CymNet: CLIP boosted SymNet for Compositional Zero-shot Learning

Jiaming Shan(*)
Department of Computer Science
Shanghai Jiao Tong University
Shanghai, China
shan_jiaming@sjtu.edu.cn

Zhaozi Wang(*)
Department of Computer Science
Shanghai Jiao Tong University
Shanghai, China
ocmykr2@sjtu.edu.cn

MingShu Zhai(*)
Department of Computer Science
Shanghai Jiao Tong University
Shanghai, China
zhaimingshuzms@sjtu.edu.cn

1

***Abstract*—In this paper, we present CymNet, a new approach for compositional zero-shot learning (CZSL) that builds upon the successful foundation of SymNet and introduces the use of CLIP to enhance its compatibility with zero-shot settings. Our research was motivated by the recognition that the potential of SymNet, which is based on the principles of symmetry and group theory, has not been fully exploited in CZSL tasks. By incorporating CLIP into SymNet, CymNet demonstrates improved performance on the UT-Zappos50K dataset, with a top-3 accuracy increase of 7%. We also conducted additional experiments to evaluate the necessity of various components of the model, including the CLIP adapter.**

Keywords—Compositional Zero-shot Learning, Attribute, SymNet, CLIP

I. INTRODUCTION

Humans could easily determine whether a closet is closed, after viewing a "closed cabinet" and a "closed wardrobe" image. We observe that some of the doors have been closed in the image, and the room has been split into two parts that separate the space inside and outside the door. But it is not an easy task for the machine because it requires a deep and abstract comprehension of the **attribute** "closed" which is highly relevant to the **object** it is describing. For Instance, for a cabinet with a large, square, blue door, and a wardrobe with a little, circular, red door, the "closed" **attribute** can be presented in very different ways. So the machine must acquire the crucial and interpretable idea of the **attribute** to do the Classification.

Attributes are interpretable, visual, significant properties of objects, such as color, texture, light, and so on. "Closed" in the example is a typical **attribute**. In **Compositional-zero-shot** setting, the machine should first learn from samples that all primitives are included, and then predict the Attribute and the Object of the given sample(for example, an image, or a piece of video).

The main differences between **Compositional-zero-shot learning** and typical **Object Classification** lie mainly in two points. Firstly, attributes and objects can compose a vastly large quantity number of classes, while the samples that can be found concentrate on a little number of classes of those. But vastly many compositions are legal and may appear in certain circumstances, and this may not be included in the training data. Secondly, the presentations of the attributes can largely differ due to the contexts, while typical modules of **Object Classification** can hardly capture such information. Those two points lead to great difficulties compared to **Object Classification.**

Figure 1: Examples of Attribute-Object Composition

*:These authors are the joint first authors of this paper, having contributed jointly to the research work presented in this paper.

One approach to address those two difficulties is SymNet[1] which is based on the observation of **Symmetry**: adding an attribute to an object that has already had the attribute would result in the object with the attribute. The advantage of this approach is its ability to separate abstract concepts from concrete concepts and learn the deeper semantics of **Attributes**. It enables it to reach the SOTA results in many datasets.

But it still has its shortcomings. The macro design of SymNet is good, but in terms of **micro implementation**, the most suitable module for the nature of zero-shot is not selected. Due to the incredible zero-shot performance of **CLIP[3]** in zero-shot setting, we propose **CymNet(C**LIP boosted S**ymNet** for Compositional Zero-shot Learning) in which, we introduce CLIP into SymNet, replacing the text-encoder and the image-encoder with the CLIP's text-encoder and image-encoder, and add CLIP-adapter to the network. **CymNet** results in better performance compared to **SymNet** and is more efficient because it can converge faster.

II. Related Work

A. Compositional Zero-Shot Learning

In the CZSL setting, the prediction dataset will contain some compositions that were unseen in the training dataset, but ensure that all attributes and objects appear enough times to learn. The model needs to learn the training dataset to predict all the samples' composition in the prediction dataset. Therefore, it is a cross-task that combined the ideas of Zero-Shot-Learning and Compositions. There have been already many approaches that considered different properties of the task. Misra et.al[2] train two SVMs respectively for objects and attributes and then use a multi-linear transformation network to learn the function that combines the two primitives. Nagarajan[4] et.al propose treating attributes as linear transformations for object embedding, and then projecting the resulting image features and transformed object embeddings into a common latent space.

Although much research has been conducted in the area, there is still a long way to go on the CZSL task to completely address the problem, because the best result that can be achieved so far is much lower than the upper bound that can be expected.

