Prompting Language-Informed Distribution for Compositional Zero-Shot Learning

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

Abstract. Compositional zero-shot learning (CZSL) task aims to recognize unseen compositional visual concepts, *e.g.*, `sliced tomatoes`, where the model is learned only from the seen compositions, *e.g.*, `sliced potatoes` and `red tomatoes`. Thanks to the prompt tuning on large pre-trained visual language models such as CLIP, recent literature shows impressively better CZSL performance than traditional vision-based methods. However, the key aspects that impact the generalization to unseen compositions, including the *diversity* and *informativeness* of class context, and the *entanglement* between visual primitives, *i.e.*, state and object, are not properly addressed in existing CLIP-based CZSL literature. In this paper, we propose a model by prompting the language-informed distribution, aka., $\mathbb{PLID}$, for the CZSL task. Specifically, the $\mathbb{PLID}$ leverages pre-trained large language models (LLM) to (*i*) formulate the language-informed class distributions which are diverse and informative, and (*ii*) enhance the compositionality of the class embedding. Moreover, a visual-language primitive decomposition (VLPD) module is proposed to dynamically fuse the classification decisions from the compositional and the primitive space. Orthogonal to the existing literature of soft, hard, or distributional prompts, our method advocates prompting the LLM-supported class distributions, leading to a better zero-shot generalization. Experimental results on MIT-States, UT-Zappos, and C-GQA datasets show the superior performance of the $\mathbb{PLID}$ to the prior arts. Our code and models are released: `https://github.com/Cogito2012/PLID`.

Keywords: Compositional Zero-shot Learning · CLIP · LLM

1 Introduction

Compositional visual recognition is a fundamental characteristic of human intelligence [13] but it is challenging for modern deep learning systems. For example, humans can easily recognize unseen `sliced tomatoes` after seeing `sliced potatoes` and `red tomatoes`. Such a compositional zero-shot learning (CZSL) capability is valuable in that, novel visual concepts from a huge combinatorial semantic space could be recognized without "seeing" any of their training data. For example, the C-GQA [29] dataset contains 413 states and 674 objects. This implies a total of at least 278K compositional classes in an open world while only

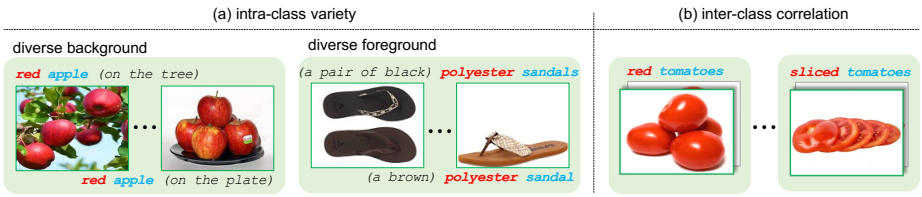


Fig. 1: Challenges of compositional recognition. (a) images of the same compositional class appear differently due to diverse visual backgrounds or foregrounds. **(b)** `red tomatoes` and `sliced tomatoes` are visually correlated because 1) both are `tomatoes` object, and 2) the object `tomatoes` is inherently entangled with the state `red`, resulting in the need of primitive decomposition.

2% of them are accessible in training. Therefore, CZSL can significantly reduce the need for large-scale training data.

Traditional vision-based methods either directly learn the visual feature of compositions, or try to first decompose the visual data into representations of simple primitives, *i.e.*, states and objects, and then learn to re-compose the compositions [1,7,10,15,25,28,29,39,49,53]. Thanks to the recent large pre-trained vision-language models (VLM) such as CLIP [35], state-of-the-art CZSL methods have been developed [6,22,31,44]. For instance, CSP [31] inherits the hard prompt template of the CLIP, *i.e.*, *a photo of* [`state`][`object`] where only the embeddings of the states and objects are trained. The following methods [6, 22, 44] use soft prompt introduced in CoOp [52], where the embeddings of the prompt template are jointly optimized, leading to a better CZSL performance. The impressive performance of CLIP-based CZSL methods benefits from the sufficiently good feature alignment between the image and text modalities, and the prompting techniques for adapting the aligned features to recognizing compositional classes.

Despite the success of existing CLIP-based methods, we find several key considerations to prompt the pre-trained CLIP for better CZSL modeling. First, the *diversity* and *informativeness* of prompts are both important to distinguish between compositional classes. CZSL can be treated as zero-shot learning on fine-grained categories, which requires a fine-grained context to prompt the CLIP model [23, 35]. However, to contextualize a class with fine granularity, the hard prompt in [35] suffers from the heuristic design of prompt templates, and a single prompt for each class lacks diversity to capture the intra-class variance of visual data (Fig. 1a). Though the ProDA [23] proposes to learn a collection of prompts that formulate class-specific distribution to address the diversity, the lack of *language informativeness* in their prompts limits their performance on fine-grained compositional categories. Second, the entanglement between visual primitives, *e.g.* `red` and `tomatoes` in Fig. 1b, incurs difficulty in learning decomposable visual representations that are useful for compositional generalization [10, 20], while such a capability is missing in [31, 44]. Though the more recent work [6, 22] learn to decompose the primitives and considers the

re-composed compositional predictions, their language-only decomposition and probability-level mixup potentially limit the generalizability in the open-world.

In this paper, we propose a novel CLIP-based method for the CZSL task by prompting the language-informed distributions ($\mathbb{PLID}$) over both the compositional and primitive categories. To learn the diverse and informative textual class representations, the $\mathbb{PLID}$ leverages off-the-shelf large language models (LLM) to build the class-specific distributions and to enhance the class embeddings. Furthermore, we propose a visual language primitive decomposition (VLPD) module to decompose the image data into simple primitives for recognition of state and objects. Eventually, the compositional classification is performed by fusing the decisions from both the compositional and primitive spaces. The proposed $\mathbb{PLID}$ shows state-of-the-art performance on CZSL benchmarks such as MIT-States [8], UT-Zappos [46], and C-GQA [29].

Note that our method is orthogonal to the existing hard prompt [35], soft prompt tuning [52], and prompt distribution learning [4, 12, 19, 23]. We advocate prompting the distribution of informative LLM-based class descriptions. From a classification perspective, this is grounded on the classification-by-description [5, 26, 27, 45], that LLM-generated text enables more informative class representations. Compared to the deterministic soft or hard prompt aforementioned, our distribution modeling could capture the intra-class diversity and inter-class correlation for better zero-shot generalization. Compared to the existing prompt distribution learning approaches, the class context is more linguistically interpretable and provides fine-grained descriptive information about the class. Our method is also parameter-efficient without the need to optimize a large collection of prompts. Specific to the CZSL task, the enhanced class embeddings by LLM descriptions enable visual language primitive decomposition and decision fusion in both compositional and primitive space, which eventually benefits the generalization to the unseen.

In summary, the contributions are as follows. (*i*) We develop a $\mathbb{PLID}$ method that advocates prompting the language-informed distribution for compositional zero-shot learning, which is orthogonal to existing soft or hard prompting and distributional prompt learning. (*ii*) We propose primitive decomposition with stochastic logit mixup to fuse the classification decision from compositional and primitive predictions. (*iii*) We empirically show that $\mathbb{PLID}$ could achieve superior performance to prior arts in both the closed-world and open-world settings on MIT-States, UT-Zappos, and C-GQA datasets.

