Learning Attention as Disentangler for Compositional Zero-shot Learning –Supplemental Material–

Shaozhe Hao Kai Han* Kwan-Yee K. Wong
The University of Hong Kong
{szhao, kykwong}@cs.hku.hk kaihanx@hku.hk

A. Coefficient β during inference

At inference time, we use a coefficient β to strike a balance between the composition score $p(c)$ and the product of the attribute and object scores $p(a)\cdot p(o)$. The final prediction score $\tilde{p}(c)$ is given by:

$$\tilde{p}(c) = p(c) + \beta \cdot p(a) \cdot p(o) \tag{1}$$

where $c \in \mathcal{C}_{test}$ and $c = (a, o) \in \mathcal{A}\times\mathcal{O}$. We first fix $\beta = 1.0$ during training, and then validate $\beta = 0.0, 0.1, \cdots, 1.0$ to choose the best $\hat{\beta}$ on the validation set. We report the chosen $\hat{\beta}$ values for different datasets in Table 1.

Datasets	Closed-world	Open-world
Clothing16K [14]	$\hat{\beta} = 0.1$	$\hat{\beta} = 0.1$
UT-Zappos50K [13]	$\hat{\beta} = 0.9$	$\hat{\beta} = 0.9$
C-GQA [8]	$\hat{\beta} = 1.0$	$\hat{\beta} = 0.7$

Table 1. The chosen $\hat{\beta}$ values for different datasets under the closed-world and the open-world settings.

B. Unseen-seen accuracy curve

For the CZSL evaluation metric, we follow the generalized evaluation protocol [1, 10]. To overcome the negative bias on seen compositions, we use a calibration term for unseen compositions. This calibration term increases unseen composition scores and leads to the following classification rule:

$$\hat{c} = \arg\max_{c\in\mathcal{C}_{test}} \tilde{p}(c) + \gamma\mathbb{I}[c \in \mathcal{C}_u] \tag{2}$$

where the prediction $\tilde{p}(c)$ is computed by Eq. (1), γ is the calibration term, $\mathbb{I}[\cdot] \in \{0, 1\}$ indicates whether or not c is an unseen composition, *i.e.*, $c \in \mathcal{C}_u$. When using different calibration terms, we can obtain different paired top-1 accuracy of seen and unseen compositions. Without any constraints, we can obtain the highest unseen accuracy by $\gamma = +\infty$ and the highest seen accuracy by $\gamma = -\infty$, leading to trivial solutions. To construct a feasible list of different calibration values, we first compute γ_i for each image i of unseen compositions:

$$\gamma_i = \max_{c\in\mathcal{C}_s} \tilde{p}(c \mid i) - \tilde{p}(c_i \mid i) \tag{3}$$

where $c_i \in \mathcal{C}_u$ is the ground-truth composition of the image i and $\tilde{p}(c \mid i)$ denotes the prediction score of composition c for the image i. A list of γ_i can be derived by applying Eq. (3) on all unseen-composition images. We then sort the list, in which the smallest value makes the highest seen accuracy and the largest value makes the highest unseen accuracy. We pick γ_i in the list with a specific interval and obtain multiple seen-unseen accuracy pairs. In this way, we can plot a curve with all scatters of seen and unseen accuracy, from which the evaluation metrics AUC (area under curve) and HM (the best harmonic mean accuracy) are obtained.

In Fig. 1, we show the unseen-seen accuracy curve of all compared CZSL methods on all datasets under the closed-world and open-world settings. With the increase of the calibration value, the classification accuracy of seen compositions decreases while the accuracy of unseen compositions increases. The evaluation metrics in the paper, *i.e.*, area under curve (AUC), the best harmonic mean value (HM), the best seen accuracy (Seen), and the best unseen accuracy (Unseen), are all derived from the unseen-seen accuracy curve. We can observe that compared to other methods, our ADE consistently achieves the best trade-off between the accuracy of seen and unseen compositions, especially on the large-scale C-GQA [8] dataset.

C. Ablation study with ResNet18 backbone

In this paper, we use ViT as our backbone, while ResNet18 is a common choice in previous works. In Table 2, we show experimental results on ablating every component in our model with both backbones to verify the effectiveness of the proposed method. We can observe that every component is crucial for both backbones. The results indicate that our model is backbone-agnostic and performs better with ViT backbone, thanks to the capability of ViT in excavating high-level sub-space information.

