Learning Graph Embeddings for Compositional Zero-shot Learning Supplementary material

1. Creating C-GQA

We introduced a new benchmark for Compositional Zero-shot Learning (CZSL) in the main manuscript. This benchmarks is based on the original GQA [2] dataset which is annotated with scene graphs where each bounding box is labelled with state-object or any other relations in the scene. For the creation of the benchmark, we only consider the bounding boxes with a single state-object relation to be consistent with existing works. Bounding boxes smaller than 112×112 are excluded because they are in half the input size of most feature extractors. From these bounding boxes, we collect the vocabulary of state and objects to remove overlapping concepts between state and object classes. We also merge plurals and synonyms using first wordnet [6] and then manual checking. This yields the final vocabulary of 457 states and 893 objects.

We define a novel composition as a state-object pair not present in the training set. We now want to generate a validation and test set consisting of seen and novel compositions. We partition the testset of GQA randomly with respect to scene graphs into the validation and test sets of C-GQA with a probability of 0.45 and 0.55 respectively. These numbers are chosen as the test set of C-GQA losses bounding boxes overlapping with the novel compositions in validation set. From the bounding boxes, we add the novel compositions in these graphs to the unseen set $\mathcal{Y}_{n-val}$ and $\mathcal{Y}_{n-test}$ respectively. However, the number of novel compositions is very small compared to the compositions in the training set represented by $\mathcal{Y}_s$. Therefore, we further divide the remaining compositions in validation randomly into $\mathcal{Y}_s$ and $\mathcal{Y}_{n-val}$. We then remove the unseen compositions of the validation set from the test set and divide the remaining compositions randomly into $\mathcal{Y}_s$ and $\mathcal{Y}_{n-test}$. Finally we remove the novel compositions $\mathcal{Y}_{n-val}$ and $\mathcal{Y}_{n-test}$ from $\mathcal{Y}_s$ and generate the images from the bounding boxes of the scene graph for the 3 sets.

This results in a training set consisting of 7882 pairs across 26k images; a validation set consists of 893 seen and 834 unseen pair across 4k images; and a testset consists of 845 seen and 705 unseen pairs across 5k images. In total C-GQA has a compositional space of over 9.5k compositional concepts making it the most extensive dataset for CZSL. With cleaner labels and a bigger label space, we hope this dataset is able to accelerate the research in the field.

2. Additional Experiments

2.1. AUC at different k.

We reported the top1 AUC for the three datasets in the Table 2 of our main manuscript. We report top 1,2,3 AUC for the 3 datasets in table 1 to allow direct comparison with older works that adopt this evaluation. We see that the trend from the main manuscript is consistent at different k. In particular on MIT-States, our model CGE achieves top3 AUC of 21.3 compared to 12.3 of the closest baseline Symnet continuing our $2\times$ improvement. On UT-Zappos, CGE maintains its lead by achieving a top3 AUC of 77.5 compared to 69.8 of TMN. Finally on C-GQA, CGE again achieves a $2\times$ improvement by achieving a top3 AUC of 6.4 compared to 3.3 of Symnet.

2.2. End-to-end training with baselines

We studied the impact of feature representations on the performance of the model in section 4 of the main paper. We showed that our model CGE benefits greatly from end to end training. Older baselines TMN and Symnet operate on a frozen ImageNet trained Resnet18 CNN backbone for feature extraction in their respective manuscripts. In this experiment, we train TMN and Symnet end to end (represented by EE) from the ImageNet- pretrained Resnet18 (the same with our CGE) and quantify if they are held back by the ImageNet representations on the validation set of MIT-States.

We see from table 2 that finetuning the CNN backbone results in worse performance than in the original implementations as they are overfitting to the training set with an AUC of 2.9 for TMN[11] and 3.9 for SymNet[4], while end-to-end training is beneficial for our CGE which achieves an AUC of 8.6 because of our graph regularization.

