

# COMMA: Co-articulated Multi-Modal Learning

Lianyu Hu<sup>1</sup>, Liqing Gao<sup>1</sup>, Zekang Liu<sup>1</sup>, Chi-Man Pun<sup>2</sup>, Wei Feng<sup>1\*</sup>

<sup>1</sup>College of Intelligence and Computing, Tianjin University, China

<sup>2</sup>Department of Computer and Information Science, University of Macau, China  
 {hly2021,lqgao,lzk100953}@tju.edu.cn,cmpun@umac.mo,wfeng@ieee.org

## Abstract

Pretrained large-scale vision-language models such as CLIP have demonstrated excellent generalizability over a series of downstream tasks. However, they are sensitive to the variation of input text prompts and need a selection of prompt templates to achieve satisfactory performance. Recently, various methods have been proposed to dynamically learn the prompts as the textual inputs to avoid the requirements of laboring hand-crafted prompt engineering in the fine-tuning process. We notice that these methods are suboptimal in two aspects. First, the prompts of the vision and language branches in these methods are usually separated or unidirectionally correlated. Thus, the prompts of both branches are not fully correlated and may not provide enough guidance to align the representations of both branches. Second, it's observed that most previous methods usually achieve better performance on seen classes but cause performance degeneration on unseen classes compared to CLIP. This is because the essential generic knowledge learned in the pretraining stage is partly forgotten in the fine-tuning process. In this paper, we propose Co-Articulated Multi-Modal Learning (COMMA) to handle the above limitations. Especially, our method considers prompts from both branches to generate the prompts to enhance the representation alignment of both branches. Besides, to alleviate forgetting about the essential knowledge, we minimize the feature discrepancy between the learned prompts and the embeddings of hand-crafted prompts in the pre-trained CLIP in the late transformer layers. We evaluate our method across three representative tasks of generalization to novel classes, new target datasets and unseen domain shifts. Experimental results demonstrate the superiority of our method by exhibiting a favorable performance boost upon all tasks with high efficiency. Code is available at <https://github.com/hulianyuyy/COMMA>

## Introduction

The increase of web data with aligned large-scale text-image pairs has greatly facilitated the development of foundation vision-language models (VLMs) such as CLIP (Radford et al. 2021). Thanks to the supervision provided by the natural language, these models have demonstrated excellent generalization performance over a series of downstream

tasks and could reason about open-vocabulary visual concepts (Gao et al. 2021; Fang et al. 2021; Cheng et al. 2021). During inference, a set of hand-crafted prompts such as 'a photo of [category]' is used as a query for the text encoder. The output text embeddings are matched with the visual embeddings generated by the image encoder to predict the output class.

Despite the impressive generalizability of CLIP over novel scenarios, its massive model scale and requirements of training data make it infeasible to fine-tune the full model in the downstream tasks. Fine-tuning the whole model also easily forgets the beneficial knowledge acquired in the training stage and overfits the downstream data. To handle the above limitations, a series of works (Radford et al. 2021; Jin et al. 2021) are dedicated to designing better hand-crafted prompts to fit downstream tasks. However, hand-crafted prompts require careful selections with intensive labors, which may also be suboptimal in depicting the characteristics of novel scenarios. Recently, many methods (Shu et al. 2022; Zhou et al. 2022a,b) propose to treat the prompts as textual embeddings and update them in the fine-tuning process to better coordinate with the VLMs. In this procedure, only the learnable prompts are updated and the original parameters of VLMs are fixed, which greatly reduces the requirements of computations.

We argue that these approaches still own two major drawbacks. First, the prompts of the vision and language branches in these methods are usually separated or unidirectionally correlated (the vision branch is uni-directionally influenced by the text branch only). As the goal of VLMs is to better match the embeddings of vision and language branches, disjointed vision and language prompts may hinder modelling the correlation of output embeddings in two branches. Second, it has been observed that most previous methods usually achieve superior performance on seen classes but demonstrate worse generalizability on unseen classes compared to CLIP. This is because the essential generic knowledge acquired in the pretraining process is partly forgotten in the fine-tuning procedure.