2 Related Work

Prompt Learning in VLM. Vision-Language Models (VLM) such as the CLIP [35] pre-trained on web-scale datasets recently gained substantial attention for their strong zero-shot recognition capability on various downstream tasks. Such a capability is typically achieved by performing prompt engineering to adapt pre-trained VLMs. Early prompting technique such as the hard prompt in CLIP uses the heuristic template "*a photo of* `[CLS]`" as the textual input. Recently,

the soft prompt tuning method in CoOp [52], CoCoOp [51], and ResPT [38] that uses learnable embedding as the textual context of class names significantly improved the model adaptation performance. This technique is further utilized in MaPLe [11] that enables multi-modal prompt learning for both image and text. However, the prompts of these methods are deterministic and lack the diversity to capture the appearance variety in fine-grained visual data, so they are prone to overfitting the training data. To handle this issue, ProDA [23] explicitly introduces a collection of soft prompts to construct the class-specific Gaussian distribution, which results in better zero-shot performance and inspires the recent success of PPL [12] in the dense prediction task. Similarly, the PBPrompt [19] uses neural networks to predict the class-specific prompt distribution and utilizes optimal transport to align the stochastically sampled soft prompts and image patch tokens. The recent work [4] assumes the latent embedding of prompt input follows a Gaussian prior and adopts variational inference to learn the latent distribution. In this paper, in order to take the merits of the *informativeness* of hard prompt and the *diversity* of distributional modeling, we adopt the soft prompt to adapt the distributions supported by LLM-generated class descriptions.

Compositional Zero-Shot Learning (CZSL). For a long period, the CZSL task has been studied from a vision-based perspective in literature. They either directly learn the compositional visual features or disentangle the visual features into simple primitives, *i.e.*, states and objects. For example, [16,29,30] performs a direct classification by projecting the compositional visual features into a common feature space, and [1,7,10,20,21,28,53] decompose the visual feature into simple primitives so that the compositional recognition can be achieved by learning to recompose from the primitives. Though the recent large-scale pre-trained CLIP model shows impressive zero-shot capability, it is found to struggle to work well for compositional reasoning [14,24,47]. Thanks to the recent prompt learning [52], the CZSL task has been dominated by CLIP-based approaches [6,17,22,31,44,50]. The common idea is to prompt the frozen CLIP model to separately learn the textual embeddings of simple primitives, which empirically show strong compositionality for zero-shot generalization. Different to [17,50] that develop primitive adapters and [6,22,44] that use learnable prompts for deterministic vision-language alignment, our method takes the benefit of learnable prompt and LLM-generated text for distributional alignment, addressing the importance of diversity and informativeness for zero-shot generalization.

3 Preliminaries

CZSL Task Formulation. The CZSL task aims to recognize images of a compositional category $y \in \mathcal{C}$, where the semantic space $\mathcal{C}$ is a Cartesian product between the state space $\mathcal{S} = \{s_1, \ldots, s_{|\mathcal{S}|}\}$ and object space $\mathcal{O} = \{o_1, \ldots, o_{|\mathcal{O}|}\}$, *i.e.*, $\mathcal{C} = \mathcal{S} \times \mathcal{O}$. For example, as shown in Fig. 1, a model trained on images of `red apple` and `sliced tomatoes` needs to additionally recognize an image of `sliced apple`. In training, only a set of **seen** compositions is available. In closed-world testing, the model needs to recognize images from both the **seen**

compositions in $\mathcal{C}^{(s)}$ and the **unseen** compositions in $\mathcal{C}^{(u)}$ that are assumed to be feasible, where the cardinality $|\mathcal{C}^{(s)} \cup \mathcal{C}^{(u)}| \ll |\mathcal{C}|$ since most of the compositions in $\mathcal{C}$ are practically not feasible. In open-world testing, the model needs to recognize images given any composition in $\mathcal{C}$.

VLMs for CZSL. Large pre-trained VLMs such as CLIP [35] have recently been utilized by CSP [31] for the CZSL task. The core idea of CSP is to represent the text embeddings of states in $\mathcal{S}$ and objects in $\mathcal{O}$ as learnable parameters and contextualize them with the hard prompt template "*a photo of* [s][o]" as the input of the CLIP text encoder, where [s] $\in \mathcal{S}$ and [o] $\in \mathcal{O}$. Given an image $\mathbf{x}$, by using the cosine similarity ($\texttt{cos}$) as the logit, the class probability of the composition y is defined as $p_{\boldsymbol{\theta}}(y|\mathbf{x}) = \texttt{softmax}(\texttt{cos}(\mathbf{v}, \mathbf{t}_y))$, where $\boldsymbol{\theta}$ are the $|\mathcal{S}| + |\mathcal{O}|$ learnable parameters, $\mathbf{v}$ and $\mathbf{t}_y$ are the image feature and class text embedding, respectively.

In training, the prediction $p_{\boldsymbol{\theta}}(\hat{y}|\mathbf{x})$ is supervised by multi-class cross-entropy loss. In CZSL testing, a test image is recognized by finding the compositional class $c \in \mathcal{C}$ which has the maximum $\texttt{cos}(\mathbf{v}, \mathbf{t}_c)$. The CSP method is simple, parameters efficient, and it largely outperforms traditional approaches. However, due to the lack of diversity and informativeness in prompting, the zero-shot capability of CLIP is not fully exploited by CSP for the CZSL task.

4 Proposed Method

Overview. Figure 2 shows an overview of the $\mathbb{PLID}$. The basic idea is to use LLMs to generate sentence-level descriptions for each compositional class and learn to prompt the class-wise text distributions (supported by the descriptions) to be aligned with image data. Besides, we introduce visual language primitive decomposition (VLPD) and stochastic logit mixup (SLM) to enable recognition at both compositional and primitive levels. In testing, an image is recognized by fusing the decisions from the directly predicted and the recomposed compositions.

4.1 Prompting Language-Informed Distribution

Motivation. To adapt the large pre-trained CLIP [35] to downstream tasks, recent distributional prompt learning [4, 12, 19, 23] shows the importance of *context diversity* by distribution modeling for strong generalization. Motivated by the inherent fine-granularity of compositional recognition in the CZSL task, we argue that not only the context diversity but also the *context informativeness* by language modeling, are both important factors to adapt CLIP to the zero-shot learning task. The insight behind this is that the sentence-level descriptions could contextualize compositional classes in a more fine-grained manner than the prior arts. Therefore, we propose to address the two factors by learning to **P**rompt the **L**anguage-**I**nformed **D**istributions ($\mathbb{PLID}$) for the CZSL task.

Compositional Class Description. To generate diverse and informative text descriptions for each compositional class, we adopt a similar way as [27] by prompting an LLM that shows instruction-following capability. An example below shows the format of the LLM instruction.

