Revisiting Visual Product for Compositional Zero-Shot Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Compositional Zero-Shot Learning (CZSL) aims to recognize compositions of objects and states in images, and generalize to the unseen compositions of objects and states. 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 [8, 13]. Examples are modeling attributes as operators modifying object representations [11, 6], predicting embeddings for each composition through simple multi-layer perceptrons [9, 7] or graph convolutions [10, 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 remove training complexity and the need for side information. Unfortunately, this method named Visual Product (VisProd) performed worse than early CZSL methods [9, 11] 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 [7, 10] and far superior to the performance of the initial works in this field [9, 11, 14, 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.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

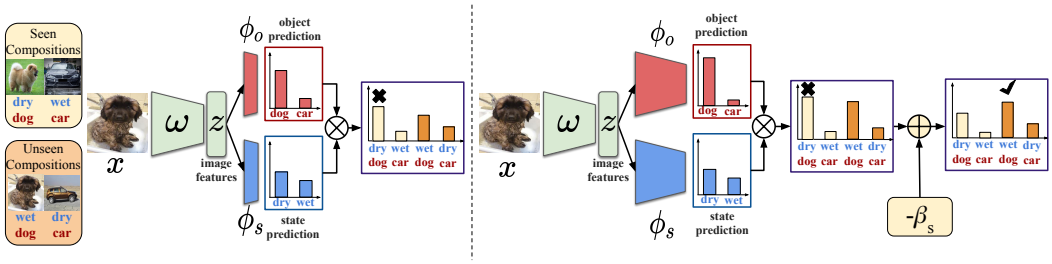


Figure 1: **Visual Product**. Previous works (left) implemented Visual Product with single-layer linear classifiers for objects and states, and did not account for the natural bias the model has towards predicting seen compositions. Our model (right) uses deeper non-linear object and state classifiers, and tackles the bias by subtracting a scalar from the scores of seen compositions.

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 [14]. 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 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 θ of a function f_θ 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 [9], we have $Y_t = Y_u$ with Y_u being a set of unseen compositions, i.e. $Y_u \cap Y_s = \emptyset$. In generalized CZSL [14] we have both seen and unseen compositions at test time, i.e. $Y_t = Y_s \cup Y_u$. More recently, in [7] 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 of all 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 $\mathcal{T}$. 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, 10]). 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 ω, mapping images into a feature space Z, i.e. $\omega : X \to Z$. We then have an object classifier $\phi_o : Z \to \Delta_O$ that assigns z to a vector in the probability simplex Δ_O, spanning all object categories. Similarly, we have another classifier that maps z to a probability over the states, i.e. $\phi_s : Z \to \Delta_S$. During training, we minimize the cross-entropy loss for both the object and state predictions:

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

Our final prediction function is

$$f_\theta = \arg\max_{y\in Y_t} \phi_o(w(x)) \cdot \phi_s(w(x)) - \mathbb{1}_{y\in Y_s}\beta_s, \tag{1}$$

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, θ contains the parameters of all functions (i.e. ω, ϕ_o, ϕ_s). In Eq. 1, β_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

Method	MIT-States				UT Zappos				C-GQA			
	S	U	HM	AUC	S	U	HM	AUC	S	U	HM	AUC
AoP	14.3	17.4	9.9	1.6	59.8	54.2	40.8	25.9	11.8	3.9	2.9	0.3
LE+	15.0	20.1	10.7	2.0	53.0	61.9	41.0	25.7	16.1	5.0	5.3	0.6
TMN	20.2	20.1	13.0	2.9	58.7	60.0	45.0	29.3	21.6	6.3	7.7	1.1
SymNet	24.2	25.2	16.1	3.0	49.8	57.4	40.4	23.4	25.2	9.2	9.8	1.8
CompCos^{CW}	25.3	24.6	16.4	4.5	59.8	62.5	43.1	28.1	27.1	10.8	11.4	2.3
CGE$_{ff}$	28.7	25.3	17.2	5.1	56.8	63.6	41.2	26.4	27.5	11.7	11.9	2.5
CGE	**32.8**	**28.0**	**21.4**	**6.5**	**64.5**	**71.5**	**60.5**	**33.5**	**31.4**	**14.0**	**14.5**	**3.6**
CompCos	25.6	22.7	15.6	4.1	59.3	61.5	40.7	27.1	26.6	8.7	9.6	1.8
VisProd$_{ff}$	20.9	20.9	12.6	2.8	54.8	56.5	41.2	25.0	24.8	8.7	9.8	1.7
VisProd	26.2	22.1	14.5	3.8	61.0	66.2	45.3	31.6	29.2	11.6	12.1	2.7
VisProd$_{ff}$++	24.6	21.8	14.0	3.5	58.0	62.0	43.7	28.7	26.0	11.2	11.3	2.3
VisProd++	28.5	22.8	15.3	4.2	61.8	66.3	46.5	32.5	29.9	12.5	13.0	3.0

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.

