Learning Graph Embeddings for Open World Compositional Zero-Shot Learning

Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata, *Member, IEEE*

Abstract—Compositional Zero-Shot learning (CZSL) aims to recognize unseen compositions of state and object visual primitives seen during training. A problem with standard CZSL is the assumption of knowing which unseen compositions will be available at test time. In this work, we overcome this assumption operating on the open world setting, where no limit is imposed on the compositional space at test time, and the search space contains a large number of unseen compositions. To address this problem, we propose a new approach, Compositional Cosine Graph Embeddings (Co-CGE), based on two principles. First, Co-CGE models the dependency between states, objects and their compositions through a graph convolutional neural network. The graph propagates information from seen to unseen concepts, improving their representations. Second, since not all unseen compositions are equally feasible, and less feasible ones may damage the learned representations, Co-CGE estimates a feasibility score for each unseen composition, using the scores as margins in a cosine similarity-based loss and as weights in the adjacency matrix of the graphs. Experiments show that our approach achieves state-of-the-art performances in standard CZSL while outperforming previous methods in the open world scenario.

Index Terms—Compositional Zero-Shot Learning, Graph Neural Networks, Open-World Recognition, Scene Understanding

1 Introduction

A ''black swan'' was ironically used as a metaphor in the 16th century for an unlikely event because the western world had only seen white swans. Yet when the European settlers observed a black swan for the first time in Australia in 1697, they immediately knew what it was. This is because humans posses the ability to compose their knowledge of known entities to generalize to novel concepts. In the literature, this task is known as Compositional Zero-Shot Learning (CZSL) [1], [2], [3], [4], [5], where the goal is to learn how to compose observed objects and their states and to generalize to novel state-object compositions.

CZSL differs from the standard Zero-Shot Learning (ZSL) in multiple aspects. ZSL aims at recognizing unseen categories given an attribute vector describing them. The main objective of ZSL is thus to learn how to map images to their representations in a given, high-dimensional, semantic space. Differently, CZSL assumes access of images containing all primitive concepts (*i.e.* objects and states) but not all their possible compositions. The goal of CZSL is thus to model how states modify objects, extrapolating this knowledge from seen to unseen compositions of the same set of objects and states.

A fundamental limitation of both standard ZSL and CZSL is the assumption of operating in a ''closed-world'', where we know a priori the test-time unseen semantic concepts and we can restrict the output space of our model accordingly. As an example, in MIT states there are 115 states and 245 objects, but only 1662 compositions out of 28175 possible combination of these states and objects are considered in the output space at test time. This assumption is unrealistic in practical, unconstrained settings, where arbitrary object categories/compositions may appear. Previous works showed that removing this assumption in ZSL, *i.e.* by considering the whole vocabulary as unseen class set, is impractical and leads to poor results [6], [7]. In this paper, we investigate this problem in CZSL, showing that we can consider all possible compositions in the output space, with a relatively small drop in performance despite having a comparable or larger search space (*e.g.* 280k for C-GQA) than the ZSL counterpart (*e.g.* 310k [6] or 21k [7] for ImageNet). We name this more realistic and challenging setting as Open World CZSL (OW-CZSL).

As in CZSL, in OW-CZSL during training we have images depicting a set of objects in a set of states and, at test time, we receive both seen and unseen compositions of the same objects and states. However, differently from CZSL, in OW-CZSL we assume that we do not know the subset of unseen compositions contained in the test images, thus the model has an output space containing *all* possible compositions of the known objects and states (*e.g.* almost 28k compositions in MIT states vs 1662 of standard CZSL). Addressing OW-CZSL requires building a representation space where seen and unseen compositions can be recognized despite the huge number of test compositions.

In this work, we tackle the OW-CZSL task with Compositional Cosine Graph Embeddings (Co-CGE), a graph-based approach for OW-CZSL. Co-CGE is based on two inductive biases. Our first inductive bias is a rich dependency structure of different states, objects and their compositions, *e.g.* learning the composition `old dog` is not only dependent on the state `old` and object `dog`, but may also require being aware of other compositions like `cute dog`, `old car`, etc. We argue that such dependency structure provides a strong regularization, while allowing the network to better

- *M. Mancini is with University of Tübingen. E-mail: massimiliano.mancini@uni-tuebingen.de*
- *M.F. Naeem is with ETH Zurich.*
- *Y. Xian is with ETH Zurich. The majority of this work was done when Y. Xian was with the Max Planck Institute (MPI) for Informatics.*
- *Z. Akata is with University of Tübingen, MPI for Informatics and MPI for Intelligent Systems.*

Manuscript submitted April 29, 2021.

generalize to novel compositions and model it through a compositional graph, connecting state, objects and their compositions. Differently from previous works [1], [2], [3], [4] that treat each state-object composition independently, our graph formulation allows the model to learn compositional embeddings that are globally consistent.

Our second inductive bias is the presence of *distractors*, *i.e.* less feasible compositions (*e.g. ripe dog*) that a model needs to either eliminate or isolate in the search space. For this purpose, we use similarities among primitives embeddings to assign a feasibility score to each unseen composition. We then use these scores as margins in a cosine-based cross-entropy loss, showing how the feasibility scores enforce a shared embedding space where unfeasible distractors are discarded, while visual and compositional domains are aligned. Since the distractors may pollute the learned representations of other unseen compositions in Co-CGE, we inject the feasibility scores also within the graph. In particular, we instantiate a weighted adjacency matrix, where the weights depend on the feasibility of each composition. Experiments show that Co-CGE is either superior or competitive with the state of the art in CZSL while being much more effective on the challenging OW-CZSL task.

Our contributions are as follows: (1) We introduce Co-CGE, a graph formulation for the new OW-CZSL problem with an integrated feasibility estimation mechanism used to weight the graph connections; (2) We exploit the dependency between visual primitives and their compositional classes and propose a multimodal compatibility learning framework that embeds related states, objects and their compositions into a shared embedding space learned through cosine logits and feasibility-based margins; (3) We improve the state-of-the-art on MIT states [8], UT Zappos [9] and the recently proposed C-GQA [5] benchmarks on both CZSL and OW-CZSL.

This paper extends our previous works [5] and [10] published in CVPR 2021 in many aspects. First, while being effective on standard CZSL, the CGE model of [5] performs poorly in the OW-CZSL, due to the noisy connections arising from the huge search space. We thus take the idea of estimating the feasibility of each composition from [10] and we inject the feasibility scores both at the loss level and within the graph connections. Our model is based on a graph convolutional neural network (GCN) [11]. This means that the embeddings as well as the feasibility scores are influenced by all other composition in the search space, rather than considered in isolation as it was the case in [10]. We extend our OW-CZSL benchmark proposed in [10] to the new C-GQA dataset proposed in [5] with 413 states, 674 objects thus a total OW-CZSL search space of almost 280k compositions. Finally, we significantly improve the state of the art on the challenging OW-CZSL setting.

2 Related works

Compositionality can loosely be defined as the ability to decompose an observation into its primitives. These primitives can then be used for complex reasoning. One of the earliest attempts in computer vision in this direction can be traced to Hoffman [12] and Biederman [13] who theorized that visual systems can mimic compositionality by decomposing objects to their parts. Compositionality at a fundamental level is already included in modern vision systems. Convolutional Neural Networks (CNN) have been shown to exploit compositionality by learning a hierarchy of features [14], [15]. Transfer learning [16], [17], [18], [19] and few-shot learning [20], [21], [22] exploit the compositionality of pretrained features to generalize to data constraint environments. Visual scene understanding [23], [24], [25], [26] aims to understand the compositionality of concepts in a scene. Nevertheless, these approaches still requires collecting data for new compositional classes.

ZSL and CZSL. Zero-Shot Learning (ZSL) aims to recognize novel classes not observed during training [27] using side information describing novel classes *e.g.* attributes [27], text descriptions [28] or word embeddings [29]. Some notable approaches include learning a compatibility function between image and class embeddings [30], [31] and learning to generate image features for novel classes [32], [33].

Compositional Zero-Shot Learning (CZSL) aims to learn the compositionality of objects and their states from the training set and generalizing to unseen combinations of these primitives. Approaches in this direction can be divided into two groups. The first group is directly inspired by [12], [13]. Some notable methods include learning a transformation upon individual classifiers of states and objects [1], modeling each state as a linear transformation of objects [2], learning a hierarchical decomposition and composition of visual primitives [34] and modeling objects to be symmetric under attribute transformations [4]. The second group argues that compositionality requires learning a joint compatibility function with respect to the image, the state and the object [3], [35], [36]. This is achieved by learning a modular networks conditioned on each composition [3], [36] that can be "rewired" for a new compositions. Finally a recent work from Atzmon et al. [35] argue that achieving generalization in CZSL requires learning the visual transformation through a causal graph where the latent representation of primitives are independent of each other.

GCN. Graph Convolutional Networks (GCN) [11], [37], [38] are a special type of neural networks that exploit the dependency structure of data (nodes) defined in a graph. Current methods [11] are limited by the network depth due to over smoothing at deeper layers of the network. The extreme case of this can cause all nodes to converge to the same value [39]. Several works have tried to remedy this by dense skip connections [40], [41], randomly dropping edges [42] and applying a linear combination of neighbor features [43], [44], [45]. Recent works in this direction combined residual connections with identity mapping [46] or used . GCNs have shown to be promising for zero-shot learning. Wang et al. [37] propose to directly regress the classifier weights of novel classes with a GCN operated on an external knowledge graph (WordNet [47]). Kampffmeyer et al. [38] improve this formulation by introducing a dense graph to learn a shallow GCN as a remedy for the Laplacian smoothing problem [39].

Our method lies at the intersection of several discussed approaches. We learn a joint compatibility function similar to [3], [35], [36] and utilize a GCN similar to [37], [38]. However, we exploit the dependency structure between

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3163667, IEEE Transactions on Pattern Analysis and Machine Intelligence

states, objects and compositions which has been overlooked by previous CZSL approaches [3], [35], [36]. Instead of using a predefined knowledge graph like WordNet [47], we propose a novel way to build a compositional graph and learn classifiers for all classes. In contrast to [35] we explicitly promote the dependency between all primitives and their compositions in our graph. This allows us to learn embeddings that are consistent with the whole graph. Furthermore, our approach estimates the feasibility of each composition, exploiting this information to re-weight the graph connections and to model the presence of distractors within the training objective. Finally, unlike all existing methods [1], [2], [3], [34], [35], [36], instead of using a fixed image feature extractor our model is trained end-to-end.

Open World Recognition. In our open world setting, all the combinations of states and objects can form a valid compositional class. This is different from an alternate definition of *Open World Recognition* (OWR) [48], [49] where the goal is to dynamically update a model trained on a subset of classes to detect unknon semantic concepts and incrementally them as new data arrives. Differently from [48] we assume a static set of objects and states, with our open world being the set of their all possible compositions.

Our definition is related to the *open set* zero-shot learning (ZSL) [7] scenario in [6], [50], proposing that expands the output space to a large vocabulary of semantic concepts. Both our work and [6] consider the lack of constraints in the output space for unseen concepts as a requirement for practical (compositional) ZSL methods. However, since we focus on the CZSL task, we have access to images of all primitives during training but not all their possible compositions. Note that this differs from [6], [50], since i) the set of objects and states are fixed and do not vary between training and test time, and ii) we have access to images of all primitives during training but not all their possible compositions, as in standard CZSL. From the latter, we can use the knowledge derived from the visual world to model the feasibility of compositions and modifying the representations in the shared visual-compositional embedding space. In this work, we explicitly model the feasibility of each unseen composition, incorporating this knowledge into our model into training.

