Revisiting Visual Product for Compositional Zero-Shot Learning

Shyamgopal Karthik
University of Tübingen

Massimiliano Mancini
University of Tübingen

Zeynep Akata
University of Tübingen
Max Planck Institute for Intelligent Systems

Abstract

Compositional Zero-Shot Learning (CZSL) aims to recognize compositions of objects and states in images, generalizing to unseen compositions of objects and states at test time. Recent works tackled this problem effectively by using side information (e.g., word embeddings) together with either consistency constraints or specific network designs modeling the relationships between objects, states, compositions, and visual features. In this work, we take a step back, and we revisit the simplest baseline for this task, i.e., Visual Product (VisProd). VisProd considers CZSL as a multi-task problem, predicting objects and states separately. Despite its appealing simplicity, this baseline showed low performance in early CZSL studies. Here we identify the two main reasons behind such unimpressive initial results: network capacity and bias on the seen classes. We show that simple modifications to the object and state predictors allow the model to achieve either comparable or superior results w.r.t. the recent state of the art in both the open-world and closed-world CZSL settings on three different benchmarks.

1 Introduction

Compositional Zero-Shot Learning (CZSL) is the task of predicting the object (e.g., tomato, dog, car) and the state (e.g., wet, dry, pureed) present in a given input image. The main challenge of CZSL stems from the inherent distribution-shift that occurs between training and test data. In fact, the training set contains only a subset of the existing state-object compositions and unseen compositions of the same objects and states appear at test time. This means that the appearance of an object may drastically change from training to test based on the associated states (e.g., wet vs pureed tomato) as well as the effect of a state may change depending on the associated objects (e.g., wet dog vs car).

Recently, multiple works tackled this problem from different perspectives, usually exploiting side information to initialize state and object representations [9, 14]. Examples are modeling attributes as operators modifying object representations [12, 6], predicting embeddings for each composition through simple multi-layer perceptrons [10, 8] or graph convolutions [11, 5], or modeling the dependency between object and state representations from a causal perspective [1].

One may wonder what happens if we get rid of specific loss functions or architectural designs and predict objects and states independently. This simple baseline would both reduce training complexity and eliminate the need for side information. Unfortunately, this method named Visual Product (VisProd) performed worse than early CZSL methods [10, 12] and hence was neglected in recent works.
Despite these results, we revisit VisProd, and we find that its performance is comparable to the recent state-of-the-art methods [8, 11] and far superior to the performance of the initial works in this field [10, 12, 15, 6] on various CZSL settings and benchmarks. Investigating this further, we find two significant reasons for this discrepancy. The first is that early VisProd models lacked enough architecture capacity to learn the objects and state predictors, something crucial for its effectiveness. The second reason pertains to the evaluation of CZSL models. The initial works directly measured the performance on both seen and unseen pairs without taking into account the bias towards seen compositions, as highlighted in [15]. If we take this bias into account and we adjust VisProd scores, its accuracy on unseen compositions is on par with the state of the art. We believe this work provides evidence to the community for re-considering VisProd as a competitive baseline for CZSL.

2 Related Work

Compositional zero-shot learning aims to recognize novel compositions of seen primitives. More specifically, we aim to recognize compositions of states and objects in images, even unseen during training. The main challenge of this setting is modeling how states modify objects, generalizing this capability to unseen compositions. Previous works have mainly focused on modeling the interactions between states and objects. LabelEmbed [10] uses an MLP to predict the parameters of the classifier of a state-object composition given the two classifiers (or embeddings) of its constituent primitives. Task-modular networks [15], follows a similar idea, using a shared compositional classifier whose connections are gated, depending on the input primitives. Other works proposed to treat states as operators modifying object embeddings. In this direction, [12] models states as matrices and objects as vectors, learning state matrices that preserve compositional principles (e.g. commutativity, inversion). Similarly, [6] predicts uses networks to add or remove states from objects/image embeddings, imposing properties such as symmetry and invertibility to the state operators.

Differently from these works, [1] tackles CZSL from a causality perspective, learning disentangled objects and states representations. [8] proposes the OW-CZSL setting, where there are no prior on the unseen compositions available at test time, and the full compositional space is considered as output space of the model, tackling these problems by modeling the feasibility of each composition. Finally, more recent approaches exploit graph convolutional networks [5] to model the interactions between state, objects and their compositions [11]. In the graph, state and objects are connected if they take part in a composition, and compositional nodes are connected with the constituent primitives [11]. In [7] this approach is revised for the OW-CZSL, modeling the feasibility of each composition.