To handle the above limitations, we propose Co-Articulated Multi-Modal Learning (COMMA) in this paper. Especially, to enhance the correlations of prompts in both branches, we generate prompts of the next layer based on preceding prompts in both branches. In this case, the

\*Corresponding author

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

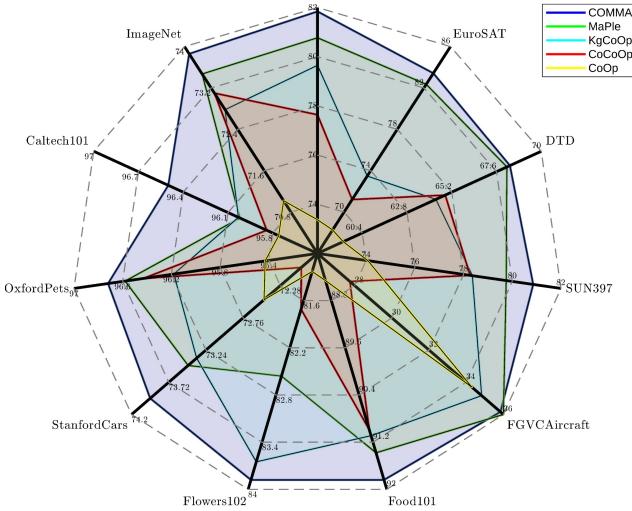


Figure 1: COMMA outperforms state-of-the-art methods across 10/11 diverse image recognition datasets on the base-to-novel generalization task.

prompt embeddings of both branches are well correlated and could provide enough guidance for the next layer to align representations of both branches. Besides, to alleviate forgetting about the essential knowledge acquired in large-scale training data, we try to minimize the discrepancy between the learned prompts and the embeddings of hand-crafted prompts in the pretrained CLIP. The generic knowledge can be better preserved and adapted to novel classes in the fine-tuning stage. Our extensive experiments on three key representative settings including base-to-novel generalization, cross-dataset evaluation, and domain generalization demonstrate the strength of COMMA. Especially, on base-to-novel generalization, our method outperforms other approaches across 10/11 datasets as shown in fig. 1. Further, our COMMA also demonstrates excellent generalizability over all datasets in the cross-dataset transfer and domain generalization settings, achieving consistent performance boost. Thanks to its streamlined design, COMMA exhibits improved training and inference efficiency compared to previous methods.

## Related Work

### Vision Language Models

Recently, the equipment of large-scale image-text pairs has greatly facilitated the development of Vision Language Models (VLMs). Previous methods usually adopt region-based (Anderson et al. 2018) or grid-based (Jiang et al. 2020; Nguyen, Goswami, and Chen 2020) approaches to model the correlations between vision and language. However, the internal relations between two modalities are not fully captured by such a design. Recently, a series of models like CLIP (Radford et al. 2021), ALIGN (Jia et al. 2021), FLIP (Yao et al. 2021) and BLIP (Li et al. 2022) are introduced to capture the correlations between image and text in a contrastive manner. They learn joint image-language repre-

sentation by maximizing the similarity of positive pairs and pushing away those negative pairs. Aided by the supervision of natural language, they have demonstrated impressive performance over a broad series of downstream tasks. However, their massive model size and requirement of training data limit their applications in resource-constrained downstream tasks. How to better exhibit their potential in those novel concepts with high efficiency is still a challenging problem. Many works have demonstrated better performance on downstream tasks by using tailored methods to adapt VLMs for few-shot image-recognition (Zhang et al. 2021; Sung, Cho, and Bansal 2022), object detection (Gu et al. 2021; Feng et al. 2022; Maaz et al. 2022), and segmentation (Ding et al. 2022; Lüdecke and Ecker 2022).

### Prompt Learning

Large language models often require instructions in the form of sentences, known as text prompts, to better understand the task. These prompts can be hand-crafted (Jin et al. 2021) or automatically learned (Houlsby et al. 2019; Liu et al. 2023) during the fine-tuning process, while the latter is referred to as prompt learning. The trend has first appeared in the natural language processing (NLP) field where some methods (Lester, Al-Rfou, and Constant 2021; Li and Liang 2021; Liu et al. 2021) propose to prepend a series of learnable prompts to the inputs or the intermediate features to adapt the learned representations to new tasks. A similar tendency has also arisen in the visual domain (Jia et al. 2022; Wang et al. 2022) and vision-language domain (Gao et al. 2021; Khattak et al. 2023; Yao, Zhang, and Xu 2023). While most previous methods separately consider prompts in multi-modal branches, we try to enhance their correlations to guide the alignment of representations in both branches.