B. CLIP

CLIP[3] is a strong neural network-based language model pretrained by contrastive pre-training in more than one hundred million samples, which achieves the SOTA results in tasks from a wide range of areas, such as language translation, language summarization, image captioning, and many other aspects. One of the key innovations of CLIP is its ability to learn to perform multiple tasks concurrently, without the need for task-specific training data. This is known as zero-shot learning, and it allows CLIP to adapt to new tasks and languages quickly and perform well without the need for additional training.

Inspired by the extraordinary zero-shot ability of CLIP and the zero-shot nature of CZSL, we introduce CLIP into SymNet by substituting the text-encoder and image-encoder of the network, expecting to reach better results.

C. SymNet

SymNet[1] is based on the observation of symmetry in the Attribute-Object Composition. Symmetry is a property that: removing some attribute from the Object without the attribute would result in the object without the Attribute while removing some attribute from the object with the attribute would result in the object without the attribute. According to Symmetry combined with the group theory, SymNet is proposed. SymNet consists of two neural networks CoN and DeCoN, respectively learn the effects of adding and removing attributes using the test dataset. In the prediction stage, SymNet calculates the relative moving distances, which are the distance between the object and the object after adding the attribute and the distance between the object after removing the attribute and compares the distances to determine whether the object has the attribute.

Due to its deep insight into the CZSL task, SymNet shows great competence in separating attributes from the objects and therefore achieves good results in many tasks.

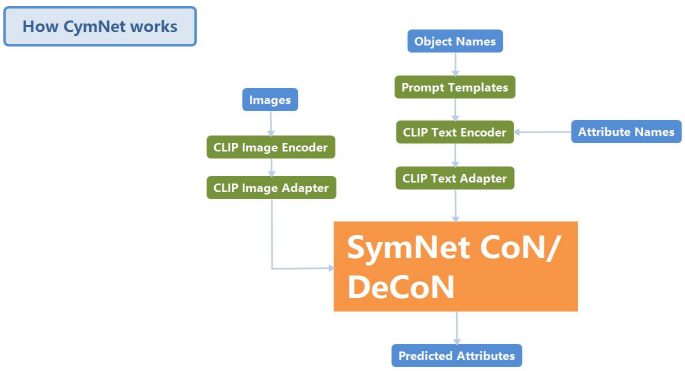
However, SymNet still has its shortcomings. The potential of the SymNet model is not fully exploited, we introduce CLIP to strengthen the model's compatibility with the Zero-shot nature of the task, and also surprisingly find that the speed of convergence is improved.

D. CLIP-Adapter

CLIP-Adapter[5] is a method to add a trainable module to zero-shot CLIP. Designing good prompts requires time and hard work and CLIP-Adapter helps CLIP adapts to new environments without prompt tuning.

The CLIP-Adapter uses a lightweight bottleneck architecture to prevent the potential overfitting issue of few-shot learning by reducing the number of parameters. CLIP Adapter only adds two additional linear layers after the last layer of the visual or language backbone, whereas the original adapter module is inserted into all layers of the language backbone. In addition, the CLIP Adapter mixes the original zero-shot visual or language embeddings with the corresponding network-adjusted features through residual connections. Through this "residual-style mixing," the CLIP Adapter can utilize both the stored knowledge from the original CLIP and the newly learned knowledge from the few-shot training samples.

III. METHODOLOGY



A. Replace the Text Encoder with the Text Transformer

The original SymNet uses one-hot encoding as its presentation of word vectors, which is easy to implement but lacks the internal information of the text. To solve this problem, we use CLIP's
text encoder to encode the object and attribute names. Also, to properly use CLIP's text encoder, you need to provide prompts rather than a single word to it to enhance its performance. We adopt CLIP's prompts designed for Imagenet to generate prompt sentences. Then these prompt sentences' encoding is averaged to get the proper encoding of a certain word.

B. Replace the Image Encoder with Vision Transformer

Vision Transformer has achieved great success in large-scale visual data processing. The original SytmNet uses ResNet to extract image features, we replace it with CLIP's image processor, more specifically, ViT/L14. We didn't use the largest model ViT/L14@336px due to the constraints of hardware, but it's believed that you will get better performance by adopting larger model.

C. Add CLIP-Adapter to both Image Processor and Text Processor

We add CLIP-Adapter to both image processor and text processor. The learnable linear layers and parameters help CLIP encoder adapt to this task. CLIP-Adapter can be represented easily by

$$A(f) = ReLU(W_1(ReLU(W_2 f))$$

where W_1, W_2 are fully connected linear layers.

