Open World Compositional Zero-Shot Learning

Massimiliano Mancini[1] *, Muhammad Ferjad Naeem[1,2] *, Yongqin Xian[3], Zeynep Akata[1,3,4]
[1]University of Tübingen [2]TU München [3]MPI for Informatics [4]MPI for Intelligent Systems

Abstract

Compositional Zero-Shot learning (CZSL) requires to recognize state-object compositions unseen during training. In this work, instead of assuming prior knowledge about the unseen compositions, we operate in the open world setting, where the search space includes a large number of unseen compositions some of which might be unfeasible. In this setting, we start from the cosine similarity between visual features and compositional embeddings. After estimating the feasibility score of each composition, we use these scores to either directly mask the output space or as a margin for the cosine similarity between visual features and compositional embeddings during training. Our experiments on two standard CZSL benchmarks show that all the methods suffer severe performance degradation when applied in the open world setting. While our simple CZSL model achieves state-of-the-art performances in the closed world scenario, our feasibility scores boost the performance of our approach in the open world setting, clearly outperforming the previous state of the art. Code is available at: https://github.com/ExplainableML/czsl.

1. Introduction

The appearance of an object in the visual world is determined by its state. A *pureed tomato* looks different from a *wet tomato* despite the shared object, and a *wet tomato* looks different from a *wet dog* despite the shared state. In Compositional Zero-Shot Learning (CZSL) [20, 21, 27, 17] the goal is to learn a set of states and objects while generalizing to unseen compositions.

Current benchmarks in CZSL study this problem in a closed space, assuming the knowledge of unseen compositions that might arise at test time. For example, the widely adopted MIT states dataset [13] contains 28175 possible compositions (in total 115 states and 245 objects), but the test time search space is limited to 1662 compositions (1262 seen and 400 unseen), covering less than 6% of the whole compositional space. This restriction on the output space is a fundamental limitation of current CZSL methods.

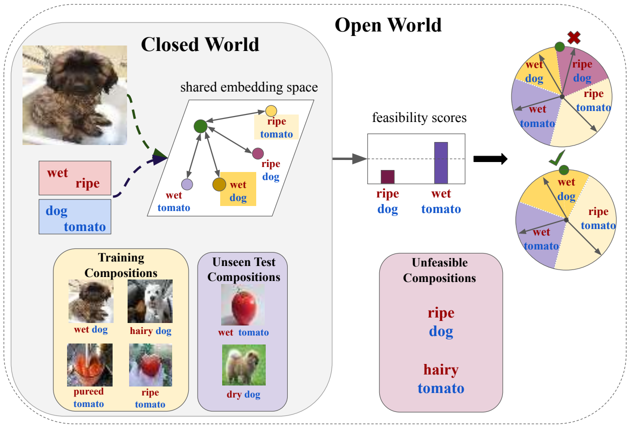

Figure 1. In closed world CZSL, the search space is *assumed to be known a priori*, *i.e.* seen (yellow) and unseen (purple) compositions are available during training/test. In our open world scenario, *no limit* on the search space is imposed. Hence, the model has to figure out implausible compositions (pink) and discard them.

In this work, we propose the more realistic Open World CZSL (OW-CZSL) task (see Figure 1), where we impose *no constraint* on the test time search space, causing the current state-of-the-art approaches to suffer severe performance degradation. To tackle this new task, we propose *Compositional Cosine Logits* (CompCos), a model where we embed both images and compositional representations into a shared embedding space and compute scores for each composition with the cosine similarity. Moreover, we treat less feasible compositions (*e.g. ripe dog*) as *distractors* that a model needs to eliminate. For this purpose, we use similarities among primitives to assign a feasibility score to each unseen composition. We then use these scores as margins in a 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. Despite its simplicity, our model surpasses the previous state-of-the-art methods on the standard CZSL benchmark as well as in the challenging OW-CZSL task.

Our contributions are as follows: (1) A novel problem formulation, Open World Compositional Zero-Shot learning (OW-CZSL) with the most flexible search space in terms of seen/unseen compositions; (2) CompCos, a novel model to solve the OW-CZSL task based on cosine logits and a

[1] First and second author contributed equally.

projection of learned primitive embeddings with an integrated feasibility estimation mechanism; (3) A significantly improved state-of-the-art performance on MIT states [13] and UT Zappos [34, 35] both on the existing benchmarks and the newly proposed OW-CZSL setting.

2. Related works

Compositional Zero-Shot Learning. Early vision works encode compositionality with hierarchical part-based models, to learn robust and scalable object representations [24, 23, 6, 38, 25, 28]. More recently, compositionality has been widely considered in multiple tasks such as compositional reasoning for visual question answering [14, 16, 12] and modular image generation [37, 30, 26].

