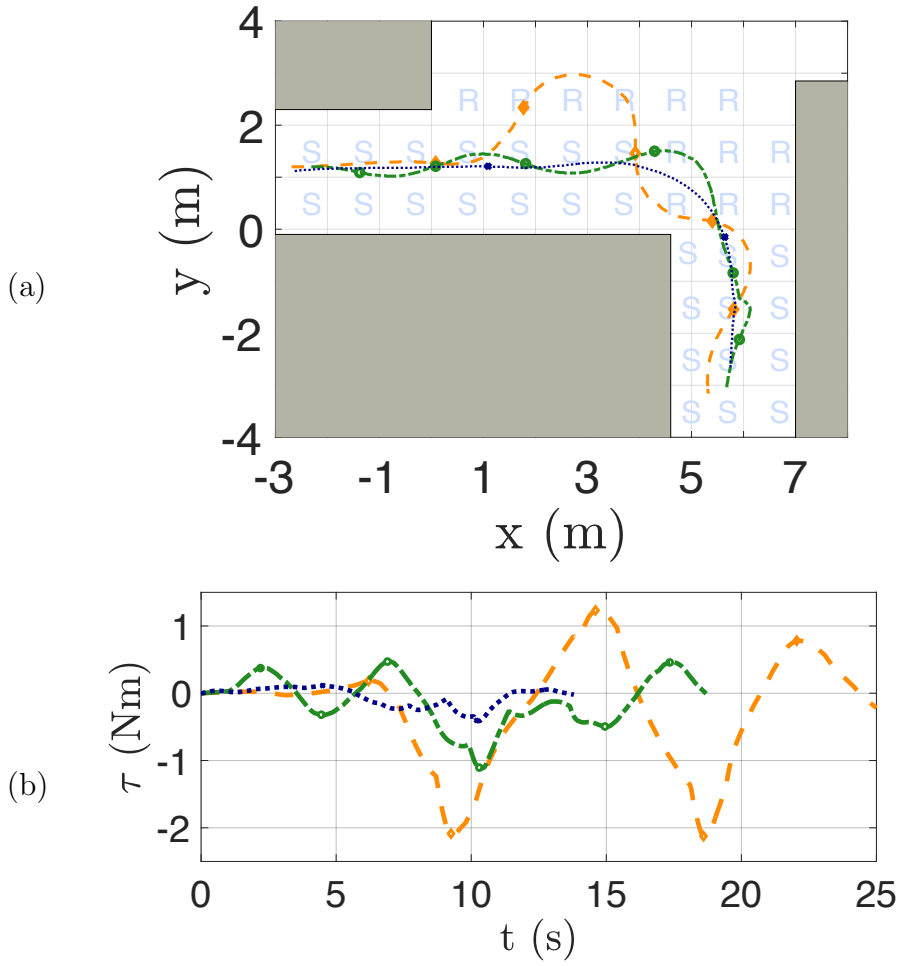


Figure 4.9: Experimental evidence of the control action behaviour in case of a user acting purposely against the desired path (orange lines), making slight deviations (green lines) or adhere to the planned path (blue lines). The resulting path (a) and the relative control actions (b) are reported.

For each technique, participants performed two tasks in the same indoor environment (see Figures 4.7 and 4.9), with identical start and end points: first, they were asked to walk compliantly to the target, and then to move erratically or even deliberately deviate from the intended path. Participants could repeat the navigation tests if they wished (none did), and were responsible for avoiding obstacles, as neither technique included obstacle avoidance.

After completing both trials, participants filled out a short questionnaire com-

Table 4.3: User evaluation (yes)

Question	Visco–elastic control	Behavioural maps control
Was it evident that was the walker to decide the path to follow?	87.5%	12.5%
Have you felt to be pulled, pushed or stuck?	37.5%	12.5%

Table 4.4: User evaluation (mean - standard deviation)

Question	Visco–elastic control	Behavioural maps control
Was experience with the walker pleasant?	3.38 - 0.92	4.75 - 0.46
You had the impression you had no control?	2.88 - 0.83	1.38 - 0.74
The walker hindered/prevented your usual way of walking?	1.63 - 0.74	1.00 - 0.00

paring their experiences with the two control strategies. The questionnaire included both yes/no questions (Table 4.3) and questions rated on a 1–5 scale (Table 4.4), where 1 means "not at all" and 5 means "extremely." The results are reported as percentages for yes/no questions and as mean and standard deviation for scaled responses.

This evaluation protocol allowed us to assess both the perceived comfort and cognitive aspects of using the walker under each control strategy, as well as overall user preference.

The results of the questionnaire are summarized in Tables 4.3 and 4.4.

From the results in Table 4.3 we can deduce that the Behavioural Maps' control is less intrusive, aiding the user's navigation without sacrificing her/his comfort. Through the questions reported in Table 4.4, we could evaluate the cognitive aspects derived from the experience of using the walker. We can observe a good level

of accordance between the 12.5% reported in both questions of Table 4.3 for the Behavioural map control and 1.38 reported in the second question of Table 4.4. Likewise, the low performance reported for the visco-elastic control in the first question of Table 4.4 is an evident consequence of its perceived level of authority and intrusiveness reported in Table 4.3. The evident conclusion is that the impression of retaining at least partial control has a clear positive impact on the user's experience.

Moreover 100% of the participants preferred the control strategy proposed in this chapter over the classic visco–elastic control. Some of the motivations were that our method gives more autonomy and freedom to perform any path while the turns were performed more softly, without forcing the participant to a particular trajectory.

In this section, we have shown a complete experimental evaluation both from the perspective of the quantitative performance and of the user experience. In both cases, the results are very good and prove that this framework provides navigation assistance, which guarantees a good level of agreement of the user trajectories with socially acceptable behaviours limiting at the same time the level of interference of the system with the user's choices.

Chapter 5

Human-Aware Motion Planning for Robot Manipulators

Predicting human motion in real time is key to safe human-robot collaboration. Traditional approaches, such as Gaussian Mixture Models (GMMs)[Mainprice and Berenson, 2013], work in controlled scenarios but struggle with complex human behaviour. Deep learning (DL) methods, including RNNs, GCNs, and Transformers, address these limitations by improving prediction accuracy[Tian et al., 2024; Yang et al., 2024], though real-time application is challenging due to computational demands.

Traditional motion planning methods ensure safety but lack adaptability for dynamic interaction. Proactive approaches anticipate human actions, with methods like temporal PRM enhancing collaboration [Hüppi et al., 2022]. This chapter compares clustering and deep learning for human motion prediction, aiming to integrate predictions into a human-aware planning framework for improved safety and responsiveness.

5.1 The UR5e Platform

The research on skeleton trajectory prediction and human-aware manipulator motion planning presented in this chapter is experimentally validated using the *UR5e* collaborative robot. The *UR5e* is a collaborative robotic arm designed to support

human-robot interaction by enabling safe operation in shared workspaces, with the robot adapting its motion based on predicted human actions. A detailed technical overview of the *UR5e* platform, including its hardware and software architecture, was provided in Chapter 2.

5.2 State-of-the-Art in Human Behaviour Understanding

This section reviews existing literature relevant to understanding and predicting human behaviour in the context of HRI, covering specific motion prediction techniques.

5.2.1 Human Motion Prediction

Predicting human motion is a fundamental aspect of effective human-robot collaboration, encompassing both understanding the spatial context of human activity and anticipating individual trajectories. In this chapter, we explore a key facet of human motion prediction: the prediction of skeleton trajectories for collaborative tasks with manipulators.

Traditional Methods

Traditional methods for human motion prediction (HMP) frequently utilize statistical models to anticipate future movements based on historical data. For example, Wiest et al. [2012] present a probabilistic framework for predicting vehicle trajectories, a concept that can be adapted to robotics for forecasting human motion paths. Mainprice et al. [2013] expand upon this approach by applying Gaussian Mixture Models (GMMs) within human-robot interaction contexts, enabling early predictions of human motion to facilitate the generation of safe robotic trajectories. Similarly, Darpino et al. [Pérez-D'Arpino and Shah, 2015] integrate GMMs with Dynamic Time Warping (DTW) to accurately forecast human reaching targets, enhancing collaboration in shared environments.

Although these methods perform well in structured environments, they face significant challenges when confronted with the inherent variability and nonlinearity of human motion. Furthermore, their scalability to large datasets and ability to generalize to unseen behaviours remain limited. This chapter seeks to address these limitations by augmenting the predictive capabilities of traditional statistical models with advanced computational methodologies.

Deep Learning Approaches

Deep learning (DL) methodologies have gained significant traction in human motion prediction (HMP) due to their capacity to capture intricate temporal and spatial interdependencies. Recurrent Neural Networks (RNNs), along with their various extensions, are frequently employed for modeling sequential pose information, whereas Convolutional Neural Networks (CNNs) are adept at extracting spatial correlations. Recent developments have introduced Graph Convolutional Networks (GCNs) and Transformer-based architectures, which harness the structural and temporal characteristics of skeleton-based data to deliver state-of-the-art results [Tian et al., 2024; Yang et al., 2024].

Graph-oriented approaches, such as Graph-Mixer, capitalize on the skeletal framework to enhance motion prediction, while Transformer-based architectures, like TransFusion, demonstrate proficiency in managing long-range dependencies. Although these advanced methods achieve notable performance, their high computational demands present obstacles for deployment in real-time scenarios. By juxtaposing these DL approaches with clustering-based techniques, this chapter examines the trade-off between computational efficiency and predictive accuracy.