3 Compositional Cosine Graph Embeddings

Let $\mathcal{S}$ be the set of possible states, with $\mathcal{O}$ being the set of possible objects, and with $\mathcal{C} = \mathcal{S} \times \mathcal{O}$ being the set of all their possible compositions. $\mathcal{T} = \{(x_i, c_i)\}_{i=1}^{N}$ is a training set where $x_i \in \mathcal{X}$ is a sample in the input (image) space $\mathcal{X}$ and $c_i \in \mathcal{C}^s$ is a composition in the subset $\mathcal{C}^s \subset \mathcal{C}$. In this formulation, $\mathcal{C}^s$ denotes the set of seen compositions. $\mathcal{T}$ is used to train a model $f : \mathcal{X} \rightarrow \mathcal{C}^t$ predicting combinations in a space $\mathcal{C}^t \subseteq \mathcal{C}$ where $\mathcal{C}^t$ may include compositions not present in $\mathcal{C}^s$ (*i.e.* $\exists c \in \mathcal{C}^t \wedge c \notin \mathcal{C}^s$).

The CZSL task entails different challenges depending on the extent of the target set $\mathcal{C}^t$. If $\mathcal{C}^t$ is a subset of $\mathcal{C}$ and $\mathcal{C}^t \cap \mathcal{C}^s \equiv \emptyset$, the task definition is of [1], where the model needs to predict only unseen compositions at test time. In case $\mathcal{C}^s \subset \mathcal{C}^t$ we are in the generalized CZSL scenario, and the output space of the model contains both seen and unseen compositions. Similar to the generalized ZSL [7], GCZSL scenario is more challenging due to the natural prediction bias of the model in $\mathcal{C}^s$, seen during training. Most recent works on CZSL consider the GCZSL scenario [3], [4], and the set of unseen compositions in $\mathcal{C}^t$ is known a priori.

In our work, the output space is the whole set of possible compositions $\mathcal{C}^t \equiv \mathcal{C}$, *i.e. Open World Compositional Zero-shot Learning* (OW-CZSL). Note that this task presents the same challenges of the GCZSL setting while being far more difficult since i) $|\mathcal{C}^t| \gg |\mathcal{C}^s|$, thus it is hard to generalize from a small set of seen to a very large set of unseen compositions; and ii) there are a large number of *distracting* compositions in $\mathcal{C}^t$, *i.e.* compositions predicted by the model but not present in the actual test set that can be close to other unseen compositions, hampering their discriminability. We highlight that, despite being similar to Open Set Zero-shot Learning [6], we do not only consider objects but also states. Therefore, this knowledge can be exploited to identify unfeasible distracting compositions (*e.g. rusty pie*) and isolate them. In the following we describe how we tackle this problem by means of compositional graph embeddings.

3.1 Compositional Graph Embedding for CZSL

In this section, we focus on the closed world setting, where $\mathcal{C}^s \subset \mathcal{C}^t \subset \mathcal{C}$. Since in this scenario $|\mathcal{C}^t| \ll |\mathcal{C}|$ and the number of unseen compositions is usually lower than the number of seen ones, this problem presents several challenges. In particular, while learning a mapping from the visual to the compositional space, the model needs to avoid being overly biased toward seen class predictions.

As states and objects are not independent *e.g.* the appearance of the state `sliced` varies significantly with the object (*e.g.* `apple` or `bread`) and learning state and object classifiers separately is prone to overfit to labels observed during training. Therefore, we model the states and objects jointly via $f : \mathcal{X} \times \mathcal{C} \rightarrow \mathbb{R}$ that learns the compatibility between an image and a state-object composition. Given a specific input image x, we predict its label $c^* = (s^*, o^*)$ as the state-object composition that yields the highest compatibility score:

$$c^* = \arg\max_{c \in \mathcal{C}^t} f(x, c) = \arg\max_{c \in \mathcal{C}^t} h(\omega(x), \phi(c)) \qquad (1)$$

where $\omega : \mathcal{X} \rightarrow \mathcal{Z}$ is the mapping from the image space to the d-dimensional shared embedding space $\mathcal{Z} \in \mathbb{R}^d$, $\phi : \mathcal{C} \rightarrow \mathcal{Z}$ embeds a composition to the same shared space and $h : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$ is a compatibility scoring function.

We implement ω as a deep neural network, ϕ as a graph convolutional neural network (GCN) [11] and h as cosine similarity. This way we exploit deep image representations and propagate information from seen to unseen concepts through a graph and while at the same time avoid bias on seen classes through cosine similarity scores. We name our model Compositional Cosine Graph Embeddings (Co-CGE).

Compositional Graph Embeddings (CGE). We encode the dependency structure of states, objects and their compositions (both seen and unseen) through a compositional graph. We map compositions into the shared embedding space by modeling ϕ as a GCN with K nodes, L layers and the output of the l_{th} layer:

$$V_{l+1} = \sigma(\hat{A} V_l W_l) \qquad (2)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3163667, IEEE Transactions on Pattern Analysis and Machine Intelligence

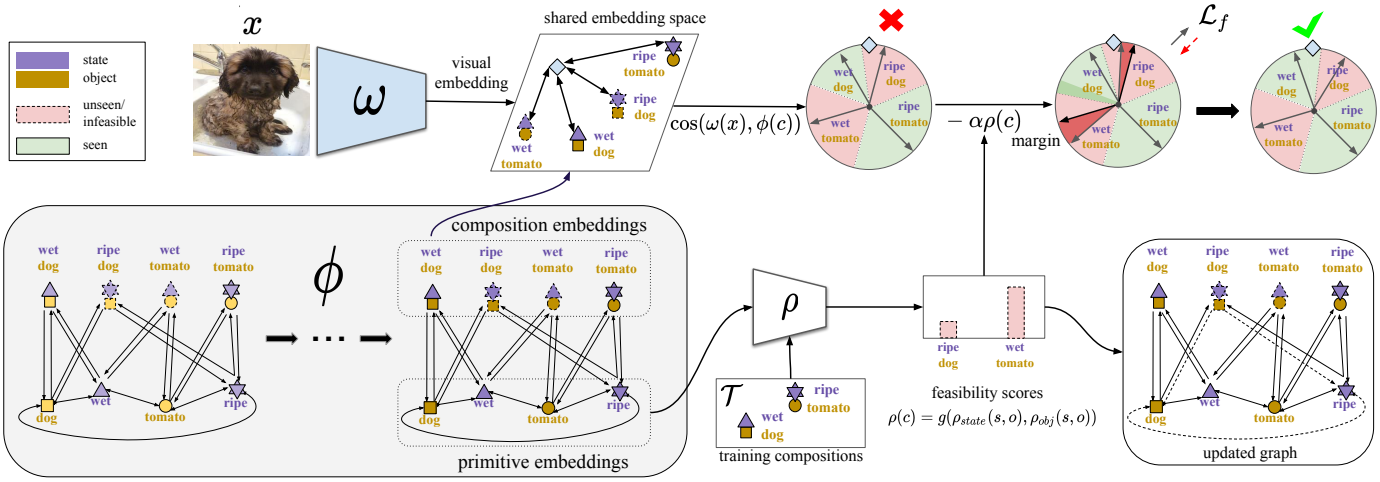

Fig. 1: **Compositional Cosine Graph Embeddings (Co-CGE).** Our approach embeds an image (top) and state-object compositions (bottom) into a shared semantic space. The state-object compositions are encoded through a graph (bottom, grey block). For OW-CZSL, we estimate a feasibility score for each of the unseen compositions, using the relation between states, objects, and the training compositions. The feasibility scores are injected as margins in the objective function (top right, purple slices) or to update the graph connections (bottom right, dashed lines).

where σ is a non-linear activation function (*i.e.* ReLU), $V_l \in \mathbb{R}^{K\times D}$ is the matrix of the D-dimensional node representations at layer l, $W_l \in \mathbb{R}^{D\times D'}$ is the trainable weight matrix at layer l and $\hat{A} \in \mathbb{R}^{K\times K}$ is the column normalized adjacency matrix A. In the CZSL task, CGE defines the set of starting nodes as $V_0 \in \mathbb{R}^{K\times m}$, with $\mathcal{K} = \mathcal{C}^t \cup \mathcal{S} \cup \mathcal{O}$ and $K = |\mathcal{K}|$. Given k_i, *i.e.* the i_{th} element of $\mathcal{K}$, the representation of its node in $\mathcal{V}_0$ is:

$$k_i = \begin{cases} \varphi(k_i) & \text{if } k_i \in \mathcal{S} \cup \mathcal{O} \\ (\varphi(s) + \varphi(o))/2 & \text{if } k_i \in \mathcal{C}^t \wedge k_i = (s,o) \end{cases} \tag{3}$$

where $\varphi : \mathcal{S} \cup \mathcal{O} \rightarrow \mathbb{R}^m$ maps the primitives, *i.e.* objects and states, into their corresponding m-dimensional embedding vectors. The input embeddings of a composition is initialized as the average of its primitive embeddings. All the node representation in $\mathcal{V}_0$ are fixed and initialized with word embeddings, *e.g.* [51]. A crucial element of the graph is the adjacency matrix A. CGE connects all states/objects to objects/states that form at least one composition in the dataset, all composition to their corresponding primitives, and vice-versa. Formally, for two elements $k_i, k_j \in \mathcal{K}$, the value of the adjacency matrix at row i, column j is:

$$A_i^j = \begin{cases} 1 & \text{if } (k_i, k_j) \in \mathcal{C}^t, \\ 1 & \text{if } (k_j, k_i) \in \mathcal{C}^t, \\ 1 & \text{if } k_j \in k_i \ \wedge \ k_i \in \mathcal{C}^t, \\ 1 & \text{if } k_i \in k_j \ \wedge \ k_j \in \mathcal{C}^t, \\ 1 & \text{if } i = j, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $k_j \in k_i$ is true if $k_i = (k_j, o) \vee k_i = (s, k_j)$. The first case in Eq. (4) denotes the connection between states and objects belonging to the same composition, while the second and the third rows denote the connections between compositions and their base primitives. We highlight that this formulation allows the model to propagate the information through the graph, obtaining better node embeddings for both the seen and unseen compositional labels. For example, the GCN allows an unseen composition *e.g. old dog* to aggregate information from its seen neighbor nodes *e.g. old*, *dog*, *cute dog*, and *old car*. Note that the operation performed to obtain the embeddings is the same of a standard GCN. However, we build the graph nodes and edges in a compositional manner, exploiting our inductive bias (*i.e.* importance of connecting states, objects and their compositions) that is specific for CZSL.