In this work, we focus on Compositional Zero-Shot Learning (CZSL) [20]. Given a training set containing a set of state-object compositions, the goal of CZSL is to recognize unseen compositions of these states and objects at test time. Some approaches address this task by learning objects and states classifier in isolation and composing them to build the final recognition model. In this context, [4] trains an SVM classifier for seen compositions and infers class weights for new compositions through a Bayesian framework. LabelEmbed [20] learns a transformation network on top of pretrained state and object classifiers. [21] proposes to encode objects as vectors and states as linear operators that change this vector. Similarly, [17] enforces symmetries in the representation of objects given their state transformations. Recently, [27] proposed a modular network where states and objects are simultaneously encoded. The network blocks are then selectively activated by a gating function, taking as input an object-state composition.

All of these works assume that the training and test-time compositions are known a priori. We show that removing this assumption causes severe performance degradation. Furthermore, we propose the first approach for Open World CZSL. There are some similarities between our model and [20] (*i.e.* primitive representations are concatenated and projected in a shared visual-semantic space). However, our loss formulation leads to significant improvements in the Open World CZSL results. More importantly, our approach is the first to estimate the feasibility of each composition and exploits this information to isolate/remove possible distractors in the shared output space.

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) [2] where the goal is to dynamically update a model trained on a subset of classes to recognize increasingly more concepts as new data arrives. Our definition of *open world* is related to the *open set* zero-shot learning (ZSL) [33] scenario in [7, 8], proposing to expand the output space to include a very large vocabulary of semantic concepts.

Our problem formulation and approach are close in spirit to that of [8] since both works consider the lack of constraints in the output space for unseen concepts as a requirement for practical (compositional) ZSL methods. However, there are fundamental differences between our work and [8]. Since we consider the problem of CZSL, we have access to images of all primitives during training but not all their possible compositions. This implies that we can use the knowledge obtained from the visual world to model the feasibility of compositions and modifying the representations in the shared visual-compositional embedding space. We explicitly model the feasibility of each unseen composition, incorporating this knowledge into training and test.

3. Compositional Cosine Logits

3.1. (OW)-CZSL Task Definition

Compositional zero-shot learning (CZSL) aims to predict a composition of multiple semantic concepts in images. Let us denote with $\mathcal{S}$ the set of possible states, with $\mathcal{O}$ the set of possible objects, and with $\mathcal{C} = \mathcal{S} \times \mathcal{O}$ 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 $\mathcal{C}^s \subset \mathcal{C}$. $\mathcal{T}$ 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 that are 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 [20], 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. Similarly to the standard generalized zero-shot learning [33], this scenario is more challenging due to the natural prediction bias of the model in $\mathcal{C}^s$, seen during training. Most recents works on CZSL consider the generalized scenario [27, 17], and the set of unseen compositions in $\mathcal{C}^t$ is assumed to be known a priori, with $\mathcal{C}^t \subset \mathcal{C}$.

In this work, we take a step further, analyzing the case where the output space of the model 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 generalized case 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 *distractor* 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