### Prompt Learning in Vision Language Models

Inspired by prompt learning in NLP, many methods propose to adapt VLMs by learning the prompt tokens through end-to-end fine-tuning. The original parameters of VLMs are fixed and only a few extra learnable prompt parameters are updated in this procedure. CoOp (Zhou et al. 2022b) first replaces the hand-crafted prompts with learnable soft prompts in the first layer to adapt to VLMs. CoCoOp (Zhou et al. 2022a) proposes to generate an image-conditional prompt to utilize the power of input features. ProGrad (Zhu et al. 2022) only updates the prompts whose gradient is aligned to the “general knowledge” generated by the original prompts. KgCoOp (Yao, Zhang, and Xu 2023) tries to align the output embeddings of the text encoder with those of the pretrained CLIP to preserve beneficial information. MaPLe (Khattak et al. 2023) learns soft prompts in both vision and language branches to better align their representations. We note that the prompts in these methods are usually separated, which hinders adopting multi-modal information to better align the representations of both branches. Besides, these methods usually cause performance degeneration over unseen classes compared to CLIP, with much worse generalizability. Our work is the first to explore gathering beneficial information from both branches to better guide the prompt generation process to well align multi-modal representations.

## Method

Our method focuses on how to improve the generalization performance of a large-scale VLM over a series of downstream tasks. To alleviate overfitting the downstream tasks and avoid incurring huge training computational costs, the parameters of both image encoder and text encoder in the original VLM are kept fixed, while only the parameters of the prompts are updated in the fine-tuning process. To better demonstrate the effects of our COMMA, we first give a brief review of VLMs by taking CLIP (Radford et al. 2021) as an example, and then recap typical visual prompt learning methods like CoOp and MaPLe to derivative our method.

## Preliminaries

We build our model based on a pre-trained VLM, CLIP, which consists of a text and vision encoder. CLIP encodes an image  $I \in \mathcal{R}^{H \times W \times 3}$  and a concurrent text description to match their output embeddings. We follow previous methods to use a vision transformer (ViT) (Dosovitskiy et al. 2020) based CLIP model.

**Encoding Image:** An image  $I \in \mathcal{R}^{H \times W \times 3}$  is first split into  $M$  patches with equal intervals, and then reshaped and projected into patch embeddings  $E_0 \in \mathcal{R}^{M \times d_v}$ . These patch embeddings are then sent into a  $K$ -layer transformer  $\mathcal{V}$  along with a learnable class token (CLS)  $c_i$ . The calculation process of each transformer layer can be represented as:

$$[c_i, E_i] = \mathcal{V}_i([c_{i-1}, E_{i-1}]), i \in [0, \dots, K-1]. \quad (1)$$

To obtain a final representation for the input image  $I$ , the CLS token  $c_{K-1}$  of the last transformer layer is extracted and projected to the common V-L latent embedding space via a projection function  $p_v$  as :

$$x = p_v(c_K). \quad (2)$$

**Encoding Text:** the input text descriptions are tokenized and then projected into word embeddings  $W_0 = [w_0^1, w_0^2, \dots, w_0^N] \in \mathcal{R}^{N \times d_t}$ , with  $N$  denoting the length of word embeddings. At each layer, the embedding  $W_{i-1}$  is sent into the  $i_{th}$  transformer layer  $\mathcal{L}_i$  of the text encoder as:

$$[W_i] = \mathcal{L}_i([W_{i-1}]), i \in [0, \dots, K-1]. \quad (3)$$

The final text representation  $z$  is obtained by projecting the last token  $w_{K-1}^N$  of the last transformer layer to the common V-L latent embedding space via a projection function  $p_L$  as :

$$z = p_L(w_{K-1}^N). \quad (4)$$

**Zero-shot inference:** during inference, the inputs for the text encoder are hand-crafted prompts (e.g., 'A photo of a [class]) by replacing the placeholder [class] with the class name of label  $y \in [1, \dots, C]$ . Then the score for  $j_{th}$  class is measured by calculating the similarity between outputs of the text encoder and image encoder via a cosine similarity function  $sim()$  with a temperature parameter  $\tau$  as:

$$p(y_j|x) = \frac{\exp(sim(x, z_j)/\tau)}{\sum_{i=1}^C \exp(sim(x, z_i))}. \quad (5)$$

The class name corresponding to the highest score is adopted as the prediction result for the input image  $I$ .