Objective function. The final element of our model is the compatibility score h. We implement h as the cosine similarity between the visual and compositional embeddings:

$$h(y,z) = \cos(y,z) = \frac{y^\intercal z}{||y||\,||z||} \tag{5}$$

to produce bounded scores and it is beneficial to avoid prediction to be influenced by the higher magnitude of scores for seen training classes [52] while generalizing better to new ones [53]. This is a greater challenge for our model compared to CGE [5] since we tailor it for the open world. Finally, we learn the mappings ϕ and ω by minimizing the cross-entropy loss over the cosine logits.

$$\mathcal{L} = -\frac{1}{|\mathcal{T}|} \sum_{(x,c)\in\mathcal{T}} \log \frac{e^{\frac{1}{T}\cdot p(x,c)}}{\sum_{y\in\mathcal{C}^s} e^{\frac{1}{T}\cdot p(x,y)}} \tag{6}$$

where T is a temperature value that scales the probabilities values for the cross-entropy loss [54] and $p(x,c) = \cos(\phi(x), \omega(c))$. By exploiting graph embeddings and bounding the classifier scores for seen and unseen compositions, our Co-CGE achieves outstanding performance on the closed world scenario. In the following we discuss how we extend Co-CGE to the more challenging OW-CZSL.

3.2 From Closed to Open World CZSL

OW-CZSL setting requires avoiding distractors, *i.e.* unlikely concepts such as *ripe dog*. The similarity among objects and states can be used as a proxy to estimate the feasibility of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3163667, IEEE Transactions on Pattern Analysis and Machine Intelligence

each composition. We can then inject the estimated feasibility into Co-CGE both as margins in the loss function and as weights within the adjacency matrix.

Estimating Compositional Feasibility. Let us consider two objects, namely *cat* and *dog*. We know, from our training set, that *cats* can be *small* and *dogs* can be *wet* since we have at least one image for each of these compositions. However, the training set may not contain images of *wet cats* and *small dogs*, which we know are feasible in reality. We conjecture that similar objects share similar states while dissimilar ones do not. Hence, it is safe to assume that the states of *cats* can be transferred to *dogs* and vice-versa.

With this idea in mind, we define the feasibility score of s composition $c = (s, o)$ with respect to the object o as:

$$\rho_{obj}(s, o) = \max_{\hat{o} \in \mathcal{O}^s} cos(\phi(o), \phi(\hat{o})) \tag{7}$$

with $\mathcal{O}^s$ being the set of objects associated with state s in the training set $\mathcal{C}^s$, *i.e.* $\mathcal{O}^s = \{o | (s, o) \in \mathcal{C}^s\}$. Note, that the score is computed as the cosine similarity between the object embedding produced by the graph and the most similar other object with the target state, thus the score is bounded in $[-1, 1]$. Training compositions get assigned the score of 1. Similarly, we define the score with respect to the state s as:

$$\rho_{state}(s, o) = \max_{\hat{s} \in \mathcal{S}^o} cos(\phi(s), \phi(\hat{s})) \tag{8}$$

with $\mathcal{S}^o$ being the set of states associated with the object o in the training set $\mathcal{C}^s$, *i.e.* $\mathcal{S}^o = \{s | (s, o) \in \mathcal{C}^s\}$. The feasibility score for a composition $c = (s, o)$ is then:

$$\rho(c) = \rho(s, o) = g(\rho_{state}(s, o), \rho_{obj}(s, o)) \tag{9}$$

where g is a mixing function, *e.g.* max operation ($g(x, y) = \max(x, y)$) or the average ($g(x, y) = (x + y)/2$), keeping the feasibility score bounded in $[-1, 1]$. Note that, while we focus on extracting feasibility from the visual information, external knowledge (*e.g.* knowledge bases [55], language models [56]) can be complementary resources.

A simple strategy to use the feasibility scores would be to consider all compositions above the threshold τ as valid and others as distractors:

$$f_{\text{HARD}}(x) = \underset{c \in \mathcal{C}^t, \rho(c) > \tau}{\arg\max} \; cos(\omega(x), \phi(c)). \tag{10}$$

However, this strategy might be too restrictive in practice. For instance, *tomatoes* and *dogs* being far in the embedding space does not mean that a state for *dog*, *e.g. wet*, cannot be applied to a *tomato*. Therefore, considering the feasibility scores as the golden standard may lead to excluding valid compositions (see Figure 1). To sidestep this issue, we inject the feasibility scores directly into both the model and the training procedure. We argue that doing so enforces separation between most and least feasible unseen compositions in the shared embedding space.

Feasibility-aware objective. First, we integrate the feasibility scores $\rho(c)$ directly within our objective function as margins, defining the new objective as:

$$\mathcal{L}_{\text{f}} = -\frac{1}{|\mathcal{T}|} \sum_{(x,c) \in \mathcal{T}} \log \frac{e^{\frac{1}{T} \cdot p_{\text{f}}(x,c)}}{\sum_{y \in \mathcal{C}} e^{\frac{1}{T} \cdot p_{\text{f}}(x,y)}} \tag{11}$$

with:

$$p_{\text{f}}(x, c) = \begin{cases} \cos(\omega(x), \phi(c)) & \text{if } c \in \mathcal{C}^s \\ \cos(\omega(x), \phi(c)) - \alpha\rho(c) & \text{otherwise} \end{cases} \tag{12}$$

where $\rho(c)$ are used as margins for the cosine similarities, and $\alpha > 0$ is a scalar factor. With Eq. (11) we include the full compositional space while training with the seen compositions data to raise awareness of the margins between seen and unseen compositions directly during training.

Note that, since $\rho(c_i) \neq \rho(c_j)$ if $c_i \neq c_j$ and $c_i, c_j \notin \mathcal{C}^s$, we have a different margin, *i.e.* $-\alpha\rho(c)$, for each unseen composition c. In this way, we penalize less the more feasible compositions, pushing them closer to the seen ones, to which the visual embedding network is biased. At the same time, we force the network to push the representation of less feasible compositions away from the compositions in $\mathcal{C}^s$ in $\mathcal{Z}$. More feasible unseen compositions will then be more likely to be predicted by the model than the less feasible ones (which are more penalized). As an example (Figure 1, top part), the unfeasible composition *ripe dog* is more penalized than the feasible *wet tomato* during training, with the outcome that the optimization procedure does not force the model to reduce the region of *wet tomato*, while reducing the one of *ripe dog* (top-right pie).

We highlight that in this stage we do not explicitly bound the revised scores $p_{\text{f}}(c)$ to $[-1, 1]$. Instead, we let the network implicitly adjust the cosine similarity scores during training. We also found it beneficial to linearly increase α till a maximum value as the training progresses, rather than keeping it fixed. This permits the model to gradually introduce the feasibility margins within the objective while exploiting improved primitive embeddings to compute them.

Feasibility-driven graph. Modelling the relationship between seen and unseen compositions via GCN is more challenging in the open world scenario, since also less feasible unseen compositions will influence the graph structure. This leads to two problems. The first is that distractors will influence the embeddings of seen compositions, making them less discriminative. The second is that the gradient flow will push unfeasible compositions close to seen ones, making harder to isolate distractors in the embedding space.

For these reasons, we modify the adjacency matrix of Eq. (4) in a weighted fashion, with the goal of reducing both the gradient flow on and the influence of less feasibile compositions. To achieve this goal, we directly exploit the feasibility scores, defining the adjacency matrix as:

$$A_i^j = \begin{cases} \max(0, \rho(k_i, k_j)) & \text{if } (k_i, k_j) \in \mathcal{C}, \\ \max(0, \rho(k_j, k_i)) & \text{if } (k_j, k_i) \in \mathcal{C}, \\ 1 & \text{if } k_j \in k_i \wedge k_i \in \mathcal{C}, \\ \max(0, \rho(k_i)) & \text{if } k_i \in k_j \wedge k_j \in \mathcal{C}, \\ 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

In Eq. (13), the connection between a state s and an object o corresponds to the feasibility of the composition (s, o), such that the higher is the feasibility of the composition and the stronger is the connection among the two constituent primitives. Similarly, the influence of a composition to its primitives (third row) corresponds to the feasibility of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3163667, IEEE Transactions on Pattern Analysis and Machine Intelligence

			Training		Validation			Test		
Dataset	s	o	sp	i	sp	up	i	sp	up	i
MIT-States [8]	115	245	1262	30k	300	300	10k	400	400	13k
UT-Zappos [9]	16	12	83	23k	15	15	3k	18	18	3k
C-GQA (Ours)	413	674	5592	27k	1252	1040	7k	888	923	5k

TABLE 1: **Dataset statistics for CZSL**: We use three datasets to benchmark our method against the baselines. C-GQA (ours): our proposed dataset splits from Stanford GQA dataset [57]. (s: # states, o: # objects, sp: # seen compositions, up: # unseen compositions, i: # images)

composition itself. We found that it is beneficial to influence the embedding of a composition $c = (s, o)$ fully by the embeddings of its primitives s and o (Eq. (13), second row). The motivation is that the mapping between compositions and primitives is not bijective: one composition corresponds to only one state and one object, but states and objects build multiple compositions. So while a composition is surely connected with its constituent primitives (second row, value 1), a state and an object are more related to existing, feasible compositions (third row).

The formulation of Eq. (13) makes the connections in the graph dependent on the feasibility of the compositions. This allows the model to reduce the impact of less feasible compositions both in the forward pass and in the backward, making the shared embedding space more discriminative and less-influenced by distractors.

Discussion. Our Co-CGE model uses a GCN to map compositions to the shared embedding space, and a cosine classifier to measure the compatibility between image features and composition embeddings. This formulation merges and extends our previous models CGE [5] and CompCos [10]. In particular, as in CGE we model the relationship between seen and unseen compositions through a graph. This allows us to perform end-to-end training of the CNN backbone without overfitting, since the feature representation is regularized by the compositional graph.

Naïvely applying CGE is not effective in the open world scenario, where we need to model the feasibility of each composition. Thus, following CompCos, we estimate the feasibility scores of each compositions and using the scores as margins in the objective function, with a cosine similarity-based classifier. We improve CompCos by modeling the feasibility of each composition also within the model by defining a weighted adjacency matrix for the GCN, with the weights dependent on the feasibility scores. Moreover, the primitive embeddings used to compute the feasibility scores, are produced by the GCN (thus influenced by the respective compositions) rather than learned in isolation, as in CompCos. These modifications allow Co-CGE to build a more discriminative shared embedding space where the compatibility function better isolates less feasible compositions. Finally, since the model is already robust enough to the presence of distractors in OW-CZSL, we do not need to use hard masking in Eq. (10).

4 Experiments

Datasets. We perform our experiments on three datasets (see Table 1). We adopt the standard split of MIT-States [8] from [3]. For the open world scenario, 26114 out of 28175 (∼93%) are not present in any splits of the dataset but are included in our open world setting. In UT-Zappos [9], [58] we follow the splits from [3]. Note that although 76 out of 192 possible compositions (∼40%) are not in any of the splits of the dataset, we consider them in our open world setting.

Both UT-Zappos and MIT-States have limitations. UT-Zappos [9], [58] is arguably not entirely compositional as states like *Faux leather* vs *Leather* are material differences not always observable as visual transformations. MIT-States instead contains images collected through older search engine with limited human annotation leading to significant label noise [35]. To address the limitations of these two datasets, in our previous work [5] we introduced a split built on top of Stanford GQA dataset [57], *i.e.* the Compositional GQA (C-GQA) dataset. In this work we extend it to the OW-CZSL task. With 413 states and 674 objects, the resulting OW-CZSL search space has almost 280K compositions making it way more challenging than other benchmarks.

Metrics. In zero-shot learning, models being trained only on seen $\mathcal{Y}_s$ labels (compositions) causes an inherent bias against the unseen $\mathcal{Y}_n$ labels. As pointed out by [3], [59], the model thus needs to be calibrated by adding a scalar bias to the activations of the novel compositions to find the best operating point and evaluate the GCZSL performance.

