# LAMM: Label Alignment for Multi-Modal Prompt Learning

**Jingsheng Gao[1], Jiacheng Ruan[1], Suncheng Xiang[2], Zefang Yu[1]**
**Ke Ji[3], Mingye Xie[1], Ting Liu[1], Yuzhuo Fu[1]***
[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China
[2] School of Biomedical Engineering, Shanghai Jiao Tong University, China
[3] School of Computer Science and Engineering, Southeast University, China
{gaojingsheng, jackchenruan, xiangsuncheng17, yuzefang, xiemingye, louisa_liu, yzfu}@sjtu.edu.cn
keji@seu.edu.cn

## Abstract

With the success of pre-trained visual-language (VL) models such as CLIP in visual representation tasks, transferring pre-trained models to downstream tasks has become a crucial paradigm. Recently, the prompt tuning paradigm, which draws inspiration from natural language processing (NLP), has made significant progress in VL field. However, preceding methods mainly focus on constructing prompt templates for text and visual inputs, neglecting the gap in class label representations between the VL models and downstream tasks. To address this challenge, we introduce an innovative label alignment method named **LAMM**, which can dynamically adjust the category embeddings of downstream datasets through end-to-end training. Moreover, to achieve a more appropriate label distribution, we propose a hierarchical loss, encompassing the alignment of the parameter space, feature space, and logits space. We conduct experiments on 11 downstream vision datasets and demonstrate that our method significantly improves the performance of existing multi-modal prompt learning models in few-shot scenarios, exhibiting an average accuracy improvement of 2.31(%) compared to the state-of-the-art methods on 16 shots. Moreover, our methodology exhibits the preeminence in continual learning compared to other prompt tuning methods. Importantly, our method is synergistic with existing prompt tuning methods and can boost the performance on top of them. Our code and dataset will be publicly available at https://github.com/gaojingsheng/LAMM.

## Introduction

Building machines to comprehend multi-modal information in real-world environments is one of the primary goals of artificial intelligence, where vision and language are the two crucial modalities (Du et al. 2022). One effective implementation method is to pre-train a foundational vision-language (VL) model on a large-scale visual-text dataset and then transfer it to downstream application scenarios (Radford et al. 2021; Jia et al. 2021). Typically, VL models employ two separate encoders to encode image and text features, followed by the design of an appropriate loss function for training. However, finetuning on extensively trained models is costly and intricate, thus making the question of how to effectively transfer pre-trained VL models to downstream tasks a inspiring and valuable issue.

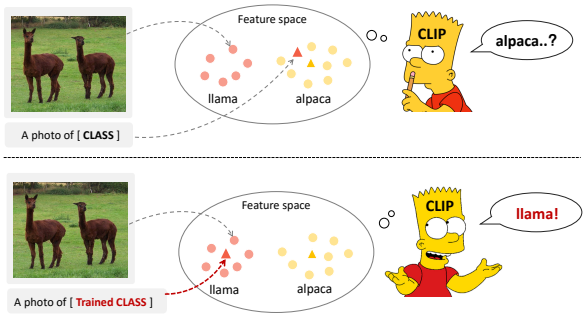


Figure 1: CLIP is more inclined to classify an image as belonging to a similar category. Altering the category text feature's position can enhance CLIP's recognition capabilities.

Prompt learning provides an effective solution to this problem, which provides downstream tasks with corresponding textual descriptions based on human prior knowledge and can effectively enhance the zero-shot and few-shot recognition capability of VL models. Through trainable templates with a small number of task-specific parameters, the process of constructing templates is further automated via gradient descent instead of manual constructions (Lester, Al-Rfou, and Constant 2021). Specifically, existing multi-modal prompt tuning methods (Zhou et al. 2022b,a; Khattak et al. 2022) use the frozen CLIP (Radford et al. 2021) model and design trainable prompts separately for the textual and visual encoders. These approaches ensure that VL models could be better transferred to downstream tasks without any changes to the VL model's parameters. However, their approach mainly focuses on the prompt template that is applicable to all categories, overlooking the feature representation of each category.

The <CLASS> token in the text template is crucial in classifying an image into the proper category. For example, as depicted in Figure 1, *llamas* and *alpacas* are two animals that resemble each other closely. In CLIP, there exists a propensity to misclassify a *llama* as an *alpaca* owing to the overrepresentation of *alpaca* data in the pre-training dataset.

*Corresponding Author.

By refining the text embedding position, CLIP can distinguish between these two species with trained feature space. Hence, identifying an optimal representation for each category in downstream tasks within the VL model is crucial. In the field of NLP, there exists the soft verbalizer (Cui et al. 2022), which enables the model to predict the representation of <MASK> in the text template to represent the category of the original sentence on its own. Unlike NLP, it is infeasible to task the text encoder of the VL model with predicting the image category directly. Nevertheless, we can optimize the category embeddings of various categories within the downstream datasets to increase the similarity between each image and its corresponding category description.

Consequently, we introduce a label alignment technique named LAMM, which automatically searches optimal <CLASS> embeddings through gradient optimization. To the best of our knowledge, the concept of trainable category token is first proposed in the pre-trained VL models. Simultaneously, to prevent the semantic features of the entire prompt template from deviating too far, we introduce a hierarchical loss during our training phase. The hierarchical loss facilitates alignment of category representations among parameter, feature and logits spaces. With these operations, the generalization ability of CLIP model can be preserved in LAMM, which makes LAMM better distinguish different categories in downstream tasks while preserving the semantics of the original category descriptions. Furthermore, given that LAMM solely fine-tunes the label embeddings within the downstream dataset, it doesn't encounter the issue of catastrophic forgetting typically encountered in conventional methods during continual learning.