Prediction of Skeleton Trajectories

In industrial settings, human motion prediction takes on a different role, focusing on the anticipation of skeleton trajectories. These trajectories are sequences of joint positions and orientations representing a person's movement over time. Accurate prediction of these trajectories is critical in collaborative tasks where manipulators, such as the UR5e, interact closely with humans.[Liu et al., 2023; Rudenko et al., 2020]

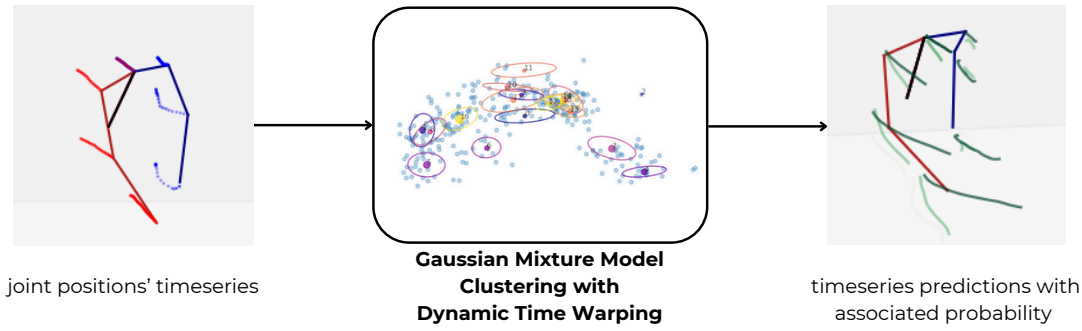


Figure 5.1: Diagram of the clustering model used for human gesture prediction.

For example, in a collaborative manufacturing process, the robot may work alongside a user assembling a product. In this scenario, the ability to predict the user's skeleton trajectory allows the robot to proactively avoid collisions while assisting in the task. By anticipating the user's next movements, the robot can adjust its path or task execution to ensure safety and efficiency. This proactive behaviour minimizes interruptions and fosters seamless human-robot collaboration.[Flowers et al., 2023; Merckaert et al., 2024]

Skeleton pose prediction relies on advanced motion modelling techniques, including machine learning approaches that analyze historical motion data and contextual cues. Neural architectures such as recurrent neural networks (RNNs) or transformers are commonly used for this purpose, as they excel at capturing temporal dependencies and modelling complex motion dynamics. By integrating these predictions into its motion planning, the manipulator achieves a level of situational awareness that is essential for safe and effective operation in shared workspaces.[Rudenko et al., 2020; Tian et al., 2024]

5.3 On-line Human Motion Prediction

Human motion prediction forms the backbone of our methodology. Human motion prediction aims to forecast future skeletal pose sequences from observed data. Formally, given an observed pose sequence $X_{1:N} = (x_1, ..., x_N) \in \mathbb{R}^{N \times J \times D}$ of length N, where J is the number of joints and D is the joint dimension, the goal is to predict

future poses $X_{N+1:N+T} = (x_{N+1}, ..., x_{N+T}) \in \mathbb{R}^{T \times J \times D}$ for T future frames. Accurate and efficient forecasting of human movements enables robotic systems to anticipate human actions and dynamically adjust their trajectories, enhancing safety, efficiency, and human-robot interaction quality. To achieve this, we implement two complementary approaches: clustering-based methods for interpretable predictions and deep learning-based methods for capturing complex spatial-temporal dependencies.

Through this comparative analysis, we aim to delineate the trade-offs between simplicity and complexity, offering insights into the optimal deployment contexts for each methodology. This holistic evaluation enables a balanced integration of both approaches, leveraging their respective advantages to enhance the overall effectiveness of human-aware motion prediction.

5.3.1 Clustering Prediction

Clustering-based approaches serve as efficient and interpretable solutions for real-time human motion prediction in repetitive tasks. These methods leverage the structured nature of human movements to group similar gestures and forecast future trajectories.

Data Segmentation and Clustering The first step in our clustering framework involves segmenting human movements based on the velocity profiles of hand trajectories. Transitions between gestures are identified using abrupt changes in velocity vectors. Each segmented motion sequence is represented as a time series, with T representing the time samples and J representing the number of human joints in 3D Cartesian space (Fig. 5.2).

To address the high dimensionality of this data, we employ Principal Component Analysis (PCA), which projects the data onto a reduced feature space of $T \times N$, where N is the number of retained principal components.

Gaussian Mixture Model Initialization and Inference Gaussian Mixture Models (GMMs) are fundamental tools for clustering and regression in our framework due to their flexibility in modelling complex, multimodal data distributions.

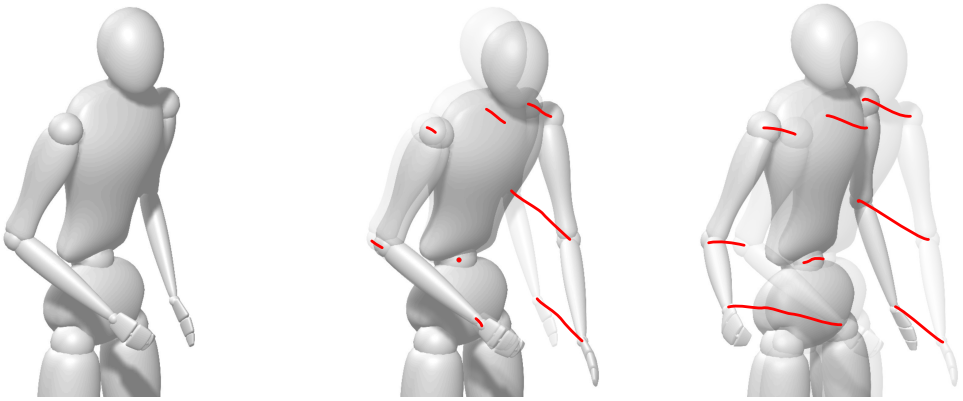


Figure 5.2: Illustration of two segments obtained from the segmentation of human movement data. The process identifies gesture transitions by analyzing velocity profiles of hand trajectories, with abrupt changes in velocity vectors marking these transitions. Each resulting segment is represented as a time-series of 3D joint positions. The specific action from which these segments are derived is shown with key poses: a starting position, an intermediate position where an object is picked from the table, and a final position similar to the start. The two segments capture distinct phases of this action, such as the movement leading to and following the object interaction.

These models combine multiple Gaussian components to approximate the underlying structure of the data. To achieve optimal performance, the initialization of the GMM plays a crucial role, as it directly impacts the convergence behaviour and the quality of the solution obtained through the Expectation-Maximization (EM) algorithm.

Two initialization strategies were investigated: random sampling and K-Means clustering. In the random sampling approach, the initial parameters of the Gaussian components are randomly assigned, leading to a diverse range of starting conditions for the EM algorithm. While this method provides simplicity, it often results in suboptimal clustering performance and requires additional iterations to converge. Conversely, the K-Means initialization technique utilizes a pre-clustering step to identify preliminary cluster centers by grouping data points based on their spatial proximity. These initial estimates serve as a more accurate starting point for the EM algorithm, significantly improving its convergence rate and reducing the risk of becoming trapped in local optima.

The results of our experiments demonstrated that K-Means initialization consistently outperformed random sampling in terms of prediction accuracy for our dataset. Specifically, the use of K-Means yielded lower prediction errors by effectively aligning the initial cluster centers with the actual distribution of the data. This alignment not only enhanced the efficiency of the EM process but also contributed to more reliable and interpretable clustering results.

During real-time inference, the closest cluster is identified based on the current motion data. Subsequently, Gaussian Mixture Regression (GMR) is applied to conditionally predict the future trajectory of the human's joints. The responsibilities and probabilities computed by the GMM are used to generate confidence scores for these predictions, allowing the system to adaptively weigh its outputs and ensure robust trajectory forecasting.

Dynamic Time Warping (DTW) for Temporal Alignment To enhance generalizability, we apply Dynamic Time Warping (DTW) during clustering. DTW allows for the alignment of trajectories with varying execution speeds, enabling a single cluster to represent multiple motion profiles that share the same geometric path. This reduces the model's complexity while maintaining prediction accuracy.

The clustering method offers prediction capabilities that can approach real-time performance, but certain limitations must be addressed to achieve consistent efficiency. Specifically, the inference time is highly sensitive to the number of clusters and the dimensionality of the feature space. As the number of clusters increases, or if Principal Component Analysis (PCA) retains too many dimensions (resulting in a high value of N), the computational cost of inference rises significantly. This can hinder real-time applicability, particularly in scenarios where rapid decision-making is critical.

To mitigate these challenges, a careful trade-off must be established between the quality of prediction and the computational efficiency of inference. Retaining a large number of clusters or high-dimensional representations generally improves the granularity and accuracy of predictions. However, this comes at the expense of increased inference time, which may not be acceptable for applications with stringent latency requirements. Conversely, reducing the number of clusters or dimensionality can enhance computational speed but risks oversimplifying the data

and degrading the predictive performance.

A key focus of this approach is to optimize the balance between these competing objectives. PCA can be employed to reduce the dimensionality of the data, but the number of principal components retained must be carefully selected to preserve the essential variability of the data while keeping the computational load manageable. Similarly, the number of clusters in the Gaussian Mixture Model (GMM) should be chosen to strike a balance between the resolution of the predictions and the time required for inference. Additionally, selecting an excessively high number of clusters can lead to overfitting, where the model becomes overly tailored to the training data and fails to generalize effectively to unseen scenarios. This overfitting not only undermines the reliability of predictions but also exacerbates the computational cost, further complicating real-time applicability. Therefore, finding an optimal number of clusters is essential to maintaining a robust and efficient prediction framework.

5.3.2 Deep Learning Prediction

Deep learning-based methods offer an alternative approach for human motion prediction, particularly when the underlying dynamics involve complex non-repetitive spatial-temporal patterns. These models leverage recent advances in graph-based and transformer architectures to achieve state-of-the-art performance. The primary motivation for exploring deep learning models is their ability to capture intricate dependencies between skeletal joints over time, which traditional clustering methods may struggle to model. The Graph-Mixer model and TransFusion architecture were selected based on their demonstrated effectiveness in handling spatial-temporal relationships.