We adopt the evaluation protocol of [3] and report the Area Under the Curve (AUC) (in %) between the accuracy on seen and unseen compositions at different operating points with respect to the bias. The best unseen accuracy is calculated when the bias term is large, *i.e.* the model predicts only the unseen labels, also known as zero-shot performance. In addition, the best seen performance is calculated when the bias term is negative, *i.e.* the model predicts only the seen labels. As a balance between the two, we also report the best harmonic mean (HM). We emphasize that our C-GQA dataset splits and the MIT-States and UT-Zappos dataset splits from [3] do no not violate the zero-shot assumption as results are ablated on the validation set. We therefore advice future works to also use our splits.

Benchmark and Implementation Details. Following [3], [4] we use a ResNet18 pretrained on ImageNet [60] as feature extractor ω and fine-tune the whole architecture with a learning rate of $5 \cdot 10^{-6}$, *i.e.* Co-CGE. For a fair comparison with the models that use a fixed feature extractor, we also perform experiments with a simplification of our model where we learn a 3 layer fully-connected (FC) network with ReLU [61], LayerNorm [62] and Dropout [63] while keeping the feature extractor fixed, *i.e.* Co-CGE$_{\text{ff}}$.

We initialize the embedding function φ with 300-dimensional word2vec [51] embeddings for UT-Zappos and C-GQA, and with 600-dimensional word2vec+fastext [64] embeddings for MIT-States, following [65], keeping the same dimensions for the shared embedding space $\mathcal{Z}$. We train both ω and ϕ using Adam [66] optimizer with a learning rate and a weight decay set to $5 \cdot 10^{-5}$. For both Co-CGE and CompCos, the margin factor α and the temperature T are set to 0.4 and 0.05 respectively for MIT-States, 1.0

Training	Method	MIT-States				UT-Zappos				C-GQA			
		S	U	HM	AUC	S	U	HM	AUC	S	U	HM	AUC
Closed	AoP [2]	14.3	17.4	9.9	1.6	59.8	54.2	40.8	25.9	17.0	5.6	5.9	0.7
	LE+ [1]	15.0	20.1	10.7	2.0	53.0	61.9	41.0	25.7	18.1	5.6	6.1	0.8
	TMN [3]	20.2	20.1	13.0	2.9	58.7	60.0	45.0	29.3	23.1	6.5	7.5	1.1
	SymNet [4]	24.2	25.2	16.1	3.0	49.8	57.4	40.4	23.4	26.8	10.3	11.0	2.1
	CompCos^{CW}	25.3	24.6	16.4	4.5	59.8	62.5	43.1	28.1	28.1	11.2	12.4	2.6
	CGE$_{ff}$	28.7	25.3	17.2	5.1	56.8	63.6	41.2	26.4	28.1	10.1	11.4	2.3
	Co-CGE$_{ff}^{CW}$	27.8	25.2	17.5	5.1	58.2	63.3	44.1	29.1	29.3	11.9	12.7	2.8
	CGE	**32.8**	28.0	**21.4**	6.5	**64.5**	**71.5**	**60.5**	33.5	**33.5**	**15.5**	**16.0**	**4.2**
	Co-CGE^{CW}	32.1	**28.3**	20.0	**6.6**	62.3	66.3	48.1	**33.9**	33.3	14.9	15.5	4.1
Open	CompCos	25.6	22.7	15.6	4.1	59.3	61.5	40.7	27.1	28.4	10.7	11.5	2.4
	Co-CGE$_{ff}$	26.4	23.3	16.1	4.3	60.1	62.1	44.0	29.2	28.7	10.0	10.7	2.2
	Co-CGE	**30.3**	**24.1**	**17.3**	**5.1**	**61.2**	**64.8**	**47.2**	**31.9**	**31.0**	**13.3**	**14.1**	**3.4**

TABLE 2: **Closed World CZSL results** on MIT-States, UT-Zappos and C-GQA. We measure best seen (S) and unseen accuracy (U), best harmonic mean (HM), and area under the curve (AUC) on the compositions. CW denotes closed-world version of the model, ff denotes frozen feature extractor.

and 0.02 for UT-Zappos, and 0.1 and 0.02 for C-GQA. We linearly increase α from 0 to these values during training, reaching them after 15 epochs. We consider the mixing function g as the average to merge state and object feasibility scores for both our model and CompCos. For CompCos we additionally use f_{HARD} as predictor, unless otherwise stated.

For ϕ we use a shallow 2-layer GCN with a hidden dimension of 4096 in the closed world experiments, but for UT-Zappos, where we use a dimension of 300. For the OW-CZSL experiments, we found beneficial to reduce the hidden dimension to the same of the input embeddings, i.e to 300 for UT-Zappos and C-GQA, 600 for MIT-States. Note that since the C-GQA search space is extremely high in the OW-CZSL setting, to test CGE and the closed world version of Co-CGE, we reduce their hidden dimension to 1024. In the tables, we denote with the superscript CW the closed-world version of the models while with the subscript ff the method with frozen feature extractor.

We compare with four state-of-the-art methods, Attribute as Operators (AOP) [2], considering objects as vectors and states as matrices modifying them [2]; LabelEmbed+ (LE+) [1], [2] training a classifier merging state and object embeddings with an MLP; Task-Modular Neural Networks (TMN) [3], modifying the classifier through a gating function receiving as input the queried state-object composition; and SymNet [4], learning object embeddings showing symmetry under different state-based transformations. We also compare Co-CGE with our previous works, CGE [5] and CompCos [10]. Note that both CGE and Co-CGE^{CW} need to produce embeddings for all unseen compositions to be applicable at inference time, thus, in OW-CZSL objects are connected to all states and with the compositions they take part in. We train each model with their default hyperparameters, reporting the closed and open world results of the models with the best AUC on the validation set. We implement our method in PyTorch [67] and train on a Nvidia V100 GPU. For baseline comparisons, we use the authors' implementations where available.

4.1 Closed World CZSL

Comparison with the State of the Art. We experiment with the closed world setting on the test sets of all three datasets. Table 2 (top) shows models trained with the closed world assumption, while Table 2 (bottom) shows the open world models, not using any prior on the unseen test compositions during training but still predicting over a closed set.

Co-CGE^{CW} achieves either comparable or superior results to the state of the art in all settings and metrics. In general, the results of Co-CGE^{CW} are comparable or superior to CGE, while surpassing by a margin the other approaches in the closed world, *e.g.* on MIT-States Co-CGE^{CW} vs CGE achieves an AUC of 6.6 vs 6.5. This signifies the importance of graph based methods for CZSL as both outperform the closest non graph baseline CompCos by a large margin. Note that the main difference between CGE and Co-CGE^{CW} is the classifier, using cosine-similarity scores in Co-CGE^{CW} while a linear operation in CGE. For this reason, we do not expect their performance to substantially differ in the closed-world setup.

Similar observations apply to UT-Zappos, where Co-CGE^{CW} is superior to all methods in AUC (33.9 vs 33.5 of CGE). However CGE outperforms our model for the best seen, unseen and HM. This signifies that while our model does not achieve the best accuracies, it is less biased between the seen and unseen classes leading to a better AUC. Our open-world model (Co-CGE) achieves lower results than the CGE counterpart (*e.g.* 33.5 AUC of CGE vs 31.9 of Co-CGE). However, Co-CGE is not using any prior on unseen classes during training. A model exploiting this prior can produce embeddings more discriminative for seen compositions, not influenced by non-existing unseen ones. Under this light our results are remarkable, since Co-CGE achieves results close to the state of the art. Note that, the impact of such prior becomes more evident with end-to-end training, since a large number of parameters can exploit it. Without end-to-end training, our open world model (Co-CGE$_{ff}$) even surpasses CGE$_{ff}$, (*e.g.* 29.2 vs 26.4 AUC).

Finally, in the challenging C-GQA dataset, Co-CGE^{CW} achieves results comparable to CGE in terms of AUC (4.1 vs 4.2), almost twice the AUC of SymNet (2.1)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3163667, IEEE Transactions on Pattern Analysis and Machine Intelligence

Connections in Graph	AUC	Best HM
a) Visual Product	3.8	14.5
b) Direct Word Embedding average	5.9	19.4
c) Direct Word Embedding concat	6.1	19.7
d) c→ p, p→ c, no self-loop on y	7.6	18.6
e) c→ p, p→ c	8.1	22.7
f) *CGE*: c→ p, p→ c, and s↔ o	**8.6**	**23.3**
g) ***Co-CGE^{CW}***: c→ p, p→ c, and s↔ o	7.9	22.5

TABLE 3: **Ablation over the graph connections** validates the structure of our proposed graph on the validation set of MIT-States dataset. We start from directly using the word embeddings as classifier weights to learning a globally consistent embedding from a GCN as the classifier weights (s: states, o: objects, p: primitives, y: compositional labels).

and almost four times the ones of TMN (1.1). Furthermore, Co-CGE$^{CW}_{ff}$ is the best among the non fine-tuned methods in all metrics (*e.g.* w.r.t. CompCos^{CW}, 2.8 AUC vs 2.6, 12.7 HM vs 12.4, 11.9 unseen accuracy vs 11.2). Remarkably, Co-CGE achieves better results than any closed-world method but CGE and Co-CGE^{CW} under all metrics (*i.e.* 3.4 AUC, 14.1 HM and 13.3 unseen accuracy), despite not exploiting any prior on unseen compositions during training.

Ablation Study. We perform an ablation study with respect to the various connections in our compositional graph on the validation set of MIT-States and report results in Table 3. We start with standard cross-entropy loss, as in CGE and we then include the cosine similarity-based classifier.

As first baselines we consider three approaches not modeling the relationship between state-objects and their compositions. These approaches are *Visual Product* (row a), classifying objects and states independently with two different predictors, and *Direct Word Embedding average* (b) and *Direct Word Embedding concat* (c), where the embeddings of each composition are initialized as the average (or concatenation) of its constituent object and state embeddings. As the results show, Visual Product achieves only 3.8 AUC and 19.5% of best HM. Direct word embeddings achieve better results, with 5.9 AUC and 19.5% HM when averaging, and 6.1 AUC and 19.7% HM when concatenating them. Interestingly, using cross-entropy as a loss function (rather than standard triplet or binary cross-entropy ones) provides already very good results w.r.t. the competitors (*e.g.* 6.1 AUC of direct word embedding concat vs 4.3 reported in [4]). This because cross-entropy directly models the relative similarity between all seen compositions and the ground-truth, opposite to the binary relationships of BCE and of sampled comparisons of the triplet loss.

Including the graph by simply connecting the primitives (*i.e.* states and objects, p) to compositional labels (y) but without self connections for the compositional label (row d) achieves an improvement on AUC (7.6) while a slight decrease on best HM (18.6%). When we include self connections in the graph for compositions, (row e) outperform all approaches treating objects and states independently by a margin, with an AUC of 8.1 and a best HM of 22.7. This demonstrates the benefit of connecting objects, states and compositions. Row (f) is the final CGE model, which additionally incorporates the connections between states (s) and objects (o) in a pair to model the dependency between them. We observe that learning a representation that is consistent with states, objects and the compositional labels increases the AUC from 8.1 to 8.6 validating the choice of connections in the graph. Finally, if we employ a cosine classifier to replace the dot product classifier of CGE, we see in row (g) that the AUC and HM are comparable. Note that, with this variant we can use the feasibility scores as margins in the objective tailoring the method for OW-CZSL.