D. Implementation Details

For two datasets, text features and image features are both generated from ViT/L14 CLIP. The CLIP-Adapter layer for the text features transforms the text features from 300-dimensional to 75- dimensional in the first linear layer and the features are sent into ReLU activation function. After that, the features are fed into a second linear layer which turns them from 75-dimensional to 00-dimensional. Finally, the features are processed by the second ReLU activation function

IV. EXPERIMENT

A. Dataset

UT-Zappos50K[6] is a dataset containing 50025 images of shoes with shoe attribute labels. We follow the train and test settings of the original SymNet paper, using 24898 images and 83 object-attribute pairs as the train set, and 4228 images and 33 pairs as the test set. The pairs the in train set will not occur in the test set.

B. Training Details

We train CymNet with Adam and SGD optimizer on a single NVIDIA RTX 3060 GPU. For UT-Zappos, the best model is trained with SGD optimizer, leaning rate 1e-4, and batch size 256 for 30 epochs. The loss weights are $\lambda_1 = 0.05, \lambda_2 = 0.03, \lambda_3 = 1, \lambda_4 = 0.5, \lambda_5 = 0.5$. Notably, we mainly adopt the parameters from the SymNet paper and we don't use cross-validation on it. The computational resource to train CymNet is rather small, you can get the performance in the Results part with 10 minutes' training on NVIDIA RTX 3060 GPU.

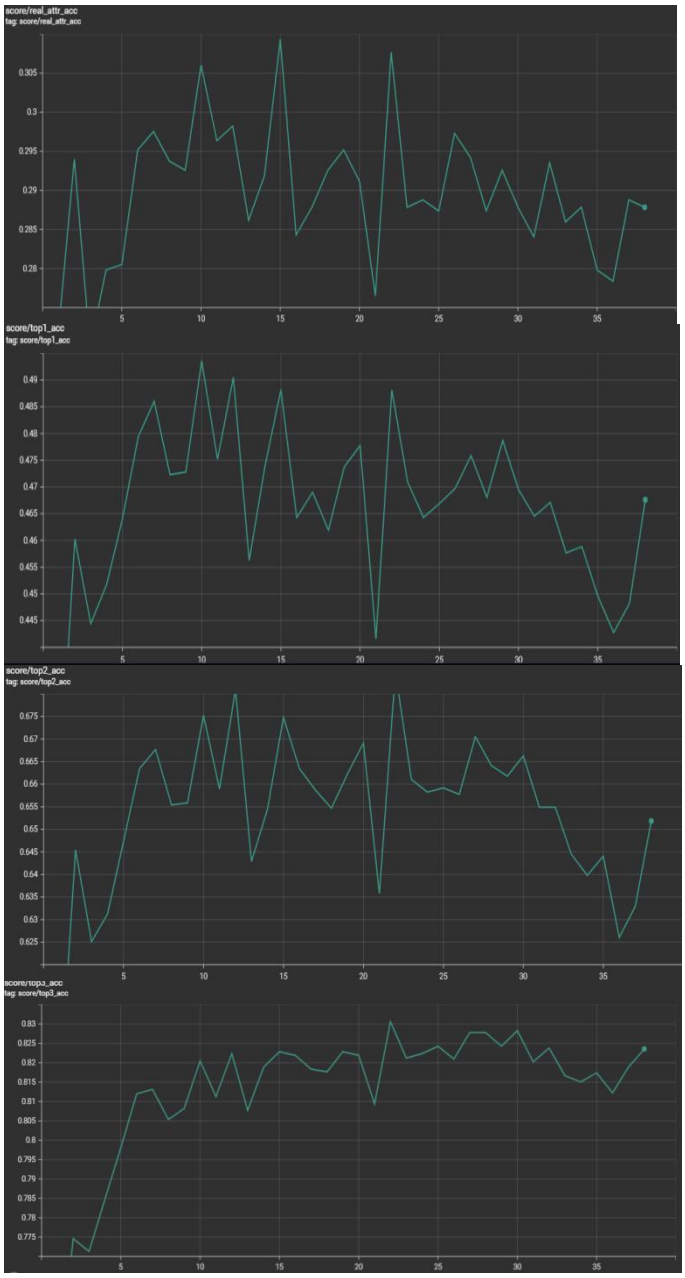
C. Results

We compare CymNet and the original SymNet to show that our modification to this model has enhanced the performance.

We test our CymNet on UT-Zappos50K, the same dataset that SymNet has tested on, and we use the Top-1, 2, 3 accuracies as the evaluation metrics, which is also the same as SymNet.The results are shown in the table.