Method	MIT-States				UT Zappos				C-GQA			
	S	U	HM	AUC	S	U	HM	AUC	S	U	HM	AUC
AoP	16.6	5.7	4.7	0.7	50.9	34.2	29.4	13.7	NA	NA	NA	NA
LE+	14.2	2.5	2.7	0.3	60.4	36.5	30.5	16.3	16.1	0.6	0.9	0.07
TMN	12.6	0.9	1.2	0.1	55.9	18.1	21.7	8.4	NA	NA	NA	NA
SymNet	21.4	7.0	5.8	0.8	53.3	44.6	34.5	18.5	24.5	1.9	2.9	0.34
CompCos^{CW}	25.3	5.5	5.9	0.9	59.8	45.6	36.3	20.8	26.2	1.0	1.9	0.22
CGE$_{ff}$	29.6	4.0	4.9	0.7	58.8	46.5	38.0	21.5	27.0	1.4	1.9	0.23
CGE	**32.4**	5.1	6.0	1.0	61.7	47.7	39.0	23.1	**31.4**	1.9	3.2	0.51
CompCos	25.4	**10.0**	**8.9**	**1.6**	59.3	46.8	36.9	21.3	26.6	2.1	3.2	0.44
VisProd$_{ff}$	20.9	5.8	5.6	0.7	54.6	42.8	36.9	19.7	24.8	1.7	2.8	0.33
VisProd	26.2	6.6	6.8	1.1	60.5	52.3	41.3	25.8	29.2	2.6	4.1	0.61
VisProd$_{ff}$++	24.6	6.7	6.6	1.0	58.3	47.1	39.3	22.8	26.0	2.0	3.2	0.40
VisProd++	28.4	7.5	7.4	1.3	**61.8**	**52.1**	**42.3**	**26.5**	29.8	**2.9**	**4.5**	**0.70**

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.

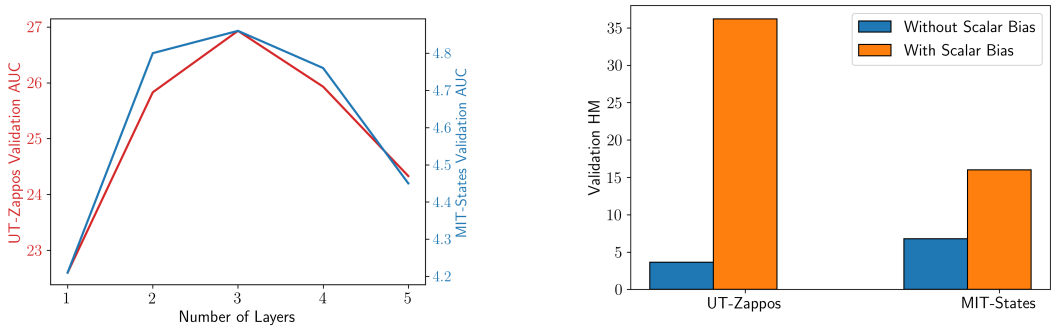
in all recent works [14, 10, 7, 6] and is also standard in generalized zero-shot learning [16]. We implement ω as a Convolutional Neural Network and ϕ_o, ϕ_s as Multi-Layer Perceptrons (MLP). To stay consistent with prior work, we refer to the model with ϕ_o and ϕ_s implemented as a linear classifier as `VisProd`, while with `VisProd++` when ϕ_o and ϕ_s have multiple fully connected layers.

3 Experiments

Implementation Details: We perform experiments on UT-Zappos [17], MIT-States [4] and the CGQA [10] datasets on both the open and closed-world settings. We compare our results against Attributes as Operators (AoP) [11], LabelEmbed (LE+) [9], Task Modular Networks (TMN) [14], Compositional Graph Embeddings (CGE) [10] as well as Compositional Cosine Classifier (CompCos) [7]. 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 [10] and CompCos [7] which employs also LayerNorm [2], Dropout [15] and ReLU [12]. As done in CGE [10], 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$_{ff}$ and VisProd$_{ff}$++. All results are the average of 5 independent runs for VisProd, while taken from [10] for the other methods.

3.1 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 (AUC of 4.2) than AoP [11] (1.6), LE+ [9] 2.0), TMN [14] (2.9), and SymNet [6] (3.0), however, CompCos [7]



(a) Effect of Classifier Depth

(b) Role of Scalar Bias

Figure 2: Ablation studies on the depth of the object and state-classifier (left) and the scalar bias (right) for UT-Zappos and MIT-States validation set in terms of AUC.

(4.5) and CGE [10] (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++ are matched 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 0.70 AUC vs 0.44 of CompCos and 0.51 of CGE. On the noisy MIT-States, the performance of VisProd++ is only surpassed by CompCos [7] (1.3 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.

3.2 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 $\text{VisProd}_{\text{ff}}$ and $\text{VisProd}_{\text{ff}}$++ 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 [11] 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 $\text{VisProd}_{\text{ff}}$ 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.

4 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 achieves performance on par or superior to the state-of-the-art in various CZSL benchmarks and settings. 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.

References

[1] Yuval Atzmon, Felix Kreuk, Uri Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition. In *NeurIPS*, 2020.

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

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

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

[5] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

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

[7] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Open world compositional zero-shot learning. In *CVPR*. IEEE, 2021.

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

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

[10] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *CVPR*. IEEE, 2021.

[11] Tushar Nagarajan and Kristen Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *ECCV*, 2018.

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

[13] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

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

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

[16] 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 TPAMI*, 41(9), 2018.

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