We conduct experiments on 11 datasets, covering a range of downstream recognition scenarios. In terms of models, we test the vanilla CLIP, CoOp (Zhou et al. 2022b), and MaPLe (Khattak et al. 2022), which currently perform best in multi-modal prompt learning. Extensive experiments demonstrate the effectiveness of the proposed method within few-shot learning, illuminating its merits in both domain generalization and continual learning. Furthermore, our approach, being compatible to prevailing multi-modal prompt techniques, amplifies their efficacy across downstream datasets, ensuring consistent enhancement.

## Related Work

**Vision Language Models** In recent years, the development of Vision-Language Pre-Trained Models (VL-PTMs) has made tremendous progress, as evidenced by models such as CLIP (Radford et al. 2021), ALIGN (Jia et al. 2021), LiT (Zhai et al. 2022) and FILIP (Yao et al. 2022). These VL-PTMs are pre-trained on large-scale image-text corpora and learn universal cross-modal representations, which are beneficial for achieving strong performance in downstream VL tasks. For instance, CLIP are pre-trained on massive collections of image-caption pairs sourced from the internet, utilizing a contrastive loss that brings the representations of matching image-text pairs closer while pushing those of non-matching pairs further apart. After the pre-trained stage, CLIP has demonstrated exceptional performance on learning universal cross-modal representation in image-recognition (Gao et al. 2021), object detection (Zang et al. 2022), image segmentation (Li et al. 2022) and vision question answering (Sung, Cho, and Bansal 2022).

**Prompt Learning** Prompt learning adapt the pre-trained language models (PLMs) by designing a prompt template to leverage the power of PLMs to unprecedented heights, especially in few-shot settings (Liu et al. 2021). With the emergence of large-scale VL models, numerous researchers have attempted to integrate prompt learning into VL scenarios, resulting in the development of prompt paradigms that are better adapted for these scenarios. CoOp (Zhou et al. 2022b) first introduces the prompt tuning approach to VL models by learnable prompt template words. Co-CoOp (Zhou et al. 2022a) improves the performance on novel classes by incorporating instance-level image information. Unlike prompt engineering in text templates, VPT (Jia et al. 2022) inserts learnable parameters into the vision encoder. MaPLe (Khattak et al. 2022) appends a soft prompt to the hidden representations at each layer of the text and image encoders, resulting in a new solid performance in few-shot image recognition.

Previous multi-modal prompts have primarily focused on the engineering of prompt templates for textual and visual inputs while neglecting the significance of label representations in such templates. However, label verbalizers have already been proven to be effective in few-shot text classification (Schick and Schütze 2021), where verbalizers aim to reduce the gap between model outputs and label words. To alleviate the required expertise and workload for constructing a manual verbalizer, Gao et al. (Gao, Fisch, and Chen 2021) design search-based methods for better verbalizer choices during the training optimization process. Some other researches (Hambardzumyan, Khachatrian, and May 2021; Cui et al. 2022) propose trainable vectors as soft verbalizers to replace label words, which eliminates the difficulty of searching the entire dictionary.

Hence, we introduce trainable vectors to substitute label words in multi-modal prompts. Our approach aims to align label representations in the downstream datasets with pre-trained VL models, reducing the discrepancy in category descriptions between downstream datasets and VL-PTMs.

## Methodology

In this section, we will introduce how our proposed LAMM can be incorporated into CLIP seamlessly, accompanied by our hierarchical loss. The whole architecture of LAMM and hierarchical alignment is shown in Figure 2.

### Preliminaries of CLIP

CLIP is a VL-PTM comprising a vision encoder $\phi$ and a text encoder $\psi$. These two encoders extract image and text information, respectively, and map them to a common feature space $R^d$, where the two feature spaces align well.

Given an input image $x$, the image encoder will extract the corresponding image representation $I_x = \phi(x)$. For each downstream dataset, there will be $k$ classes and each class will be filled in a manual prompt template, such as "a photo of <CLASS>". Then the text encoder will generate feature