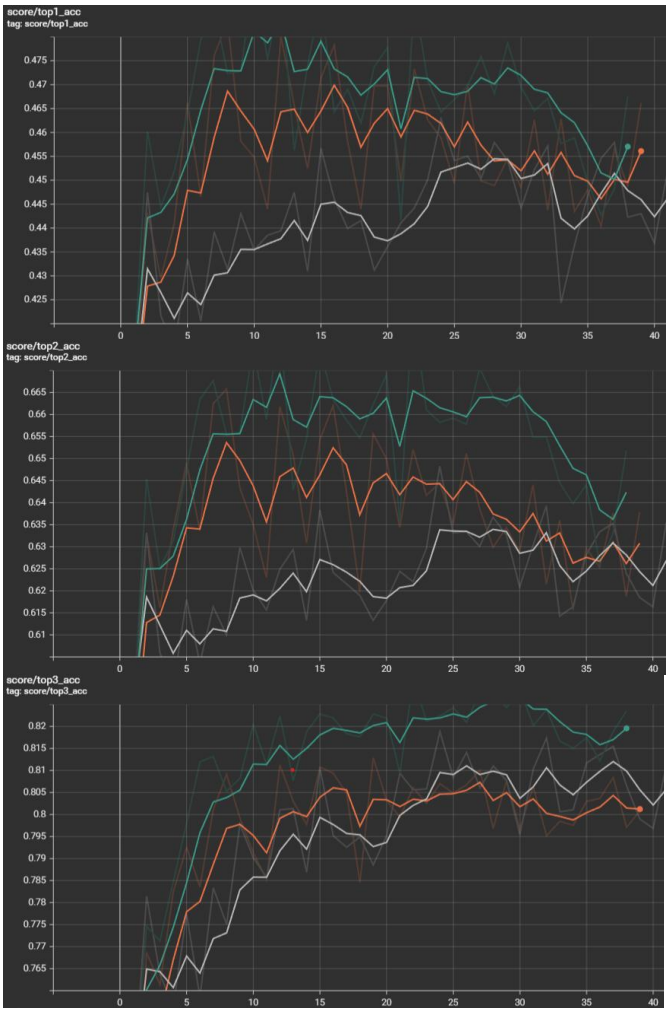
Method	**UT-Zappos**		
	Top-1	***Top-2***	***Top-3***
SymNet	52.1	67.8	76.0
CymNet/SGD(Ours)	51.8	70.9	83.0
CymNet/Adam(Ours)	49.4	67.5	82.1

The convergence of CymNet on UT-Zappos datasets using Adam optimizer is shown below, the x-axis is the epoch number:



We also compare CymNet's text encoder performances on different prompts. We test the performance on the UT-Zappos dataset using the same hyperparameters but different prompts.

The orange line indicates the performance with no prompt engineering, just a single word to encode. The grey line indicates the performance with the prompt "a photo of a {}". The green line indicates the performance with prompts that the original CLIP designed for ImageNet dataset. Note that the model in the following chart is trained with different parameters with the best performance parameter sets, they are designed to show the gaps between different prompts.



D. Performance Analysis

We have made some progress in the UT-Zappos dataset. Compared to SymNet with the computational resources, we just train our model with 40 epochs, and SymNet is trained with 600 epochs. CymNet performs better than SymNet with less training time, the results show the potential of the model.

Prompts do matter in text feature generation through CLIP text encoder, but sometimes a single handmade prompt is useless in some datasets. CymNet gains great enhancement in performance by adding plenty of prompts. But a single prompt "a photo of a {}" seems not to work on the UT-Zappos dataset. The UT-Zappos dataset is composed of clear images consisting of a single shoe in each one, its simplicity may account for the low performance of a single prompt.

V. CONCLUSION

Inspired by the zero-shot nature of **CZSL**, we propose a SymNet-based model **CymNet** which is constructed by replacing SymNet's text-encoder and image-encoder with CLIP's text-encoder and image-encoder. **CymNet** improves the performance of the original modelvy a remarkable degree and can converge in a short time which confirms our conjecture that introducing parts that are compatible with

zero-shot setting into SymNet would improve its ability to learn deeper semantics of attributes.

A. Authors and Affiliations

[1] Yong-Lu Li, Yue Xu, Xiaohan Mao, Cewu Lu; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11316-11325

[2] Ishan Misra, Abhinav Gupta, and Martial Hebert. 2017. From red wine to red tomato: Composition with context. In CVPRI. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International Conference on Machine Learning. PMLR, 2021.

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

[5] Gao, Peng, et al. "Clip-adapter: Better vision-language models with feature adapters." arXiv preprint arXiv:2110.04544 (2021).

[6] Aron Yu and Kristen Grauman. "Semantic Jitter: Dense Supervision for Visual Comparisons via Synthetic Images". In ICCV, 2017.

AUTHORS' BACKGROUND

Your Name	Title*	Research Field	Personal website
单佳铭	master student	Interpretable Machine Learning	shanjiaming.github.io
王照梓	master student	Online Algorithm, Theory Computer Science	/
翟明舒	master student	Computer vision	/