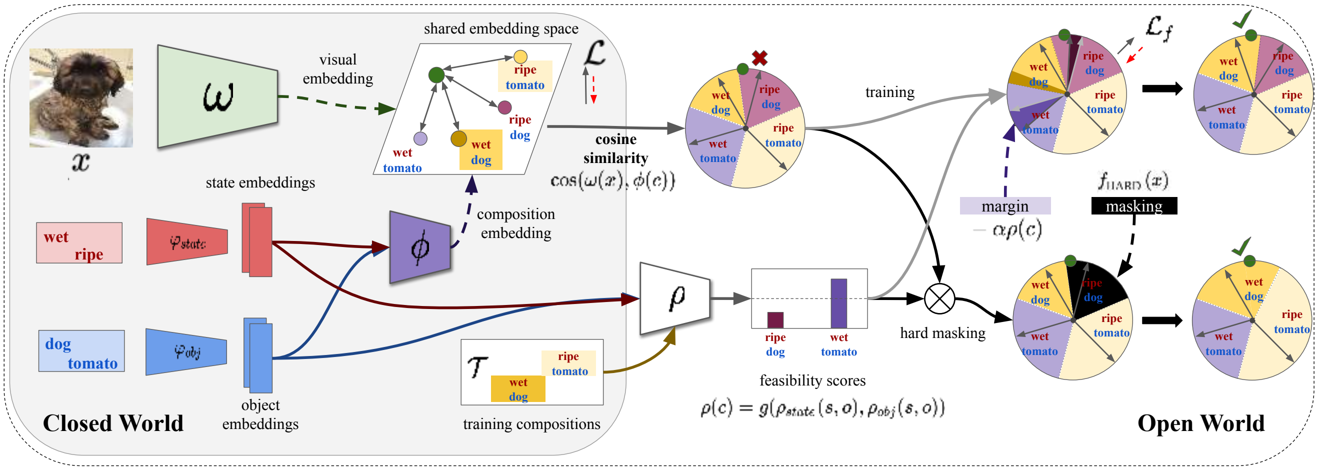

Figure 2. **Compositional Cosine Logits (CompCos).** Our approach embeds an image (top) and state-object compositions (bottom) into a shared semantic space defined by the cosine-similarity between image features and composition embeddings. In the open world model, we estimate a feasibility score for each of the unseen compositions, using the relation between states, objects, and the training compositions. The feasibility scores are injected to the model either by removing less feasible compositions (*e.g. ripe dog*) from the output space (bottom, black slice) or by adding a bias to the cosine similarities computed during training (top, purple slices).

Open Set Zero-shot Learning [8], we do not only consider objects but also states. Therefore, this knowledge can be exploited to identify unfeasible distractor compositions (*e.g. rusty pie*) and isolate them. Figure 2 shows an overview of our approach for both closed and open world scenarios.

3.2. CompCos: A Closed World Model

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. Inspired by incremental learning [11] and generalized few-shot learning [9], we reduce this problem by replacing the logits of the classification layer with cosine similarities between the image features and the composition embeddings in the shared embedding space:

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

where $\omega : \mathcal{X} \to \mathcal{Z}$ is the mapping from the image space to the shared embedding space $\mathcal{Z} \in \mathbb{R}^d$ and $\phi : \mathcal{C} \to \mathcal{Z}$ embeds a composition to the same space. $cos(y,z) = \frac{y^\intercal z}{||y|| \, ||z||}$ is the cosine similarity among the two embeddings. As sketched Figure 2 (left), the visual embedding ω (green block) maps an image to the shared embedding space, while states (red) and objects (blue) embeddings, are embedded to the shared space by ϕ (purple block).

Visual embedding. We use a standard deep neural network, e.g. ResNet-18 [10], with an additional embedding function ω mapping, the feature extracted from the backbone to $\mathcal{Z}$. The embedding function is a simple 2-layer MLP, where dropout [29], LayerNorm [1] and a ReLU non-linearity [22] are applied after the first layer. During training, we freeze the main backbone, updating only the final MLP.

Composition embedding. The function $\varphi : \mathcal{S} \cup \mathcal{O} \to \mathbb{R}^d$ maps the primitives, *i.e.* objects and states, into their corresponding embedding vectors. The embedding of a given composition $c = (s, o)$ is a simple linear projection of the embeddings of its primitives:

$$\phi(c) = [\varphi(s) \; \varphi(o)]^\top W$$

with $W \in \mathbb{R}^{2d \times d}$, where we consider $\mathcal{Z} \in \mathbb{R}^d$ for simplicity. We chose a linear embedding function since we found it works well in practice. Moreover, it applies a strong constraint to the compositional space, making the embedding less prone to overfitting and more suitable for generalizing in a scenario where we might have $|\mathcal{C}^t| \gg |\mathcal{C}^s|$. During training, we update both the embedding matrix W and the atomic embeddings of φ, after initializing the latter with word embeddings.

Objective function. We define a cross-entropy loss on top of the cosine logits:

$$\mathcal{L} = -\frac{1}{|\mathcal{T}|} \sum_{(x,c) \in \mathcal{T}} \log \frac{e^{\frac{1}{T} \cdot p(x,c)}}{\sum_{y \in \mathcal{C}^s} e^{\frac{1}{T} \cdot p(x,y)}} \quad (2)$$

where T is a temperature value that balances the probabilities for the cross-entropy loss [36] and $p(x, c) = \cos(\phi(x), \omega(c))$. In the following we discuss how to extend our Compositional Cosine Logits (CompCos) model from the closed to the more challenging open world scenario.

3.3. CompCos: from Closed to Open World

Although CompCos is an effective CZSL algorithm in the standard closed world scenario, performing well on the OW-CZSL requires tackling different challenges, such as avoiding distractors. We consider distractors as less-likely concepts, e.g. *pureed dog*, *hairy tomato*, and argue that the similarity among objects and states can be used as a proxy to estimate the feasibility of each composition. The least feasible compositions can then be successfully isolated, improving the representations of the feasible ones.