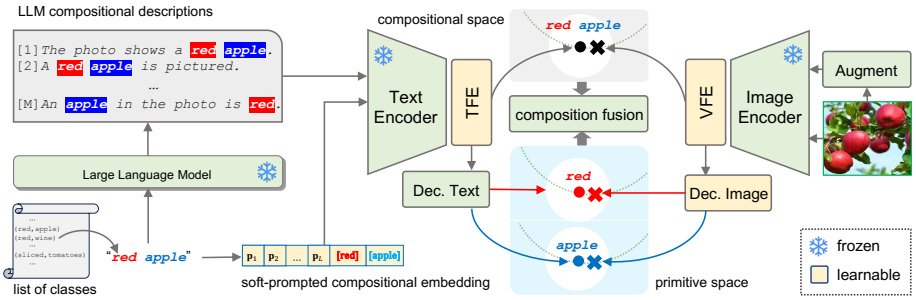


Fig. 2: Overview of $\mathbb{PLID}$. The model is developed for the CZSL task by aligning the semantics of image $\mathbf{x}$ (*e.g.*, image on the right) and compositional class $y = (s, o)$ (*e.g.*, "red apple") via a frozen CLIP [35]. It constructs language-informed text distributions in both compositional and primitive (attribute and object) spaces (middle part) by soft prompting and LLM-generated class descriptions (left part). The features of the image and text are enhanced by text and visual feature enhancement (TFE and VFE). Eventually, the compositional decisions from the two spaces are fused as the prediction.

```
Keywords: sliced, potato, picture
Output: The picture features a beautifully arranged plate of thinly
    sliced potatoes.
###
```

See the Supplement B for more details. For each composition $y = (s, o)$, we generate M descriptions denoted as $S^{(y)} = \{S_1^{(y)}, \ldots, S_M^{(y)}\}$ where $S_m^{(y)}$ is a linguistically complete sentence. Different to [27] that aims to interpret the zero-shot recognition by attribute phrases from LLMs, we utilize the LLM-based sentence-level descriptions in the CZSL task for two benefits: (i) provide diverse and informative textual context for modeling the class distributions, and (ii) enhance the class embedding with fine-grained descriptive information.

Language-Informed Distribution (LID). For both the image and text modalities, we use the frozen CLIP model and learnable feature enhancement modules to represent the visual and language features, which are also adopted in existing CZSL literature [6, 22].

Specifically, for the text modality, each composition y is tokenized and embedded by CLIP embedding layer and further prompted by concatenating with learnable context vectors, *i.e.*, "$[\mathbf{p}_1] \ldots [\mathbf{p}_L][\mathbf{s}][\mathbf{o}]$", where $\mathbf{p}_{1:L}$ is initialized by "`a photo of`" and shared with all classes. Followed by the frozen CLIP text encoder $\mathcal{E}_T$, the embedding of class y is $\mathbf{q}_y = \mathcal{E}_T([\mathbf{p}_1] \ldots [\mathbf{p}_L][\mathbf{s}][\mathbf{o}])$ where $\mathbf{q}_y \in \mathbb{R}^d$. Following the CZSL literature [22, 44], here the soft prompt $\mathbf{p}_{1:L}$ and primitive embeddings $[\mathbf{s}][\mathbf{o}]$ are learnable while $\mathcal{E}_T$ is frozen in training.

To simultaneously address the lack of diversity and informativeness of the soft prompts, we propose to formulate the class-specific distributions supported by the texts $S^{(y)}$ and learn to prompt these distributions. Specifically, we encode $S^{(y)}$ by the frozen CLIP text encoder: $\mathbf{D}^{(y)} = \mathcal{E}_T(S^{(y)})$, where $\mathbf{D}^{(y)} \in \mathbb{R}^{M \times d}$. Then, we use $\mathbf{D}^{(y)}$ to enhance $\mathbf{q}_y$ by $\mathbf{t}_y = \Psi_{\text{TFE}}(\mathbf{q}_y, \mathbf{D}^{(y)})$ where Ψ_{TFE} is the text

feature enhancement (**TFE**) implemented by a single-layer cross attention Transformer [41]. Similarly, given an image $\mathbf{x}$, to mitigate the loss of fine-grained cues, we augment it with N views to be $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$. Followed by the frozen CLIP visual encoder $\mathcal{E}_V$, the feature of $\mathbf{x}$ is enhanced by $\mathbf{v} = \Psi_{\text{VFE}}(\mathcal{E}_V(\mathbf{x}), \mathcal{E}_V(\mathbf{X}))$ where Ψ_{VFE} is the visual feature enhancement (**VFE**) by cross attention [41], implemented with the same structure as TFE for simplicity.

We treat the enhanced text feature $\mathbf{t}_y$ of class y as the class mean and $\mathbf{t}_y + \mathbf{D}^{(y)}$ as the distribution support points (**DSP**) that follow the Gaussian $\mathcal{N}(\mathbf{t}_y, \boldsymbol{\Sigma}_y)$ where $\boldsymbol{\Sigma}_y$ is the text variance of the class y. The motivation of $\mathbf{t}_y + \mathbf{D}^{(y)}$ is to enable the flexibility of DSP to traverse around in the d dimensional space in training since $\mathbf{t}_y$ is trainable while $\mathbf{D}^{(y)}$ are pre-trained. For all $|\mathcal{C}^{(s)}|$ (denoted as C) seen compositional classes, we build joint Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_{1:C}, \boldsymbol{\Sigma}_{1:C})$ similar to ProDA [23], where the means $\boldsymbol{\mu}_{1:C} \in \mathbb{R}^{C \times d}$ are given by $\mathbf{t}_y$ over C classes, and the covariance $\boldsymbol{\Sigma}_{1:C} \in \mathbb{R}^{d \times C \times C}$ is defined across C classes for each feature dimension from DSP.

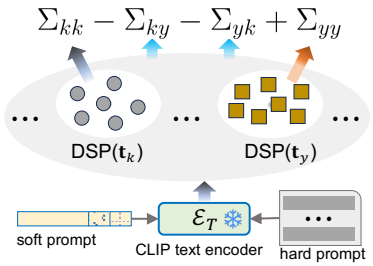


Fig. 3: Prompting for intra- and inter-class covariance optimization.

Discussions. Compared to the ProDA [23] that learns a collection of non-informative prompts, our DSPs are language-informed by $\mathbf{D}^{(y)}$ that provides more fine-grained descriptive information to help recognition and decomposition. Besides, our method is more parameter-efficient than ProDA since we only have a single soft prompt to learn. This is especially important for the CZSL task where there is a huge number of compositional classes. Lastly, we highlight the benefit of performing the intra- and inter-class covariance optimization induced by the learning objective of distribution modeling, which will be introduced below.