*Corresponding author

					ViT				ResNet18			
	CA	AA	OA	Reg	AUC	HM	Seen	Unseen	AUC	HM	Seen	Unseen
(0)	✗	✗	✗	✗	64.3	69.1	97.5	71.8	48.3	58.7	96.7	54.8
(1)	self	✗	✗	✗	65.8	71.6	98.2	71.6	48.8	58.7	96.9	56.7
(2)	self	self	self	✗	67.3	**74.3**	98.5	72.1	50.8	61.7	95.7	58.2
(3)	self	cross	cross	✗	67.3	73.0	98.7	72.7	52.3	61.3	**97.2**	60.4
(4)	self	cross	cross	✓	**68.0**	74.2	**99.0**	**73.1**	**53.7**	**64.1**	**97.2**	**60.7**

Table 2. Ablate the components in ADE on open-world Clothing16K with both backbones. CA, AA, and OA denote composition, attribute, and object attention. Reg denotes the regularization term. We test self- or cross-attention for AA and OA.

	Composition			Train		Val		Test	
Datasets	$\|\mathcal{A}\|$	$\|\mathcal{O}\|$	$\|\mathcal{A}\| \times \|\mathcal{O}\|$	$\|\mathcal{C}_s\|$	$\|\mathcal{X}\|$	$\|\mathcal{C}_s\|$ / $\|\mathcal{C}_u\|$	$\|\mathcal{X}\|$	$\|\mathcal{C}_s\|$ / $\|\mathcal{C}_u\|$	$\|\mathcal{X}\|$
Clothing16K [14]	9	8	72	18	7242	10 / 10	5515	9 / 8	3413
UT-Zappos50K [13]	16	12	192	83	22998	15 / 15	3214	18 / 18	2914
C-GQA [8]	413	674	278362	5592	26920	1252 / 1040	7280	888 / 923	5098
Vaw-CZSL [12]	440	541	238040	11175	72203	2121 / 2322	9524	2449 / 2470	10856

Table 3. Comparison of data split statistics.

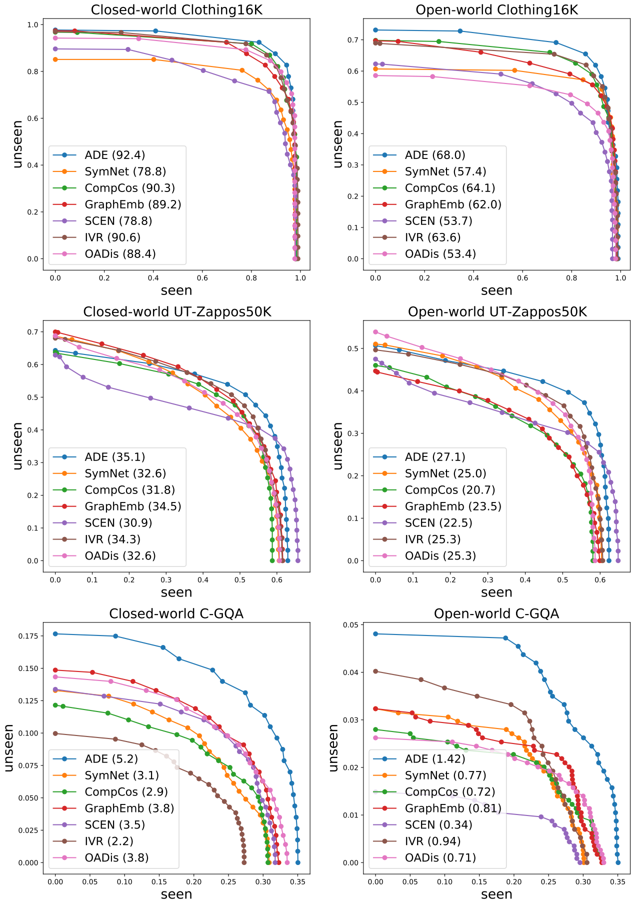

Figure 1. Unseen-seen accuracy curve on Clothing16K [14], UT-Zappos50K [13], and C-GQA [8] under the closed-world and open-world settings. We compare our ADE with SymNet [6], CompCos [7], GraphEmb [8], SCEN [5], IVR [14], and OADis [12]. Area under curve (AUC) is reported in brackets.