In this work, we take a different direction, revisiting the Visual Product (VisProd) baseline that independently predicts objects and states for CZSL. Despite the lack of any compositionality-specific design, we show that VisProd is effective strategy not only for the standard CZSL, but also for the more challenging OW-CZSL setting.

3 Method

Problem Formulation: The CZSL problem can be formalized as follows. Let $\mathcal{T} = \{(x, y) | x \in X, y \in Y_s\}$ be a training set, where $x$ denotes an image in the space $X$ and $y$ is its label in the set of
seen compositions \(Y_s\). Each label is a tuple \(y = (s, o)\) of a state \(s \in S\) and an object \(o \in O\) with \(S\) and \(O\) being the set of states and objects respectively. The goal of CZSL is to learn the parameters \(\theta\) of a function \(f_\theta\) that can assign to an image one of the unseen compositions in a set \(Y_t \subseteq Y\), with \(Y\) being the set of all possible compositions, i.e. \(Y = S \times O\). The set \(Y_t\) depends on the particular CZSL setting. In standard CZSL \[10\], we have \(Y_t = Y_u\) with \(Y_u\) being a set of unseen compositions, i.e. \(Y_t \cap Y_s = \emptyset\). In generalized CZSL \[15\] we have both seen and unseen compositions at test time, i.e. \(Y_t = Y_s \cup Y_u\). More recently, in \[8\] the problem of open-world CZSL (OW-CZSL) has been studied, where the set of unseen pairs \(Y_u\) is not known apriori to the model during inference. In this setting, the output space contains all of possible compositions, i.e. \(Y_t = Y\).

**Visual Product:** Given an image \(x\), we want to model the joint probability distribution \(p(s, o|x)\), from the partial view given by \(T\). The simplifying assumption made by visual product is:

\[
p(s, o|x) \sim p(s|x) \cdot p(o|x).
\]

With this formulation, we assume state and objects to be independent. This contrasts with the main idea of recent works, explicitly modeling the relationship between objects and states (e.g. \[6, 11\]). Additionally, we are completely disregarding side information (e.g. word embeddings), tackling the task by relying solely on the visual cues.

Practically, given an image \(x\), we extract its feature representation \(z = \omega(x)\) through a function \(\omega\), mapping images into a feature space \(Z\), i.e. \(\omega : X \rightarrow Z\). We then have an object classifier \(\phi_o : Z \rightarrow \Delta_O\) that assigns \(z\) to a vector in the probability simplex \(\Delta_O\), spanning all object categories. Similarly, we have another classifier that maps \(z\) to a probability over the states, i.e. \(\phi_s : Z \rightarrow \Delta_S\).

During training, we minimize the cross-entropy loss for both the object and state predictions:

\[
\min_{\theta} \sum_{(x, (s, o)) \in T} \ell_{\text{obj}} + \ell_{\text{state}} = \min_{\theta} \sum_{(x, (s, o)) \in T} \ell_{\text{CE}}(\phi_o(\omega(x)), o) + \ell_{\text{CE}}(\phi_s(\omega(x)), s).
\]

During inference, our prediction function is

\[
f_\theta(x) = \arg \max_{y \in Y_t} \phi_o(\omega(x)) \cdot \phi_s(\omega(x)) - 1_{y \in Y, \beta_s},
\]

with \(Y_t\) being the target output space (e.g. \(Y_u\) for original CZSL, \(Y_s \cup Y_u\) for generalized CZSL, \(Y\) for OW-CZSL). Note that, for simplicity, \(\theta\) contains the parameters of all functions (i.e. \(\omega, \phi_o, \phi_s\)). In Eq. 1, \(\beta_s\) is a scalar applied only on the scores of seen compositions, to take into account the inherent bias that models have toward them. Note that this hyperparameter is used during evaluation in all recent works \[15, 11, 8, 6\] and is also standard in generalized zero-shot learning \[17\]. We implement \(\omega\) as a Convolutional Neural Network and \(\phi_o, \phi_s\) as Multi-Layer Perceptrons (MLP). To stay consistent with prior work, we refer to the model with \(\phi_o\) and \(\phi_s\) implemented as a linear classifier as VisProd, while with VisProd++ when \(\phi_o\) and \(\phi_s\) have multiple fully connected layers.