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 for *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, given a composition $c = (s, o)$ we define its feasibility score with respect to the object o as:

$$\rho_{obj}(s, o) = \max_{\hat{o} \in \mathcal{O}^s} cos(\varphi(o), \varphi(\hat{o})) \tag{3}$$

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 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:

$$\rho_{state}(s, o) = \max_{\hat{s} \in \mathcal{S}^o} cos(\varphi(s), \varphi(\hat{s})) \tag{4}$$

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:

$$\rho(c) = \rho(s, o) = g(\rho_{state}(s, o), \rho_{obj}(s, o)) \tag{5}$$

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 [18], language models [31]) can be complementary resources.

Exploiting Compositional Feasibility. A first simple strategy is applying a threshold on the feasibility scores, considering all compositions above the threshold as valid and others as distractors (e.g. *ripe dog*, as shown in the black pie slice of Figure 2):

$$f_{\text{HARD}}(x) = \underset{c \in \mathcal{C}^t, \rho(c) > \tau}{\arg\max} \; cos(\omega(x), \phi(c)) \tag{6}$$

where τ is the threshold, tuned on a validation set. While this strategy is effective, it 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. To sidestep this issue, we propose to inject the feasibility scores directly into the training procedure. We argue that doing so can enforce separation between most and least feasible unseen compositions in the shared embedding space.

To inject the feasibility scores $\rho(c)$ directly within our objective function, we can define:

$$\mathcal{L}_{\text{f}} = -\frac{1}{|\mathcal{T}|} \sum_{(x,c) \in \mathcal{T}} \log \frac{e^{\frac{1}{T} \cdot p^f(x,c)}}{\sum_{y \in \mathcal{C}} e^{\frac{1}{T} \cdot p^f(x,y)}} \tag{7}$$

with:

$$p^f(x, c) = \begin{cases} \cos(\omega(x), \phi(c)) & \text{if } c \in \mathcal{C}^s \\ \cos(\omega(x), \phi(c)) - \alpha\rho(c) & \text{otherwise} \end{cases} \tag{8}$$

where $\rho(c)$ are used as margins for the cosine similarities, and $\alpha > 0$ is a scalar factor. With Eq. (7) 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. This is because most feasible compositions should be closer to the seen ones (to which the visual embedding network is biased) than less feasible ones. By doing that, we force the network to push the representation of less feasible compositions away from the representation of compositions in $\mathcal{C}^s$ in $\mathcal{Z}$. On the other hand, the less penalized feasible compositions benefit from the updates produced for seen training compositions containing the same primitives. More feasible compositions will then be more likely to be predicted by the model, even without being present in the training set. As an example (Figure 2, 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 score $p^f(c)$ 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 us to gradually introduce the feasibility margins within our objective while exploiting improved primitive embeddings to compute them.

4. Experiments

Datasets. We experiment with two standard CZSL benchmark datasets. MIT states [13] contains 53K images of 245 objects in 115 possible states. We adopt the standard split from [27]. For the closed world experiments, the output space contains 1262 seen and 300/400 unseen (validation/test) compositions. For the open world scenario, we consider all possible 28175 compositions as present in the search space. Note, that 26114 out of 28175 ($\sim$93%) are not present in any splits of the dataset but are included in our open world setting.

UT Zappos [34, 35] contains 12 different shoe types (objects) and 16 different materials (states). In the closed world setting, the output space is constrained to the 83 seen and to additional 15 and 18 unseen compositions for validation and test respectively. Although 76 out of 192 possible compositions ($\sim$40%) are not in any of the splits of the dataset, we consider them in our open world setting.

Metrics. For the primitives, we report the object and state classification accuracies. Since we focus on the generalized scenario and the model has an inherent bias for seen compositions, we follow the evaluation protocol from [27]. We consider the performance of the model with different bias factors for the unseen compositions, reporting the results as best accuracy on only images of seen compositions (*best seen*), best accuracy on only unseen compositions (*best unseen*), best harmonic mean (*best HM*) and Area Under the Curve (AUC) for seen and unseen accuracies at bias values.

Benchmark and Implementation Details. As in [27, 17] our image features are extracted from a ResNet18 pretrained on ImageNet [5] and we learn the visual embedding module ω on top of these features. We initialize the embedding function φ with 300-dimensional word2vec [19] embeddings for UT Zappos and with 600-dimensional word2vec+fastext [3] embeddings for MIT states, following [32], keeping the same dimensions for the shared embedding space $\mathcal{Z}$. We train both ω (φ and W) and ϕ using Adam [15] optimizer with a learning rate and a weight decay set to $5 \cdot 10^{-5}$. The margin factor α and the temperature T are set to 0.4 and 0.05 respectively for MIT states and 1.0 and 0.02 for UT Zappos. We linearly increase α from 0 to the previous values during training, reaching the values after 15 epochs. We consider the mixing function g as the average to merge state and object feasibility scores and f_{HARD} as predictor for OW-CZSL, unless otherwise stated.