Graph-Mixer Model The Graph-Mixer model is a sophisticated framework designed for processing 3D pose sequences, leveraging graph-based methodologies to capture spatial and temporal dependencies effectively. The model architecture comprises three key components:
(1) an Adaptive Spatial Graph Convolution layer that encodes pose information by embedding joint features into a spatial graph representation,

(2) a Spatial-Temporal Graph-Mixer that integrates spatial and temporal information to model both the interdependencies among joints and their temporal dynamics across sequences, and
(3) a prediction head responsible for generating future poses based on the learned representations.

The model employs three distinct adjacency matrices to represent the relationships between joints: a predefined matrix based on the skeletal structure, a learnable matrix optimized during training, and an adaptive matrix that dynamically adjusts based on input data. These matrices serve to encode varying levels of connectivity, capturing both static anatomical relationships and dynamic interactions. Despite the inclusion of these advanced features, experimental results indicate that relying solely on the predefined skeletal adjacency matrix achieves comparable accuracy to configurations using all three matrices. This simplification not only maintains predictive performance but also substantially reduces computational complexity, making the model more efficient and scalable for real-time applications.

Furthermore, the design of the Graph-Mixer emphasizes adaptability and interpretability. By leveraging a graph-based representation, the model can naturally incorporate the hierarchical structure of the human skeleton while capturing nuanced joint interactions. The spatial-temporal Graph-Mixer component effectively disentangles spatial correlations from temporal progression, enabling the model to generalize across diverse motion patterns.

TransFusion: Transformer-Based Diffusion Model TransFusion represents a cutting-edge approach to long-term motion prediction, integrating Transformer layers with a diffusion process to effectively model complex temporal dependencies in motion data. The architecture leverages the strengths of Transformers in capturing long-range correlations and the denoising capabilities of diffusion models to generate accurate and coherent motion sequences.

The input motion data undergoes preprocessing through the Discrete Cosine Transform (DCT), which projects the temporal sequence into the frequency domain. This transformation serves two primary purposes: it reduces noise by filtering out high-frequency components often associated with measurement artifacts

and compresses the data, effectively lowering its dimensionality. By focusing on the dominant frequency components, the model can efficiently learn the underlying motion patterns without being hindered by irrelevant noise or excessive computational demands.

At the core of TransFusion is a denoiser network, which iteratively refines the predictions generated during the diffusion process. The Transformer layers within this network excel at capturing the intricate temporal relationships present in the motion data, enabling the model to reconstruct highly accurate and plausible motion trajectories. Once the denoising process is complete, the predictions are mapped back to the temporal domain using the inverse DCT, restoring the original time-series representation of the motion.

By leveraging these models, our system is designed to achieve a balance between predictive accuracy and computational efficiency, making it suitable for a wide range of motion prediction tasks. Among the approaches considered, the Graph-Mixer model has emerged as the more practical choice for real-time applications, owing to its ability to maintain robust performance while significantly reducing computational overhead.

To validate this claim, we conduct a detailed comparison between the Gaussian Mixture Model (GMM)-based method and the Graph-Mixer model, focusing on their respective trade-offs between accuracy and inference time. The GMM method offers a lightweight and interpretable framework, particularly advantageous for systems with strict computational constraints. However, its performance can degrade in scenarios involving complex motion dynamics or high-dimensional data, where the Graph-Mixer's ability to model joint dependencies and temporal relationships proves superior.

While the TransFusion model demonstrates high predictive accuracy, its computational demands make it less suitable for real-time scenarios and therefore outside the scope of our real-time system analysis. By narrowing our focus to the GMM and Graph-Mixer methods, we aim to highlight the strengths and limitations of these models in the context of real-time motion prediction.

We trained and tested our system using the publicly available HA4M dataset [Cicirelli et al., 2022]. The HA4M (Human Action Multi-Modal Monitoring in Manufacturing) dataset provides multi-modal data from subjects performing a realistic

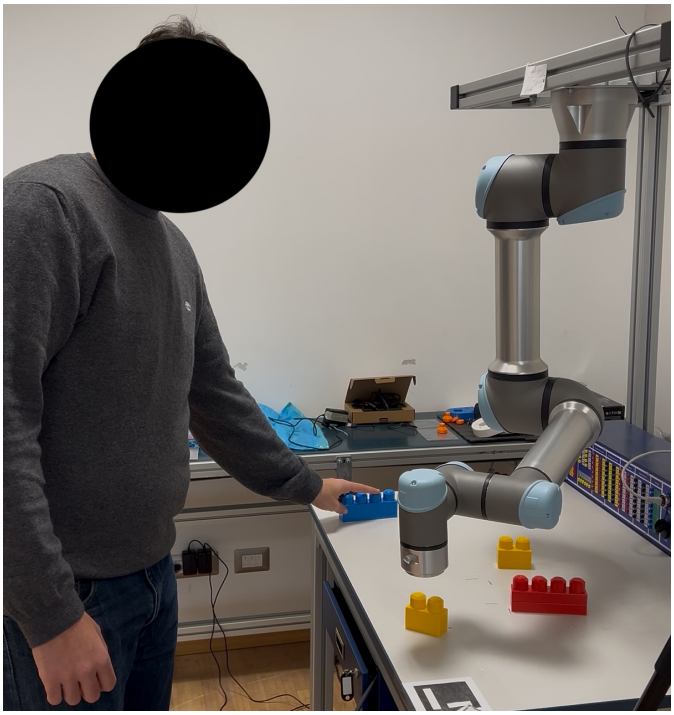


Figure 5.3: Experimental setup for human-aware motion planning. A human operator and the UR5e manipulator (mounted upside-down) are positioned one in front of the other, each performing concurrent pick-and-place tasks on a table. The close proximity between human and robot facilitates collaborative interaction and provides a challenging scenario for evaluating predictive planning strategies.

assembly task. For this chapter, we specifically utilized the skeleton tracking data, which provides the 3D positions ($D = 3$) of $J = 25$ body joints recorded at approximately 30 frames per second. This data captures the relatively complex, high-dimensional movements involved in the assembly process.

In this chapter, we utilize the skeleton data from this dataset to simulate a concurrent pick-and-place task. In this scenario (Fig. 5.3), a human and a robot are positioned opposite each other, each independently performing repetitive pick-and-place actions on different objects. The primary objective is to ensure that both agents effectively avoid interfering with each other's operations, thereby facilitating seamless human-aware interaction.

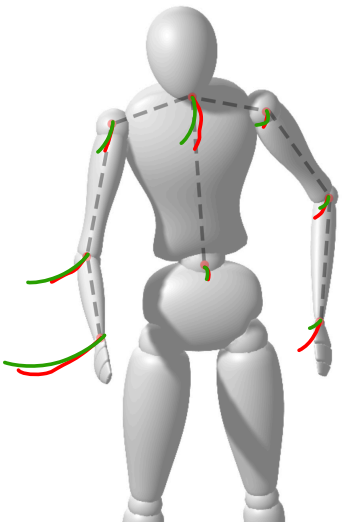
5.3.3 Training of the Models

The models were trained and evaluated using multiple sequences of the assembly task available in the HA4M dataset. For the GMM approach, we decided to keep 10 principal components after the PCA and create 100 clusters. These hyper-parameters were selected based on an ablation study balancing performance and complexity. The length of the segmented gestures used for clustering is variable (due to natural variations in execution speed) and handled by the DTW alignment, but typically corresponds to approximately 2 seconds of skeleton trajectory data recorded at 30 fps.

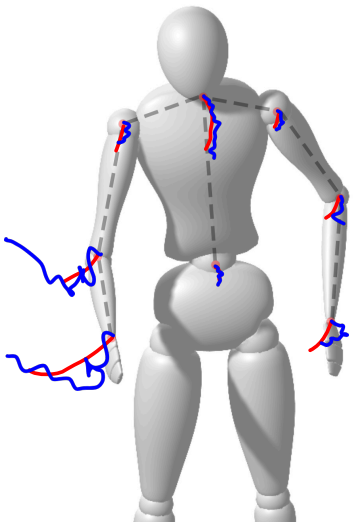
For the deep learning (DL) approaches, we standardized the input and output lengths. We decided to use 12 frames of historical motion (approximately 0.4 seconds) as input, chosen to provide sufficient context for capturing the dynamics of the assembly task gestures without excessive computational cost. The models were trained to predict 60 frames (corresponding to 2 seconds, matching the approximate gesture length and providing a challenging "long-term motion" prediction horizon) of future motion. The Graph-Mixer network was trained for 20 epochs, while the TransFusion network was trained for 500 epochs.

5.3.4 Evaluation of Human Motion Prediction Models

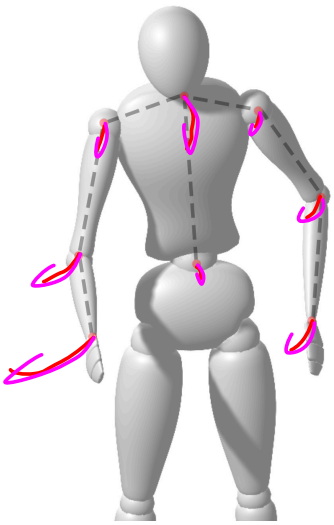
A qualitative evaluation and comparison of the GMM, TransFusion, and Graph-Mixer predictions is shown in Figure 5.4. From Figure 5.4a (GMM method), we observe that the GMM method effectively predicts the general direction of human motion, although it exhibits higher error rates for limbs with less movement. In contrast, as shown in Fig. 5.4b, TransFusion introduces high-frequency noise in its predictions, rendering it unsuitable for real-time applications. Additionally, TransFusion suffers from discontinuities over time—while not evident in the figure, the predictions can change significantly from one frame to the next, further diminishing its reliability for consistent motion prediction. The Graph-Mixer model, illustrated in Fig. 5.4c, demonstrates smooth predictions while accurately capturing the overall motion patterns. Notably, it also predicts the returning motion, highlighting the model's ability to effectively understand and anticipate motion dynamics. The HA4M dataset has complex dynamics and high-dimensional data,



(a) GMM method predictions (green trails).