D. Comparison with CNN-based models

OADis [12] and ProtoProp [11] are two CZSL methods, which heavily depend on the spatial structure of convolutional features extracted from CNN models, *e.g.*, ResNet18 [3]. ProtoProp [11] extracts local prototypes of attribute and object features in the spatial dimension propagated through a GCN-based compositional graph. OADis [12] uses attribute and object affinity modules to capture the high-similarity regions in the spatial features of images with the same attribute or object. We compare our ADE with these two CNN-based models in Table 4. We can observe that our model consistently outperforms other methods on all datasets. ADE increases AUC by 6.3 on Clothing16K, 1.1 on UT-Zappos59K, and 2.1 on C-GQA. In the meanwhile, ADE increases the best harmonic mean value (HM) by 4.0% on Clothing16K, 2.3% on UT-Zappos50K, and 4.4% on C-GQA. The experimental results demonstrate ViT-based ADE is more efficient than the current CNN-based state-of-the-art models.

Saini *et al.* [12] propose a new CZSL dataset, named Vaw-CZSL, a subset of Vaw [9], which is a multi-label attribute-object dataset. Saini *et al.* [12] sample one attribute per image, leading to a much larger dataset in comparison to previous datasets as shown in Table 3. We compare ADE with all ViT-adapted methods in the main paper and CNN-based OADis [12] on Vaw-CZSL [12] in Table 5. Similar to the results on standard CZSL datasets, ADE outperforms all the other models. ADE increases AUC by 0.3 (∼27.3% relatively) and increases the best harmonic mean value (HM) by 1.2% (∼14.8% relatively). Overall, ADE achieves stable state-of-the-art performance across various small-scale and large-scale datasets.

E. Additional qualitative results

We show additional qualitative results of ADE in this section. We follow the main paper to conduct more experiments of text-to-image retrieval, image-to-text retrieval, and visual concept retrieval, adding some results on Vaw-CZSL [12].

In Fig. 3, we retrieve the top-5 closest images for texts of attribute-object compositions. For the relatively easier

	Clothing16K						UT-Zappos50K						C-GQA					
Models	AUC	HM	Seen	Unseen	Attr	Obj	AUC	HM	Seen	Unseen	Attr	Obj	AUC	HM	Seen	Unseen	Attr	Obj
ProtoProp† [11]	86.1	84.1	97.7	93.4	86.6	89.7	34.0	48.8	60.6	**66.8**	**48.3**	74.7	2.0	11.0	26.4	9.4	11.2	26.6
OADis† [12]	85.5	84.7	96.7	94.1	84.9	92.5	30.0	44.4	59.5	65.5	46.5	**75.5**	3.1	13.6	30.5	12.7	10.6	30.7
ADE (ours)	**92.4**	**88.7**	**98.2**	**97.7**	**90.2**	**93.6**	**35.1**	**51.1**	**63.0**	64.3	46.3	74.0	**5.2**	**18.0**	**35.0**	**17.7**	**16.8**	**32.3**

Table 4. Comparison results of ADE and two CNN-based models. We conduct experiments on Clothing16K [14], UT-Zappos50K [13], and C-GQA [8] under the closed-world setting. The superscript † denotes the model using ResNet18 [3] as the backbone.

	Vaw-CZSL					
Models	AUC	HM	Seen	Unseen	Attr	Obj
SymNet [6]	0.89	7.4	12.3	10.2	9.9	32.4
CompCos [7]	0.92	7.5	14.2	8.7	8.4	30.5
GraphEmb [8]	1.02	7.8	14.1	9.9	10.8	29.8
SCEN [5]	0.84	7.1	14.2	8.1	7.6	30.0
IVR [14]	0.91	7.4	13.0	9.6	8.9	31.9
OADis† [12]	0.87	7.1	13.6	9.4	9.7	31.4
OADis [12]	1.10	8.1	15.2	10.1	9.9	31.6
ADE (ours)	**1.40**	**9.3**	**15.5**	**12.0**	**11.5**	**33.8**

Table 5. Experimental results on Vaw-CZSL [12]. We compare ADE with baseline models in the main paper and OADis [12]. The superscript † denotes the model using ResNet18 [3] as the backbone. The others use ViT-B-16 [2] as the backbone.