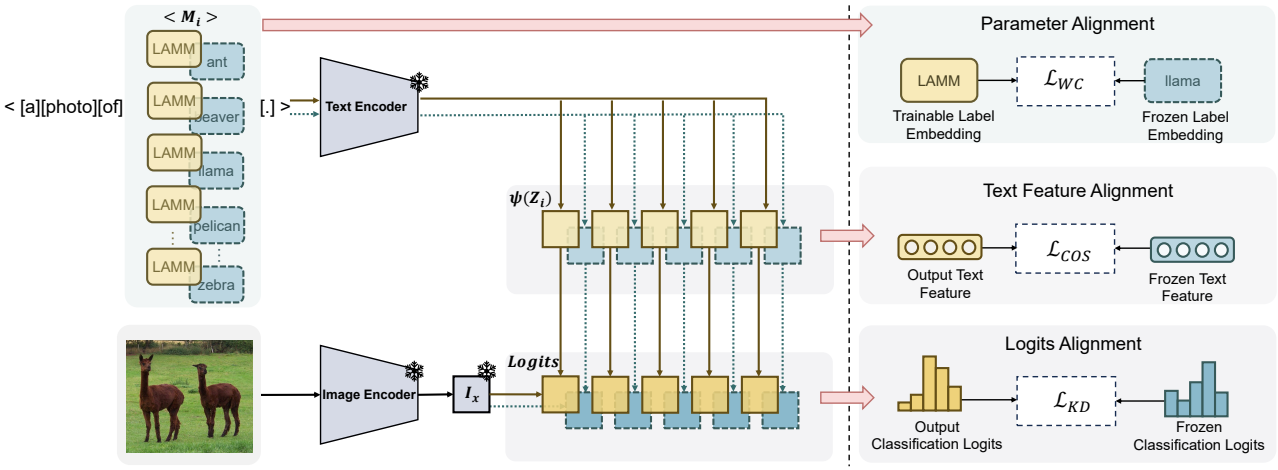


Figure 2: The whole architecture of LAMM. We replace the category tokens in the downstream dataset with trainable vectors and incorporate a hierarchical loss to preserve the CLIP's generalization ability of each category. The gray boxes represent the frozen model and image feature, the blue boxes indicate the original label embeddings/features/logits, and the yellow ones denote the labe embeddings/features/logits during training.

representations for each category by further processing, resulting in each category feature. During training, CLIP maximizes the cosine similarity between image representation and its corresponding category representation, while minimizing the cosine similarity between unmatched pairs. During zero-shot inference, the prediction probability of $i$-th category is computed as:

$$p(y=i \mid I)=\frac{\exp\left(\cos\left(I_x, \psi(y_i)\right)/\tau\right)}{\sum_{j=1}^{k}\exp\left(\cos\left(I_x, \psi(y_j)\right)/\tau\right)} \quad (1)$$

where $\tau$ is a temperature parameter acquired by CLIP, while function $\cos$ represents cosine similarity.

## Label Alignment

Although CLIP has strong zero-shot performance, providing downstream tasks with corresponding textual descriptions can effectively enhance the zero-shot and few-shot recognition capability of CLIP. Previous prompt tuning work on text template mainly focus on the training of "a photo of", while neglecting the optimization of "<CLASS>". To effectively align class labels in downstream tasks to pre-trained models, we propose LAMM, which automatically optimizes the label embedding through end-to-end training. We take CLIP as an example, LAMM on CLIP only finetunes the class embedding representation for downstream tasks. In this way, the prompt template of LAMM converts to:

$$z_i = [a][photo][of][< M_i >][.] \quad (2)$$

where $< M_i >$ (i=1, 2, ..., k) represents a learnable token of the $i$-th category. Similar to Equation 1, the prediction probability of LAMM is computed as:

$$p(y=i \mid I)=\frac{\exp\left(\cos\left(I_x, \psi(z_i)\right)/\tau\right)}{\sum_{j=1}^{k}\exp\left(\cos\left(I_x, \psi(z_i)\right)/\tau\right)} \quad (3)$$

During training, we only update the category vectors $\{< M_i >\}_{i=1}^{k}$ in each downstream dataset, which will decrease the gap between image representation and its corresponding category representation.

Futhermore, LAMM can be applied into existing multimodal prompting methods. Take CoOp for example, the difference between CoOp and vanilla CLIP is replacing the prompt template with M learnable tokens. Thus, the prompt template of CoOp+LAMM becomes:

$$z_i^* = [V_1][V_2]...[V_4][< M_i >] \quad (4)$$

By replacing the class representation with a trainable vector, we can integrate our method into any existing multimodal prompt approach.

## Hierarchical Loss

A well-aligned feature space is the key of strong zero-shot ability of CLIP, which also facilitates the learning of downstream tasks. Given that LAMM does not introduce any modifications or additions to the CLIP model's parameters, the image representation within the aligned feature space remains fixed, while the trainable embedding of each category changes the text representation within the feature space. Nevertheless, a single text representation corresponds to multiple images of a given category in the downstream datasets, despite the entire training process being conducted under few-shot settings. This situation may lead to overfitting of the trainable class embeddings to the limited number of images in the training set. Hence, we propose a hierarchical loss (HL) to safeguard the generalization ability in the parameter space, feature space and logits space.

**Parameter Space** To mitigate the risk of overfitting in models, numerous machine learning techniques employ parameter regularization to enhance the generalization ability on unseen samples. In this regard, the weight consolidation (WC) (Kirkpatrick et al. 2017) loss is employed as follows:

$$\mathcal{L}_{\mathrm{WC}} = \sum_i \left(\theta_i - \bar{\theta}_i\right)^2 \quad (5)$$

where $\theta$ is the trainable parameters of the current model, and $\bar{\theta}$ is the reference ones. In LAMM, $\theta$ represents the trainable label embeddings, whereas $\bar{\theta}$ represents the original label embeddings. While parameter regularization can address the issue of overfitting, excessive regularization may hinder the model's ability to adequately capture the features and patterns present in the training data. We set the coefficient of $\mathcal{L}_{\mathrm{WC}}$ inversely proportional to the number of training shots, suggesting that fewer shots will be accompanied by a more strict regularization process within parameter space.