Learning Objective. Given the visual feature $\mathbf{v} \in \mathbb{R}^d$ of image $\mathbf{x}$ and the text embeddings $\mathbf{t}_{1:C}$ from class-wise joint distributions $\mathcal{N}(\boldsymbol{\mu}_{1:C}, \boldsymbol{\Sigma}_{1:C})$, minimizing the cross-entropy loss is equivalent to minimizing the upper bound of negative log-likelihood (NLL):

$$-\log \mathbb{E}_{\mathbf{t}_{1:C}} p(y|\mathbf{v}, \mathbf{t}_{1:C}) \leq -\log \frac{\exp(h_y/\tau)}{\sum_{k=1}^{C} \exp((h_k + h_{k,y}^{(m)})/\tau)} := \mathcal{L}_y(\mathbf{x}, y), \quad (1)$$

where the compositional logit $h_y = \texttt{cos}(\mathbf{v}, \mathbf{t}_y)$, the pairwise margin $h_{k,y}^{(m)} = \mathbf{v}^\top \mathbf{A}_{k,y} \mathbf{v}/(2\tau)$ and $\mathbf{A} \in \mathbb{R}^{d \times C \times C}$ is given by $\mathbf{A}_{k,y} = \boldsymbol{\Sigma}_{kk} + \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{ky} - \boldsymbol{\Sigma}_{yk}$. The covariance $\mathbf{A}_{k,y}$ indicates the correlation between the k-th out of C classes and the target class y on each of d feature dimensions. The insight of minimizing $\mathcal{L}_y(\mathbf{x}, y)$ is illustrated in Fig. 3, which encourages minimizing intra-class variance by $\boldsymbol{\Sigma}_{yy}$ and $\boldsymbol{\Sigma}_{kk}$, and maximizing inter-class separability indicated by $\boldsymbol{\Sigma}_{ky}$ and $\boldsymbol{\Sigma}_{yk}$. In Supplement C, we discuss the case when C is too large to compute $\mathbf{A}$, our workaround by covariance sharing within each object group leads to negligible performance decrease.

4.2 Primitives Decomposition for Fused Recognition

Motivation. Considering the fundamental challenge in the CZSL task, that the visual primitives are inherently entangled in an image, an unseen composition in testing can hardly be identified if its object (or its state) embedding is overfitted to the visual data of seen compositions. To this end, it is better to inherit the benefits of the decompose-recompose paradigm [10, 20, 53] by decomposing visual features into simple primitives, *i.e.*, states and objects, from which the recomposed decision can be leveraged for zero-shot recognition. Thanks to the compositionality of CLIP [40, 43], such motivation can be achieved by the visual-language primitive decomposition (**VLPD**). See Fig. 4 and we explain it below. Based on VLPD, we propose the stochastic logit mixup to fuse the directly learned compositions and the recomposed ones.

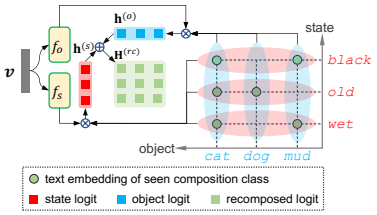


Fig. 4: VLPD for recomposing.

VLPD. Specifically, we use two parallel neural networks f_s and f_o to decompose $\mathbf{v}$ into the state visual feature $f_s(\mathbf{v})$ and object visual feature $f_o(\mathbf{v})$, respectively. To get the primitive-level supervisions, given the training compositions $\mathcal{C}^{(s)}$ (see the circle dots in Fig. 4), we group their enhanced embeddings $\{\mathbf{t}_y\}$ over the subset $\mathcal{Y}_o$, in which all compositions share the same given object o (see vertical ellipses in Fig. 4), and group $\{\mathbf{t}_y\}$ over the subset $\mathcal{Y}_s$, in which all compositions share the same given state s (see horizontal ellipses in Fig. 4). Thus, given a state s and an object o, the predicted object logit h_s and state logit h_o are computed by

$$h_s = \texttt{cos}\left(f_s(\mathbf{v}), \frac{1}{|\mathcal{Y}_s|}\sum_{y\in\mathcal{Y}_s}\mathbf{t}_y\right), \quad h_o = \texttt{cos}\left(f_o(\mathbf{v}), \frac{1}{|\mathcal{Y}_o|}\sum_{y\in\mathcal{Y}_o}\mathbf{t}_y\right). \tag{2}$$

Different from DFSP [22] that only decomposes text features, we additionally use f_s and f_o to decompose visual features $\mathbf{v}$ and empirically show the superiority of performing both visual and language decomposition (see Tab. 6).

Following the spirit of distribution modeling, we also introduce the distributions over state and object categories, where the corresponding DSP, denoted as $\mathbf{D}^{(s)}$ and $\mathbf{D}^{(o)}$, are obtained by grouping $\mathbf{D}^{(y)}$ over $\mathcal{Y}_s$ and $\mathcal{Y}_o$, respectively. This leads to the following upper-bounded cross-entropy losses:

$$\begin{aligned}\mathcal{L}_s(x,s) &= -\log\frac{\exp(h_s/\tau)}{\sum_{k=1}^{|\mathcal{S}|}\exp((h_k+h_{k,s}^{(m)})/\tau)},\\ \mathcal{L}_o(x,o) &= -\log\frac{\exp(h_o/\tau)}{\sum_{k=1}^{|\mathcal{O}|}\exp((h_k+h_{k,o}^{(m)})/\tau)},\end{aligned} \tag{3}$$

where $h_{k,s}^{(m)}$ and $h_{k,o}^{(m)}$ are determined the same way as $h_{k,y}^{(m)}$ in Eq. (1). See details in Supplement D. In this way, the merits of language-informed distribution

modeling, *i.e.*, the inter- and intra-class covariance optimization constraints, can be introduced into primitive space for fused recognition as introduced below.

Composition Fusion. With the individually supervised f_s and f_o, we have $p(y|\mathbf{v}) = p(s|\mathbf{v}) \cdot p(o|\mathbf{v})$ according to conditional independence, that induces $p(y|\mathbf{v}) \propto \exp((h_s + h_o)/\tau)$. Therefore, the recomposed logit matrix $\mathbf{H}^{(rc)} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{O}|}$ is a Cartesian (element-wise combinatorial) sum between $\mathbf{h}^{(s)} \in \mathbb{R}^{|\mathcal{S}|}$ and $\mathbf{h}^{(o)} \in \mathbb{R}^{|\mathcal{O}|}$, *i.e.*, $\mathbf{H}^{(rc)} = \mathbf{h}^{(s)} \oplus \mathbf{h}^{(o)\top}$, where $\mathbf{h}^{(s)}$ contains all state logits and $\mathbf{h}^{(o)}$ contains all object logits. See the red and blue squares in Fig. 4.

Given the recomposed logit $h_y^{(rc)} \in \mathbf{H}^{(rc)}$ and the directly learned compositional logit h_y by Eq. (1), we propose to stochastic fusion method in training by sampling a coefficient λ from a Beta prior distribution:

$$\tilde{h}_y = (1-\lambda)h_y + \lambda h_y^{(rc)}, \quad \lambda \sim \text{Beta}(a, b), \tag{4}$$

where (a, b) are hyperparameters indicating the prior preference for each decision. In training, we replace the h_y and h_k of Eq. (1) with the mixed logit $\tilde{h}_y$ and $\tilde{h}_k$, respectively. In testing, no stochasticity is needed so we use the Beta expectation of λ which is $a/(a+b)$ to get the fused logit $\tilde{h}_y$.