We compare with four state-of-the-art methods, Attribute as Operators (AOP) [21], considering objects as vectors and states as matrices modifying them [21]; LabelEmbed+ (LE+) [20, 21] training a classifier merging state and object embeddings with an MLP; Task-Modular Neural Networks (TMN) [27], modifying the classifier through a gating function receiving as input the queried stat-object composition; and SymNet [17], learning object embeddings showing symmetry under different state-based transformations. 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.

4.1. Comparing with the State of the Art

We compare CompCos and the state of the art in the standard closed world and the proposed open world setting.

Closed World Results. The results analyzing the performance of CompCos in the standard closed world scenario are reported on the left side of Table 1 for the MIT states and UT Zappos test sets. Although being a substantially simple approach, our CompCos achieves remarkable results. On MIT states our model either outperforms or is comparable to all competitors in all metrics. In particular, while obtaining a comparable best harmonic mean with SymNet, it achieves 4.5 AUC, which is a significant improvement over 3.0 from SymNet. This highlights how our model is more robust to the bias on unseen test compositions. Compared to the closest method to ours, LabelEmbed+ (LE+), CompCos shows clear advantages for all metrics (from 2 to 4.5 AUC, and from 10.7% to 16.4% on the best harmonic mean), underlying the impact of our embedding functions and the cross-entropy loss on cosine logits.

On UT Zappos, our model is superior to almost all methods (except TMN in two cases). It is particularly interesting how CompCos surpasses AoP, LE+, and SymNet with more than 2% on best harmonic mean and at least by 2.8 in AUC. In comparison with TMN, while achieving a lower best harmonic mean (-1.9%) and a slightly lower AUC (-0.6), it achieves the best unseen accuracy (+2.5%) and improves of 4% the accuracy in recognizing each primitive in isolation. These results show that CompCos is less sensitive to the value of the bias applied to the unseen compositions in the generalized scenario, thanks to the use of cosine similarity as prediction score. We would like to highlight that our model uses a magnitude lower trainable parameters, e.g. 0.8M vs 2.3M for TMN, to achieve these results.

Open World Results. The results on the challenging OW-CZSL setting are reported on the right side of Table 1. As expected, the first clear outcome is the severe decrease in performance of every method. In fact, the OW-CZSL performances (*e.g.* best unseen, best HM, and AUC) are less than half of CZSL performances in MIT states. The largest decrease in performance is on the best unseen metric, due to the presence of a large number of distractors. As an example, LE+ goes from 20.1% to 2.5% of best unseen accuracy and even the previous state of the art, SymNet, loses 18.2%, confirming that the open world scenario is significantly more challenging than the closed world setting.

In the OW-CZSL setting, our model (CompCos) is more robust to the distractors, due to the injected feasibility-based margins which shape the shared embedding space during

Method	**Closed World**												**Open World**											
	MIT states						**UT Zappos**						**MIT states**						**UT Zappos**					
	Sta.	Obj.	S	U	HM	auc	Sta.	Obj.	S	U	HM	auc	Sta.	Obj.	S	U	HM	auc	Sta.	Obj.	S	U	HM	auc
AoP[21]	21.1	23.6	14.3	17.4	9.9	1.6	38.9	69.9	59.8	54.2	40.8	25.9	15.4	20.0	16.6	5.7	4.7	0.7	25.7	61.3	50.9	34.2	29.4	13.7
LE+[20]	23.5	26.3	15.0	20.1	10.7	2.0	41.2	69.3	53.0	61.9	41.0	25.7	10.9	21.5	14.2	2.5	2.7	0.3	**38.1**	68.2	**60.4**	36.5	30.5	16.3
TMN[27]	23.3	26.5	20.2	20.1	13.0	2.9	40.8	69.5	58.7	60.0	**45.0**	**29.3**	6.1	15.9	12.6	0.9	1.2	0.1	14.6	61.5	55.9	18.1	21.7	8.4
SymNet[17]	26.3	28.3	24.2	**25.2**	16.1	3.0	41.3	68.6	49.8	57.4	40.4	23.4	17.0	26.3	21.4	7.0	5.8	0.8	33.2	70.0	53.3	44.6	34.5	18.5
CompCos	**27.9**	**31.8**	**25.3**	24.6	**16.4**	**4.5**	**44.7**	**73.5**	**59.8**	**62.5**	43.1	28.7	**18.8**	**27.7**	**25.4**	**10.0**	**8.9**	**1.6**	35.1	**72.4**	59.3	**46.8**	**36.9**	**21.3**

Table 1. Closed and Open World CZSL results on MIT states and UT Zappos. We measure states (Sta.) and objects (Obj.) accuracy on the primitives, best seen (S) and unseen accuracy (U), best harmonic mean (HM), and area under the curve (auc) on the compositions.