Clothing16K [14] dataset, all the retrieved images are correct. For the more challenging large-scale Vaw-CZSL [12] dataset with more complicated semantics of attributes and objects, some wrong images may be retrieved but they are highly semantically-related to the given text. Taking the "flying plane" (row 5) as an example, the mismatched images are the "in-the-air jet", the "metal plane", the "diagonal jet", and the "in-the-air plane". These images are labelled with synonyms or from a different perspective, but they are essentially images of a "flying plane". We can observe that ADE performs equally well for seen and unseen compositions.

In Fig. 4, we retrieve the top-5 closest compositional texts for images of seen and unseen compositions. For seen compositions, it is difficult to retrieve the ground-truth label in the top-1 closest result, but all the retrieved texts are related to the image, giving the reasonable attribute-object compositions which the ground-truth label fails to incorporate. For unseen compositions, although it is quite hard to retrieve the unseen ground-truth label because of the learning bias on seen compositions, the retrieved texts are mostly reasonable to describe the given image. These results indicate ADE efficiently connects the compositional texts and the corresponding images by transferring knowledge from seen concepts to unseen compositions.

The property of ADE to disentangle concept-exclusive features enables us to conduct visual concept retrieval experiments. In Fig. 2, we retrieve the attribute-related or the object-related images for the given image based on their visual concept feature distances. We report the top-5 retrieval results of four images by their attribute-exclusive and object-exclusive features. The results show that ADE effectively disentangles the attribute and object concepts from visual images and produces reliable concept-exclusive features.

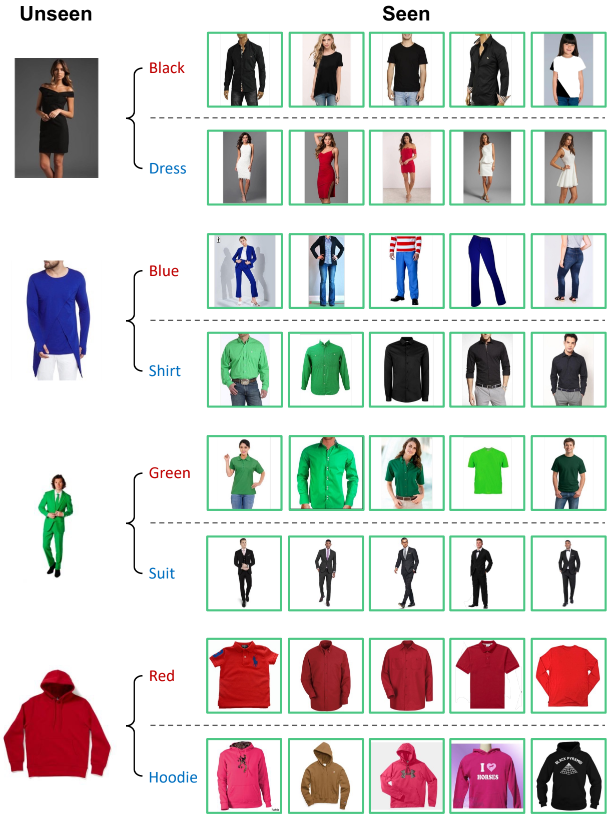

Figure 2. Retrieve *seen* compositions for *unseen* compositions based on the visual concept feature distance. We report the top-5 retrieval results on Clothing16K [14]. All the retrieved images for the corresponding concept are correct (in the green box).

F. Pseudocode for ADE

ADE is simple and easy to implement. For reproducibility, we show the PyTorch-style pseudocode of ADE for training in Algorithm 1 and for inference in Algorithm 2. The complete source code of ADE is available: `https://github.com/haoosz/ade-czsl`.

