Disentangling Visual Embeddings for Attributes and Objects - Supplementary

Nirat Saini Khoi Pham Abhinav Shrivastava
University of Maryland, College Park

1. Dataset issues of C-GQA [10]

GraphEmb [10] proposes a new benchmark for compositional zero-shot learning. However, there are some issues with the dataset have been raised on their official github page [1, 2]. These issues are related to (1) the attribute-object pairs being placed into the incorrect train, validation, and test subset, and (2) there are missing images for a decent amount of pairs (20%), which could potentially affect the final experiment results. Due to [10] being unable to provide a corrected version of the dataset in time before the CVPR 2022 deadline, we were unable to run any experiments for C-GQA. Post the deadline, we did run some preliminary results where our method outperformed GraphEmb [10]. Although, a major issue we observed was for OADis, C-GQA [10] training set did not have similar attributes and objects samples for constructing I_{attr} and I_{obj}. However, we propose for learning compositional concepts, firstly disentangled concepts must be learnt, and for that, we require I_{attr} and I_{obj}. Hence, we do not report results on C-GQA for OADis.

2. Dataset Creation: VAW-CZSL

We propose a new benchmark for the compositional zero-shot learning task (CZSL), focusing on images of objects and attributes in the wild that span across a much larger number of categories. We select the VAW dataset [12] to create our benchmark. VAW contains images originally from Visual Genome (thus objects and attributes in the wild). Every image of an object instance contains an object label and one (or possibly multiple) attribute labels. In the followings, we describe our steps in creating the VAW-CZSL benchmark, which shares some similarities with the C-GQA dataset.

Different from C-GQA, we consider object instances whose bounding boxes are larger than 50 x 50. C-GQA selected instances whose boxes are larger than 112 x 112, which could possibly leave out small, narrow objects that are still recognizable from images. For every object instance, among its possibly multiple attributes label, we keep only one attribute that has the lowest frequency in the dataset (*i.e.*, the uncommon attribute) to be consistent with the standard CZSL benchmark. By keeping the most uncommon attribute and using the top-3 & 5 evaluation metrics, all methods will be evaluated based on whether they are able to rank this uncommon (but still representative) attribute in its top-3 & 5 predictions rather than always predicting the most frequent attributes. From this, we follow the similar steps from [10] to merge plurals and synonyms (e.g., {*airplane, plane, aeroplane, airplanes...*}, {*rock, stone, rocks...*}). We then keep only those attribute and object categories with frequency greater than 30 to make sure all primitive concepts have a decent amount of data for training and evaluating.

We use images in VAW-training as our training set, and use images in VAW-val and VAW-test for creating the validation and testing splits following the standard generalized benchmark in CZSL. We first merge VAW-val and VAW-test in one set, and follow similar steps mentioned in [10] to create a validation and test set of seen and unseen attribute-object pairs. At the end, we remove objects and attributes that no longer appear in the training set. This is because a model that has never seen an attribute (or object) will find it impossible to generalize to unseen pairs containing this attribute (or object). This problem happens with the C-GQA dataset where 8% of attribute and 22% of object categories do not exist in their training set. More details about dataset can be found in Table 1. The dataset splits are made publicly available at https://github.com/nirat1606/OADis.

3. Implementation Details

Following baselines, we use ResNet18 [5] pre-trained on Imagenet [4] as backbone feature extractor. Since, proposed auxiliary losses leverage image features, we use a single convolutional layer with Batch Normalization, ReLU and dropout for Image embedder with output dimension 1024 and dropout as 0.3. Note that we extract ResNet features before average pool. For word embeddings, we initialize with GLoVe [11]. Object Conditioned network, uses multiple linear layers, first for objects and attributes separately, then for concatenated features. Label embedder takes 1024-d feature, performs AveragePool and finally embeds in a 300-d space. Each loss uses compatibility function, i.e.

cosine similarity, followed by cross-entropy loss over the compatibility function. Object similarity and attribute similarity modules also use two linear layers with dropout 0.05. On UT-Zappos, because the dataset is very small, we find using a linear layer (a smaller and simpler module than OCN) with dropout 0.1 results in better performance. We use Adam optimizer with weight decay $5e^{-5}$, and learning rate $2.5e^{-6}$ for the GLoVe embedding. The learning rate for the rest of the model is $3e^{-4}$ on MIT-States, and $1e^{-4}$ on UT-Zappos and VAW-CZSL. We decay the learning rate by 10 at epoch 30 and 40 on MIT-States, at epoch 50 on UT-Zappos, and at epoch 70 on VAW-CZSL. OADis needs to be trained for 70-150 epochs depending on the dataset, and training time is comparable with other methods (5-7 hours). These implementation details are also provided in our released source code.