### 4 Experiments

**Implementation Details:** We perform experiments on UT-Zappos \[18\], MIT-States \[4\] and the CGQA \[11\] datasets on both the open and closed-world settings. We compare our results against Attributes as Operators (AoP) \[12\], LabelEmbed (LE+) \[10\], Task Modular Networks (TMN) \[15\], Compositional Graph Embeddings (CGE) \[11\] as well as Compositional Cosine Classifier (CompCos) \[8\]. We use a ResNet18 backbone \[3\] for extracting the image features. For VisProd++ we use a 3-layer MLP for the object and state classifiers. This architecture is very similar to the one used in both CGE \[11\] and CompCos \[8\] which employs also LayerNorm \[2\], Dropout \[16\] and ReLU \[13\].

As done in CGE \[11\], we benchmark the performance of Visual Product with the frozen ResNet backbone as well as end-to-end training. We refer to the version of Visual Product with a frozen backbone as VisProd\textsubscript{ff} and VisProd\textsubscript{ff}++. All results are the average of 5 independent runs for VisProd, while taken from \[11\] for the other methods. For C-GQA, we take the results and the updated splits of \[7\].

#### 4.1 Comparison with the state of the art

**Closed-World Results** The results for the closed-world setting are presented in Table 1. On MIT-States, the best performing Visual Product model (VisProd++), achieves better performance
Table 1: **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.

<table>
<thead>
<tr>
<th>Method</th>
<th>MIT-States</th>
<th>UT Zappos</th>
<th>C-GQA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td>U</td>
<td>HM</td>
</tr>
<tr>
<td>AoP [12]</td>
<td>14.3</td>
<td>17.4</td>
<td>9.9</td>
</tr>
<tr>
<td>LE+ [10]</td>
<td>15.0</td>
<td>20.1</td>
<td>10.7</td>
</tr>
<tr>
<td>TMN [15]</td>
<td>20.2</td>
<td>20.1</td>
<td>13.0</td>
</tr>
<tr>
<td>SymNet [6]</td>
<td>24.2</td>
<td>25.2</td>
<td>16.1</td>
</tr>
<tr>
<td>CompCos CW [8]</td>
<td>25.3</td>
<td>24.6</td>
<td>16.4</td>
</tr>
<tr>
<td>CGE ff [11]</td>
<td>32.8</td>
<td>28.0</td>
<td>21.4</td>
</tr>
<tr>
<td>CompCos [8]</td>
<td>25.6</td>
<td>22.7</td>
<td>15.6</td>
</tr>
</tbody>
</table>

Table 2: **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.

<table>
<thead>
<tr>
<th>Method</th>
<th>MIT-States</th>
<th>UT Zappos</th>
<th>C-GQA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td>U</td>
<td>HM</td>
</tr>
<tr>
<td>AoP [12]</td>
<td>16.6</td>
<td>5.7</td>
<td>4.7</td>
</tr>
<tr>
<td>LE+ [10]</td>
<td>14.2</td>
<td>2.5</td>
<td>2.7</td>
</tr>
<tr>
<td>TMN [15]</td>
<td>12.6</td>
<td>0.9</td>
<td>1.2</td>
</tr>
<tr>
<td>SymNet [6]</td>
<td>21.4</td>
<td>7.0</td>
<td>5.8</td>
</tr>
<tr>
<td>CompCos CW [8]</td>
<td>25.3</td>
<td>5.5</td>
<td>5.9</td>
</tr>
<tr>
<td>CGE ff [11]</td>
<td>29.6</td>
<td>4.0</td>
<td>4.9</td>
</tr>
<tr>
<td>CGE [11]</td>
<td>32.4</td>
<td>5.1</td>
<td>6.0</td>
</tr>
<tr>
<td>CompCos [8]</td>
<td>25.4</td>
<td><strong>10.0</strong></td>
<td><strong>8.9</strong></td>
</tr>
<tr>
<td>VisProd</td>
<td>20.9</td>
<td>5.8</td>
<td>5.6</td>
</tr>
<tr>
<td>VisProd</td>
<td>26.2</td>
<td>6.6</td>
<td>6.8</td>
</tr>
<tr>
<td>VisProd++</td>
<td>24.6</td>
<td>6.7</td>
<td>6.6</td>
</tr>
<tr>
<td>VisProd++</td>
<td>28.1</td>
<td>7.5</td>
<td>7.3</td>
</tr>
</tbody>
</table>