G. Broader Impacts

Compositional zero-shot learning is a new topic of learning visual features for the objects and the corresponding attributes. Our work efficiently disentangles attribute features and object features to learn the compositionality of

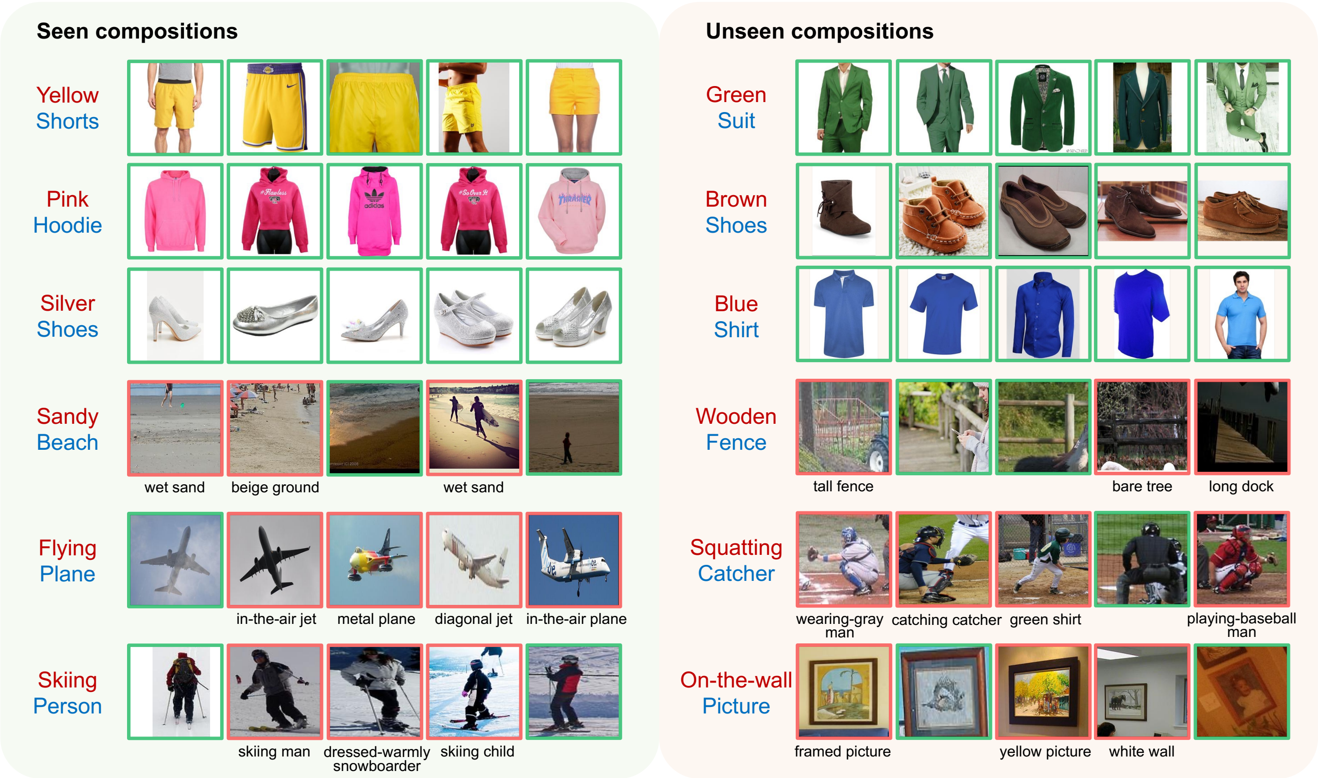

Figure 3. Text-to-image retrieval of seen (left) and unseen (right) compositions. We report the top-5 closest retrieval results on Clothing16K [14] (top three rows) and Vaw-CZSL [12] (bottom three rows). The correct image is in the green box, and the wrong image is in the red box with its ground-truth label below (black text).