4. Ablation studies (extension)

As mentioned in the paper, we show ablation for various other parameters. All the ablations are done for MIT-states [6], for one random seed initialization, and are consistent for other datasets as well.

4.1. Choice of word embeddings

Prior works [7,8,10] experiment with various kinds of word embeddings. In fact, GraphEmb [10] has more advantages over all other baselines, since they use a combination of word embeddings word2vec [9] and fasttext [3], whereas rest of the works use GloVe [11] only. To keep the results fair between all methods, we run all the baselines, even GraphEmb [10] with only GloVe [11], and report the accuracy in Table 1, in the main paper. Results for using different embedding combinations is shown in Table 2. Overall, since our method uses word embeddings for visual disentanglement, the choice of word embeddings does not impact the performance much. Although, empirically, we found our model performs best when GloVe embeddings are used.

4.2. Object-conditioned network

We experiment with different networks on top of word embeddings, namely Linear, MLP and Object-Conditioned. Object conditioned network uses word embedding for object to concatenate with attribute-object composition embeddings. We show in Figure 1, the diagrammatic representation of different networks.

4.3. Values for λ and δ

We find the temperature variables λ and δ empirically. The values $\lambda = 10$ and $\delta = 0.05$ works best for OADis. Table 3 shows the results for all the different configurations. To understand the effect of each temperature variable, we keep all the rest of the parameters constant and only change the studied parameter.

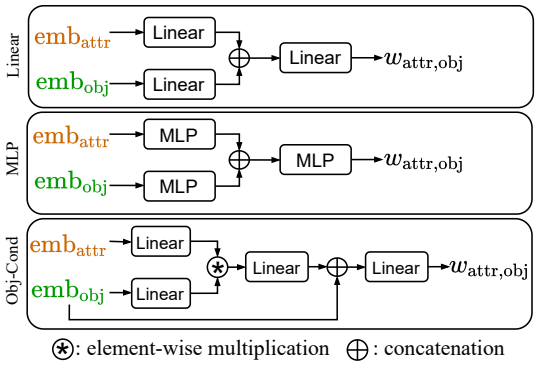

Figure 1. We show the different networks used on top of word embeddings. Empirically and following our intuition, Object-Conditioned network works best among the three (rest two are Linear and MLP). (Sec 4.2)

4.4. Different weights for losses

We mention different weights for each loss function in the paper, in Section 3.3. Each α value is empirically found, and is used in the following equation for final loss function:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \alpha_1 \mathcal{L}_{\text{attr}} + \alpha_2 \mathcal{L}_{\text{obj}} + \alpha_3 \mathcal{L}_{\text{seen}} + \alpha_4 \mathcal{L}_{\text{unseen}}$$

Note that $\mathcal{L}_{\text{cls}}$ is the main branch. The object and attribute losses are complementary, as shown in paper (Table 4). Hence, α_1 and α_2, which are the weights for $\mathcal{L}_{\text{attr}}$ and $\mathcal{L}_{\text{obj}}$ share the same values, *i.e.* 0.5. Finally, α_4 and α_5 have the same value since both are composition losses for seen and unseen pairs, *i.e.* 0.05. The chosen weights for α values are in bold in Table 4.

5. Qualitative results

We show more qualitative results to support our architecture for different datasets.

5.1. UT-Zappos.

We show nearest neighbor results in paper for MIT-States [6] (Fig. 4(a)). Here, we show similar study for UT-Zappos [13] in Figure 3. Using the hallucinated composed features of unseen pairs, we find the top 5 nearest neighbors from test set. The red boxes show incorrect labels, where green show the correct labels.

5.2. Attention Maps

In Figure 2 and 4, we show the qualitative results on MIT-States [6] and VAW-CZSL, with examples f and f_{attr} and overlayed feature maps. To re-iterate, for images with features f and f_{attr}, $\mathbf{m}_{\text{attr}} \cdot f$ shows how the regions in f which are most similar to f_{attr}, and $\mathbf{m} \cdot f_{\text{attr}}$ shows the regions in f_{attr} which are most similar to regions in f. Lastly, $\mathbf{m}'_{\text{obj}} \cdot f_{\text{attr}}$ shows the regions of f_{attr} which are most dissimilar to f.

Table 1. Dataset Details: This table shows the statistics for different datasets and their splits. The proposed VAW-CZSL benchmark significantly increases the number of attributes and objects.