(AUC of 4.2) than AoP [12] (1.6), LE+ [10] (2.0), TMN [15] (2.9), and SymNet [6] (3.0), however, CompCos [8] (4.5) and CGE [11] (6.5) achieve significantly better performance. In this setting, we believe VisProd is penalized by the label noise of MIT-States [1], especially for state predictions. On the cleaner UT-Zappos and C-GQA, the results of VisProd++ is surpassed only by the state-of-the-art CGE. In particular, on UT-Zappos, VisProd++ is surpassed only by CGE (e.g. 32.5 vs 33.5 in AUC), outperforming by a margin the second best competitor (29.3 AUC of TMN). We also observe a similar trend on C-GQA where the results of Visual Product is only outperformed by CGE with both a frozen backbone as well as with end-to-end training (i.e. 3.0 vs 3.6). This is remarkable given that VisProd is based on a strong independence assumption and completely ignores side information.

**Open-World Results:** We report the results on the open-world setting in Table 2. Most notably, on UT-Zappos, VisProd++ surpasses the performance of all previous models (i.e. AUC of 26.5 vs 23.1 of CGE) and achieves the new state of the art. Similarly, on C-GQA, the performance achieved by VisProd++ is significantly higher than of prior works, achieving 0.75 AUC vs 0.39 of CompCos and 0.47 of CGE. On the noisy MIT-States, the performance of VisProd++ is only surpassed by CompCos [8] (1.2 vs 1.6), which uses feasibility scores to improve its performance on the open-world setting. Our results indicate that modeling states and objects independently may be an effective approach to deal with the very large output space of OW-CZSL.
4.2 Ablation studies

We investigate the two crucial aspects behind VisProd’s performance: the depth of the object and state classifier and the bias on seen compositions.

Effect of depth of the classifier: We ablate the impact of the depth in Fig 2a for VisProd$_I$ and VisProd$_{I+}$ in both MIT-States and UT-Zappos validation sets. The validation AUC on both UT-Zappos and MIT-States rapidly increases with the depth of the classifier. This shows the importance of having a powerful predictor with enough capacity to learn a complex classification function. However, after 3 layers, the performance degrades (i.e., going from 26.9 to 24.3 on UT-Zappos when depth is increased from 3 to 5 layers), possibly due to overfitting.

Effect of bias during evaluation: We evaluate the harmonic mean between the performance on the seen and unseen compositions in Fig 2b. The results demonstrate that using a bias to penalize seen compositions plays a vital role in the performance of VisProd. Interestingly, AoP [12] had found that VisProd had the best performance when evaluated only on the unseen classes. The performance degraded when evaluated on both seen and unseen compositions due to the inherent bias of VisProd toward the seen ones. However, subtracting a bias to their predicted value solved this issue, with the HM reported by VisProd$_I$ improving from 3.63 to 36.2 on UT-Zappos and from 6.8 to 16.0 on MIT-States. These results show that modeling the bias on seen compositions is crucial in VisProd.

5 Conclusion

In this work, we revisit the traditional baseline Visual Product for the task of Compositional Zero-Shot Learning. We find that by increasing the representation power of its state and object classifiers and taking into account its inherent bias toward training compositions during the evaluation, Visual Product (VisProd++) achieves performance on par or superior to the state-of-the-art in various CZSL benchmarks and settings. Remarkably, this method is particularly effective in open-world CZSL (OW-CZSL) where there are no priors on the unseen compositions and the full compositional space is considered as output space at test time. In this setting, VisProd surpasses all competitors in the challenging C-GQA dataset, with almost 280k compositions. We hope that our work will revive interest in visual-only methods for Compositional Zero-Shot Learning and re-consider treating objects and states separately as a competitive approach.

Acknowledgments This work has been partially funded by the ERC (853489 - DEXIM) and by the DFG (2064/1 – Project number 390727645). The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Shyamgopal Karthik.
References