The insights behind the stochasticity are that the Beta distribution indicates a prior preference to h_y or $h_y^{(rc)}$. It provides the flexibility of which compositional decision to trust in, and the stochasticity of the coefficient λ inherently introduces a regularization effect in training [3]. Moreover, compared to softmax probability mixup [6], our logit mixup avoids the limitation of softmax normalization over a huge number of compositional classes, that rich information of class relationship is lost after softmax normalization according to [2]. Such class relationships are even more important in the CZSL problem as indicated in [29].

5 Experiments

Datasets. We perform experiments on three CZSL datasets, *i.e.*, MIT-States [8], UT-Zappos [46], and C-GQA [29], following the standard splitting protocols in CZSL literature [22, 31, 34]. MIT-States consists of 115 states and 245 objects, with 53,753 images in total. Following [22, 31, 34], it is split into 1,262 seen and 300/400 unseen compositions for training and validation/testing, respectively. UT-Zappos contains 16 states and 12 objects for 50,025 images in total, and it is split into 83 seen and 15/18 unseen compositions for training and validation/testing. C-GQA contains 453 states and 870 objects for 39,298 images, and it is split into 5,592 seen and 1,040/923 unseen compositions for training and validation/testing, respectively, resulting in 7,555 and 278,362 target compositions in closed- and open-world settings.

Evaluation. We report the metrics in both closed-world (**CW**) and open-world (**OW**) settings, including the best seen accuracy (**S**), the best unseen accuracy (**U**), the best harmonic mean (**H**) between the seen and unseen accuracy, and the area under the curve (**AUC**) of unseen versus seen accuracy. For OW evaluation, following the CSP [31], we adopt the feasibility calibration by GloVe [33] to filter out infeasible compositions.

Table 1: CZSL results of Closed- and Open-World settings on three datasets. Baseline results are from published literature except for ProDA. Note that "–" indicates no results reported by the PCVL paper or not applicable by ProDA for more than 278K compositional classes on the C-GQA dataset.

	Method	*MIT-States*				*UT-Zappos*				*C-GQA*			
		S	U	H	AUC	S	U	H	AUC	S	U	H	AUC
Closed	CLIP [35]	30.2	46.0	26.1	11.0	15.8	49.1	15.6	5.0	7.5	25.0	8.6	1.4
	CoOp [52]	34.4	47.6	29.8	13.5	52.1	49.3	34.6	18.8	20.5	26.8	17.1	4.4
	ProDA[1] [23]	37.4	51.7	32.7	16.1	63.7	60.7	47.6	32.7	–	–	–	–
	CSP [31]	46.6	49.9	36.3	19.4	64.2	66.2	46.6	33.0	28.8	26.8	20.5	6.2
	PCVL [44]	48.5	47.2	35.3	18.3	64.4	64.0	46.1	32.2	–	–	–	–
	HPL [42]	47.5	50.6	37.3	20.2	63.0	68.8	48.2	35.0	30.8	28.4	22.4	7.2
	DFSP [22]	46.9	52.0	37.3	20.6	66.7	**71.7**	47.2	36.0	38.2	32.0	27.1	10.5
	𝕡𝕃𝕀𝔻	**49.7**	**52.4**	**39.0**	**22.1**	**67.3**	68.8	**52.4**	**38.7**	**38.8**	**33.0**	**27.9**	**11.0**
Open	CLIP [35]	30.1	14.3	12.8	3.0	15.7	20.6	11.2	2.2	7.5	4.6	4.0	0.3
	CoOp [52]	34.6	9.3	12.3	2.8	52.1	31.5	28.9	13.2	21.0	4.6	5.5	0.7
	ProDA[1] [23]	37.5	18.3	17.3	5.1	63.9	34.6	34.3	18.4	–	–	–	–
	CSP [31]	46.3	15.7	17.4	5.7	64.1	44.1	38.9	22.7	28.7	5.2	6.9	1.2
	PCVL [44]	48.5	16.0	17.7	6.1	64.6	44.0	37.1	21.6	–	–	–	–
	HPL [42]	46.4	**18.9**	19.8	6.9	63.4	48.1	40.2	24.6	30.1	5.8	7.5	1.4
	DFSP [22]	47.5	18.5	19.3	6.8	66.8	**60.0**	44.0	30.3	38.3	7.2	10.4	2.4
	𝕡𝕃𝕀𝔻	**49.1**	18.7	**20.4**	**7.3**	**67.6**	55.5	**46.6**	**30.8**	**39.1**	**7.5**	**10.6**	**2.5**

Implementation Details. We implement the 𝕡𝕃𝕀𝔻 based on the CSP codebase in PyTorch. The CLIP architecture ViT-L/14 is used by default. On the MIT-States, we generate $M = 64$ texts and augment an image with $N = 8$ views, and adopt Beta$(1, 9)$ as prior. The dropout rates of cross-attention layers in TFE and VFE are set to 0.5, and the dropout rate to 0.3 for the learnable state and object embeddings. For the soft prompt embeddings, we set the context length of the text encoder to 8 for all datasets. Following [22], we use Adam optimizer with base learning rate 5e-5 and weight decay 2e-5, and step-wise decay it with the factor of 0.5 every 5 training epochs for a total of 20 epochs. Complete training hyperparameters on three datasets are in the Supplement E.

5.1 Main Results

The results are reported in Tab. 1. We compare with the CZSL baselines that are developed on the same frozen CLIP model. The table shows that under both the closed-world and open-world test settings, our proposed 𝕡𝕃𝕀𝔻 method achieves the best performance in most metrics on the three datasets. Note that ProDA [23] also formulates the class-wise Gaussian distributions to address the intra-class diversity, but it can only outperform CLIP and CoOp on all metrics.

[1] ProDA is re-implemented for the CZSL setting. Limited by the GPU memory, ProDA is not applicable to the C-GQA dataset which consists of more than 278K compositional classes.

Table 2: Ablation study. (a): the baseline that uses mean pooling of text embeddings from T5-generated sentences. (b): add language-informed distribution (LID). (c): use text and visual feature enhancement module (FE). (d): change the LLM from T5-base to the OPT-1.3B. (e): apply primitive decomposition for fused decision (PDF).

	LID	FE	OPT	PDF	H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
(a)					35.41	18.56	17.37	5.56
(b)	✓				37.06	20.43	18.65	6.50
(c)	✓	✓			37.87	21.09	19.70	6.95
(d)	✓	✓	✓		38.80	21.67	19.61	7.01
(e)	✓	✓	✓	✓	**38.97**	**22.12**	**20.41**	**7.34**

This indicates the importance of both diversity and informativeness for the CZSL task. On the UT-Zappos dataset, the $\mathbb{PLID}$ outperforms the DFSP in terms of S, H, and AUC by 0.6%, 5.2%, and 2.7% respectively, while inferior to the DFSP on the best unseen metric. The potential reason is that DFSP fuses the text features into the image images, which better preserves the generalizability of CLIP for the small downstream UT-Zappos dataset. Note that the HPL method uses prompt learning and recognition at both compositional and primitive levels, but it performs only slightly better than CSP and way worse than our method, indicating that traditional prompt learning helps but is not enough to adapt the CLIP model to the CZSL task.