Effect of the Margins		Seen	Unseen	HM	AUC
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
Effect of Primitives		Seen	Unseen	HM	AUC
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

Table 2. Results on MIT states validation set for different ways of applying the margins (top) and different ways of computing the feasibility scores (bottom) for CompCos with f as predictor.

training. This is clear in MIT states, where CompCos outperforms the state of the art for all metrics. Remarkably, it obtains double the AUC of the best competitor, SymNet, going from 0.8 to 1.6 with a 3.1% improvement on the best HM and 3.0% on best unseen accuracy.

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, CompCos improves by 2.8 in AUC and 2.4 in best unseen accuracy over SymNet, showing the highest results according to all compositional metrics but the seen accuracy, where it performs comparably to LE+. This is expected, since the output space of all previous works is limited to seen classes during training, thus the models are discriminative for seen compositions.

4.2. Ablation studies

We investigate the impact of the feasibility-based margins, how we obtain them (without f_{HARD}), and the benefits of limiting the output space during inference using f_{HARD}. We perform our analyses on MIT states' validation set. Note that, in the tables, CompCos$^{\text{CW}}$ is the closed world CompCos model, as described in Sec. 3.2.

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 2 (Top) shows, 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 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 Seen and moderate improvements on both Unseen and HM.

Finally, linearly increasing α (*i.e.* warmup α) further improves the harmonic mean due to both the i) improved margins that CompCos estimates 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. (3), states as in Eq. (4)) or both as in Eq. (5) to estimate the feasibility score for each unseen composition. Here we consider all these choices, showing their impact on the results in Table 2 (Bottom), with f as predictor.

The results show that computing feasibility on the primitives 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

	Mask	Seen	Unseen	HM	AUC
LE+		14.8	3.1	3.2	0.3
	✓		5.0	4.6	0.5
TMN		15.9	1.3	1.7	0.1
	✓		4.1	4.1	0.4
SymNet		23.6	7.9	7.6	1.2
	✓		7.9	7.7	1.2
CompCos$^{\text{CW}}$		28.0	6.0	7.0	1.2
	✓		8.1	8.7	1.6
CompCos		27.1	11.0	10.8	2.1
	✓		11.2	11.0	2.2

Table 3. Results on MIT states validation set for applying our feasibility-based binary masks (f_{HARD}) on different models.

particular, computing the scores starting from state information (ρ_{state}) brings good best unseen and HM results while under-performing on the best seen accuracy. On the other hand, using similarities among objects (ρ_{obj}) performs well on the seen classes while achieving slightly lower performances on unseen ones and HM.

Nevertheless 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. We ascribe this behavior to the fact that, with the average, 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. (7) 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 Masking. We consider mainly two ways of using the feasibility scores: during training as margins and/or during inference as masks on the predictions. We analyze the impact of applying the mask during inference, *i.e.* using as prediction function Eq. (6) in place of Eq. (1), with the threshold τ computed empirically. We run this study on our closed world model CompCos$^{\text{CW}}$, our full model CompCos and three CZSL baselines, LE+, TMN, and SymNet. Note that, since seen compositions are not masked, best seen performances are shared across a single model.

As shown in Table 3, if we apply binary masks on top of CompCos$^{\text{CW}}$ the AUC increases by 0.5, best harmonic mean by 1.7%, and best unseen accuracy by 2.1%. This is because our masks filter out the less feasible compositions, rather than just restricting the output space. At the same time, the improvements are not as pronounced for the full CompCos model. Indeed, including the feasibility scores as margins during training makes the model already robust to the distractors. The hard masking still provides a slight benefit over all compositional metrics, with an 11% on accuracy on unseen compositions. This confirms the importance of restricting the search space under a criterion taking into account the probability of a composition of being a distractor, such as our feasibility scores.

An interesting observation is that applying our feasibility-based binary masks on top of other approaches (*i.e.* LE+, TMN, and SymNet) is beneficial. SymNet, modeling states and objects separately, sees a minor increase in performance in HM while maintaining the AUC and unseen accuracy. However, LE+ and TMN, which learn a joint compatibility between compositions and images, see a big increase in performance with the introduction of feasibility. TMN improves from 0.1 to 0.4 in the AUC while seeing a big increase in the HM from 1.7 to 4.1. Similarly, LE+ improves from 0.3 AUC to 0.5 with a significant increase in the HM, from 3.2 to 4.6.