4.2 Open World CZSL

Comparing with the State of the Art. As shown in Table 4, the first clear effect of moving to the more challenging OW-CZSL setting is the severe decrease in performance for every method. The largest decrease in performance is on the best unseen metric, due to the presence of a large number of distractors. As an example, in MIT states LE+ goes from 20.1% to 2.5% of best unseen accuracy and even the previous state of the art, CGE, loses 22.9%. Similarly, in C-GQA the best seen accuracy drops of 6.8% for SymNet, 8.7% for CompCos^{CW} and even the end-to-end trained CGE and Co-CGE^{CW} lose 11.8% and 12.3% respectively.

Compared to the baselines, our models, Co-CGE$_{ff}$ and Co-CGE are more robust to the presence of distractors, *e.g.* particularly for the best HM performance on MIT-States, Co-CGE$_{ff}$ surpasses CompCos by 1.2% at the same feature extractor. This demonstrates the importance of explicitly modeling the different feasibility of the compositions in the whole compositional space, injecting them within the objective function and the graph connections. Similar considerations apply to Co-CGE, that achieves the best results in MIT-States and C-GQA wrt all metrics. Remarkably, it achieves a 0.78 of AUC on C-GQA which is comparable to the closed world results of early CZSL methods, such as AoP (0.7) and LE+ (0.8). Over CompCos, the improvements are clear also in the accuracy on unseen classes (+1.8% on MIT-States, +1.2 % on C-GQA) and harmonic mean (+1.8% on MIT-States, and +2.0% on C-GQA). Interestingly, MIT states contains label noise, mostly linked to ambiguities in attribute annotations [35]. Under this light, the performance of Co-CGE are remarkable, showing also robustness to label noise even in OW-CZSL. In UT-Zappos the performance gap with the other approaches is more nuanced. This is because the vast majority of compositions in UTZappos are feasible, thus it is hard to see a clear gain from injecting the feasibility scores into the training procedure. Nevertheless, Co-CGE achieves the best HM mean (40.8%) and AUC (23.3). However, a good closed world model performs well in this scenario, as showed by the performance of CGE, achieving the best accuracy on unseen classes (47.7%). However, the overall results being lower than the closed setting indicates that OW-CZSL setting poses an open challenge.

For C-GQA, we observe that due to the huge search space (almost 280k compositions) achieving good OW-CZSL performance is extremely hard compared to the much smaller search space of MIT-States (almost 30k compositions) and UT-Zappos (192). It is interesting to highlight how the relative performance degradation of Co-CGE in moving to the OW-CZSL is 27.0% for UT-Zappos (31.9 vs 23.3 AUC), 54.9% on MIT-States (5.1 vs 2.3 AUC), and 77.1%

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3163667, IEEE Transactions on Pattern Analysis and Machine Intelligence

Training	Method	MIT-States				UT-Zappos				C-GQA			
		S	U	HM	AUC	S	U	HM	AUC	S	U	HM	AUC
Closed	AoP [2]	16.6	5.7	4.7	0.7	50.9	34.2	29.4	13.7	NA	NA	NA	NA
	LE+ [1]	14.2	2.5	2.7	0.3	60.4	36.5	30.5	16.3	19.2	0.7	1.0	0.08
	TMN [3]	12.6	0.9	1.2	0.1	55.9	18.1	21.7	8.4	NA	NA	NA	NA
	SymNet [4]	21.4	7.0	5.8	0.8	53.3	44.6	34.5	18.5	26.7	2.2	3.3	0.43
	$\text{CompCos}^{\text{CW}}$	25.3	5.5	5.9	0.9	59.8	45.6	36.3	20.8	28.0	1.0	1.6	0.20
	CGE_{ff}	29.6	4.0	4.9	0.7	58.8	46.5	38.0	21.5	28.3	1.3	2.2	0.30
	$\text{Co-CGE}^{\text{CW}}_{\text{ff}}$	28.2	6.0	6.5	1.1	59.5	41.5	36.1	20.1	28.9	1.2	2.1	0.29
	CGE	**32.4**	5.1	6.0	1.0	61.7	**47.7**	39.0	23.1	**32.7**	1.8	2.9	0.47
	$\text{Co-CGE}^{\text{CW}}$	31.1	5.8	6.4	1.1	**62.0**	44.3	40.3	23.1	32.1	2.0	3.4	0.53
Open	CompCos	25.4	10.0	8.9	1.6	59.3	46.8	36.9	21.3	28.4	1.8	2.8	0.39
	$\text{Co-CGE}_{\text{ff}}$	26.4	10.4	10.1	2.0	60.1	44.3	38.1	21.3	28.7	1.6	2.6	0.37
	Co-CGE	30.3	**11.2**	**10.7**	**2.3**	61.2	45.8	**40.8**	**23.3**	32.1	**3.0**	**4.8**	**0.78**

TABLE 4: **Open World CZSL results** on MIT-States, UT-Zappos and C-GQA. We measure best seen (S) and unseen accuracy (U), best harmonic mean (HM), and area under the curve (AUC) on the compositions. CW denotes closed-world version of the model, ff denotes frozen feature extractor.

			S	U	HM	AUC
Margins	$\text{CompCos}^{\text{CW}}$		28.0	6.0	7.0	1.2
	CompCos	$\alpha = 0$	25.4	10.0	9.7	1.7
		+ $\alpha > 0$	27.0	10.9	10.5	2.0
		+ warmup α	27.1	11.0	10.8	2.1
Primitives	CompCos	ρ_{state}	26.6	10.2	10.2	1.9
		ρ_{obj}	27.2	10.0	9.9	1.9
		$\max(\rho_{state}, \rho_{obj})$	27.2	10.1	10.1	2.0
		$(\rho_{state} + \rho_{obj})/2$	27.1	11.0	10.8	2.1
Graph Connections		s↔o p→ c c→ p				
	$\text{Co-CGE}_{\text{ff}}$	1 1 ρ	28.4	9.1	9.3	1.8
		1 ρ ρ	26.6	8.1	8.4	1.5
		ρ 1 ρ	28.2	8.9	9.0	1.7
		ρ ρ ρ	29.3	9.7	10.0	2.0
		1 1 1	28.4	11.3	11.6	2.4
		1 ρ 1	28.2	10.8	11.0	2.3
		ρ 1 1	27.2	11.9	11.6	2.3
		ρ ρ 1	29.5	11.5	12.0	2.5
	Co-CGE	end-to-end	30.4	12.4	12.6	2.8

TABLE 5: Results on MIT-States validation set for different ways of applying the margins (top), computing the feasibility scores (middle) and using the scores within the graph (bottom) for CompCos and $\text{Co-CGE}_{\text{ff}}$. (S: seen, U: unseen)

on C-GQA (3.4 vs 0.78 AUC). These results confirm how the larger is the compositional space, the larger is the decrease in performance of CZSL models. Finally, Table 4 shows two interesting trends. The first is the importance of end-to-end training, with CGE and $\text{Co-CGE}^{\text{CW}}$ surpassing all other methods but Co-CGE. The second, is the results of SymNet being slightly superior than $\text{Co-CGE}_{\text{ff}}$ in AUC, *i.e.* SymNet is the only method modeling states and objects separately at classification levels. Therefore, as the search spaces grows, it may beneficial to predict each primitive independently to get an initial estimate of the composition.

In the following experiments, we use MIT-States' validation set to analyze the different choices for our feasibility scores. In particular, we investigate the impact of the feasibility-based margins within the loss functions (starting from CompCos), how they are computed, and how they should be injected within the graph connections. Finally we check the eventual benefit that limiting the output space during inference using f_{HARD} may bring to different models.

Importance of the feasibility-based margins. We check the impact of including all compositions in the objective function (without any margin) and of including the feasibility margin but without any warmup strategy for α. As the results in Table 5 (Top) show, including all unseen compositions in the cross-entropy loss without any margin (*i.e.* $\alpha = 0$) increases the best unseen accuracy by 4% and the AUC by 0.5. This is a consequence of the training procedure: since we have no positive examples for unseen compositions, including unseen compositions during training makes the network push their representation far from seen ones in the shared embedding space. This strategy regularizes the model in presence of a large number of unseen compositions in the output space. Note that this problem is peculiar in the open world scenario since in the closed world the number of seen compositions is usually larger than the unseen ones. The CompCos ($\alpha = 0$) model performs worse than $\text{CompCos}^{\text{CW}}$ on seen compositions, as the loss treats all unseen compositions equally.

Results increase if we include the feasibility scores during training (*i.e.* $\alpha > 0$). The AUC goes from 1.7 to 2.0, with consistent improvements over the best seen and unseen accuracy. This is a direct consequence of using the feasibility to separate the unseen compositions from the unlikely ones. In particular, this brings a large improvement on S and moderate improvements on both U and HM.

Finally, linearly increasing α (*i.e.* warmup α) further improves the harmonic mean due to both the i) improved margins estimated from the updated primitive embeddings and ii) the gradual inclusion of these margins in the objective. This strategy improves the bias between seen and unseen classes (as for the better on harmonic mean) while slightly enhancing the discriminability on seen and unseen compositions in isolation.

Effect of Primitives. We can either use objects as in Eq. (7), states as in Eq. (8)) or both as in Eq. (9) to estimate the feasibility score for each unseen composition. Here we show the impact of these choices on the results in Table 5 (Middle).

We observe that computing feasibility on the primitives

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3163667, IEEE Transactions on Pattern Analysis and Machine Intelligence

alone is already beneficial (achieving an AUC of 1.9) since the dominant states like *caramelized* and objects like *dog* provide enough information to transfer knowledge. In particular, computing the scores starting from state information (ρ_{state}) brings improves the best U and HM. Using similarities among objects (ρ_{obj}) performs well on S while achieving slightly lower performances on U and HM.

Introducing both states and objects give the best result at AUC of 2.1 as it combines the best of both. Merging objects and states scores through their maximum (ρ_{max}) maintains the higher seen accuracy of the object-based scores, with a trade-off between the two on unseen compositions. However, merging objects and states scores through their average brings to the best performance overall, with a significant improvement on unseen compositions (almost 1%) as well as the harmonic mean. This is because the model is less-prone to assign either too low or too high feasibility scores for the unseen compositions, smoothing their scores. As a consequence, more meaningful margins are used in Eq. (11) and thus the network achieves a better trade-off between discrimination capability on the seen compositions and better separating them from unseen compositions (and distractors) in the shared embedding space.

Effect of Graph Connections. For each graph connection, we have two choices: either keeping it unaltered (*i.e.* value 1) or replacing it with the feasibility scores (ρ), as in Eq. (13). In Table 5 (bottom), we analyze these choices, *i.e.* symmetric state and objects connections (s↔o), the connection from primitives to compositions (p→c) and viceversa (c→p).

A clear observation is the importance of keeping the influence of primitives on compositions (c→p) unaltered (equal 1). We conjecture that since most of unseen compositions will get feasibility scores lower than 1, their representations would be updated mainly through their self-connection. However, the seen compositions for which we have supervision fully exploit the representations of their primitives. This causes the representations for unseen compositions to have an inherent distribution shift and being i) poorer with respect to seen one and ii) less discriminative. This is clearly shown in the table from the low HM and best unseen class accuracy of Co-CGE whenever c→p is different than 1. Keeping the connections c→p as 1 allows the model to keep its discrimination capability on unseen classes and a best trade-off between accuracy on seen and unseen classes, with an average improvement of 0.63 in AUC.