5.2 Model Analysis

To comprehensively analyze the proposed $\mathbb{PLID}$, we perform extensive ablation study and design analysis on the middle-sized MIT-States dataset in this section. More ablation results are provided in the Supplement E.

Major Components. In Tab. 2, we show the contribution of the major components in the $\mathbb{PLID}$ model. It is clear that they are all beneficial. We highlight some important observations: (1) The LID method in row (b) significantly improves the performance compared to the baseline (a) that does not formulate Gaussian distribution in training, and they are much better than ProDA (20.43% vs 16.1% of AUC_{cw}) when referring to Tab. 1. This implies that addressing the context **diversity** by modeling the Gaussian distribution like the ProDA is not sufficient, but context **informativeness** is critical and preferred for the CZSL task. (2) Rows (c)(d) show that feature enhancement (FE) and the better LLM OPT-1.3B can also bring performance gains. (3) Rows (e) show that the primitive decomposition for fused decision (PDF) could further improve the CZSL performance in both closed- and open-world settings. In the following paragraphs, we further validate the effect or design choices of these components in detail.

Effect of LID. In Tab. 3, we investigate at which semantic level the language-informed distribution (LID) should be applied. Denote the Gaussian distribution on state, object, and composition as $\mathcal{N}_s$, $\mathcal{N}_o$, and $\mathcal{N}_y$, respectively. The Tab. 3 results clearly show the superiority of applying LID on all three semantic levels.

Table 3: Effect of LID on states ($\mathcal{N}_s$), objects ($\mathcal{N}_o$), and compositions ($\mathcal{N}_y$). The first row indicates the model without LID.

$\mathcal{N}_s$	$\mathcal{N}_o$	$\mathcal{N}_y$	H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
			38.44	21.67	19.53	6.99
✓	✓		38.30	21.62	19.49	6.95
		✓	38.49	21.90	19.93	7.20
✓	✓	✓	**38.97**	**22.12**	**20.41**	**7.34**

Table 4: Effect of LLMs. Note that GPT-3.5 is not open-sourced so that we use its API call to get text descriptions.

LLMs	H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
Mistral-7B	37.22	20.78	19.22	6.74
GPT-3.5	37.38	20.61	19.38	6.80
T5-base	38.41	21.53	**20.46**	7.34
OPT-1.3B	**38.97**	**22.12**	20.41	**7.34**

Table 5: Design choices of feature enhancement (FE). We explore the use of text or visual feature enhancement (TFE/VFE) and the number of their cross-attention layers.

TFE	VFE	layers	H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
✓		1	37.89	21.07	19.37	6.78
	✓	1	37.48	21.04	19.43	6.72
✓	✓	3	37.46	20.65	19.15	6.70
✓	✓	1	**38.97**	**22.12**	**20.41**	**7.34**

Table 6: Effect of VLPD and fusion strategies. We explore the modalities (text or image) of the decomposition, and whether deterministic (det.) or stochastic (stoc.) compositional fusion.

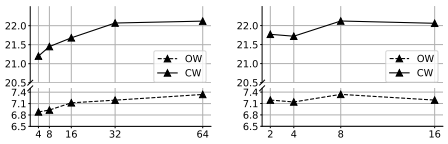
VLPD		fusion		H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
text	image	det.	stoc.				
			✓	37.94	20.98	19.67	6.98
✓			✓	38.40	21.31	19.99	7.13
✓	✓			38.42	21.69	20.24	7.31
✓	✓	✓		38.67	21.90	19.99	7.15
✓	✓		✓	**38.97**	**22.12**	**20.41**	**7.34**

This indicates the generality of LID towards many potential zero-shot or open-vocabulary recognition problems.

Effect of LLM. In Tab. 4, we analyze the choice of LLMs by comparing $\mathbb{PLID}$ variants using different LLMs, including the T5-base [36], OPT-1.3B [48], GPT-3.5 [32], and Mistral-7B [9]. It shows the performance varies across different LLMs. Note that the capacity of GPT-3.5 and Mistral-7B on general language processing tasks is much better than T5-base and OPT-1.3B. However, we do not see improvements by using these generally larger and better LLMs, but a small OPT-1.3B is sufficient to achieve the best performance. We provide some examples of the generated texts by these LLMs in Supplement B.

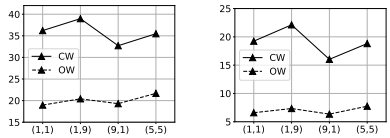
TFE and VFE. In Tab. 5, we explore the design choices of the text and visual feature enhancement (TFE and VFE) modules. The results show that using one layer of randomly initialized cross-attention for both TFE and VFE performs the best. Using more cross-attention layers will cause a significant performance drop (see the 3rd row). We attribute the cause to the over-fitting issue when more learnable parameters are introduced to aggregate text descriptions or visual features.

VLPD and Fusion. In Tab. 6, we validate the design choices of visual language primitive decomposition (VLPD) and the stochastic compositional fusion. Compared with the results of the first two rows, it shows clear advantages of primitive decomposition over both image and text modalities. Note that DFSP [22] also has primitive decomposition but only on text modality. Our better performance than DFSP and the results in Tab. 6 thus tell the need for decomposition on both visual and image. Besides, to validate our stochastic



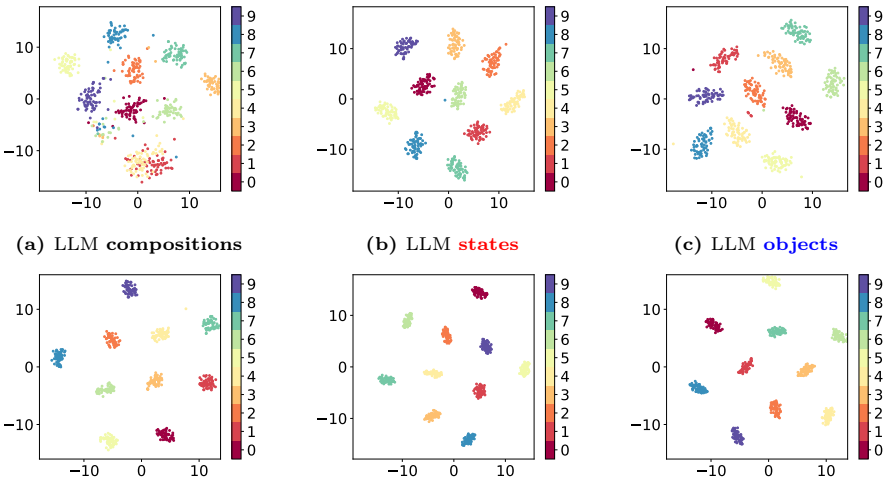
(a) AUC vs. M (b) AUC vs. N

Fig. 5: Impact of M and N. We set $N = 8$ for the Fig. 5a, while we set $M = 64$ for the Fig. 5b.



(a) best HM (b) best AUC

Fig. 6: Impact of (a, b). Here $(1, 1)$ implies random sampling while $(5, 5)$ implies equally trusted.