4.3. Qualitative results

We show example composition predictions for a set of images of our CompCos$^{\text{CW}}$and our CompCos in the open world setting. Furthermore, we show examples of most- and least-feasible compositions determined by our model.

Compositions Corrected due to Feasibility Scores. We qualitatively analyze the reasons for the improvements of CompCos over CompCos$^{\text{CW}}$, by looking at the example predictions of both models on simple sample images of MIT states. We compare predictions on samples where the closed world model is "distracted" by a distractor while the open world model is able to predict the correct class label.

As shown in Figure 3, 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. broken dog*, *molten chicken*, *unripe boulder*, *mossy bear*, *rusty pie*) but the associated state is not only wrong but also making the compositions unfeasible. In other cases, the state prediction is almost correct (*e.g. cracked fan*,*barren wave*, *inflated eggs*) but the associated object is unrelated, making the composition unfeasible. All these problems are less severe in our full CompCos model since our feasibility-driven objective helps in isolating the implausible distractors in the shared embedding space, reducing the possibility to predict them.

Discovered Most and Least Feasible Compositions. The most crucial advantage of our method is its ability to estimate the feasibility of each unseen composition, to later inject these estimates into the learning process. Our assumption is that our procedure described in Section 3.3 is robust

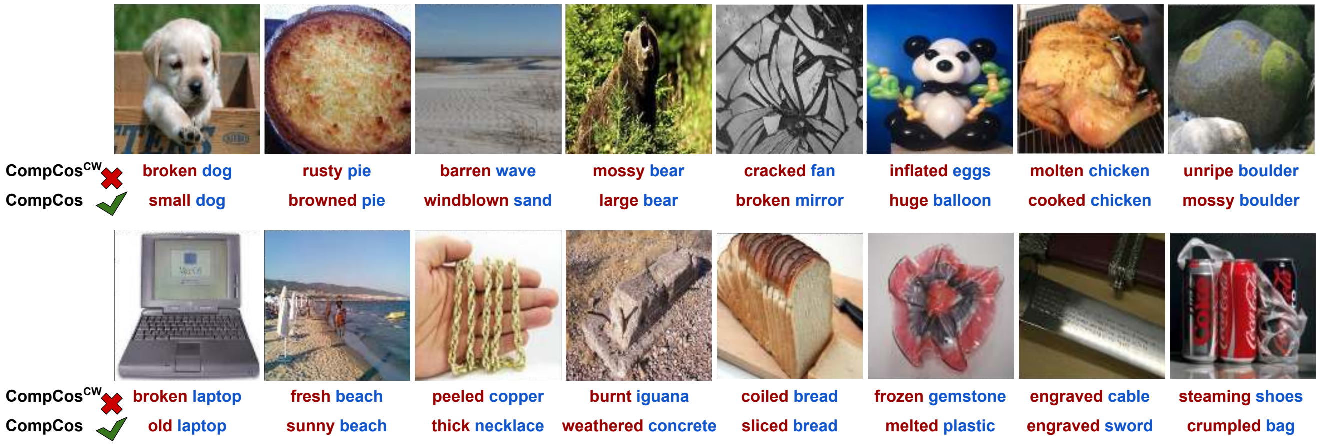

Figure 3. Examples correct predictions of CompCos in the OW-CZSL scenario when the CompCos^{CW} fails. The first row shows the predictions of the closed world model, the bottom row shows the results of CompCos. The images are randomly selected.

	Compositions	
	Most Feasible (Top-1)	Least Feasible (Bottom-1)
	browned tomato	short lead
	caramelized potato	cloudy gemstone
	thawed meat	standing vegetable
	small dog	full nut
	large animal	blunt milk
Objects	**States**	
	Most Feasible (Top-3)	Least Feasible (Bottom-3)
tomato	browned, peeled, diced	tight, full, standing
dog	small, old, young	fallen, toppled, standing
cat	wrinkled, huge, large	viscous, smooth, runny
laptop	small, shattered, modern	cloudy, sunny, dull
camera	tiny, huge, broken	diced, caramelized, cloudy

Table 4. Unseen compositions wrt their feasibility scores (Top: Top-5 compositions on the left, Least-5 on the right; Bottom: Top-3 highest and Bottom-3 lowest feasible state per object.

enough 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 would like to highlight that here we are focusing mainly on visual information to extract the relationships. This information can in principle be coupled with knowledge bases (*i.e.* [18]) and language models (*i.e.* [31]) to further refine the scores.