	Train set				Val set			Test set		
Datasets:	Attr.	Obj.	Seen Pairs.	# Images	Seen Pairs	Unseen Pairs	# Images	Seen Pairs	Unseen Pairs	# Images
MIT-States [6]	115	245	1262	30338	300	300	10420	400	400	12995
UT-Zappos [13]	16	12	83	22998	15	15	3214	18	18	2914
VAW-CZSL [12]	440	541	11175	72203	2121	2322	9524	2449	2470	10856

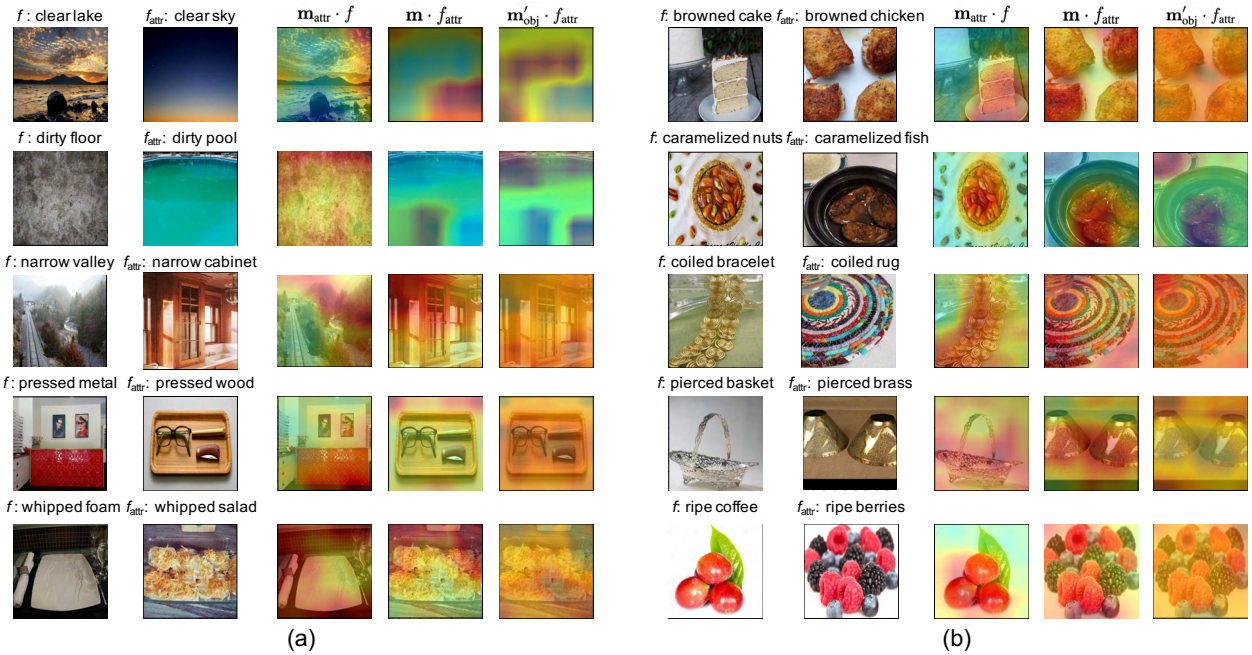

Figure 2. **(a) Failure Cases:** Shows the image pairs, f and f_{attr}, and the similarity and dissimilarity map overlayed (details in Sec 2). Moreover, we show for some cases for MIT-States, the examples are very vague or incorrect to actually capture attribute and object concepts separately. For instance, in `clear lake` and `clear sky`, it is very difficult to distinguish lake and sky. Hence the similarity and dissimilarity maps do not perform very well. Other examples are also of failure cases where the overlayed similarity and dissimilarity maps do not make sense. **(b) Correct Examples:** This shows some good examples, where the similarity and dissimilarity maps capture the attibutes and objects correctly for MIT-States.

Table 2. Results with pre-trained word-embeddings. GloVe [11] performs the best, and is therefore used for OADis. (Sec 4.1)

Word Embs	Val AUC@1	Test AUC@1
Glove	**7.6**	**5.9**
Fasttext	7.4	5.3
Word2vec	7.5	5.4
Glove+fasttext	7.4	5.5
Glove+word2vec	7.5	5.6
Fasttext+word2vec	7.4	5.6

Table 3. Results with pre-trained word-embeddings. GloVe [11] performs the best, and is therefore used for OADis. (Sec 4.3)

λ	Val AUC@1	Test AUC@1
0.01	7.5	5.6
0.1	7.5	5.7
1	7.4	5.7
10	**7.6**	**5.9**
100	7.4	5.7
δ	Val AUC@1	Test AUC@1
0.01	6.4	4.8
0.05	**7.6**	**5.9**
0.1	6.7	5.2

Although, the overlayed attention maps for similarity and dissimilarity make sense most of the times (Figure 2(b)), due to some inconsistencies in dataset, we still find some samples where is it difficult to disentangle the attribute and object features. The main reasons why this happens is:

Figure 3. We show the top 5 nearest neighbors using the hallucinated unseen composition features for UT-Zappos. All the neighbors with correct labels are represented by green, whereas incorrect ones are represented with red outline.