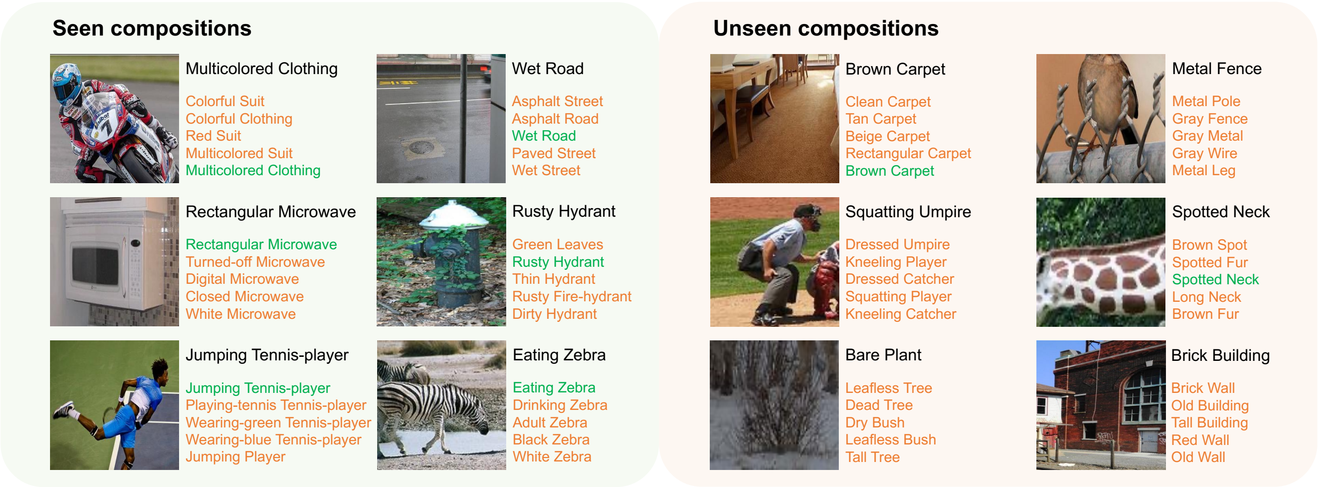

Figure 4. Image-to-text retrieval of seen (left) and unseen (right) compositions. We report the top-5 closest retrieval results on Vaw-CZSL [12]. The black text denotes the ground-truth label, the green text denotes the correct result, and the orange text denotes the wrong result.

visual images in the real life. Our work can be used to recognize attribute-object compositions, significantly extending the traditional object recognition, which has various positive implications, *e.g.*, object detection, fine-grained recognition, and action recognition. Besides, our work also contributes to the explainability of deep learning models by exploring how they learn unseen things in the real world. However, there are negative impacts as well. Although it seems far from becoming true, models may be used for harmful purposes, *e.g.*, building weapons and conducting surveillance. When the learning ability is no longer constrained by the training data and the specific training task, it is possible to train models with normal and harmless data for evil implementations because we have no idea what compositional

Algorithm 1: PyTorch-style pseudocode for training

```
# emb_a, emb_o, emb_c: embeddings of
 attributes, objects, and compositions
# f: visual encoder
# attn_a, attn_o, attn_c: attribute,
 object, and composition attention blocks
# proj_a, proj_o, proj_c: projection with
 attribute, object, composition embedders
# emd: adapted EMD at the attention level

# initialize attribute/object embeddings;
 compose them to composition embeddings
emb_a = init(all_attr)
emb_o = init(all_obj)
emb_c = compos(emb_a, emb_o)
# load 3 images and labels
for x, x_a, x_o, c in train_loader:
   # composition, attribute, object label
   y, y_a, y_o = c
   # encoded tokens
   z, z_a, z_o = f(x), f(x_a), f(x_o)
   # concept features and attention maps
   out_a1, amap_aa1 = attn_a(z, z_a)
   out_a2, amap_aa2 = attn_a(z_a, z)
   out_o1, amap_oo1 = attn_o(z, z_o)
   out_o2, amap_oo2 = attn_o(z_o, z)
   out_c, _ = attn_c(z, z)
   # when inputs are of no interest
   _, amap_ao1 = attn_a(z, z_o)
   _, amap_ao2 = attn_a(z_o, z)
   _, amap_oa1 = attn_o(z, z_a)
   _, amap_oa2 = attn_o(z_a, z)
   # probabilities
   p_a1 = proj_a(out_a1) @ emb_a.T
   p_a2 = proj_a(out_a2) @ emb_a.T
   p_o1 = proj_o(out_o1) @ emb_o.T
   p_o2 = proj_o(out_o2) @ emb_o.T
   p_c = proj_c(out_c) @ emb_c.T
   # cross entropy losses
   l_a1 = cross_entropy(p_a1, y_a)
   l_a2 = cross_entropy(p_a2, y_a)
   l_o1 = cross_entropy(p_o1, y_o)
   l_o2 = cross_entropy(p_o2, y_o)
   l_c = cross_entropy(p_c, y)
   # adapted EMDs
   s_aa = emd(amap_aa1, amap_aa2)
   s_oo = emd(amap_oo1, amap_oo2)
   s_ao = emd(amap_ao1, amap_ao2)
   s_oa = emd(amap_oa1, amap_oa2)
   # loss
   l_ce = l_a1 + l_a2 + l_o1 + l_o2 + l_c
   l_reg = s_ao + s_oa - s_aa - s_oo
   loss = l_ce + l_reg
   # optimization step
   loss.backward()
   optimizer.step()
```