**Feature Space** In addition to the parameter space, it is crucial for the text feature of our trained category to align with the characteristics of the training images. During the training process, the text feature of our trained category gradually converges towards the characteristics present in the training images. However, if the representations of the few-shot training images for a particular category do not align with those of the entire image dataset, it can lead to the label semantics overfitting to specific samples. For example, consider the label embedding of the "llama" image illustrated in Figure 2. The label embedding may overfit to the background grass information, even though llamas may not always be associated with grass in all instances. In order to mitigate the overfitting of text features for each category, we employ a text feature alignment loss to restrict the optimization region of the text feature. Drawing inspiration from previous work on similarity at the semantic level (Gao, Yao, and Chen 2021), we employ a cosine similarity loss for alignment. In this approach, for a category template $z_i$ in Equation 2, the original prompt template $y_i$ from Equation 1 serves as the center of its optimization region. The cosine loss is formulated as follows:

$$\mathcal{L}_{COS} = \sum_i 1 - \cos(\psi(z_i), \psi(y_i)) \tag{6}$$

**Logits Space** The strong generalization ability of CLIP plays a crucial role in the effectiveness of multi-modal prompting methods within few-shot scenarios. While the previous two losses enhance the generalization capability of LAMM through regularization in the parameter and feature spaces, it is desirable to minimize the distribution shift of logits between image representations and different text representations, as compared to the zero-shot CLIP. Therefore, we introduce a knowledge distillation loss in the classification logits space, which allows for the transfer of generalization knowledge from CLIP to LAMM. The distillation loss can be formulated as follows:

$$\mathcal{L}_{\mathrm{KD}} = -\sum \cos\left(I_x, \psi\left(z_i\right)\right) \log\left(\cos\left(I_x, \psi\left(y_i\right)\right)\right) \tag{7}$$

**Total Loss** To train the LAMM for downstream tasks, cross-entropy (CE) loss is applied to the similarity score as same to finetuning the CLIP model:

$$\mathcal{L}_{\mathrm{CE}} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{CE}\left(\tau \cdot \cos(I_x, \psi(z_i)), y_i\right) \tag{8}$$

where $\tau$ is a parameter learned during the pre-training. In this way, the total loss is:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_1 * \mathcal{L}_{WC} + \lambda_2 * \mathcal{L}_{COS} + \lambda_3 * \mathcal{L}_{\mathrm{KD}} \tag{9}$$

where $\lambda_1, \lambda_2, \lambda_3$ are hyper-parameters. To prevent the redundancy of adjusting parameter, we set $\lambda_1 = 1/n, \lambda_2 = 1, \lambda_3 = 0.05$ for all our experiments empirically, where $n$ represents the number of training shots.

# Experiments

## Few-shot Settings

**Datasets** We follow the datasets used in previous works (Zhou et al. 2022b; Khattak et al. 2022) and evaluate our method on 11 image classification datasets, including Caltech101 (Fei-Fei, Fergus, and Perona 2007), ImageNet (Deng et al. 2009), OxfordPets (Parkhi et al. 2012), Cars (Krause et al. 2013), Flowers102 (Nilsback and Zisserman 2008), Food101 (Bossard, Guillaumin, and Gool 2014), FGVC (Maji et al. 2013), SUN397 (Xiao et al. 2010), UCF101 (Soomro, Zamir, and Shah 2012), DTD (Cimpoi et al. 2014) and EuroSAT (Helber et al. 2019). Besides, we follow (Radford et al. 2021; Zhou et al. 2022b) to set up the few-shot evaluation protocol for our few-shot learning experiments. Specifically, we use 1, 2, 4, 8, and 16 shots for training respectively, and evaluate the models on the full test sets. All experimental results are the average of the results obtained from running the experiments on seeds 1, 2 and 3.

**Baselines** We compare the results across LAMM, CoOp, and MaPLe. Furthermore, we incorporate LAMM into CoOp and MaPLe to prove the compatibility of LAMM. To maintain fair controlled experiments, all prompt templates in our experiments are initialized from "a photo of <CLASS>". Besides, the pre-trained model adopted here is ViT-B/16 CLIP since MaPLe can only be adopted in transformer-based VL-PTMs. We keep the same training parameters (e.g., learning rate, epochs, and other prompt parameters) of each model in their original settings, where the epoch of CoOp is 50 and MaPLe is 5. As for vanilla CLIP + LAMM, we follow the settings of CoOp. All of our experiments are conducted on a single NVIDIA A100. The corresponding hyper-parameters are fixed across all datasets in our work. Moreover, adjusting parameters for different training shots and datasets can boost performance.

## Comparision to the State-of-the-art Methods

**Main Results on 11 Datasets.** We compare LAMM, CoOp, MaPLe and zero-shot CLIP on the 11 datasets as mentioned above, demonstrated in in Figure 3. We can observe that LAMM yield best performance among all shots compared to the state-of-the-art multi-modal prompt methods. LAMM only replaces <CLASS> for each category with a trainable vector, while CoOp replaces "a photo of" with trainable vectors. However, compared to CLIP, LAMM demonstrates an enhancement of +1.02, +2.11, +2.65, +2.57, and +2.57(%), on the 1, 2, 4, 8, and 16 shots, respectively. The preceding result highlights the importance of finetuning label embeddings for downstream tasks, compared to the previous focus solely on prompt template learning in multi-modal prompt learning. It suggests that label embeddings are even more crucial than prompt templates for pre-trained models' transferability to downstream tasks. Furthermore,