The text prompts in the text encoder are usually hand-crafted which require labor-intensive manual search, and may not be optimal for the downstream task. Thus, recent methods propose to treat the prompts as textual embeddings and optimize them in the fine-tuning process. We next briefly introduce two typical soft-prompt-based methods.

**CoOp.** CoOp replaces the hand-crafted prompts in the text encoder as learnable soft prompts and directly updates them in the fine-tuning process. Specifically, CoOp introduces  $M$  learnable prompt vectors  $\{p_0, p_1, \dots, p_{M-1}\}$ , and concatenate them with the token embedding  $c_i$  of  $i_{th}$  class as the text input embeddings:  $t_i^{\text{CoOp}} = \{p_0, p_1, \dots, p_{M-1}, c_i\}$ . These embeddings are then sent into the text encoder to obtain the final text representation.

**MaPLe.** MaPLe argues that using prompting in a single branch of VLM may not be optimal since it doesn't allow adjusting representations of both branches to better match their output embeddings. Besides, only inserting prompts in the first layer may not be enough to encode beneficial information of various hierarchies. Thus, MaPLe proposes to insert prompts into both the vision and text branches up to  $J$  layers to enable deep prompting. In each layer, the prompts of the image encoder are uni-directionally generated by the prompts of the text encoder.

## Proposed Method

With considerable performance boost achieved by previous methods over downstream tasks compared to CLIP, they still have two major limitations. First, the prompts in the vision and text branches are usually separated or uni-directionally generated. Since the goal of VLM is to match the output embeddings of different modalities to describe their relationships, the prompts of both branches should be closely related to provide proper guidance to align the representations of both branches. Second, despite impressive performance boosts on the seen classes achieved by recent methods, they usually cause performance degeneration on the unseen classes compared to CLIP (Yao, Zhang, and Xu 2023; Zhou et al. 2022a), demonstrating worse generalization to novel concepts. This is harmful to real-life scenarios as a large number of novel classes may appear. The reason is that the generic knowledge is partly forgotten in the fine-tuning process. To handle the above limitations, we combine beneficial information from both branches to generate the prompts to well match their output embeddings. We also propose to preserve the generic representations of pretrained CLIP in the fine-tuning process to alleviate overfitting. An overview of our proposed method is given in fig. 2.

**Correlated prompt generation.** To better guide and align the representations of two branches, we present to compute prompts based on preceding prompts of both branches to aggregate beneficial multi-modal information. Especially, following previous methods (Khattak et al. 2023), we insert learnable prompts in both vision and text branches up to a specific depth  $J$ . Taking the vision branch as an example, the input embeddings are denoted as  $\{P_0^v, c_0, E_0\}$ , with  $P$  representing the  $M$ -length learnable prompts. The calculation process of  $i_{th}$  ( $i \in [0, \dots, K-1]$ ) transformer layer  $\mathcal{V}_i$