(b) TransFusion method predictions (blue trails).



(c) Graph-Mixer method predictions (magenta trails).

Figure 5.4: Qualitative comparison of human motion prediction methods. For all subfigures: black dashed lines are human body links, red trails are ground truth joint trajectories. Predictions for each method are shown with distinct colored trails.

since it involves 25 joints moving in 3D space during an assembly task, recorded at 30 fps. We have "long-term motion" predictions since our task forecasts 60 frames (2 seconds) into the future based on 12 frames (0.4 seconds) of history, which is challenging due to the potential for significant changes in movement over that horizon.

We evaluated our methods using the Mean Per Joint Position Error (MPJPE), a standard metric in 3D human pose prediction. This metric calculates the average Euclidean distance between the predicted and actual joint positions in 3D space. The MPJPE is calculated by computing the error for each joint, averaging across all joints, and then averaging across all frames in a sequence. Formally:

$$MPJPE = \frac{1}{N}\sum_{i=1}^{N}\|\hat{J}_i - J_i\| \tag{5.1}$$

where N is the number of joints, $\hat{J}_i$ is the predicted 3D position of the i-th joint, J_i is the ground truth 3D position of the i-th joint, and $\|\cdot\|$ denotes the Euclidean distance between the predicted and ground truth joint positions. In Table 5.1 we report the MPJPE, variance, the 5th percentile, and the 95th percentile results for the three models.

Table 5.1: Human motion prediction methods comparison

Method	MPJPE	Standard Deviation	5th Percentile	95th Percentile
Clustering	71.60	1.46	7.13	182.25
TransFusion	100.81	76.51	20.57	251.47
Graph-Mixer	111.27	101.89	14.44	343.57

Note: All values are in millimeters [mm]. Results are derived from evaluating the models on multiple sequences of a realistic assembly task from the HA4M dataset [Cicirelli et al., 2022], involving 25 body joints performing pick-and-place-like movements. The dataset was recorded at 30 fps, and the natural speeds of these assembly movements are inherent in the evaluation data.

The Graph-Mixer model demonstrates robust predictive capabilities, achieving a mean positional error of 111.27 mm when forecasting 60 future poses at a frequency of 30 frames per second. This performance highlights its ability to balance accuracy and efficiency, which is critical for applications requiring rapid and

reliable motion prediction, such as human-robot interaction or real-time motion analysis.

TransFusion achieves a good level of accuracy, with a mean positional error of 100.81 mm for long-term motion predictions. This result demonstrates its capability to model complex motion dynamics effectively, making it suitable for applications requiring motion forecasting. However, the model's computational time of 0.1 seconds per prediction introduces a significant limitation for real-time applications, particularly in scenarios demanding rapid decision-making or low-latency responses, such as human-robot interaction.

The clustering method, in contrast, achieves the lowest mean positional error of 71.60 mm, demonstrating its effectiveness in capturing the dynamics of human motion in the assembly task. However, its computational time of 0.05 seconds is not optimal and is highly dependent on the chosen parameters, such as the number of clusters and the extent of PCA dimensionality reduction. Increasing the number of clusters or retaining more principal components improves prediction granularity but significantly increases inference time, which can hinder real-time applicability.

State-of-the-Art in Manipulator Motion Planning

Manipulator motion planning focuses on determining optimal paths for robotic manipulators, ensuring safe and efficient operation, especially crucial in collaborative settings like the one addressed in the following section. These methods are broadly categorized into passive, reactive, and proactive approaches, each with distinct advantages and limitations.

Passive Methods Passive methods involve predefined strategies and constraints, often relying on speed and separation monitoring (SSM) [International Organization for Standardization, 2016]. Byner et al. [Byner et al., 2019] explore continuous adaptation of robot velocity to improve SSM, offering notable productivity gains over traditional zone-based approaches. However, such methods lack real-time adaptability to dynamic environments like those involving unpredictable human movements, limiting their effectiveness in close human-robot collaboration.

Reactive Methods Reactive methods adapt robot trajectories in response to dynamic obstacles detected in real time. For example, Xie et al. [2018] enhance artificial potential fields to avoid moving obstacles, enabling manipulators to track dynamic goals effectively. Merckaert et al. [2024] integrate Rapidly-exploring Random Trees (RRT) with Explicit Reference Governors (ERG) to address dynamic constraints, ensuring safety in shared workspaces. These methods, while responsive to immediate changes, often struggle with anticipatory planning needed for seamless and efficient collaboration, as they only react once a potential conflict is detected.

Proactive Methods Proactive planning anticipates future environmental changes, including predicted human motion, to optimize robot trajectories beforehand. Zhao et al. [2008] extend the classical A* algorithm for time-dependent path planning, allowing consideration of future obstacle positions. The RRTX algorithm [Otte and Frazzoli, 2016] introduces efficient replanning for dynamic environments, quickly adapting paths as predictions update. Temporal PRM [Hüppi et al., 2022] incorporates temporal dynamics directly into roadmap construction, facilitating multi-query handling and smooth path generation in dynamic settings by evaluating path validity over time. Proactive methods, which enable robots to anticipate and adapt to human actions based on perceived nonverbal cues, offer the potential for smoother, more efficient HRI by avoiding unnecessary stops or sharp deviations. Understanding social norms, including proxemics—the study of spatial separation humans maintain during interactions—is vital for developing such proactive behaviours [Rios-Martinez et al., 2015; Saunderson and Nejat, 2019]. Models incorporating proxemic zones can help robots predict human reactions to their movements and plan paths that are not only collision-free but also socially acceptable [Agand et al., 2022]. This area of proactive robotics is extensively reviewed by Sirithunge et al. [2019].

In collaborative manipulation tasks, RL (section 4.1.4) provides a mechanism for learning task-specific strategies that ensure safe and efficient interaction. For example, an RL policy can be trained to predict and avoid potential collisions by learning from past interactions and environmental feedback. Model-free approaches, such as Q-learning and DQN, have demonstrated success in adapting

robot behaviour without requiring an explicit model of the environment.

The following section builds upon proactive planning principles, integrating insights from human motion prediction to enable anticipatory collision avoidance.

5.4 Human-Aware Motion Planning

This section details our human-aware motion planning framework, which leverages the predictive insights from the Human Motion Prediction (HMP) techniques discussed in Section 5.3 to ensure safe and efficient collaboration in shared human-robot workspaces. Our approach achieves this by integrating both reactive and proactive motion planning strategies.

In our framework, these strategies are combined: reactive components are utilized for their capacity to make rapid adjustments to the robot's trajectory in light of unforeseen environmental changes or imminent collision risks. Proactive components, crucially informed by the HMP outputs, allow the robot to anticipate likely human movements and adapt its path preemptively. This synergistic combination is designed to bridge the gap between immediate, reflexive responses and longer-term, predictive planning. The goal is to enhance the robot's ability to navigate dynamic environments effectively, thereby maintaining safety, optimizing task efficiency, and fostering a more fluid and natural human-robot interaction.

Safety is ensured through a collision-avoidance algorithm that continuously monitors predicted trajectory overlaps between the robot and humans. This algorithm integrates human pose predictions with advanced obstacle avoidance methods to maintain a safe separation distance in real time. For instance, the HMP framework employs Temporal PRM to refine trajectories in anticipation of future changes while relying on reactive potential fields for immediate obstacle avoidance.

The integration of these methods results in a robust planning framework that dynamically responds to human movements while anticipating future actions. The robot leverages predicted human motion to adapt its path, ensuring smooth and collision-free operation.

Specifically, the simulation (Fig. 5.5) showcases the interaction between the UR5e manipulator, mounted upside-down, and a human subject represented by a virtual twin model with 8 degrees of freedom. The virtual twin's predicted joint

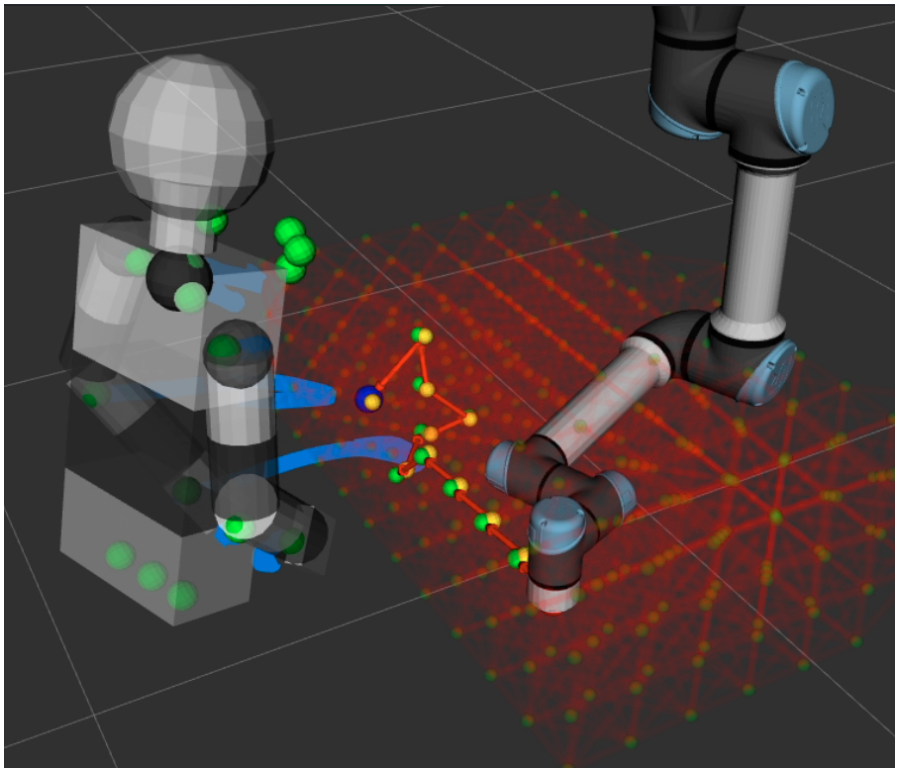


Figure 5.5: Simulation of human-aware motion planning. The human's virtual twin, with 8 degrees of freedom, is shown alongside predicted joint trajectories (blue paths) in 3D space. The UR5e manipulator, mounted upside-down, plans its trajectory (yellow spheres connected by red arrows) while avoiding the predicted human motion. The PRM grid, composed of green dots and red lines, supports the planning process.

trajectories are visualized as blue paths, providing insights into future human motion in 3D space.