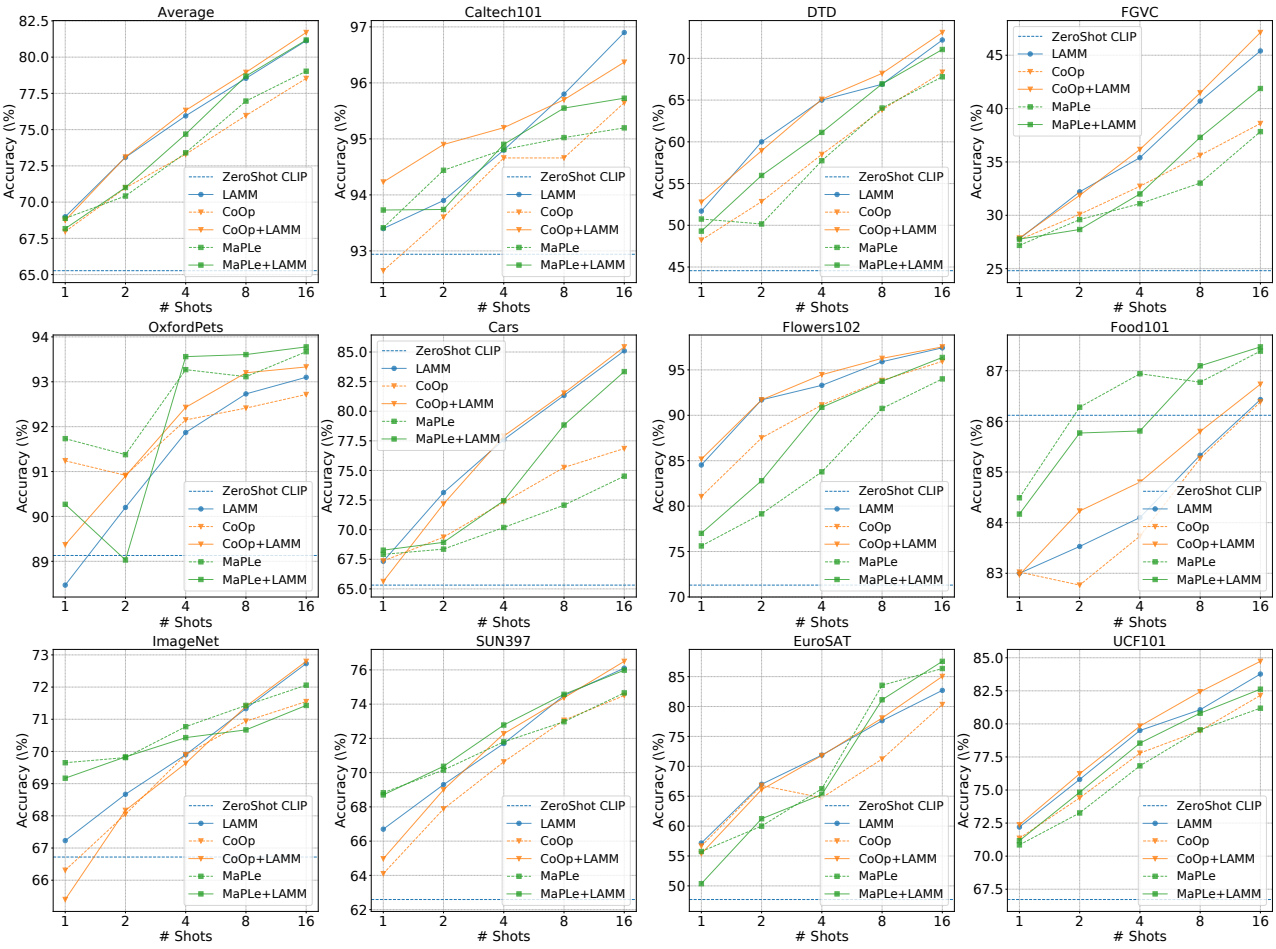


Figure 3: Main results over 11 datasets under the few-shot learning setting. We report the average accuracy (%) of 1/2/4/8/16 shots over three runs. Overall, the proposed LAMM enhances the performance of CLIP, CoOp, and MaPLe.

as the number of shot increases, the improvement observed with LAMM becomes more clear. This can be attributed to the challenge LAMM faces in learning representative embeddings for categories in downstream tasks when provided with fewer shots.

**Domain Generalization** We evaluate the cross-dataset generalization ability of LAMM by training it on ImageNet and evaluating on ImageNetV2 (Recht et al. 2019) and Imagenet-Sketch (Wang et al. 2019), following Khattak et al. (2022). The evaluating datasets have same categories with the training set. But the three datasets are different in domain distribution. The experimental results are shown in Table 1. LAMM achieves the best performance, which surpasses MaPLe (Khattak et al. 2022) 1.06% on ImageNet-V2 and falls behind MaPLe 1.04% on ImageNet-Sketch. On average, LAMM exceeds MaPLe 0.91%. This indicates the LAMM is also capable of out-of-distribution tasks.

| Method | Source | Target | | Average |
|---|---|---|---|---|
| | ImageNet | -V2 | -Sketch | |
| CLIP | 66.73 | 60.83 | 46.15 | 57.90 |
| CoOp | 71.51 | 64.20 | 47.99 | 61.23 |
| CoCoOp | 71.02 | 64.07 | 48.75 | 61.28 |
| MaPLe | 70.02 | 64.07 | **49.15** | 61.08 |
| LAMM | **72.73** | **65.13** | 48.11 | **61.99** |

Table 1: Comparison of LAMM with existing methods in domain generalization setting. LAMM shows highest performance on average.

