Prompting Language-Informed Distribution for Compositional Zero-Shot Learning (Supplementary Material)

Wentao Bao[1], Lichang Chen[2], Heng Huang[2], and Yu Kong[1]

[1] Department of Computer Science and Engineering, Michigan State University
[2] Department of Computer Science, University of Maryland
{baowenta,yukong}@msu.edu, {bobchen,heng}@umd.edu

A Broader Impact and Limitations

Broader Impact. The method in is work can be broadly extended to more other multi-modality applications, such as general zero-shot learning, cross-modality compositional retrieval and generation, etc. Besides, the central idea of LLM-based modality alignment is not limited to text and image, but any modality that could reveal the semantic categories in practice is promising to explore in the future. The potential negative societal impact is that, the developers should be cautious by carefully examining the societal biases indicated by the generated class descriptions, though the LLMs we used are publicly accessible.
Limitations. One limitation is that the primitive decomposition could be difficult to learn when the states are non-visual concepts like `smelly`, `hot`, etc., even by the pre-trained CLIP model. Another limitation is that the generated descriptions by LLMs are not grounded to the image such that some distraction from generated descriptions could be introduced.

B Generating Compositional Class Descriptions

In this work, we choose T5-base, OPT-1.3B, GPT-3.5, and Mistral-7B models as the LLMs for compositional class description generation. For the T5 model, we follow the same setting as [1] that uses the T5-base model for word-to-sentence generation. The T5-base model was pre-trained on the Colossal Clean Crawled Corpus dataset [6] and finetuned on the CommonGen dataset [2]. Take the `painted ceiling` as an example, the results from T5-base model are:

```
- A very old but beautifully decorated ceiling.
- A remodeled interior with a painted ceiling.
- A painted ceiling at a restaurant.
- Stained glass windows and a carved pattern on the ceiling.
- Painted ceilings and a fireplace.
- This apartment has a painted ceiling.
- A painted ceiling was a huge hit.
- A chandelier is painted in the middle of the ceiling.
```

```
- A stained glass window in a bathroom with a white painted
    ceiling.
- The ceiling of a hotel is painted.
- ...
```

For the OPT model, we adopt the pre-trained OPT-1.3B model which is small but still could generate sentences of sufficiently good quality. Considering that it is a decoder-only LLM that relies on a few examples as the prompt for word-to-sentence generation, we design the prompt template with 4 examples as shown below.

```
Keywords: sliced, potato, picture
Output: The picture features a beautifully arranged plate of
    thinly sliced potatoes.
###
Keywords: red, apple, photo
Output: In the photo, a bright red apple is the central focus,
    captured in stunning detail.
###
Keywords: leather, shoes, image
Output: The image showcases a sleek pair of leather shoes,
    meticulously designed to impress.
###
Keywords: broken, car, photo
Output: The photo captures the aftermath of a car accident with a
    broken vehicle lying on the road.
###
```

where the `Keywords` is followed by the words of the state, object, and the word randomly picked from the set {`photo`, `image`, `picture`}. Using the same example *painted ceiling* as T5-base model, the generated sentences are:

```
- The painting of the ceiling features an intricate pattern of
    intricate gold-colored paint and is framed by a white
    background.
- The ceiling has been painted with the pictures of these three
    characters, all arranged together.
- In the picture, the ceiling is covered in bright, colorful
    paintwork that has been painted on by hands painted white. The
    colors have been selected carefully.
- In the picture, the ceiling features painted decoration. The
    decoration resembles the surface of the sea, and has been
    painted in shades of blue.
- The photograph captures both the bright colors of the painting
    atop the ceiling and the subtle shades of light reflecting off
    of it.
- The large picture shows a large pattern painted onto the
    ceiling. The blue line shows paint dripping down.
- The wall behind the picture shows three different painted
    ceilings, in bright contrasting colors. A vibrant sky and blue
    skies are depicted against the dark brick wall.
```

```
- The ceiling of the room depicted in the painting could very well
    be painted in a few hours. The details of each object are
    clearly defined in its placement and position.
- Another photo of the same scene, this time featuring a ceiling
    painted in a stunning, white color.
- A painted ceiling is shown, painted according to a specific
    design. this is a typical design that can also include
    decorative or functional elements.
- ...
```

It is clear that the generated class descriptions are much more diverse and informative than those of the OPT model.

C Covariance Sharing

For the CZSL task, the spatial complexity of computing the covariance matrix $\boldsymbol{\Sigma}_{1:C}$ is $O(|C^{(s)}|^2 d)$ which could be too heavy to compute if the number of the compositions is too large. For example, the C-GQA dataset contains 278K seen compositions which result in around 6×10^{13} floating elements of $\boldsymbol{\Sigma}_{1:C}$ for 768-dim text features. To handle this issue, we instead implement the $\boldsymbol{\Sigma}_{1:C}$ by sharing the covariance across attributes given the same object. This implies that the model is encouraged to learn the object-level distributions.

Specifically, similar to the VLPD module of the main paper, we compute the mean $\boldsymbol{\mu}_{1:|\mathcal{O}|}$ and covariance $\boldsymbol{\Sigma}_{1:|\mathcal{O}|}$ over the objects by grouping $\mathbf{t}_y$ and $\mathbf{D}^{(y)}$ with object labels:

$$\mathbf{t}_o = \frac{1}{|\mathcal{Y}_o|} \sum_{y \in \mathcal{Y}_o} \mathbf{t}_y, \quad \mathbf{D}^{(o)} = \frac{1}{|\mathcal{Y}_o|} \sum_{y \in \mathcal{Y}_o} \mathbf{D}^{(y)}, \tag{1}$$

where $\mathcal{Y}_o$ is the subset of compositions in $\mathcal{Y}$ that contains the same object as y. Then, all the pairwise margins $\mathbf{H}_o^{(m)} \in \mathbb{R}^{|\mathcal{O}| \times |\mathcal{O}|}$ in object space can be mapped back to $\mathbf{H}^{(m)} \in \mathbb{R}^{C \times C}$ in a compositional space by sharing it with all compositions in $\mathcal{Y}_o$. This could significantly reduce the computation load of the covariance while compromising the accuracy of distribution modeling.