Importantly, the simulation scenario depicted in Fig. 5.5 is designed to match exactly the experimental setup shown in Fig. 5.3, ensuring consistency between simulated and real-world conditions for human-aware motion planning.

The PRM (Probabilistic Roadmap) grid, displayed as green dots connected by red lines, serves as the foundation for path planning. The robot's planned trajectory, represented by yellow spheres connected by red arrows, avoids collisions with predicted human joint paths, ensuring safe and efficient navigation.

By uniting reactive and proactive planning, the system enhances collaboration

fluidity and addresses the limitations of traditional motion planning systems. This approach lays a foundation for the development of advanced human-aware motion planning systems capable of operating effectively in dynamic, shared environments.

5.4.1 Preliminary Evaluation of the Human-Aware Motion Planning Framework

The human-aware motion planning framework presented, which integrates real-time human motion predictions with proactive and reactive planning strategies, offers several notable advantages for human-robot collaboration. The current simulation (Fig. 5.5) demonstrates the framework's potential in a single, illustrative pick-and-place task. However, a comprehensive evaluation of its relevance and robustness requires acknowledging current limitations and outlining further steps towards more extensive validation, initially through a broader simulation campaign, followed by real-world experimental validation.

Advantages

The primary advantages of this integrated planning framework include:

- **Enhanced Safety:** Proactive collision avoidance, informed by human motion predictions, allows the robot to anticipate and mitigate potential risks before they become imminent, leading to safer interactions.
- **Improved Efficiency and Fluidity:** By minimizing abrupt stops or purely reactive manoeuvres, the robot's actions become smoother and more predictable. This can lead to increased task efficiency and a more natural flow of collaboration.
- **Increased Adaptability:** The framework allows the robot to dynamically adjust its plans in response to human movements, making it suitable for less structured and more dynamic collaborative environments.
- **Potentially More Intuitive Collaboration:** A robot that proactively adapts to human presence and movement can be perceived as more intelligent

and considerate, potentially improving the human's comfort and trust in the collaborative system.

Limitations and Challenges

Despite its potential, the framework faces several limitations and challenges:

- **Dependency on Prediction Accuracy:** The efficacy of the proactive planning component is heavily reliant on the accuracy and timeliness of the human motion prediction. Significant prediction errors could lead to suboptimal, inefficient, or, in worst-case scenarios, unsafe robot actions.
- **Computational Complexity:** The continuous cycle of predicting human motion, evaluating potential collisions, and replanning robot trajectories can be computationally intensive. Achieving real-time performance, especially with sophisticated prediction models and complex planning algorithms, remains a significant challenge for deployment on resource-constrained robotic hardware.
- **Scalability:** The current framework is demonstrated in a single-human, single-robot scenario. Scaling the approach to environments with multiple humans or robots would introduce additional complexities in prediction and coordination.
- **Handling of Highly Unpredictable Behaviour:** While prediction models aim to capture typical human movements, highly erratic, sudden, or novel human actions not well-represented in the training data may not be accurately forecasted, potentially reducing the effectiveness of proactive measures.
- **Fidelity of Models:** The accuracy of the human model (e.g., virtual twin, skeleton representation) and the environmental model is crucial. Discrepancies between the models and reality can impact planning outcomes.
- **Robustness to Real-World Conditions:** Sensor noise, occlusions affecting human tracking, and communication latencies are real-world factors that can degrade the performance of both the prediction and planning modules.

- **Preliminary Simulation Scope:** The current simulation results (Fig. 5.5) are illustrative and based on a single task. A more extensive quantitative evaluation across diverse scenarios and tasks is needed to fully assess the framework's capabilities and limitations.

Integration in Real Experiments for Enhanced Relevance Evaluation

Before, or in parallel with, embarking on full-scale real-world experiments, a more extensive simulation campaign is warranted to further quantify the framework's performance and establish its boundaries. Such a campaign should involve:

- **Diverse Scenarios:** Testing the integrated prediction and planning system across a wider variety of collaborative tasks (e.g., handover, co-manipulation, sequential assembly steps) and environmental layouts.
- **Parametric Variation:** Systematically varying parameters such as human movement speed, predictability of human actions (e.g., introducing unexpected movements), robot task goals, and workspace clutter.
- **Quantitative Benchmarking:** Collecting and analyzing comprehensive quantitative metrics. This includes not only task success rates and completion times, but also detailed statistics on collision avoidance (e.g., frequency of interventions, proximity to human), path efficiency, computational load (prediction and replanning times), and the impact of prediction errors on planning outcomes.
- **Stress Testing:** Identifying failure modes and understanding the framework's robustness under challenging or near-limit conditions.

This more thorough simulation-based evaluation would provide a stronger quantitative foundation for the framework's capabilities, help refine algorithms, and offer clearer insights into its operational envelope before committing to the significant resources required for extensive real-world human-robot interaction studies.

To more comprehensively evaluate the relevance and effectiveness of this human-aware motion planning framework, integration into real experiments with the UR5e manipulator is essential. This would involve:

- **System Integration:** Setting up a physical testbed with the UR5e, motion capture systems (e.g., depth cameras, marker-based systems) for real-time human skeleton tracking, and robust communication links between the perception, prediction, planning, and control modules.
- **Development of Safety Protocols:** Implementing stringent safety measures, including emergency stops, speed limitations, and potentially safety zones, especially during initial HRI experiments.
- **Real-Time Performance Optimization:** Profiling and optimizing the entire pipeline to meet the demands of real-time operation on the target robotic platform.
- **Quantitative Evaluation Metrics:** Assessing performance in real collaborative tasks using metrics such as:
  - Task completion time (for human, robot, and collaborative task).
  - Number of robot stops or significant slowdowns.
  - Minimum human-robot separation distance maintained.
  - Fluency metrics (e.g., idle times for human/robot).
- **Qualitative Evaluation Metrics:** Conducting user studies to gather feedback on:
  - Perceived safety and comfort during interaction.
  - Intuitiveness and predictability of robot behaviour.
  - Overall effectiveness and perceived intelligence of the collaborative system.

Such real-world experiments would provide invaluable data beyond simulation, offering a more robust validation of the framework's practical relevance, its ability to handle real-world uncertainties, and its impact on the quality of human-robot collaboration. This empirical evidence is crucial for demonstrating the tangible benefits and identifying further areas for improvement.

Chapter 6

Conclusions and Future Perspectives

This thesis explored the application of human-aware robotics across two different domains, each aiming to improve the safety, accuracy, and efficiency of mobile robots and human-robot interaction. We provided different approaches to address these challenges, combining learning-based methods with control systems for optimal performance. Below, we summarize the key findings, contributions, and directions for future research.

6.1 ViT-Based Semantic Maps for Human Occupancy Analysis

Human occupancy analysis in navigation systems plays a pivotal role in ensuring the safety of mobile robots. We introduced a Vision Transformer (ViT) backbone for human occupacy prediction, enabling the model to capture spatial relationships between nearby areas starting from a semantic map, thereby reconstructing a more global view of the environment.

Our results demonstrate that using ViT significantly improves prediction accuracy while maintaining an acceptable computation time for real-time applications. This solution is promising for future integration into robotic navigation systems, where environmental awareness is critical for autonomous operation. Moving for-

ward, we plan to extend our approach by testing it on more diverse and larger datasets to evaluate its generalization capacity. Furthermore, it will be important to investigate the effect of environmental variables—such as time of day, day of the week, month, or year—on human behaviour, and whether there is any seasonality in how people move within a given environment. Other factors, such as weather conditions, lighting, special events, or temporary obstacles, may also influence human behaviour. Given the same semantic map, human behaviour could vary significantly depending on these factors. Further, integrating this model into a full robot navigation framework and exploring its potential in different application domains are important next steps in our research.

6.2 Shared Control in Robot-Assisted Navigation

We also examined a robot-assisted navigation scenario where the control authority is shared between the human and the robot. Our primary contribution was the development of a decision-making framework that adapts the robot's behaviour based on the human's actions. By leveraging a combination of learning and control techniques, we were able to dynamically adjust the robot's guidance system to match the level of confidence in the human's behaviour, resulting in a safer and more adaptable navigation system.

In the future, we aim to refine this system by removing the need for pre-existing map knowledge. Integrating semantic map-based human behaviour priors into the shared control system could enable more robust and context-aware navigation, reducing reliance on simulated data and improving adaptability to unknown environments. We envision a framework where the robot can autonomously classify environmental features and match them with behavioural templates in real-time. Additionally, we plan to expand the system's capabilities to better distinguish between turns and straight paths. These considerations will require further investigation, particularly in understanding the impact of such deviations on the system's behaviour. Lastly, we will explore dynamic goal changes during online execution, ensuring the system remains responsive and capable of adapting without

inducing erratic behaviour.