Table 4 (Top) shows qualitative examples of the most and least feasible compositions discovered by CompCos. As an example, it correctly ranks *browned tomato* and *small dog* as one of the most feasible compositions, while *full nut* and *blunt milk* among the least feasible ones. Since the dataset has a lot of food classes, we see that the top and bottom are mostly populated by them. However, the presence of relevant and irrelevant states with these objects is promising and shows the potential of our feasibility estimation strategy.

Table 4 (Bottom) shows the top-3 most feasible compositions and bottom-3 least feasible compositions given five randomly selected objects. 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. diced camera*) as well as atmospheric conditions (*e.g. sunny laptop*). Similarly, states usually associated with substances are considered unfeasible for animals (*e.g. runny cat*). At the same time, size and ages are mostly linked with animals (*e.g. young dog*) while cooking states are correctly associated with food (*e.g. diced tomato*). Interestingly, the top states for *cat* are all present with *dog* as seen compositions, thus the similarities between the two classes has been used to transfer these states from *dog* to *cat*, following Eq. (3).

5. Conclusions

In this work, we propose a new benchmark for CZSL that extends the problem from the closed world to an open world where all the combinations of states and objects could potentially exist. We show that state-of-the-art methods fall short in this setting as the number of unseen compositions significantly increases. We argue that not all combinations are valid classes but it is unrealistic to assume that test set pairs are the only valid compositions. 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, CompCos, that incorporates this feasibility and achieves state-of-the-art performance in both closed and open world on two real-world datasets.

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

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[2] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR*, 2015. 2

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *ACL*, 2017. 5

[4] Chao-Yeh Chen and Kristen Grauman. Inferring analogous attributes. In *CVPR*, 2014. 2

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[6] Sanja Fidler and Ales Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, 2007. 2

[7] Yanwei Fu and Leonid Sigal. Semi-supervised vocabulary-informed learning. In *CVPR*, 2016. 2

[8] Yanwei Fu, Xiaomei Wang, Hanze Dong, Yu-Gang Jiang, Meng Wang, Xiangyang Xue, and Leonid Sigal. Vocabulary-informed zero-shot and open-set learning. *IEEE T-PAMI*, 2019. 2, 3

[9] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[11] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019. 3

[12] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. In *ICLR*, 2018. 2

[13] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015. 1, 2, 5

[14] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 2

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 5

[16] Jayanth Koushik, Hiroaki Hayashi, and Devendra Singh Sachan. Compositional reasoning for visual question answering. In *ICML*, 2017. 2

[17] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *CVPR*, 2020. 1, 2, 5, 6

[18] Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004. 4, 8

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013. 5

[20] Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *CVPR*, 2017. 1, 2, 5, 6

[21] Tushar Nagarajan and Kristen Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *ECCV*, 2018. 1, 2, 5, 6

[22] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 3

[23] Bjorn Ommer and Joachim Buhmann. Learning the compositional nature of visual object categories for recognition. *IEEE T-PAMI*, 32(3):501–516, 2009. 2

[24] Bjorn Ommer and Joachim M Buhmann. Learning the compositional nature of visual objects. In *CVPR*, 2007. 2

[25] Patrick Ott and Mark Everingham. Shared parts for deformable part-based models. In *CVPR*, 2011. 2

[26] Dim P Papadopoulos, Youssef Tamaazousti, Ferda Ofli, Ingmar Weber, and Antonio Torralba. How to make a pizza: Learning a compositional layer-based gan model. In *CVPR*, 2019. 2

[27] Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. In *ICCV*, 2019. 1, 2, 5, 6

[28] Zhangzhang Si and Song-Chun Zhu. Learning and-or templates for object recognition and detection. *IEEE T-PAMI*, 35(9):2189–2205, 2013. 2

[29] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014. 3

[30] Fuwen Tan, Song Feng, and Vicente Ordonez. Text2scene: Generating compositional scenes from textual descriptions. In *CVPR*, 2019. 2

[31] Chenguang Wang, Mu Li, and Alexander J Smola. Language models with transformers. *arXiv preprint arXiv:1904.09408*, 2019. 4, 8

[32] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *CVPR*, 2019. 5

[33] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE T-PAMI*, 41(9):2251–2265, 2018. 2

[34] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014. 2, 5

[35] Aron Yu and Kristen Grauman. Semantic jitter: Dense supervision for visual comparisons via synthetic images. In *CVPR*, 2017. 2, 5

[36] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *CVPR*, 2019. 3

[37] Bo Zhao, Bo Chang, Zequn Jie, and Leonid Sigal. Modular generative adversarial networks. In *ECCV*, pages 150–165, 2018. 2

[38] Long Leo Zhu, Yuanhao Chen, Antonio Torralba, William Freeman, and Alan Yuille. Part and appearance sharing: Recursive compositional models for multi-view multi-object detection. In *CVPR*, 2010. 2