Since the distribution modeling for both our $\mathbb{PLID}$ and ProDA is not applicable to the C-GQA dataset, we use the MIT States dataset to show the negative impact of sharing the covariance (see Tab. 1). It shows that the covariance sharing can significantly save the GPU memory (17.6 vs 32.5 GB), while still performing much better than ProDA.

D Primitive-level Gaussian Modeling

To formulate the Gaussian distributions over the state classes and the object classes, we group the text embeddings of composition descriptions $\mathbf{D}$ by Eq. (1), resulting in the distribution support points (DSP) $\mathbf{t}_o + \mathbf{D}^{(o)}$ and $\mathbf{t}_s + \mathbf{D}^{(s)}$ for a given object class o and state class s, respectively. The DSPs are assumed

Variants	Mem.(GB)	H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
ProDA [4]	32.5	32.71	16.11	17.30	5.11
PLID (w. ShareCov)	**17.6**	38.50 (-0.47%)	21.69 (-0.43%)	19.81 (-0.60%)	7.04 (-0.30%)
PLID (full)	22.2	**38.97**	**22.12**	**20.41**	**7.34**

Table 1: Effect of covariance sharing on MIT-States dataset. All methods use the same batch size of 64 for a fair comparison of GPU memory.

Hyperparameters	MiT-States	UT-Zappos	C-GQA
max epochs	20	25	20
base learning rate	0.00005	0.0001	0.00001
weight decay	0.00002	0.00001	0.00001
number of text descriptions	64	32	64
number of image views	8	8	8
attention dropout	0.5	0.1	0.1
weights of primitive loss	0.1	0.01	0.01

Table 2: Hyperparameters of model implementation.

to follow the state distribution $\mathcal{N}(\mathbf{t}_s, \boldsymbol{\Sigma}_s)$ or the object distribution $\mathcal{N}(\mathbf{t}_o, \boldsymbol{\Sigma}_o)$, where the covariances $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_o$ are determined by $\mathbf{D}^{(s)}$ and $\mathbf{D}^{(o)}$, respectively.

Eventually, given the decomposed state visual features $f_s(\mathbf{v})$ and object visual features $f_o(\mathbf{v})$, the logit margin terms are defined as

$$h_{k,s}^{(m)} = f_s(\mathbf{v})^\top \mathbf{A}_{k,s} f_s(\mathbf{v}), \quad \text{and} \quad h_{k,o}^{(m)} = f_o(\mathbf{v})^\top \mathbf{A}_{k,o} f_o(\mathbf{v}), \tag{2}$$

where the index k ranges within $[1, |\mathcal{S}|]$ for computing the state classification loss $\mathcal{L}_s$, and ranges within $[1, |\mathcal{O}|]$ for computing the object classification loss $\mathcal{L}_o$, respectively.

E More Implementation Details and Results

Implementation. The training hyperparameters of our final model on each dataset are listed in Tab. 2.

More Ablation Analysis. In Table 3, we show more ablation study results on the design choices of our model. The first is to answer: *Should we learn both the compositional and primitive feature space?* This is interesting because if the primitive space can be learned by the proposed VLPD, intuitively the original compositional space is redundant. In the first line of Table 3, we show that if we remove the compositional space but only learn primitive space to recompose, the performance experiences a large drop in all metrics. This can be explained by the intuition that, without a direct compositional recognition, the merits of *explicitly* learned separatability and *implicitly* learned compositionality will be totally lost. These are the keys to the success of the pioneering CZSL method CSP [5].

model variants		H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
recompose only		30.02	13.88	15.46	4.35
w/o soft prompt		38.57	21.67	20.00	7.17
3-layers FE	TFE only	36.89	19.93	18.77	6.42
	VFE only	36.55	19.80	19.06	6.51
	TFE+VFE	37.46	20.65	19.15	6.70
full model		38.97	22.12	20.41	7.34

Table 3: More ablation study results.

Besides, in Table 3 line 2, we investigate whether the soft prompt is still useful or not based on our model, though it has been validated in prior CZSL literature [3]. It shows that without the soft prompt, the performance decreases but not too much. However, it is still necessary as it drives the LLM text distributions to align with visual features in training.

Lastly, in Table 3 lines 3-5, we further analyze the impact of TFE and VFE modules if they are implemented with the three-layer cross-attention Transformers. The two modules still show contributions to the performance gain. Moreover, compared to the default one-layer setting, using more Transformer layers does not improve the performance, even performing worse.

References

1. He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., Bai, S., Qi, X.: Is synthetic data from generative models ready for image recognition? In: ICLR (2023)
2. Lin, B.Y., Zhou, W., Shen, M., Zhou, P., Bhagavatula, C., Choi, Y., Ren, X.: CommonGen: A constrained text generation challenge for generative commonsense reasoning. In: EMNLP (2020)
3. Lu, X., Liu, Z., Guo, S., Guo, J.: Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In: CVPR (2023)
4. Lu, Y., Liu, J., Zhang, Y., Liu, Y., Tian, X.: Prompt distribution learning. In: CVPR (2022)
5. Nayak, N.V., Yu, P., Bach, S.H.: Learning to compose soft prompts for compositional zero-shot learning. In: ICLR (2023)
6. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR **21**(1), 5485–5551 (2020)