Algorithm 2: PyTorch-style pseudocode for inference

```
# emb_a, emb_o, emb_c: embeddings of
 attributes, objects, and compositions
# f: visual encoder
# attn_a, attn_o, attn_c: attribute,
 object, and composition attention blocks
# proj_a, proj_o, proj_c: projection with
 attribute, object, composition embedders
# p: a dictionary storing probabilities
# beta: probability coefficient

# initialize attribute/object embeddings;
 compose them to composition embeddings
emb_a = init(all_attr)
emb_o = init(all_obj)
emb_c = compos(emb_a, emb_o)
# encoded tokens
z = f(x)
# concept features
out_a, _ = attn_a(z, z)
out_o, _ = attn_o(z, z)
out_c, _ = attn_c(z, z)
# probabilities
p_a = proj_a(out_a) @ emb_a.T
p_o = proj_o(out_o) @ emb_o.T
p_c = proj_c(out_c) @ emb_c.T
# initialize an empty p
p = {}
# enumerate all compositions
for c in all_comp:
   a, o = c # c = (a, o)
   # combine 3 probabilities
   p[c] = p_c[c] + beta * p_a[a] * p_o[o]
return p # return final probabilities
```

information we can derive from the training data. In general, learning attribute and object features for compositional zero-shot learning has both positive and negative impacts, depending on how people implement this technology.

H. Data licences

Clothing16K[1] [14] is a split of a public dataset in Kaggle competitions under CC0 license. UT-Zappos50K[2] is collected by Yu *et al.* [13], allowing non-commercial research use. C-GQA [8] is a split built on top of Stanford GQA dataset[3] [4], which is free for non-commercial research use. Vaw-CZSL [12] is a subset of Vaw[4] [9] under MIT license.

[1] https://www.kaggle.com/datasets/kaiska/apparel-dataset

[2] https://vision.cs.utexas.edu/projects/finegrained/utzap50k/

[3] https://cs.stanford.edu/people/dorarad/gqa/index.html

[4] https://vawdataset.com/

References

[1] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3

[4] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019. 5

[5] Xiangyu Li, Xu Yang, Kun Wei, Cheng Deng, and Muli Yang. Siamese contrastive embedding network for compositional zero-shot learning. In *CVPR*, 2022. 2, 3

[6] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *CVPR*, 2020. 2, 3

[7] M Mancini, MF Naeem, Y Xian, and Zeynep Akata. Open world compositional zero-shot learning. In *CVPR*, 2021. 2, 3

[8] MF Naeem, Y Xian, F Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *CVPR*, 2021. 1, 2, 3, 5

[9] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *CVPR*, 2021. 2, 5

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

[11] Frank Ruis, Gertjan Burghouts, and Doina Bucur. Independent prototype propagation for zero-shot compositionality. In *NeurIPS*, 2021. 2, 3

[12] Nirat Saini, Khoi Pham, and Abhinav Shrivastava. Disentangling visual embeddings for attributes and objects. In *CVPR*, 2022. 2, 3, 4, 5

[13] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014. 1, 2, 3, 5

[14] Tian Zhang, Kongming Liang, Ruoyi Du, Xian Sun, Zhanyu Ma, and Jun Guo. Learning invariant visual representations for compositional zero-shot learning. In *ECCV*, 2022. 1, 2, 3, 4, 5