For the other two types of connections, (s↔o) and (p→c), the best results are achieved when their weights are set as in Eq. (13), using the feasibility scores. This allows the model to achieve the best AUC (2.5%), HM (12%) and seen accuracy (29.5%) while being slightly inferior to the top method in best unseen accuracy (-0.4%). Note that the advantage of this combination is consistent also for c→p set through ρ.

As a final experiment, we check the benefits of fine-tuning the while representation end-to-end with the best performing combination (c→p and c→p through ρ, c→p). As expected this brings to the best results for all metrics. In particular, the learned representations results more discriminative for both seen and unseen compositions, achieving an improvement of 0.9% on both best seen and best unseen accuracies and a consequent gain of 0.6% in HM.

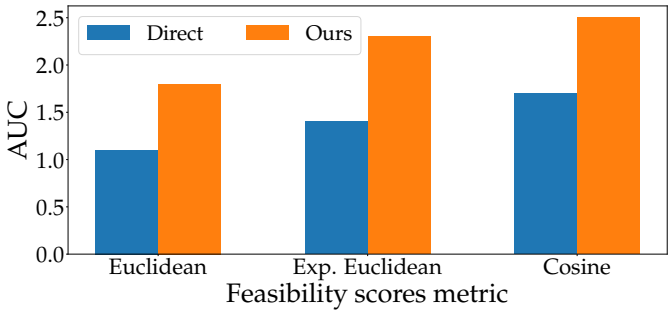

Fig. 2: AUC on MIT-States validation set by computing feasibility scores through different strategies: direct similarity between state and object embeddings (*Direct*), *Ours* (Section 3.2), negative (*Euclidean*) and exponential negative (*Exp. Euclidean*) Euclidean distance, and cosine similarity (*Cosine*).

	Mask	S	U	HM	AUC		Mask	S	U	HM	AUC
			3.1	3.2	0.3				6.0	7.0	1.2
a.	CompCos	14.8	5.0	4.6	0.5	d.	CompCos	28.0	8.1	8.7	1.6
	Co-CGE		6.1	5.4	0.6		Co-CGE		9.3	9.4	1.8
			1.3	1.7	0.1				7.1	8.3	1.6
b.	CompCos	15.9	4.1	4.1	0.4	e.	CompCos	31.2	8.5	9.5	2.0
	Co-CGE		4.7	5.3	0.5		Co-CGE		11.1	11.5	2.6
			7.9	7.6	1.2				12.4	12.6	2.8
c.	CompCos	23.6	7.9	7.7	1.2	f.	CompCos	30.4	12.5	12.6	2.8
	Co-CGE		8.7	8.4	1.4		Co-CGE		12.7	12.8	2.9

TABLE 6: Results on MIT-States validation set by applying feasibility-based binary masks (f_{HARD}) on the methods a. LE+, b. TMN, c. SymNet, d. CompCos$^{\text{CW}}$, e. Co-CGE$^{\text{CW}}$, f. Co-CGE. (S: seen, U: Unseen)

Effect of the Feasibility Computation strategy. In Section 3.2, we consider the cosine similarity to compare the primitive embeddings for computing the feasibility scores, propagating feasible states for an object using its similarity with other objects, and viceversa for the feasible objects for certain states. In this part, we explore alternatives to this choice. First, we test whether directly comparing object and state embeddings (*Direct*) can outperform our strategy (Sec. 3.2). Second, we check if cosine similarity is the best similarity metric in this scenario, testing as alternative either negative Euclidean distance (Euclidean) or its negative exponential version (Exp. Euclidean), using in both cases the negative exponential of the distance to set the weights of the graph connections (to keep them positive). Note that, in the Euclidean case, the classifier is a non-bounded one (*i.e.* linear) while in Exp. Euclidean we keep our cosine classifier (since scores are bounded in the range $[0, 1]$).

Results are shown in Figure 2 for the validation set of MIT-States. As the Table shows, our strategy (*Ours*) largely outperforms the Direct counterpart for all similarity metrics, with an average improvement of 0.8 in AUC (*e.g.* 1.7 vs 2.5 in AUC for *Cosine*). We ascribe this behaviour to the different nature of objects and states: while objects are physical entities, states are modifiers modifying objects' appearance [2] and the representation of a state is influenced by all objects it can be applied to. This leads to high feasibility scores for compositions with high correlation between objects and states (*e.g. inflated balloon*) while low feasibility scores whenever a state can be applied to a large variety of objects (*e.g. wet*). Our strategy sidesteps this problem by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3163667, IEEE Transactions on Pattern Analysis and Machine Intelligence

(a) Positive qualitative results.

(b) Failure cases.

Fig. 3: **Qualitative results.** Positive (left) and negative (right) examples of Co-CGE predictions in the OW-CZSL scenario for MIT-States (top) and C-GQA (bottom), compared with Co-CGE^{CW}.

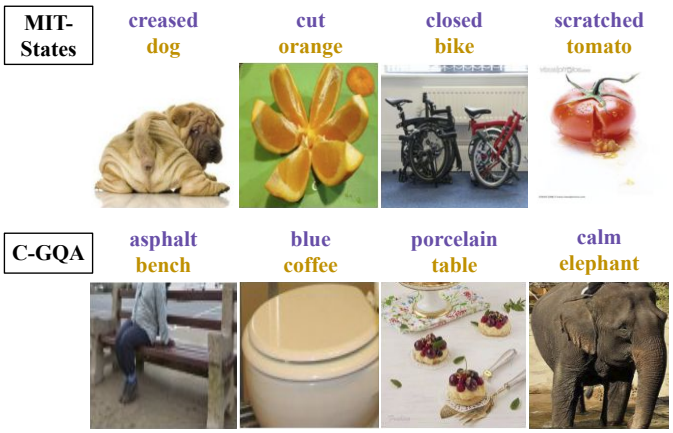

Fig. 4: **Retrieving images from labels.** Top-1 retrieved images by Co-CGE for state-object compositions not present in MIT-States (top) and C-GQA (bottom).

Training / Test	MIT-States C-GQA			C-GQA MIT-States		
	S	U	HM	S	U	HM
SymNet [4]	6.5	0.93	0.83	0.44	0.21	0.10
CGE$_{ff}$	6.3	1.1	1.0	0.38	0.21	0.13
Co-CGE$_{ff}^{CW}$	6.2	1.1	1.0	0.91	0.33	0.15
CGE	6.3	1.4	1.1	**1.5**	0.19	0.14
Co-CGE^{CW}	**7.3**	1.7	1.1	1.0	0.31	0.20
CompCos	6.3	2.5	1.5	0.59	**0.49**	0.17
Co-CGE$_{ff}$	5.5	**3.2**	1.6	0.80	0.31	0.19
Co-CGE	**7.3**	3.0	**2.6**	1.3	0.30	**0.23**

TABLE 7: **Cross-dataset CZSL results** on MIT-States compositions from C-GQA models and vice-versa. We measure best seen (S) and unseen accuracy (U), and best harmonic mean (HM) between the two.

exploiting priors from training compositions and treating similarity between objects and states independently.

For what concerns the similarity metrics, that cosine similarity achieves the best results, independently on the way feasibility scores are computed (*i.e.* 1.7 AUC on Direct vs 2.3 of Exp. Euclidean and 1.8 of Euclidean, 2.5 AUC on *Ours* vs *1.4* of Exp. Euclidean and 1.1 of Euclidean). These results show the importance of bounding the feasibility scores in a predefined range to: i) directly compare their values with the model's output, which is beneficial for the loss function, and ii) easily pruning graph connections by considering as feasible every composition with a score falling in the positive half of the range. Interestingly, also Exp. Euclidean achieves good results, confirming the importance of using bounded feasibility scores as margins in the loss function.

Effect of Hard Thresholding. In Section 3, we described that the feasibility scores can also be used during inference to mask the predictions, *i.e.* using as prediction function Eq. (10) in place of Eq. (1), with the threshold τ computed empirically. We study the impact of this masking computed either with CompCos or with Co-CGE feasibility scores on Co-CGE and the closed world models Co-CGE^{CW}, LE+, TMN, SymNet and CompCos^{CW}, showing the results in Table 6. Note that, since seen compositions are not masked, best S performances do not change.

We observe that applying either ours or CompCos feasibility-based binary masks on top of all closed world approaches is beneficial. This is because the masks are able to filter out the less feasible compositions, rather than simply restricting the output space. In particular, CompCos-based mask brings an average improvement of 0.3 of AUC, 1.4% of HM and 1.6% of best unseen accuracy, while Co-CGE-based improves the AUC of the base model of 0.5 in average, 2.4 in harmonic mean and 2.9 in best unseen accuracy. This suggests that Co-CGE estimates more precise feasibility scores than CompCos, and it can better filter out compositions from the output space.

Interestingly, masking largely improves Co-CGE^{CW} (+1.0 AUC), despite being the best performing closed-world method. On the other hand, SymNet does not benefit of CompCos-based masking (*e.g.* +0.1 HM) and marginally does with Co-CGE mask (*i.e.* +0.8% on unseen accuracy). This suggests that masking the output space with feasibility scores is more beneficial for models predicting object and states together at inference time (*e.g.* LE+, Co-CGE^{CW}) than for those predicting them in isolation (*e.g.* SymNet). Finally, the improvements are minimal (*i.e.* +0.1% AUC) for our open world Co-CGE model, being already robust to the presence of distractors. Since f_{HARD} requires tuning an additional hyperparameter, we do not apply any masking to Co-CGE$_{ff}$ and Co-CGE in the experiments.

4.3 Cross-dataset results

In Section 4.1 and Section 4.2 we tested CZSL models in standard scenarios, where training and test images belong to the same data collection. However, since the final goal is to perform CZSL in the wild (*e.g.* on web images, robotics applications), we would like such models to show robustness to distribution-shifts, *i.e.* changes between training and test distributions. For these reasons, in this section we benchmark CZSL methods in cross-dataset experiments.

To perform such experiment, we consider two datasets, MIT-States and C-GQA. The two datasets share 149 objects and 68 states but have different acquisition conditions: while C-GQA contains images of objects cropped from natural images (with eventual loss in resolution), MIT-States contains images downloaded from the web (where the target object is not centered and with possible label noise). For the experiments, we train on the standard training set of one dataset and test on the other. For testing, we consider all the images of the second dataset where the label is a composition of states and objects included in the first. This amounts on testing with 156 seen and 579 unseen compositions on MIT-States, and 131 seen and 687 unseen compositions on C-GQA. Since there is no standard validation set, we take the best models in the OW-CZSL setting of Table 4, directly applying them on the cross-dataset scenario.