6.3 Human-Aware Motion Planning Framework

In the domain of human-robot collaboration with manipulators, this thesis presented a human-aware motion planning framework designed to enhance safety and efficiency. We explored and compared clustering-based (GMM) and deep learning-based (Graph-Mixer, TransFusion) methods for real-time prediction of human skeleton trajectories, using the HA4M dataset for evaluation. The insights from these prediction models were then integrated into a motion planning system for the UR5e robot, combining proactive and reactive strategies. This allows the robot to anticipate human movements and adapt its trajectory to avoid collisions while working towards its task goals, as demonstrated in a simulated pick-and-place scenario. The approach aims to create a more fluid and intuitive interaction by enabling the robot to be responsive to the human collaborator.

It is important to emphasize that this work is extremely preliminary, and many aspects remain to be addressed before the framework can be considered mature or suitable for deployment in real-world scenarios. The current results are limited to simulation, and several components, such as experimental validation, robustness to sensor noise, and integration with real-time perception systems, have yet to be fully developed and rigorously tested.

For future work, we plan to test the framework experimentally, using the ZED2 camera and a Universal Robots UR5e. We plan to extend our approach by incorporating probabilistic scenario-based planning techniques [Oleinikov et al., 2024], to enhance our system's ability to handle multiple predictions with associated probabilities, since our Clustering-Based prediction method outputs multiple predictions with their corresponding probability. This extension will enable more robust and adaptive motion planning by generating and optimizing trajectories based on probabilistic human motion scenarios.

6.4 Integration of Environment-Driven and Motion-Driven Approaches

A key contribution of this thesis is the demonstration of how environment-driven semantic priors and motion-driven forecasting can be used to enhance human-robot interaction across diverse platforms. By predicting human behaviour priors from static semantic maps, robots gain a global understanding of likely human activities and spatial usage patterns. When combined with real-time trajectory prediction and adaptive planning, this could enable robots to anticipate human actions more accurately and to plan their own behaviour in a safer, more context-aware manner.

The synergy between these methods paves the way for future research on unified frameworks that seamlessly blend semantic understanding with adaptive, data-driven control in real-world human-robot collaboration scenarios.

Bibliography

Abbink, D. A., Carlson, T., Mulder, M., de Winter, J. C. F., Aminravan, F., Gibo, T. L., & Boer, E. R. (2018). A Topology of Shared Control Systems—Finding Common Ground in Diversity [Conference Name: IEEE Transactions on Human-Machine Systems]. *IEEE Transactions on Human-Machine Systems*, *48*(5), 509–525. https://doi.org/10.1109/THMS.2018.2791570

Agand, P., Taherahmadi, M., Lim, A., & Chen, M. (2022). Human navigational intent inference with probabilistic and optimal approaches. *2022 International Conference on Robotics and Automation (ICRA)*, 8562–8568.

Aigner, P., & McCarragher, B. (1997). Human integration into robot control utilising potential fields. *Proceedings of International Conference on Robotics and Automation*, *1*, 291–296 vol.1. https://doi.org/10.1109/ROBOT.1997.620053

Andreetto, M., Divan, S., Ferrari, F., Fontanelli, D., Palopoli, L., & Zenatti, F. (2018). Simulating passivity for Robotic Walkers via Authority-Sharing. *IEEE Robotics and Automation Letters*, *3*(2), 1306–1313. https://doi.org/10.1109/LRA.2018.2797321

Andreetto, M. et al. (2019a). Authority-sharing control of assistive robotic walkers. *PhD Thesis*.

Andreetto, M., Divan, S., Ferrari, F., Fontanelli, D., Palopoli, L., & Prattichizzo, D. (2019b). Combining Haptic and Bang-Bang Braking Actions for Passive Robotic Walker Path Following. *IEEE Transactions on Haptics*, *12*(4), 542–553. https://doi.org/10.1109/TOH.2019.2912570

Antonucci, A., Papini, G. R., Bevilacqua, P., Palopoli, L., & Fontanelli, D. (2021). Efficient Prediction of Human Motion for Real-Time Robotics Applications

with Physics-inspired Neural Networks. *IEEE Access*, *10*, 144–157. https://doi.org/10.1109/ACCESS.2021.3138614

Antonucci, A., Bevilacqua, P., Leonardi, S., Palopoli, L., & Fontanelli, D. (2021). Humans as path-finders for safe navigation. *arXiv preprint arXiv:2107.03079*.

Arechavaleta, G., Laumond, J.-P., Hicheur, H., & Berthoz, A. (2008a). On the nonholonomic nature of human locomotion. *Autonomous Robots*, *25*(1), 25–35.

Arechavaleta, G., Laumond, J.-P., Hicheur, H., & Berthoz, A. (2008b). An optimality principle governing human walking. *IEEE Transactions on Robotics*, *24*(1), 5–14. https://doi.org/10.1109/TRO.2008.915449

Argall, B. D. (2018). Autonomy in Rehabilitation Robotics: An Intersection. *Annual Review of Control, Robotics, and Autonomous Systems*, *1*(1), 441–463. https://doi.org/10.1146/annurev-control-061417-041727

Backman, K., Kulic, D., & Chung, H. (2021). Learning to Assist Drone Landings. *IEEE Robotics and Automation Letters*, *6*(2), 3192–3199. https://doi.org/10.1109/LRA.2021.3062572

perche non è meglio fare un sistema completamente autonomo??

Baratta, A., Cimino, A., Gnoni, M. G., & Longo, F. (2023). Human robot collaboration in industry 4.0: A literature review. *Procedia Computer Science*, *217*, 1887–1895.

Bertolazzi, E., & Frego, M. (2018). On the g2 hermite interpolation problem with clothoids. *Journal of Computational and Applied Mathematics*, *341*, 99–116.

Bevilacqua, P., Frego, M., Fontanelli, D., & Palopoli, L. (2018). Reactive Planning for Assistive Robots. *IEEE Robotics and Automation Letters*, *3*(2), 1276–1283. https://doi.org/10.1109/LRA.2018.2795642

Bevilacqua, P., Frego, M., Bertolazzi, E., Fontanelli, D., Palopoli, L., & Biral, F. (2016). Path planning maximising human comfort for assistive robots. *2016 IEEE Conference on Control Applications (CCA)*, 1421–1427. https://doi.org/10.1109/CCA.2016.7588006

Bevilacqua, P., Frego, M., Palopoli, L., & Fontanelli, D. (2020). Activity planning for assistive robots using chance-constrained stochastic programming. *IEEE Transactions on Industrial Informatics*, *17*(6), 3950–3961.

Bieber, G., Chodan, W., Bader, R., Hölle, B., Herrmann, P., & Dreher, I. (2019). Roro: A new robotic rollator concept to assist the elderly and caregivers. *Proceedings of the 12th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, 430–434. https://doi.org/10.1145/3316782.3322779

Birku, Y., & Agrawal, H. (2018). Survey on fall detection systems. *Int. J. Pure Appl. Math*, *118*(18), 2537–2543.

Borenstein, J., & Ulrich, I. (1997). The guidecane—a computerized travel aid for the active guidance of blind pedestrians. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, *2*, 1283–1288.

Byner, C., Matthias, B., & Ding, H. (2019). Dynamic speed and separation monitoring for collaborative robot applications – concepts and performance. *Robotics and Computer-Integrated Manufacturing*, *58*, 239–252. https://doi.org/https://doi.org/10.1016/j.rcim.2018.11.002

Carlson, T., Leeb, R., Chavarriaga, R., & Millán, J. d. R. (2012). Online modulation of the level of assistance in shared control systems [ISSN: 1062-922X]. *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 3339–3344. https://doi.org/10.1109/ICSMC.2012.6378307

Cicirelli, G., Marani, R., Romeo, L., García-Domínguez, M., Heras, J., Perri, A., & D'Orazio, T. (2022). The ha4m dataset: Multi-modal monitoring of an assembly task for human action recognition in manufacturing. *Scientific Data*, *9*. https://doi.org/10.1038/s41597-022-01843-z

Crandall, J., & Goodrich, M. (2002). Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, *2*, 1290–1295 vol.2. https://doi.org/10.1109/IRDS.2002.1043932

d'Addato, G., Falqueto, P., Palopoli, L., & Fontanelli, D. (2024). Socially-aware opinion-based navigation with oval limit cycles. https://arxiv.org/abs/2411.04678

Dani, A. P., Salehi, I., Rotithor, G., Trombetta, D., & Ravichandar, H. (2020). Human-in-the-Loop Robot Control for Human-Robot Collaboration: Human Intention Estimation and Safe Trajectory Tracking Control for Collaborative Tasks [Conference Name: IEEE Control Systems Magazine]. *IEEE*

Control Systems Magazine, *40*(6), 29–56. https://doi.org/10.1109/MCS.2020.3019725

Dautenhahn, K. (2007). Methodology & Themes of Human-Robot Interaction: A Growing Research Field [Publisher: SAGE Publications]. *International Journal of Advanced Robotic Systems*, *4*(1), 15. https://doi.org/10.5772/5702

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Doellinger, J., Spies, M., & Burgard, W. (2018). Predicting occupancy distributions of walking humans with convolutional neural networks. *IEEE Robotics and Automation Letters*, *3*(3), 1522–1528. https://doi.org/10.1109/LRA.2018.2800780

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

Dragan, A. D., & Srinivasa, S. S. (2013). A policy-blending formalism for shared control. *The International Journal of Robotics Research*, *32*(7), 790–805. https://doi.org/10.1177/0278364913490324

Eraslan, E., Yildiz, Y., & Annaswamy, A. M. (2020). Shared Control Between Pilots and Autopilots: An Illustration of a Cyberphysical Human System [Conference Name: IEEE Control Systems Magazine]. *IEEE Control Systems Magazine*, *40*(6), 77–97. https://doi.org/10.1109/MCS.2020.3019721

Falqueto, P., Antonucci, A., Palopoli, L., & Fontanelli, D. (2024). Humanising robot-assisted navigation. *Intelligent Service Robotics*, *17*(2), 155–165.