## Combination with State-of-the-art methods

**The improvement of incorporating LAMM** We compare existing methods CoOp and MaPLe with and without LAMM, which uses trainable vectors and a hierarchical loss function. From Figure 3, we observe that LAMM notably enhances performance in few-shot scenarios, except in 1-shot learning. For CoOp, the average accuracy variations across 11 datasets with 1, 2, 4, 8, and 16 shots are +0.77, +2.13, +3.03, +2.98, +3.17(%), respectively. As for MaPLe, the variations are -0.71, +1.79, +1.29, +1.72, +2.15(%). The incorporation of LAMM has yielded exceedingly favorable outcomes across all three methodologies, reaffirming

| Method | Subset | ImageNet | Caltech101 | DTD | FGVC | Cars | Flowers102 | OxfordPets | Food101 | EuroSAT | UCF101 | SUN397 | Average | Degradation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Zero-shot CLIP | Set1 | 72.43 | 96.84 | 53.24 | 27.19 | 63.37 | 72.08 | 91.17 | 90.10 | 56.48 | 70.53 | 69.36 | 69.34 | 69.34 (**0**) |
| | Set2 | 68.14 | 94.00 | 59.90 | 36.29 | 74.89 | 77.80 | 97.26 | 91.22 | 64.05 | 77.50 | 75.35 | 74.22 | |
| CoOp | Set1 | 71.77 | 97.07 | 54.63 | 23.40 | 61.63 | 61.97 | 93.63 | 87.10 | 72.37 | 71.03 | 72.53 | 69.74 | 82.69 (**-12.95**) |
| | Set2 | 73.67 | 96.37 | 76.57 | 54.53 | 87.07 | 97.50 | 97.70 | 91.47 | 90.27 | 88.13 | 83.77 | 85.18 | |
| MaPLe | Set1 | 74.37 | 97.10 | 70.83 | 30.27 | 65.77 | 77.53 | 95.03 | **90.6** | 82.73 | 78.17 | 77.60 | 76.36 | 82.28 (**-5.92**) |
| | Set2 | 74.13 | **96.50** | 77.40 | 53.73 | 83.27 | 97.00 | **98.17** | **92.27** | **92.60** | 88.33 | 83.95 | 85.21 | |
| LAMM | Set1 | **77.23** | **98.40** | **83.30** | **43.27** | **81.63** | **97.80** | **95.17** | 89.83 | **90.60** | **86.17** | **82.13** | **84.14** | 84.14 (**0**) |
| | Set2 | **74.57** | 96.03 | **79.47** | **61.47** | **91.97** | **98.33** | 97.93 | 91.73 | 91.80 | **89.23** | **84.80** | **87.03** | |

Table 2: Comparison of LAMM and other prompting methods on 16-shot classs incremental learning. Initially, models are trained on Set 1, followed by training on Set 2. The contents within the parentheses of term "Degradation" refers to the decline in evaluation results on Set 1 subsequent to further training on Set 2.

LAMM's status as an exceptional few-shot learner.

Besides, the enhanced performance of LAMM improves significantly with the increase in shots, and the reason behind this is that our approach primarily operates on label representation. When the number of training sample is small, such as in the case of 1-shot learning, the label representation learns numerous specific features associated with the single image, which constitutes a significant portion of noise rather than the representation of the entire category. For instance, the significant reduction in accuracy of MaPLe with LAMM in the case of 1-shot learning is primarily due to the decrease of 5.38% accuracy on the EuroSAT (Krause et al. 2013). This is owing to the fact that the EuroSAT is a satellite-image dataset, which suggests that the gap between the feature spaces of images and text is considerable. Consequently, it makes LAMM more susceptible to overfitting on the features of limited-sample images. Moreover, MaPLe has more trainable parameters than CoOp, which makes it more prone to overfitting than CoOp.

**Comparisons of different LAMMs** After incorporating LAMM into previous methods, we evaluate the performance of LAMM, CoOp+LAMM, and MaPLe+LAMM. From Table 3, we can observe that CoOp+LAMM achieving the best results. To be more specific, LAMM achieves a reduction of +0.25, -0.02, -0.38, -0.41, -0.60(%) compared to CoOp+LAMM on 1, 2, 4, 8, and 16 shots. We conclude that CoOp+LAMM can achieve superior results due to CoOp's ability to search a soft template that is more effective than "a photo of", without significantly altering the semantic premise of the prompt template. In contrast, MaPLe introduces more trainable parameters, which may result in a greater deviation of the semantic meaning of the soft prompt template from that of "a photo of".

## Class Incremental Learning

Since LAMM exclusively modifies the training class embedding without altering any parameters of the CLIP model, it is confined to using the unmodified CLIP and the untrained novel class embedding during base-to-novel testing. In this way, LAMM's performance for novel classes is same to that of the zero-shot CLIP. However, except MaPLe exceed zero-shot CLIP 0.92% on novel class testing, other prompt tuning methods falls behind zero-shot CLIP.

Apart from base-to-novel testing, incremental learning is also an significant issue when aiming to broaden a model's knowledge base. Following MaPLe, we partition the datasets into base and novel classes, designating the base classes as Set 1 and the novel ones as Set 2. Initially, each model is trained on Set 1, after which the continual training persists on Set 2. Finally, we evaluate the performance of each model on both Set 1 and Set 2. The results are shown in Table 2. The term 'Degradation' the disparity in performance on Set 1 prior to and subsequent to incremental training on Set 2. Such a decline represents the forgetting of prior tasks during the incremental learning process.