2.3. Ablation over the GCN

We reported ablation over various components of the Graph Convolutional Network (GCN) used in our model in the section 4.2 of the main manuscript. We ablate over

	MIT-States						UT-Zap50K						C-GQA					
	Val AUC			Test AUC			Val AUC			Test AUC			Val AUC			Test AUC		
Top $k\rightarrow$	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
AttOp[9]	2.5	6.2	10.1	1.6	4.7	7.6	21.5	44.2	61.6	25.9	51.3	67.6	0.8	2.2	3.4	0.4	1.1	1.7
LE+[8]	3.0	7.6	12.2	2.0	5.6	9.4	26.4	49.0	66.1	25.7	52.1	67.8	0.9	2.2	3.2	0.4	1.0	1.5
TMN[11]	3.5	8.1	12.4	2.9	7.1	11.5	36.8	57.1	69.2	29.3	55.3	69.8	1.7	4.2	6.4	0.8	1.8	2.9
Symnet[4]	4.3	9.8	14.8	3.0	7.6	12.3	25.9	50.9	64.5	23.9	48.2	64.4	2.9	5.1	7.4	1.0	2.3	3.3
CGE_{ff} (Ours)	6.8	14.6	20.2	5.1	11.8	17.3	38.7	60.2	73.2	26.4	55.6	71.0	3.5	7.4	10.4	1.4	3.2	4.5
CGE (Ours)	**8.6**	**17.6**	**24.9**	**6.5**	**14.7**	**21.3**	**43.2**	**64.5**	**77.7**	**33.5**	**64.2**	**77.5**	**5.3**	**10.7**	**14.6**	**2.5**	**4.6**	**6.4**

Table 1: AUC in percentage on MIT-States, UT-Zap50K and GQA. We consistently outperform the baselines by a significant margin.

Method	**AUC**	**Best HM**
TMN EE [11]	2.9	13.0
Symnet EE [4]	3.9	15.3
CGE (Ours)	**8.6**	**23.4**

Table 2: End-to-End training results

Dataset	**Embedding Model**					
	gl	w2v	ft	gl+w2v	ft+w2v	
MIT-States [3]	6.4	6.4	6.5	6.6	6.6	**6.8**
UT-Zappos [12]	38.6	**38.7**	37.5	37.0	36.2	38.1
C-GQA (Ours)	3.4	**3.5**	3.2	3.4	3.3	3.5

Table 3: **Ablation over embedding**: We use three popular word embedding models. (ft: Fasttext[1], w2v: Word2Vec[5] and gl: Glove[10])

	Num layers				
Hidden dim	2	4	6	8	10
1024	6.53	6.13	5.58	5.07	4.33
2056	6.59	6.20	6.14	5.68	5.10
4096	**6.80**	6.30	6.20	5.83	4.95
8184	6.59	6.27	6.27	5.63	4.71

(a) Ablation over GCN

	Num layers				
Hidden dim	2	4	6	8	10
1024	5.56	5.21	5.44	5.43	5.12
2056	6.00	6.10	6.00	5.92	5.84
4096	6.11	6.00	6.22	6.11	5.76
8184	6.54	6.14	6.00	5.61	5.32

(b) Ablation over GCNII

Table 4: Ablation over the depth and hidden dimension of the GCN on CGE_{ff}

the remaining components of the GCN. For these experiments, we use the fixed feature extractor version of our model CGE_{ff} to quantify the improvements directly from the graph wrt to the word embeddings used for initialization and the learnable GCN configuration.

Choice of embedding. We test three popular word embedding models and the concatenation of their features for every word to study their impact on the performance of our model. We report the results in Table 3. We see that MIT-States benefits most from the concatenation of fasttext and word2vec models as these models are closely related to achieve a AUC of 6.8. While UT-Zappos and C-GQA achieve the best results with Word2Vec at 38.7 and 3.5 AUC respectively.

Graph architecture. We ablate over the learnable architecture of GCN at different depth and hidden dimension on the validation set of MIT-States and report results in table 4a. We observe that increasing the hidden dimension is generally beneficial when we go from 1024 to 4096 as the performance increases from an AUC of 6.53 to 6.80. However, increasing the hidden dimension from 4096 to 8192 decreases the AUC to 6.59 at 2 layers of GCN. Increasing the depth of the GCN network generally results in a decrease in performance across all hidden dimensions. In particular, at 4096 the AUC decreases from 6.80 AUC to 4.95. In order to validate if this is a consequence of laplacian smoothing we use a recent version of graph convolution called GCNII[7]. We see from table 4b that the performance decrease across columns is less pronounced at different hidden dimensions for this model. However, the best AUC achieved at 6.54 is less than we achieved with the original GCN indicating that this version of graph convolution is less beneficial for our problem.

References

[1] 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

[2] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional

question answering. In *CVPR*, 2019. 1

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

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

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 2

[6] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990. 1

[7] Zhewei Wei Ming Chen, Bolin Ding Zengfeng Huang, and Yaliang Li. Simple and deep graph convolutional networks. In *ICML*, 2020. 2

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

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

[10] 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)*, pages 1532–1543, 2014. 2

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

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