Falqueto, P., Sanfeliu, A., Palopoli, L., & Fontanelli, D. (2024). Learning priors of human motion with vision transformers [©2024 IEEE]. *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, 382–389. https://doi.org/10.1109/COMPSAC61105.2024.00060

FANUC Corporation. (Accessed: 2024-07-04). CRX Series Collaborative Robots [Manufacturer Website].

Farina, F., Fontanelli, D., Garulli, A., Giannitrapani, A., & Prattichizzo, D. (2017). Walking ahead: The headed social force model. *PloS one*, *12*(1), e0169734.

Fei Shi, Qixin Cao, Chuntao Leng, & Hongbing Tan. (2010). Based on force sensing-controlled human-machine interaction system for walking assistant robot. *2010 8th World Congress on Intelligent Control and Automation*, 6528–6533. https://doi.org/10.1109/WCICA.2010.5554167

Flemisch, F., Abbink, D. A., Itoh, M., Pacaux-Lemoine, M.-P., & Weßel, G. (2019). Joining the blunt and the pointy end of the spear: Towards a common framework of joint action, human–machine cooperation, cooperative guidance and control, shared, traded and supervisory control. *Cognition, Technology & Work*, *21*(4), 555–568. https://doi.org/10.1007/s10111-019-00576-1

Flemisch, F. O., Adams, C. A., Conway, S. R., Goodrich, K. H., Palmer, M. T., & Schutte, P. C. (2003). *The h-metaphor as a guideline for vehicle automation and interaction* (tech. rep.). RWTH Aachen University.

Flowers, J., Faroni, M., Wiens, G., & Pedrocchi, N. (2023). Spatio-temporal avoidance of predicted occupancy in human-robot collaboration. https://arxiv.org/abs/2307.03909

Gaz, C., Cognetti, M., Oliva, A., Giordano, P. R., & De Luca, A. (2019). Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization. *2019 International Conference on Robotics and Automation (ICRA)*, 9163–9169.

Fujioka, T., Shirano, Y., & Matsushita, A. (1999). Driver's behavior under steering assist control system. *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*, 246–251. https://doi.org/10.1109/ITSC.1999.821062

Gerdes, J. C., & Rossetter, E. J. (1999). A Unified Approach to Driver Assistance Systems Based on Artificial Potential Fields. *Journal of Dynamic Systems, Measurement, and Control*, *123*(3), 431–438. https://doi.org/10.1115/1.1386788

Goodrich, M. A., & Schultz, A. C. (2007). Human-Robot Interaction: A Survey. *Foundations and Trends® in Human-Computer Interaction*, *1*(3), 203–275. https://doi.org/10.1561/1100000005

Goswami, A., Peshkin, M., & Colgate, J. (1990). Passive robotics: An exploration of mechanical computation. *Proceedings., IEEE International Conference*

on Robotics and Automation, 279–284 vol.1. https://doi.org/10.1109/ROBOT.1990.125987

Graf, B., Hans, M., & Schraft, R. D. (2009). Care-o-bot 3 vision–a versatile mobile service robot platform for research. *2009 IEEE International Conference on Robotics and Automation*, 187–194.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2018). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2021). Masked autoencoders are scalable vision learners.

Helbing, D., & Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical review E*, *51*(5), 4282.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2018). Densely connected convolutional networks.

Hüppi, M., Bartolomei, L., Mascaro, R., & Chli, M. (2022). T-prm: Temporal probabilistic roadmap for path planning in dynamic environments. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10320–10327. https://doi.org/10.1109/IROS47612.2022.9981739

International Organization for Standardization. (2016). *Iso/ts 15066:2016 robots and robotic devices – collaborative robots*. Standard. International Organization for Standardization. Geneva, CH.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation.

Kaleci, B., Şenler, Ç. M., Dutağacı, H., & Parlaktuna, O. (2020). Semantic classification of mobile robot locations through 2d laser scans. *Intelligent Service Robotics*, *13*(1), 63–85.

Kanda, T., Glas, D. F., Shiomi, M., & Hagita, N. (2009). Abstracting people's trajectories for social robots to proactively approach customers. *IEEE Transactions on Robotics*, *25*(6), 1382–1396.

Kartoun, U., Stern, H., & Edan, Y. (2010). A Human-Robot Collaborative Reinforcement Learning Algorithm. *Journal of Intelligent & Robotic Systems*, *60*(2), 217–239. https://doi.org/10.1007/s10846-010-9422-y

Kavraki, L., Svestka, P., Latombe, J.-C., & Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, *12*(4), 566–580. https://doi.org/10.1109/70.508439

Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, *32*(11), 1238–1274. https://doi.org/10.1177/0278364913495721

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097–1105.

Bischoff, R., Kurth, J., Schreiber, G., Koeppe, R., Albu-Schäffer, A., Beyer, A., & Hirzinger, G. (2010). The kuka-dlr lightweight robot arm–a new reference platform for robotics research. *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, 1–7.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86. https://doi.org/10.1214/aoms/1177729694

Kyrarini, M., Lygerakis, F., Rajavenkatanarayanan, A., Sevastopoulos, C., Nambiappan, H. R., Chaitanya, K. K., Babu, A. R., Mathew, J., & Makedon, F. (2021). A Survey of Robots in Healthcare. *Technologies*, *9*(1), 8. https://doi.org/10.3390/technologies9010008

Laumond, J.-P., Arechavaleta, G., Truong, T.-V.-A., Hicheur, H., Pham, Q.-C., Berthoz, A., & Berthoz, A. (2010). The words of the human locomotion. *ISRR*. https://doi.org/10.1007/978-3-642-14743-2_4

LaValle, S. M. (2006). *Planning algorithms*. Cambridge university press.

Li, S., Bowman, M., & Zhang, X. (2020). A General Arbitration Model for Robust Human-Robot Shared Control with Multi-Source Uncertainty Modeling [arXiv: 2003.05097]. *arXiv:2003.05097 [cs]*. Retrieved September 22, 2021, from http://arxiv.org/abs/2003.05097

Liu, W., Liang, X., & Zheng, M. (2023). Task-constrained motion planning considering uncertainty-informed human motion prediction for human–robot col-

laborative disassembly. *IEEE/ASME Transactions on Mechatronics*, *28*(4), 2056–2063. https://doi.org/10.1109/TMECH.2023.3275316

Losey, D. P., McDonald, C. G., Battaglia, E., & O'Malley, M. K. (2018). A Review of Intent Detection, Arbitration, and Communication Aspects of Shared Control for Physical Human–Robot Interaction. *Applied Mechanics Reviews*, *70*(1), 010804. https://doi.org/10.1115/1.4039145

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. https://arxiv.org/abs/1711.05101

Lu, S., & Xia, Y. (2020). Dual supervised autoencoder based trajectory classification using enhanced spatio-temporal information. *IEEE Access*, *8*, 173918–173932.

Mainprice, J., & Berenson, D. (2013). Human-robot collaborative manipulation planning using early prediction of human motion. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 299–306. https://doi.org/10.1109/IROS.2013.6696368

Marayong, P., Bettini, A., & Okamura, A. (2002). Effect of virtual fixture compliance on human-machine cooperative manipulation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, *2*, 1089–1095 vol.2. https://doi.org/10.1109/IRDS.2002.1043876

Marcano, M., Diaz, S., Perez, J., & Irigoyen, E. (2020). A Review of Shared Control for Automated Vehicles: Theory and Applications. *IEEE Transactions on Human-Machine Systems*, *50*(6), 475–491. https://doi.org/10.1109/THMS.2020.3017748

Mavrogiannis, C., Baldini, F., Wang, A., Zhao, D., Trautman, P., Steinfeld, A., & Oh, J. (2023). Core challenges of social robot navigation: A survey. *ACM Transactions on Human-Robot Interaction*, *12*(3), 1–39.

Merckaert, K., Convens, B., Nicotra, M. M., & Vanderborght, B. (2024). Real-time constraint-based planning and control of robotic manipulators for safe human–robot collaboration. *Robotics and Computer-Integrated Manufacturing*, *87*, 102711. https://doi.org/https://doi.org/10.1016/j.rcim.2023.102711

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *44*(7), 3523–3542.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529–533.

Mohebbi, A. (2020). Human-Robot Interaction in Rehabilitation and Assistance: A Review. *Current Robotics Reports*, *1*(3), 131–144. https://doi.org/10.1007/s43154-020-00015-4

Murray, B., & Perera, L. P. (2020). A dual linear autoencoder approach for vessel trajectory prediction using historical ais data. *Ocean Engineering*, *209*, 107478.

Nazemzadeh, P., Fontanelli, D., Macii, D., & Palopoli, L. (2017). Indoor Localization of Mobile Robots through QR Code Detection and Dead Reckoning Data Fusion. *IEEE/ASME Transactions on Mechatronics*, *22*(6), 2588–2599. https://doi.org/10.1109/TMECH.2017.2762598

Oh, Y., Wu, S.-W., Toussaint, M., & Mainprice, J. (2020). Natural Gradient Shared Control. *IEEE International Conference*, 7.