It is evident that LAMM exhibits superior performance on both Set 1 and Set 2, particularly on Set 1. The observed decline in performance of CoOp and MaPLe on Set 1 can be attributed to the phenomenon of forgetting during continual learning. Especially for CoOp, its performance on Set 1 closely approximates that of zero-shot CLIP. This indicates that the adaptable text template is highly sensitive to variations in testing tasks. Consequently, CoOp necessitates an entirely new template upon encountering novel classes. Although MaPLe outperforms CoOp, it still undergoes an average degradation of -5.92%.

As LAMM solely manipulates the embedding of new classes while preserving the existing class embeddings, the performance of LAMM on previous classes remains stable during continual training on new classes. Conversely, in the case of CoOp and MaPLe, when the number of categories to be learned increases, the model's performance will significantly decline if retraining from scratch on all categories is not undertaken. This unique prowess of LAMM in incremental learning implies its suitability and desirability for deployment in downstream applications.

**Influence of Hierarchical Loss** To validate the effect of the proposed loss function, we conduct a comprehensive study

| Method | 1-shot | 2-shot | 4-shot | 8-shot | 16-shot |
|---|---|---|---|---|---|
| **Zero-shot CLIP** | 65.27 | 65.27 | 65.27 | 65.27 | 65.27 |
| **LAMM*(w/o HL)*** | 61.39 | 66.80 | 72.13 | 76.21 | 79.93 |
| **LAMM** | **68.99** | 73.09 | 75.95 | 78.54 | 81.13 |
| **CoOp** | 67.97 | 70.98 | 73.30 | 75.97 | 78.53 |
| **+LAMM*(w/o HL)*** | 61.23 | 66.33 | 72.06 | 76.19 | 79.94 |
| **+LAMM** | 68.74 | **73.11** | **76.33** | **78.95** | **81.71** |
| **MaPLe** | 68.88 | 69.22 | 73.40 | 76.97 | 79.03 |
| **+LAMM*(w/o HL)*** | 61.25 | 67.70 | 73.39 | 77.64 | 80.74 |
| **+LAMM** | 68.17 | 71.01 | 74.69 | 78.69 | 81.18 |

Table 3: Comparision of with or without hierarchical loss among vanilla CLIP, CoOp, and MaPLe.

| $\mathcal{L}_{CE}$ | $\mathcal{L}_{WC}$ | $\mathcal{L}_{COS}$ | $\mathcal{L}_{KD}$ | Average |
|---|---|---|---|---|
| ✓ | | | | 79.93 |
| ✓ | ✓ | ✓ | ✓ | 81.13 |
| ✓ | | ✓ | ✓ | 80.98 |
| ✓ | ✓ | | ✓ | 80.86 |
| ✓ | ✓ | ✓ | | 81.08 |
| ✓ | | | ✓ | 80.82 |
| ✓ | | ✓ | | 80.75 |
| ✓ | ✓ | | | 80.63 |

Table 4: Ablations on hierarchical loss function on 16-shot.

| Setting | Random | Words |
|---|---|---|
| 1-shot | 62.71 | 68.99 |
| 2-shot | 62.71 | 73.09 |
| 4-shot | 71.68 | 75.95 |
| 8-shot | 76.65 | 78.54 |
| 16-shot | 80.71 | 81.13 |

Table 5: Comparision of initializations from random and category words.

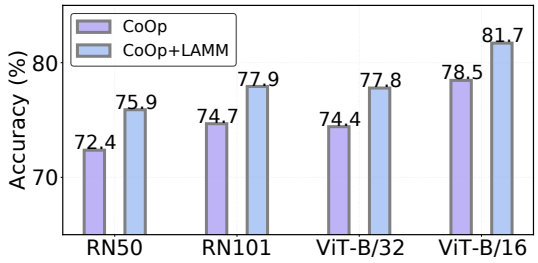


Figure 4: Ablations among different vision backbones

on three models among all the 11 datasets. For comparison, we consider LAMM, CoOp+LAMM, and MaPLe+LAMM with or without hierarchical loss. Table 3 presents the averaged results over three runs. We observe that the results exhibit a significant difference upon the introduction of the loss, with hierarchical loss proving to be highly essential in enhancing the performance of LAMM. Furthermore, the degree of improvement brought by hierarchical loss becomes more apparent when there are fewer shots involved. This is due to that finetuning to align the label with few samples, especially with a single image, can result in the label overfitting to other noise information within the image. Since hierarchical loss incorporates strong constraints over parameter space, text feature space and logits space, which ensures that the semantic meaning of the label does not deviate too far from its original semantic meaning after training, it can improve LAMM's performance more significantly with fewer samples compared to without hierarchical loss.

## Ablation Experiments

**Ablations of Hierarchical Loss** In the context of the hierarchical loss, it is composed of losses from three dimensions: parameter space, feature space, and logits space. Therefore, we conducted an ablation study on these varying levels of loss in the context of 16-shot, as illustrated in Table 4. It is evident that each of the three losses contributes positively to the final outcome. However, their cumulative effects cannot be simply combined, given that the loss from one feature space can influence the characteristics of another space. The universal objective of these losses is to retain the generalization capability of CLIP within LAMM, preventing overfitting on specific samples.

**Initialization** We conduct a comparison between initialization from category words and random initialization. The former uses the original word embedding of each category as the initialized category embedding, while the latter randomly initializes the category embeddings. Table 5 shows the results, which are conducted on LAMM among 11 datasets and suggest that random initialization is inferior to category word initialization. This illustrates that training representations for each category from scratch in few-shot training is quite challenging, and this is reflected in the increasing discrepancy between random initialization and word initialization as the shot number decreases.