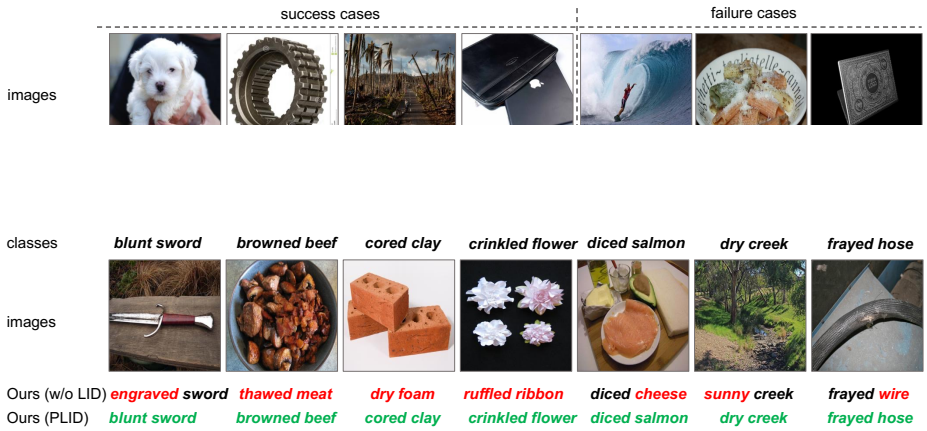
(a) LLM **compositions** (b) LLM **states** (c) LLM **objects**

(d) Learned **composition** DSP (e) Learned **state** DSP (f) Learned **object** DSP

Fig. 7: tSNE visualization of the text embeddings with (the 2nd row) and without (the 1st row) learnable distribution modeling over compositions (the 1st column), states (the 2nd column), and objects (the 3rd column). This figure clearly shows that our method achieves good performance by distribution modeling.

compositional fusion, we compare it with the model without fusion in the 3rd row and the model with only deterministic fusion (weighted average without Beta sampling) in the 4th row. They also show the benefit of fusion with stochasticity.

Hyperparameters. In Fig. 5, we show the impact of the number of generated text descriptions M and the number of augmented image views N. It shows that the best performance is achieved when $M = 64$ and $N = 8$. We note that more augmented image views slightly decrease the performance, which could be attributed to the overfitting of the seen compositions. In Fig. 6, we show the impact of the Beta prior parameters (a, b). We set them to $(1, 1)$ for random sampling, $(1, 9)$ for preference to the composition, $(9, 1)$ for preference to recomposition, and $(5, 5)$ for equal preference, respectively. It reveals that trusting more of the directly learned composition by Beta$(1, 9)$ achieves the best results.



(b) Comparison with model without LID.

Fig. 8: Case studies. In Fig. 8a, we show the cases from MiT-States dataset that our method succeeds or fails. In Fig. 8b, we compare the proposed method with and without language-informed distribution (LID) modeling. Correct predictions are in **green** color, while incorrect predictions on state or object part are marked in **red**.

Class Distributions. We use the tSNE to visualize the generated text embeddings $\mathbf{D}$ and the learned DSP from or $\mathbb{PLID}$ model in Fig. 7, where the same set of 10 compositional (or state/object) classes are randomly selected from MIT-States dataset. It shows that by learning the distribution of each composition, state, and object from LLM-generated texts using Eq. (1) and (3) and TFE module, class embeddings can be distributed more compactly in each class (small intra-class variance), and better separated among multiple classes (large inter-class distance). This clearly shows why our proposed language-informed distribution modeling works in the CZSL task.

Case Study. In Fig. 8a, we show some success and failure cases of our $\mathbb{PLID}$ model. For example, the `heavy water` case indicates an incorrect label while $\mathbb{PLID}$ could correctly predict it as `huge wave`. This shows the robustness of $\mathbb{PLID}$ against noisy labels. The last two failure cases reveal $\mathbb{PLID}$ still could make mistakes on the state prediction (`cooked pasta`) and object prediction (`engraved floor`), which indicates there are still rooms for improvement. In Fig. 8b, we show that $\mathbb{PLID}$ could work much better than the model without LID. For example, the `sunny creek` and `frayed wire` are incorrect potentially due to the lack of handling (i) intra-class variety, as the `dry creek` images can be `sunny` and the `frayed hose` class could contain `wire` images, and (ii) inter-class correlation, as the `sunny` (or `wire`) is correlated to both the `dry creek` images (or `frayed hose` images) and other `sunny` images (or `wire` images).

6 Conclusion

In this work, we propose a novel CLIP-based compositional zero-shot learning (CZSL) method named $\mathbb{PLID}$. It leverages the generated text description of each class from large language models to formulate the class-specific Gaussian distributions. By softly prompting these language-informed distributions, $\mathbb{PLID}$ could achieve diversified and informative class embeddings for fine-grained compositional classes. Besides, we decompose the visual embeddings of image data into simple primitives that contain the basic states and objects, from which the re-composed predictions are derived to calibrate the prediction by our proposed stochastic logit mixup strategy. Experimental results show the superiority of the $\mathbb{PLID}$ method to prior arts on all common CZSL datasets.

Acknowledgements

WT Bao and Y Kong were partially supported by the Office of Naval Research (ONR) grant N00014-23-1-2046 and N00014-23-1-2417. LC Chen and H Huang were partially supported by NSF IIS 2347592, 2347604, 2348159, 2348169, DBI 2405416, CCF 2348306, CNS 2347617. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ONR or NSF.

References

1. Atzmon, Y., Kreuk, F., Shalit, U., Chechik, G.: A causal view of compositional zero-shot recognition. In: Adv. Neural Inform. Process. Syst. (2020)
2. Bang, D., Baek, K., Kim, J., Jeon, Y., Kim, J.H., Kim, J., Lee, J., Shim, H.: Logit mixing training for more reliable and accurate prediction. In: IJCAI (2022)
3. Carratino, L., CissÃ©, M., Jenatton, R., Vert, J.P.: On mixup regularization. JMLR **23**(325) (2022)
4. Derakhshani, M.M., Sanchez, E., Bulat, A., da Costa, V.G.T., Snoek, C.G.M., Tzimiropoulos, G., Martinez, B.: Bayesian prompt learning for image-language model generalization. In: ICCV (2023)
5. 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)
6. Huang, S., Gong, B., Feng, Y., Lv, Y., Wang, D.: Troika: Multi-path cross-modal traction for compositional zero-shot learning. In: CVPR (2024)
7. Huynh, D., Elhamifar, E.: Compositional zero-shot learning via fine-grained dense feature composition. In: Adv. Neural Inform. Process. Syst. (2020)
8. Isola, P., Lim, J.J., Adelson, E.H.: Discovering states and transformations in image collections. In: CVPR (2015)
9. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7b. arXiv preprint arXiv:2310.06825 (2023)
10. Karthik, S., Mancini, M., Akata, Z.: Kg-sp: Knowledge guided simple primitives for open world compositional zero-shot learning. In: CVPR (2022)

11. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: CVPR (2023)
12. Kwon, H., Song, T., Jeong, S., Kim, J., Jang, J., Sohn, K.: Probabilistic prompt learning for dense prediction. In: CVPR (2023)
13. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. Behavioral and brain sciences **40** (2017)
14. Lewis, M., Yu, Q., Merullo, J., Pavlick, E.: Does clip bind concepts? probing compositionality in large image models. arXiv preprint arXiv:2212.10537 (2022)
15. Li, X., Yang, X., Wei, K., Deng, C., Yang, M.: Siamese contrastive embedding network for compositional zero-shot learning. In: CVPR (2022)
16. Li, Y.L., Xu, Y., Mao, X., Lu, C.: Symmetry and group in attribute-object compositions. In: CVPR (2020)
17. Li, Y., Liu, Z., Chen, H., Yao, L.: Context-based and diversity-driven specificity in compositional zero-shot learning. In: CVPR (2024)
18. 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)
19. Liu, X., Wang, D., Li, M., Duan, Z., Xu, Y., Chen, B., Zhou, M.: Patch-token aligned bayesian prompt learning for vision-language models. arXiv preprint arXiv:2303.09100 (2023)
20. Liu, Z., Li, Y., Yao, L., Chang, X., Fang, W., Wu, X., Yang, Y.: Simple primitives with feasibility-and contextuality-dependence for open-world compositional zero-shot learning. arXiv preprint arXiv:2211.02895 (2022)
21. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: ECCV (2016)
22. Lu, X., Liu, Z., Guo, S., Guo, J.: Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In: CVPR (2023)
23. Lu, Y., Liu, J., Zhang, Y., Liu, Y., Tian, X.: Prompt distribution learning. In: CVPR (2022)
24. Ma, Z., Hong, J., Gul, M.O., Gandhi, M., Gao, I., Krishna, R.: Crepe: Can vision-language foundation models reason compositionally? In: CVPR (2023)
25. Mancini, M., Naeem, M.F., Xian, Y., Akata, Z.: Open world compositional zero-shot learning. In: CVPR (2021)
26. Maniparambil, M., Vorster, C., Molloy, D., Murphy, N., McGuinness, K., O'Connor, N.E.: Enhancing clip with gpt-4: Harnessing visual descriptions as prompts. arXiv preprint arXiv:2307.11661 (2023)
27. Menon, S., Vondrick, C.: Visual classification via description from large language models. In: ICLR (2023)
28. Misra, I., Gupta, A., Hebert, M.: From red wine to red tomato: Composition with context. In: CVPR (2017)
29. Naeem, M.F., Xian, Y., Tombari, F., Akata, Z.: Learning graph embeddings for compositional zero-shot learning. In: CVPR (2021)
30. Nagarajan, T., Grauman, K.: Attributes as operators: factorizing unseen attribute-object compositions. In: ECCV (2018)
31. Nayak, N.V., Yu, P., Bach, S.H.: Learning to compose soft prompts for compositional zero-shot learning. In: ICLR (2023)
32. OpenAI: OpenAI GPT-3.5 API [gpt-3.5-turbo-0125]. `https://openai.com/blog/chatgpt`, accessed: 2023
33. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP (2014)

34. Purushwalkam, S., Nickel, M., Gupta, A., Ranzato, M.: Task-driven modular networks for zero-shot compositional learning. In: ICCV (2019)
35. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
36. 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)
37. 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)
38. Razdaibiedina, A., Mao, Y., Hou, R., Khabsa, M., Lewis, M., Ba, J., Almahairi, A.: Residual prompt tuning: Improving prompt tuning with residual reparameterization. In: ACL (2023)
39. Tokmakov, P., Wang, Y.X., Hebert, M.: Learning compositional representations for few-shot recognition. In: ICCV (2019)
40. Trager, M., Perera, P., Zancato, L., Achille, A., Bhatia, P., Xiang, B., Soatto, S.: Linear spaces of meanings: the compositional language of vlms. arXiv preprint arXiv:2302.14383 (2023)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Adv. Neural Inform. Process. Syst. (2017)
42. Wang, H., Yang, M., Wei, K., Deng, C.: Hierarchical prompt learning for compositional zero-shot recognition. In: IJCAI (2023)
43. Wolff, M., Brendel, W., Wolff, S.: The independent compositional subspace hypothesis for the structure of clip's last layer. In: ICLR Workshop (2023)
44. Xu, G., Kordjamshidi, P., Chai, J.: Prompting large pre-trained vision-language models for compositional concept learning. arXiv preprint arXiv:2211.05077 (2022)
45. Yan, A., Wang, Y., Zhong, Y., Dong, C., He, Z., Lu, Y., Wang, W., Shang, J., McAuley, J.: Learning concise and descriptive attributes for visual recognition. arXiv preprint arXiv:2308.03685 (2023)
46. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: CVPR (2014)
47. Yuksekgonul, M., Bianchi, F., Kalluri, P., Jurafsky, D., Zou, J.: When and why vision-language models behave like bags-of-words, and what to do about it? In: ICLR (2023)
48. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., et al.: Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068 (2022)
49. Zhang, T., Liang, K., Du, R., Sun, X., Ma, Z., Guo, J.: Learning invariant visual representations for compositional zero-shot learning. In: ECCV (2022)
50. Zheng, Z., Zhu, H., Nevatia, R.: Caila: Concept-aware intra-layer adapters for compositional zero-shot learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1721–1731 (2024)
51. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: CVPR (2022)
52. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. IJCV (2022)
53. Zou, Y., Zhang, S., Chen, K., Tian, Y., Wang, Y., Moura, J.M.: Compositional few-shot recognition with primitive discovery and enhancing. In: ACM MM (2020)

Appendix

A Broader Impact and Limitations

Broader Impact. The method in this work can be broadly extended to more 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 [5] 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 [37] and finetuned on the CommonGen dataset [18]. 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.
- ...
```

Table 7: Effect of covariance sharing on MIT-States dataset. All methods use the same batch size of 64 for a fair comparison of GPU memory.

Variants	Mem.(GB)	H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
ProDA [23]	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**

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:

$$\mathbf{t}_o = \frac{1}{|\mathcal{Y}_o|} \sum_{y \in \mathcal{Y}_o} \mathbf{t}_y, \quad \mathbf{D}^{(o)} = \frac{1}{|\mathcal{Y}_o|} \sum_{y \in \mathcal{Y}_o} \mathbf{D}^{(y)}, \tag{5}$$

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 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. 7). 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. (5), 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

Table 8: Hyperparameters of model implementation.

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

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

$$h_{k,s}^{(m)} = f_s(\mathbf{v})^\top \mathbf{A}_{k,s} f_s(\mathbf{v}), \quad \text{and} \quad h_{k,o}^{(m)} = f_o(\mathbf{v})^\top \mathbf{A}_{k,o} f_o(\mathbf{v}), \tag{6}$$

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. 8.

More Ablation Analysis. In Table 9, 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 9, 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 [31].

Besides, in Table 9 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 [22]. 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 9 lines 3-5, we further analyze the impact of TFE and VFE modules if they are implemented with the three-layer cross-attention Transformers.

Table 9: More ablation study results.

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

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.