We report the results in Table 7, measuring them in terms of best seen accuracy (S) best unseen accuracy (U), and best harmonic mean of the two (HM). A first highlight from the table is the much lower accuracy of all models in predicting seen compositions. For instance, CGE performs 1.5% on MIT-States and 6.3% on C-GQA vs 32.4% and 32.7% on the same datasets in OW-CZSL (Table 7). This confirms that there is a significant distribution-shift between the two domains. Despite the shift, Co-CGE shows the best overall results in both settings, being always the best in HM and either best or second-best in the other metrics. In detail, when training on MIT-States and testing on C-GQA, our model shows good discriminability on seen classes (7.3%), being on par with its closed-world counterpart and superior to other closed-world models, such as SymNet (6.5%) and CGE (6.5%). On unseen classes, all OW-CZSL methods shows better results, with Co-CGE (3.0%) and Co-CGE$_{\text{ff}}$ (3.2%) achieving the best results, almost doubling the performance of the best closed-world model (CGE, 1.7%). Co-CGE shows the best results overall, with 2.6% HM vs 1.5% of the best OW-CZSL competitor (CompCos) and 1.1% of the best closed-world one (CGE and Co-CGE$^{\text{CW}}$).

Results are consistent when training on C-GQA and testing on MIT-States, with our model achieving competitive seen accuracy (1.3% vs 1.5% of CGE), and unseen accuracy (0.30% vs 0.31% of Co-CGE$^{\text{CW}}$) with closed-world models. With respect to open-world models, CompCos shows remarkable unseen accuracy (0.49%) but much lower discriminability on seen compositions (0.59%). Overall, our model still achieves the best tradeoff between discriminating seen and unseen compositions, achieving 0.23% HM vs 0.20% of the best closed-world model (Co-CGE$^{\text{CW}}$) and 0.17% of CompCos. These results show that the efficacy of our model is not restricted to standard OW-CZSL but generalize across distributions. Nevertheless, the overall results of all models are much lower than the ones without distribution-shift, and future works might specifically focus on improving CZSL across domains.

4.4 Qualitative results

Influence of Feasibility Scores on Predictions. We analyze the reasons for the improvements of Co-CGE over Co-CGE$^{\text{CW}}$, by showing output examples of both models on images of MIT states (top) and C-GQA (bottom). In Figure 3 we compare predictions on unseen compositions for samples where the closed world model is "distracted" while the open world one predicts the correct label (green, Fig. 3a) and examples where also our model fails (red, Fig. 3b).

We observe that the closed world model is generally not capable of dealing with the presence of distractors. For instance, there are cases where the object prediction is correct (*e.g. coiled elephant*, *wilted silk*, *full dog*, *rusty snake*, *viscous cheese*, *swinging uniform*, *messy bottle*, *concrete water*) but the associated state is not only wrong but also making the compositions unfeasible. In other cases, the state prediction is either correct (*e.g. eroded*) or reasonable (*e.g. melted*) but the associated object is not, eventually resulting in unfeasible predictions (*e.g. parked crowd* vs *airplane* in the background). In some cases, both state and object predictions are incorrect for Co-CGE$^{\text{CW}}$, being either unfeasible (*e.g. steaming necklace*, *young chain*, *potted couch*) or confused in the large open world search space (*molten coin*, *old bottle*). All these problems are less severe in our full Co-CGE model since injecting the feasibility of each composition within the objective and the graph connections helps in both reducing the possibility to predict implausible distractors and improving the structure of the compositional space, better discriminating the constituent visual primitives. This occurs even when the predictions of Co-CGE are wrong, being either close to the ground-truth (*i.e. small snake*, *sliced chicken*) or referring to another concept of the image (*i.e. small dog*, *black pole*).

Despite these results, the wrong predictions in Figure 3b highlight some of the limitations of our current model. In particular, the model currently has no localization constraints on the object and state predictions. As a consequence, the predicted state might be linked to an object different than the predicted one (*e.g. eroded building*, *white bottle*). Moreover, Co-CGE does not model foreground information and might focus on different parts of the image than the target ones (*e.g. small dog*, *black pole*, *yellow noodles*). Furthermore, the focus on modeling the distractors might make the model less confident in discriminating known concepts, producing imprecise predictions (*e.g. white bird* vs *white goose*). In the future it will be interesting to revisit the trade-off between discriminating seen compositions and isolating distractors as well as developing constraints to ensure that state and object predictions refer to the same part of the image. Finally, it will be crucial to break the limits of the current CZSL problem formulation, designing an approach that can predict multi-states/objects per image.

Discovering Most and Least Feasible Compositions. The biggest advantage of our method is its ability to estimate the feasibility of each unseen composition, to later inject these estimates into the learning process and the model. Our procedure described in Section 3.2 needs to be robust enough

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3163667, IEEE Transactions on Pattern Analysis and Machine Intelligence

MIT-States		**States**
	Most Feasible (Top-3)	Least Feasible (Bottom-3)
cat	huge, tiny, small	cloudy, browned, standing
tomato	diced, peeled, mashed	full, fallen, heavy
house	ancient, painted, grimy	mashed, wilted, browned
banana	diced, browned, fresh	dull, barren, unpainted
knife	blunt, curved, wide	viscous, standing, runny
C-GQA		**States**
	Most Feasible (Top-3)	Least Feasible (Bottom-3)
dog	small, drinking, toy	horizontal, ridged, styrofoam
wine	pink, red, black	rubber, hard, angled
fruit	unripe, sliced, peeled	angled, glossy, toilet
jacket	tight, closed, sleeveless	porcelain, shaped, miniature
window	glass, beige, purple	packaged, greasy, boiled

TABLE 8: Top-3 highest and Bottom-3 lowest feasible state per object on MIT-States (top) and C-GQA (bottom) as given by Co-CGE.

to model which compositions should be more feasible in the compositional space and which should not, isolating the latter in the shared embedding space. We highlight that here we are focusing mainly on visual information to extract the relationships. This information can be coupled with knowledge bases (*e.g.* [55]) and language models (*e.g.* [56]) to further refine the scores.

Table 8 shows the top-3 most feasible compositions and bottom-3 least feasible compositions given five randomly selected objects for MIT-States (top) and C-GQA (bottom). These objects specific results show a tendency of the model to relate feasibility scores to the subgroups of classes. For instance, cooking states are considered as unfeasible for standard objects (*e.g. mashed house*, *boiled window*) as well as substance-specific states (*e.g. runny knife*). Similarly, states usually associated with substances are considered unfeasible for animals (*e.g. runny cat*). At the same time, size and actions are mostly linked with animals (*e.g. small cat*, *drinking dog*) while cooking states are correctly associated with food (*e.g. diced tomato*, *sliced fruit*).

Interestingly, in MIT-States the top states for *knife* are all present with *blade* as seen compositions, and in C-GQA the top states for *dog* are all present with animals as seen compositions (*e.g. drinking cat*, *small cat*, *small horse*, *toy cat*). This shows that our model exploits the similarities between two objects to transfer these states, *e.g.* from *blade* to *knife* and from *cat* to *dog*, following Eq. (7). Furthermore, the state *standing* is considered as unfeasible for *cat* in MIT-States while being feasible (6th top) for *dog* in C-GQA. This is because the state *standing* has different meanings in the two datasets, *i.e.* buildings (*e.g. standing tower*) in MIT-States, animals and persons (*e.g. standing cat*) in C-GQA. This highlights the strict dependency of the feasibility scores estimated by our model to the particular dataset, with the impossibility to capture polysemy if the dataset does not account for it. These limitations can be overcomed by integrating external information from knowledge bases [55] and language [56].

Retrieving compositions in the open world. In the OW-CZSL scenario there is no limitation in the output space of the model. Thus, we can predict arbitrary state-object compositions at test time and, eventually, retrieve the closest images to arbitrary concepts. Here we explore the latter scenario and we check which images are the closest to the embeddings of random compositions for state and objects not present in the original datasets. The results are shown in Figure 4 for MIT-States (top) and C-GQA (bottom). When the composition is feasible (*e.g. calm elephant*, *cut orange*) the model retrieves an image depicting the exact concept. When the composition is inexact for the real world (*e.g. closed* vs *folded bike*, *scratched* vs *broken tomato*, *creased* vs *wrinkled dog*) the model still retrieves reasonable images, showing its ability to capture the underlying effect that a state is supposed to have on the particular object it refers. Finally, when the composition has an unclear meaning, the model tends to retrieve images containing both state and objects, even if present in isolation. This is the case of *asphalt bench*, where the *bench* is close to the an *asphalt* road, and of *porcelain table*, where the image shows a *table* with *porcelain* crockery. Clearly, our model is not perfect and may retrieve images less representative of the queried concept. An example is *blue coffee*, where the retrieved image contains a cup with blue lines.

Training	**Test**	**MIT-States**				**C-GQA**				**Average**		
		MIT-Sta.		**C-GQA**		**MIT-Sta.**		**C-GQA**				
		S	U	S	U	S	U	S	U	S	U	HM
CGE_{ff}		23.3	22.5	9.9	4.2	14.1	6.7	12.2	3.8	14.9	9.3	10.9
Co-CGE_{ff}^{CW}		26.8	**30.8**	13.7	4.8	11.5	5.4	11.1	2.9	15.8	**11.0**	11.9
CGE		25.5	24.8	12.2	**5.1**	**16.0**	5.4	12.5	**4.6**	16.6	9.9	11.7
Co-CGE^{CW}		**30.5**	26.0	**15.3**	**5.1**	9.6	**7.3**	**12.8**	3.6	**17.1**	10.5	**12.4**
CompCos		26.2	25.0	**13.7**	3.9	**15.4**	6.0	**12.6**	**3.6**	**17.0**	9.6	11.5
Co-CGE_{ff}		**28.0**	**30.0**	12.2	4.1	9.0	**6.9**	10.5	2.8	14.9	**11.0**	11.8
Co-CGE		25.0	27.3	**13.7**	**6.1**	12.8	5.7	12.4	3.3	16.0	10.6	**11.9**

TABLE 9: **Compositional retrieval results** on MIT-States and C-GQA, including cross-dataset tests. We report the top-1 accuracy on seen (S) and unseen (U) compositions.

4.5 Image-retrieval from compositions: results and limitations

In the previous section, we reported a qualitative study of retrieving compositions of arbitrary objects and states using our OW-CZSL model. A natural question is whether the learned compositional embeddings can be used to reliably retrieve images of the corresponding concepts. We conducted this study on MIT-States and C-GQA performing the experiments both within and across dataset, following the setup of Section 4.3 for the latter. In this setting, we benchmark the best methods of Table 4 that explicitly learn compositional embeddings: CGE, Co-CGE^{CW} and their frozen counterparts among the closed-world CZSL methods, and all the OW-CZSL ones (*i.e.* CompCos, Co-CGE_{ff} and Co-CGE). We measure the results according to accuracy on seen compositions (S), unseen compositions (U) and their average across all settings, together with their average harmonic mean (HM). Since focusing only on seen compositions may lead to more discriminative compositional embeddings, we highlight the results of the best models trained in the closed- and open-world settings separately.