**Vision Backbone** Figure 4 illustrates the average results on 11 datasets of various visual backbones, including CNN and ViT architectures. The results demonstrate that LAMM can consistently improve the performance of CoOp across different visual backbones. This further indicates that LAMM can enhance the VL model's transferability to downstream tasks in a stable manner.

## Conclusion

Compared to traditional few-shot learning methods, prompt learning based on VL-PTMs has demonstrated strong transferability in downstream tasks. Our research reveals that in addition to prompt template learning, reducing the gap between VL-PTMs and downstream task label representations is also a significant research issue. Our paper provides a comprehensive study on how to align label representations in downstream tasks to VL-PTMs in a plug-and-play way. Our proposed LAMM has demonstrated a significant improvement in the performance of previous multi-modal prompt methods in few-shot scenarios. In particular, by simply incorporating LAMM into vanilla CLIP, we can achieve better results than previous multi-modal prompt methods. LAMM also demonstrates robustness in out-of-distribution scenarios, along with its superiority in the incremental learning process. These findings further demonstrate the immense potential of optimizing the transfer of VL-PTMs to downstream tasks, not only limited to image recognition, but also encompassing visually semantic tasks such as image segmentation, object detection, and more. We hope that the insights gained from our work on label representation learning will facilitate the development of more effective transfer methods for VL-PTMs.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant No.61977045).

## References

Bossard, L.; Guillaumin, M.; and Gool, L. V. 2014. Food-101 - Mining Discriminative Components with Random Forests. In *Proc. of ECCV*.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing Textures in the Wild. In *Proc. of CVPR*.

Cui, G.; Hu, S.; Ding, N.; Huang, L.; and Liu, Z. 2022. Prototypical Verbalizer for Prompt-based Few-shot Tuning. In *Proc. of ACL*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proc. of CVPR*.

Du, Y.; Liu, Z.; Li, J.; and Zhao, W. X. 2022. A Survey of Vision-Language Pre-Trained Models. In *Proc. of IJCAI*.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2007. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Comput. Vis. Image Underst.*

Gao, P.; Geng, S.; Zhang, R.; Ma, T.; Fang, R.; Zhang, Y.; Li, H.; and Qiao, Y. 2021. CLIP-Adapter: Better Vision-Language Models with Feature Adapters. *CoRR*.

Gao, T.; Fisch, A.; and Chen, D. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *Proc. of ACL*.

Gao, T.; Yao, X.; and Chen, D. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Hambardzumyan, K.; Khachatrian, H.; and May, J. 2021. WARP: Word-level Adversarial ReProgramming. In *Proc. of ACL*.

Helber, P.; Bischke, B.; Dengel, A.; and Borth, D. 2019. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.; Parekh, Z.; Pham, H.; Le, Q. V.; Sung, Y.; Li, Z.; and Duerig, T. 2021. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *Proc. of ICML*.

Jia, M.; Tang, L.; Chen, B.; Cardie, C.; Belongie, S. J.; Hariharan, B.; and Lim, S. 2022. Visual Prompt Tuning. In *Proc. of ECCV*.

Khattak, M. U.; Rasheed, H. A.; Maaz, M.; Khan, S.; and Khan, F. S. 2022. MaPLe: Multi-modal Prompt Learning. *CoRR*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3D Object Representations for Fine-Grained Categorization. In *Proc. of ICCV*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proc. of EMNLP*.

Li, B.; Weinberger, K. Q.; Belongie, S. J.; Koltun, V.; and Ranftl, R. 2022. Language-driven Semantic Segmentation. In *Proc. of ICLR*.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M. B.; and Vedaldi, A. 2013. Fine-Grained Visual Classification of Aircraft. *CoRR*.

Nilsback, M.; and Zisserman, A. 2008. Automated Flower Classification over a Large Number of Classes. In *Sixth Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008*.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. V. 2012. Cats and dogs. In *Proc. of CVPR*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proc. of ICML*.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2019. Do imagenet classifiers generalize to imagenet? In *Proc. of ICML*.

Schick, T.; and Schütze, H. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proc. of EACL*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR*.

Sung, Y.; Cho, J.; and Bansal, M. 2022. VL-ADAPTER: Parameter-Efficient Transfer Learning for Vision-and-Language Tasks. In *Proc. of CVPR*.

Wang, H.; Ge, S.; Lipton, Z.; and Xing, E. P. 2019. Learning robust global representations by penalizing local predictive power. *Proc. of NeurIPS*.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. SUN database: Large-scale scene recognition from abbey to zoo. In *Proc. of CVPR*.

Yao, L.; Huang, R.; Hou, L.; Lu, G.; Niu, M.; Xu, H.; Liang, X.; Li, Z.; Jiang, X.; and Xu, C. 2022. FILIP: Fine-grained Interactive Language-Image Pre-Training. In *Proc. of ICLR*.

Zang, Y.; Li, W.; Zhou, K.; Huang, C.; and Loy, C. C. 2022. Open-Vocabulary DETR with Conditional Matching. In *Proc. of ECCV*.

Zhai, X.; Wang, X.; Mustafa, B.; Steiner, A.; Keysers, D.; Kolesnikov, A.; and Beyer, L. 2022. LiT: Zero-Shot Transfer with Locked-image text Tuning. In *Proc. of CVPR*.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022a. Conditional Prompt Learning for Vision-Language Models. In *Proc. of CVPR*.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022b. Learning to Prompt for Vision-Language Models. *Int. J. Comput. Vis.*