Oleinikov, A., Soltan, S., Balgabekova, Z., Bemporad, A., & Rubagotti, M. (2024). Scenario-based model predictive control with probabilistic human predictions for human–robot coexistence. *Control Engineering Practice*, *142*, 105769. https://doi.org/https://doi.org/10.1016/j.conengprac.2023.105769

Open Robotics. (2014). *Robot operating system (ros)*. Retrieved December 9, 2024, from https://www.ros.org

Otte, M., & Frazzoli, E. (2016). Rrtx: Asymptotically optimal single-query sampling-based motion planning with quick replanning. *The International Journal of Robotics Research*, *35*(7), 797–822. https://doi.org/10.1177/0278364915594679

Palopoli, L., Argyros, A., Birchbauer, J., Colombo, A., Fontanelli, D., Legay, A., Garulli, A., Giannitrapani, A., Macii, D., Moro, F., Nazemzadeh, P., Panteleris, P., Passerone, R., Poier, G., Prattichizzo, D., Rizano, T., Rizzon, L., Scheggi, S., & Sedwards, S. (2015). Navigation assistance and guidance

of older adults across complex public spaces: The dali approach. *Intelligent Service Robotics*, *8*, 77–92. https://doi.org/10.1007/s11370-015-0169-y

Papudesi, V., & Huber, M. (2003). *Learning from Reinforcement and Advice Using Composite Reward Functions.* [Pages: 365].

Patoglu, V., Li, Y., & O'Malley, M. K. (2009). On the Efficacy of Haptic Guidance Schemes for Human Motor Learning. In O. Dössel & W. C. Schlegel (Eds.), *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany* (pp. 203–206). Springer. https://doi.org/10.1007/978-3-642-03889-1_55

Pérez-D'Arpino, C., & Shah, J. A. (2015). Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 6175–6182. https://doi.org/10.1109/ICRA.2015.7140066

Polydoros, A. S., & Nalpantidis, L. (2017). Survey of Model-Based Reinforcement Learning: Applications on Robotics. *Journal of Intelligent & Robotic Systems*, *86*(2), 153–173. https://doi.org/10.1007/s10846-017-0468-y

Rákos, O., Aradi, S., Bécsi, T., & Szalay, Z. (2020). Compression of vehicle trajectories with a variational autoencoder. *Applied Sciences*, *10*(19), 6739.

Reddy, S., Dragan, A. D., & Levine, S. (2018). Shared Autonomy via Deep Reinforcement Learning [arXiv: 1802.01744]. *arXiv:1802.01744 [cs]*. Retrieved September 22, 2021, from http://arxiv.org/abs/1802.01744
Comment: Accepted to the Robotics: Science and Systems (RSS) 2018 conference

Rios-Martinez, J., Spalanzani, A., & Laugier, C. (2015). From proxemics theory to socially-aware navigation: A survey. *International Journal of Social Robotics*, *7*, 137–153.

Robicquet, A., Sadeghian, A., Alahi, A., & Savarese, S. (2016). Learning social etiquette: Human trajectory prediction in crowded scenes. *European Conference on Computer Vision (ECCV)*.

Rosenberg, L. (1993). Virtual fixtures: Perceptual tools for telerobotic manipulation. *Proceedings of IEEE Virtual Reality Annual International Symposium*, 76–82. https://doi.org/10.1109/VRAIS.1993.380795

Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, *40*, 99–121. https://doi.org/10.1023/A:1026543900054

Rudenko, A., Palmieri, L., Herman, M., Kitani, K. M., Gavrila, D. M., & Arras, K. O. (2020). Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, *39*(8), 895–935. https://doi.org/10.1177/0278364920917446

Rudenko, A., Palmieri, L., Doellinger, J., Lilienthal, A. J., & Arras, K. O. (2021). Learning occupancy priors of human motion from semantic maps of urban environments. *IEEE Robotics and Automation Letters*, *6*(2), 3248–3255. https://doi.org/10.1109/LRA.2021.3062010

Santhanaraj, K. K., M.M., R., & D., D. (2021). A survey of assistive robots and systems for elderly care. *Journal of Enabling Technologies*, *15*(1), 66–72. https://doi.org/10.1108/JET-10-2020-0043

Saunderson, S., & Nejat, G. (2019). How robots influence humans: A survey of nonverbal communication in social human–robot interaction. *International Journal of Social Robotics*, *11*(4), 575–608.

Sheridan, T. B., & Verplank, W. L. (1978). *Human and Computer Control of Undersea Teleoperators* (tech. rep.) [Section: Technical Reports]. MASSACHUSETTS INST OF TECH CAMBRIDGE MAN-MACHINE SYSTEMS LAB. Retrieved September 27, 2021, from https://apps.dtic.mil/sti/citations/ADA057655

Singamaneni, P. T., Bachiller-Burgos, P., Manso, L. J., Garrell, A., Sanfeliu, A., Spalanzani, A., & Alami, R. (2024). Advances and challenges in human-aware social robot navigation: A survey. *International Journal Robotics Research*. https://doi.org/10.1177/02783649241230562

Sirithunge, C., Jayasekara, A. B. P., & Chandima, D. (2019). Proactive robots with the perception of nonverbal human behavior: A review. *IEEE Access*, *7*, 77308–77327. https://doi.org/10.1109/ACCESS.2019.2921861

Song, K.-T., Jiang, S.-Y., & Wu, S.-Y. (2017). Safe Guidance for a Walking-Assistant Robot Using Gait Estimation and Obstacle Avoidance. *IEEE/ASME Transactions on Mechatronics*, *22*(5), 2070–2078. https://doi.org/10.1109/TMECH.2017.2742545

Strudel, R., Garcia, R., Laptev, I., & Schmid, C. (2021). Segmenter: Transformer for semantic segmentation.

Tang, A., & Cao, Q. (2012). Motion control of walking assistant robot based on comfort. *Industrial Robot: An International Journal*, *39*(6), 564–579. https://doi.org/10.1108/01439911211268778

Tian, S., Zheng, M., & Liang, X. (2024). Transfusion: A practical and effective transformer-based diffusion model for 3d human motion prediction. *IEEE Robotics and Automation Letters*, *9*(7), 6232–6239. https://doi.org/10.1109/LRA.2024.3401116

Tong, Y., & Liu, J. (2021). Review of Research and Development of Supernumerary Robotic Limbs. *IEEE/CAA Journal of Automatica Sinica*, *8*(5), 929–952. https://doi.org/10.1109/JAS.2021.1003961

Tran, A., Somanath, S., & Sharlin, E. (2018). Using supernumerary robotic arms for background tasks. *Extended Abstracts of the 2018 GI Conference Graphics Interface. ACM.*

Universal Robots A/S and FZI Forschungszentrum Informatik. (Accessed: 2024-07-04). Universal Robots ROS Driver [GitHub Repository].

Universal Robots Inc. (2018). *Ur5e - lightweight, versatile cobot.* Retrieved October 17, 2024, from https://www.universal-robots.com/products/ur5e

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention is all you need.

Watkins, C. (1989). *Learning From Delayed Rewards* (Doctoral dissertation). King's College.

Wei, X., & Zhang, X. (2017). Research on the technology of walking intention identification for walking assistant robot. *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 835–838. https://doi.org/10.1109/URAI.2017.7992838

Wiest, J., Höffken, M., Kreßel, U., & Dietmayer, K. (2012). Probabilistic trajectory prediction with gaussian mixture models. *2012 IEEE Intelligent Vehicles Symposium*, 141–146. https://doi.org/10.1109/IVS.2012.6232277

Xie, L., & Liu, S. (2018). Dynamic obstacle-avoiding motion planning for manipulator based on improved artificial potential filed. *Kongzhi Lilun Yu*

Yingyong/Control Theory and Applications, *35*, 1239–1249. https://doi.org/10.7641/CTA.2018.70187

Yanco, H., & Drury, J. (2004). Classifying human-robot interaction: An updated taxonomy. *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, *3*, 2841–2846. https://doi.org/10.1109/ICSMC.2004.1400763

Yang, S., Li, H., Pun, C.-M., Du, C., & Gao, H. (2024). Adaptive spatial-temporal graph-mixer for human motion prediction. *IEEE Signal Processing Letters*, *31*, 1244–1248. https://doi.org/10.1109/LSP.2024.3392686

Yu, W., Alqasemi, R., Dubey, R., & Pernalete, N. (2005). Telemanipulation Assistance Based on Motion Intention Recognition [ISSN: 1050-4729]. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 1121–1126. https://doi.org/10.1109/ROBOT.2005.1570266

Zachiotis, G. A., Andrikopoulos, G., Gornez, R., Nakamura, K., & Nikolakopoulos, G. (2018). A survey on the application trends of home service robotics. *2018 IEEE international conference on Robotics and Biomimetics (ROBIO)*, 1999–2006.

Zhang, D., Tron, R., & Khurshid, R. P. (2021). Haptic Feedback Improves Human-Robot Agreement and User Satisfaction in Shared-Autonomy Teleoperation [arXiv: 2103.03453]. *arXiv:2103.03453 [cs, eess]*. Retrieved September 22, 2021, from http://arxiv.org/abs/2103.03453

Zhang, L., Lv, Y., Zhao, J., Zhang, T., Mu, Y., Shahzad, A., Cui, D., Dong, H., & Gao, X. (2016). Design of an active and passive type walking assistant robot for human. *2016 Chinese Control and Decision Conference (CCDC)*, 4403–4408. https://doi.org/10.1109/CCDC.2016.7531778

Zhao, L., Ohshima, T., & Nagamochi, H. (2008). A* algorithm for the time-dependent shortest path problem.

Zurek, M., Bobu, A., Brown, D. S., & Dragan, A. D. (2021). Situational Confidence Assistance for Lifelong Shared Autonomy [arXiv: 2104.06556]. *arXiv:2104.06556 [cs]*. Retrieved September 22, 2021, from http://arxiv.org/abs/2104.06556 Comment: In proceedings ICRA 2021

List of Figures

List of Tables