Table 4. We show empirical weights of each loss function in this table. (Sec 4.4)

α_1 and α_2	α_3 and α_4	Val AUC@1	Test AUC@1
0.1	0.05	7.1	5.7
0.5	0.1	7.0	5.3
0.1	0.05	7.5	5.8
0.5	**0.05**	**7.6**	**5.9**
1.0	0.05	7.3	5.6

- Some concepts are abstract, such as `clear sky`, `pressed metal`, `dirty floor` (fig. 2(a)), since it is very difficult to separate `dirty` from `floor`. Hence, the attention maps for similarity and dissimilarity do not make much sense.
- Some images in MIT-States and even in other dataset are mislabelled (*e.g.* `whipped foam` in fig. 2(a)), which makes it difficult to learn attributes from those.
- Finally, for some cases, like `narrow valley`, our method fails to disentangle attribute and object similarity, due to various objects in the scene. For future work, using a foreground and background separator before finding similarities and dissimilarities between features can be helpful.

6. Negative Impact of our work

Our work is a new initiative in the direction of learning visual features for objects and it's attributes. We present it as a prototype, or an alternative direction for understanding attributes-object pairs. Similar to any other work in vision, learning attributes of objects can have various positive implications, *e.g.* in object detection, knowing attributes can provide additional knowledge about the objects. However, knowing the additional information about attributes, it can be used for persuasion for marketing policies, for even worse factors. Even though it seems very far fetched ideas, but using attribute classification along with object detection, knowing the attributes people can build weapons and ammunition to either counter attack the present ammunition. Attribute classification can also be used on humans, to detect certain traits of human for bypassing large-scale surveillance applications. In general, attributes provide additional information for objects, which can be used negatively or positively.

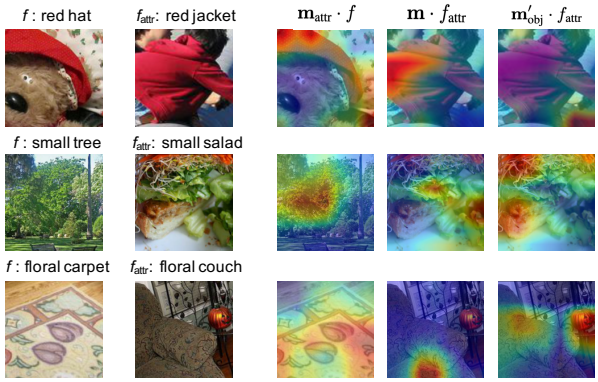

Figure 4. **Correct Examples:** We show the similarity and dissimilarity attention maps overlayed on images for VAW-CZSL as well. To re-iterate, for images with features f and f_{attr}, $\mathbf{m}_{attr} \cdot f$ shows how the regions in f which are most similar to f_{attr}, and $\mathbf{m} \cdot f_{attr}$ shows the regions in f_{attr} which are most similar to regions in f. Lastly, $\mathbf{m}'_{obj} \cdot f_{attr}$ shows the regions of f_{attr} which are most dissimilar to f.

7. Dataset license

Because we are creating the VAW-CZSL dataset based on the existing VAW dataset, as per the guideline of CVPR 2022, we provide the VAW dataset URL and license as follows:

- URL: https://vawdataset.com
- License: https://github.com/adobe-research/vaw_dataset/blob/main/LICENSE.md

References

[1] Official github for c-gqa. https://github.com/ExplainableML/czsl/issues/4. Accessed: 2021-11-22. 1

[2] Official github for c-gqa. https://github.com/ExplainableML/czsl/issues/3. Accessed: 2021-11-22. 1

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. 2

[4] Jia Deng, W. Dong, R. Socher, L. Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*, 2009. 1

[5] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1

[6] Phillip Isola, Joseph J. Lim, and E. Adelson. Discovering states and transformations in image collections. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1383–1391, 2015. 2, 3

[7] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11313–11322, 2020. 2

[8] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Learning graph embeddings for open world compositional zero-shot learning. *ArXiv*, abs/2105.01017, 2021. 2

[9] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 2

[10] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. *ArXiv*, abs/2102.01987, 2021. 1, 2

[11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 1, 2, 3

[12] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3

[13] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014. 2, 3