We report the results of our experiment in Table 9. As a first observation, there is no clear winner across all settings. For instance, Co-CGE^{CW} achieves the best results on seen compositions across all settings but the cross-dataset experiment C-GQA to MIT-States, where it achieves poor performance (9.6% vs 16.0% of CGE). Similarly, Co-CGE shows the best unseen accuracy on the cross-dataset

experiment MIT-States to CGQA, but lower than its non-finetuned counterpart when testing on MIT-States (*i.e.* 30.0% vs 27.3% for within, and 6.9% vs 5.7% for cross dataset experiments). CompCos shows good retrieval results when trained on C-GQA, achieving the best accuracy on seen classes among OW-CZSL models and either best or second-best on unseen ones. Overall, our Co-CGE$^{\text{CW}}$ and Co-CGE shows the best trade-off between seen and unseen accuracy in all setting, achieving a slightly better average HM than the best competitors (*i.e.* 11.9% Co-CGE vs 11.8 Co-CGE$_{\text{ff}}$ and 11.5% CompCos, 12.4% of Co-CGE$^{\text{CW}}$ vs 11.9% of Co-CGE$^{\text{CW}}_{\text{ff}}$ and 12.4%), with cosine-based classifiers always surpassing their non-cosine counterpart in all metrics (*i.e.* CGE vs Co-CGE$^{\text{CW}}$ and CGE$_{\text{ff}}$ vs Co-CGE$^{\text{CW}}_{\text{ff}}$).

These results show that better CZSL results do not necessarily lead to better compositional embeddings. For instance, while in Table 4 Co-CGE and clearly surpasses CompCos (*i.e.* 0.78 AUC on C-GQA, 2.3 on MIT-States vs 0.39 and 1.6 respectively) the margins in the retrieval setting are smaller (*e.g.* 10.6% vs 9.6% on average unseen accuracy) and the best model may even change (*e.g.* 17.0% average seen accuracy for CompCos vs 16.0% of Co-CGE). Refining the compositional embeddings by exploiting advances on metric learning (*e.g.* [68], [69]) may improve results on both compositional-retrieval and standard CZSL.

5 Conclusions

In this work, we address the compositional zero-shot learning (CZSL) problem, where the goal is recognizing unseen compositions of seen state-object primitives. We focus on the open world compositional zero-shot learning (OW-CZSL) task, where all the combinations of states and objects could potentially exist. We propose a way to model the feasibility of a state-object composition by using the visual information available in the training set. This feasibility is independent of an external knowledge base and can be directly incorporated in the optimization process. We propose a novel model, Co-CGE, that models the relationship between primitives and compositions through a graph convolutional neural network. Co-CGE incorporates the feasibility scores in two ways: as margins for a cross-entropy loss and as weights for the graph connections. Experiments show that our approach is either superior or comparable to the state of the art in closed world CZSL while improving it by margin in the open world setting. In the future, we plan to study new ways of computing our feasibility scores, as well as the effectiveness of our compositional graph in other problems with interacting sets (*e.g.* human-object interaction, action recognition). We also plan to investigate different GCN architectures (*e.g.* [70]) to improve our compositional graph embeddings. Finally, it would be interesting to merge the advances in ZSL for recognizing unseen semantic concepts (*i.e.* new primitives) with CZSL ones in modeling the interactions among primitives, to develop models that can extrapolate compositional capabilities to not only unseen compositions of known concepts, but also to compositions of unseen objects and states.

Acknowledgments

This work has been partially funded by the ERC (853489-DEXIM) and the DFG (2064/1–Project number 390727645).

References

[1] I. Misra, A. Gupta, and M. Hebert, "From red wine to red tomato: Composition with context," in *CVPR*, 2017.
[2] T. Nagarajan and K. Grauman, "Attributes as operators: factorizing unseen attribute-object compositions," in *ECCV*, 2018.
[3] S. Purushwalkam, M. Nickel, A. Gupta, and M. Ranzato, "Task-driven modular networks for zero-shot compositional learning," in *ICCV*, 2019.
[4] Y.-L. Li, Y. Xu, X. Mao, and C. Lu, "Symmetry and group in attribute-object compositions," in *CVPR*, 2020.
[5] M. F. Naeem, Y. Xian, F. Tombari, and Z. Akata, "Learning graph embeddings for compositional zero-shot learning," in *CVPR*. IEEE, 2021.
[6] Y. Fu, X. Wang, H. Dong, Y.-G. Jiang, M. Wang, X. Xue, and L. Sigal, "Vocabulary-informed zero-shot and open-set learning," *IEEE T-PAMI*, 2019.
[7] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly," *IEEE TPAMI*, vol. 41, no. 9, 2018.
[8] P. Isola, J. J. Lim, and E. H. Adelson, "Discovering states and transformations in image collections," in *CVPR*, 2015.
[9] A. Yu and K. Grauman, "Fine-grained visual comparisons with local learning," in *CVPR*, 2014.
[10] M. Mancini, M. F. Naeem, Y. Xian, and Z. Akata, "Open world compositional zero-shot learning," in *CVPR*. IEEE, 2021.
[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[12] D. D. Hoffman and W. A. Richards, "Parts of recognition," *Cognition*, vol. 18, no. 1-3, 1984.
[13] I. Biederman, "Recognition-by-components: a theory of human image understanding." *Psychological review*, vol. 94, no. 2, 1987.
[14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.
[15] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, 1989.
[16] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, 1997.
[17] J. Choi, M. Rastegari, A. Farhadi, and L. S. Davis, "Adding unlabeled samples to categories by learned attributes," in *CVPR*, 2013.
[18] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-scale object classification using label relation graphs," in *ECCV*, 2014.
[19] N. Patricia and B. Caputo, "Learning to learn, from transfer learning to domain adaptation: A unifying perspective," in *CVPR*, 2014.
[20] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *CVPR*, 2017.
[21] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *ICLR*, 2017.
[22] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost," in *ECCV*, 2012.
[23] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *CVPR*, 2015.
[24] B. Dai, Y. Zhang, and D. Lin, "Detecting visual relationships with deep relational networks," in *CVPR*, 2017.
[25] S. Jae Hwang, S. N. Ravi, Z. Tao, H. J. Kim, M. D. Collins, and V. Singh, "Tensorize, factorize and regularize: Robust visual relationship learning," in *CVPR*, 2018.
[26] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *ECCV*, 2016.
[27] C. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," in *TPAMI*, 2013.
[28] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *CVPR*, 2016.
[29] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *NIPS*, 2013.
[30] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for attribute-based classification," in *CVPR*, 2013.
[31] L. Zhang, T. Xiang, and S. Gong, "Learning a deep embedding model for zero-shot learning," in *CVPR*, 2017.

[32] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *CVPR*, 2018.

[33] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," in *CVPR*, 2018.

[34] M. Yang, C. Deng, J. Yan, X. Liu, and D. Tao, "Learning unseen concepts via hierarchical decomposition and composition," in *CVPR*, 2020.

[35] Y. Atzmon, F. Kreuk, U. Shalit, and G. Chechik, "A causal view of compositional zero-shot recognition," in *NeurIPS*, 2020.

[36] X. Wang, F. Yu, T. Darrell, and J. E. Gonzalez, "Task-aware feature generation for zero-shot compositional learning," *arXiv preprint arXiv:1906.04854*, 2019.

[37] X. Wang, Y. Ye, and A. Gupta, "Zero-shot recognition via semantic embeddings and knowledge graphs," in *CVPR*, 2018.

[38] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, and E. P. Xing, "Rethinking knowledge graph propagation for zero-shot learning," in *CVPR*, 2019.

[39] Q. Li, Z. Han, and X.-M. Wu, "Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning," in *AAAI*, 2018.

[40] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," *arXiv preprint arXiv:1806.03536*, 2018.

[41] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *CVPR*, 2019.

[42] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *ICLR*, 2019.

[43] F. Wu, T. Zhang, A. H. d. Souza Jr, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," *arXiv preprint arXiv:1902.07153*, 2019.

[44] J. Klicpera, S. Weißenberger, and S. Günnemann, "Diffusion improves graph learning," in *NeurIPS*, 2019.

[45] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," *arXiv preprint arXiv:1810.05997*, 2018.

[46] Z. W. Ming Chen, B. D. Zengfeng Huang, and Y. Li, "Simple and deep graph convolutional networks," in *ICML*, 2020.

[47] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International journal of lexicography*, vol. 3, no. 4, 1990.

[48] A. Bendale and T. Boult, "Towards open world recognition," in *CVPR*, 2015.

[49] M. Mancini, H. Karaoguz, E. Ricci, P. Jensfelt, and B. Caputo, "Knowledge is never enough: Towards web aided deep open world recognition," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[50] Y. Fu and L. Sigal, "Semi-supervised vocabulary-informed learning," in *CVPR*, 2016.

[51] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[52] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *CVPR*, 2019.

[53] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *CVPR*, 2018.

[54] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "Adacos: Adaptively scaling cosine logits for effectively learning deep face representations," in *CVPR*, 2019.

[55] H. Liu and P. Singh, "Conceptnet—a practical commonsense reasoning tool-kit," *BT technology journal*, vol. 22, no. 4, 2004.

[56] C. Wang, M. Li, and A. J. Smola, "Language models with transformers," *arXiv preprint arXiv:1904.09408*, 2019.

[57] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *CVPR*, 2019.

[58] A. Yu and K. Grauman, "Semantic jitter: Dense supervision for visual comparisons via synthetic images," in *ICCV*, 2017.

[59] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," in *ECCV*, 2016.

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[61] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[62] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *JMLR*, vol. 15, no. 1, 2014.

[64] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, 2017.

[65] Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata, "Semantic projection network for zero-and few-label semantic segmentation," in *CVPR*, 2019.

[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[67] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[68] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, "Deep metric learning to rank," in *CVPR*, 2019.

[69] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen, "Revisiting training strategies and generalization performance in deep metric learning," in *ICML*, 2020.

[70] L. Yang, C. Wang, J. Gu, X. Cao, and B. Niu, "Why do attributes propagate in graph convolutional neural networks?" in *AAAI*, 2021.

Massimiliano Mancini is a post-doctoral researcher at the Cluster of Excellence in Machine Learning of the University of Tübingen, in the Explainable Machine Learning group, lead by Prof. Zeynep Akata. He completed his PhD in Engineering in Computer Science at the Sapienza University of Rome in 2020. During the Ph.D. he has been a member of the ELLIS PhD program, the Technologies of Vision lab at Fondazione Bruno Kessler, and the Visual Learning and Multimodal Applications Laboratory of the Italian Institute of Technology. His research interests include transfer learning across domains and learning from low supervision.

Muhammad Ferjad Naeem is a PhD candidate at the Computer Vision Lab at ETH Zurich supervised by Prof. Luc Van Gool. He completed his Masters at the Technical University of Munich. During his masters, he visited the Explainable Machine Learning group with Prof. Zeynep Akata to work on his master thesis. His Research interests include compositionaility, zero-shot learning and robustness in Machine Learning.

Yongqin Xian is a post-doctoral researcher at ETH Zurich, Switzerland. He received a bachelor degree from Beijing Institute of Technology (China) in 2013, a M.Sc. degree with honors from Saarland University (Germany) in 2016 and PhD degree (summa cum laude) from the Max Planck Institute for Informatics (Germany) in 2020. His research interests include zero-shot and few-shot learning for computer vision tasks.

Zeynep Akata is a professor of Computer Science at the Cluster of Excellence Machine Learning in the University of Tübingen. After her PhD in INRIA Rhone Alpes (F) in 2014, she worked as a post-doctoral researcher at the Max Planck Institute for Informatics (DE) between 2014-2017, at UC Berkeley (USA) between 2016-2017 and as an assistant professor at the University of Amsterdam (NL) between 2017-2019. She received a Lise-Meitner Award for Excellent Women in Computer Science in 2014 and an ERC Starting Grant in 2019. Her research interests include multimodal learning in low-data regimes and explainable machine learning focusing on vision and language.