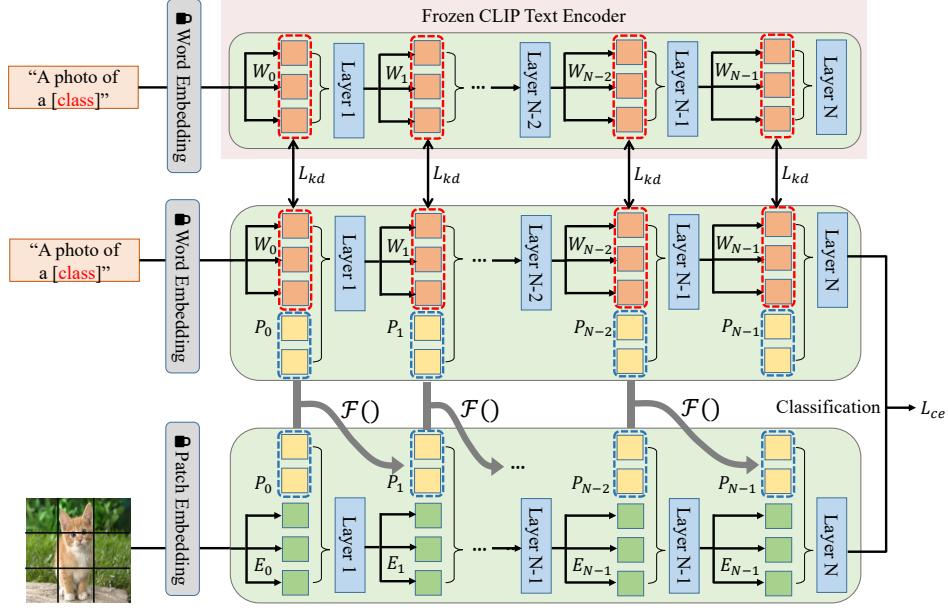


Figure 2: The overview for COMMA. Here,  $\mathcal{L}_{ce}$  denotes the cross-entropy loss and  $\mathcal{L}_{kd}$  represents the knowledge distillation loss between two branches. COMMA generates the prompts of the vision branch based on preceding prompts of both branches to aggregate multi-modal beneficial information to guide their representation alignment. Besides, it let the learned prompts approximate the hand-crafted prompts in the pre-trained CLIP model to preserve generic knowledge.

could be expressed as:

$$[-, c_i, E_i] = \mathcal{V}_i([P_{i-1}^v, c_{i-1}, E_{i-1}]). \quad (6)$$

Instead of leaving the prompts in both branches separated or uni-directionally controlled, we leverage multi-modal information by computing the prompts in the image branch based on the preceding prompts of both branches. Specifically, for  $i_{th}$  transformer layer  $\mathcal{V}_i$ , its prompts are dynamically generated by treating prompts  $P_{i-1}^v$  of  $(i-1)_{th}$  layer in the vision branch as a query, and the prompts  $P_{i-1}^l$  of  $(i-1)_{th}$  layer in the text branch as key and value. This procedure is expressed as :

$$P_i^v = \text{softmax}\left(\frac{P_{i-1}^v \cdot P_{i-1}^l}{\sqrt{P}}\right) P_{i-1}^l. \quad (7)$$

We perform aggregation along the token dimension. In this sense, the prompts in the vision encoder aggregate complementary information from the text branch to guide the alignment path of their representations. Practically, we only generate the prompts in the vision branch with guidance from the high-level semantics in the text branch, and leave the prompts in the text branch randomly initialized for back propagation to avoid hurting their high-level semantic representations.

**Alleviating Forgetting Generic Knowledge.** Previous works (Khattak et al. 2023; Yao, Zhang, and Xu 2023) have witnessed that the generic knowledge contained in pre-trained CLIP models is easily forgotten in the fine-tuning process. We find that the similarity between the learned prompts and the hand-crafted prompts in pretrained CLIP

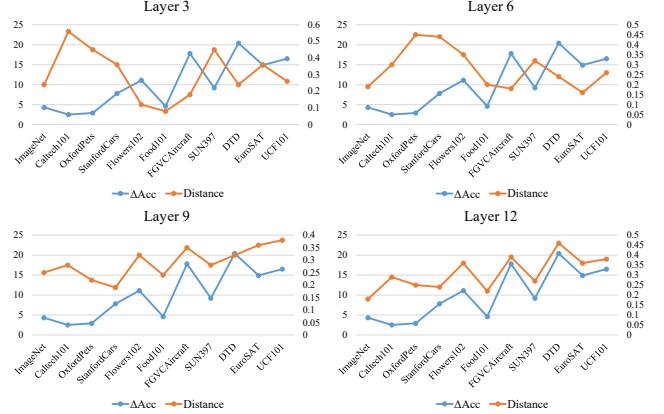


Figure 3: Relationships concerning the degree of performance degradation  $\Delta\text{Acc}$  with the distance between the learnable prompts in CoOp and the hand-crafted prompts in the pretrained CLIP with their performance gap  $\Delta\text{Acc}$  across different layers. It's observed that as the layers go deeper, the degree of performance degradation  $\Delta\text{Acc}$  is more consistent with the prompt embedding distance. Specifically, in the first few layers (e.g., Layer 3 & 6) the correlations between  $\Delta\text{Acc}$  and the prompt distance are irregular, while in the last several layers (e.g., Layer 9 &

is positively correlated with their performance on novel classes. Fig. 3 depicts the relationship concerning the distance between the learnable prompts in CoOp and the hand-crafted prompts in the pretrained CLIP with their performance gap  $\Delta\text{Acc}$  across different layers. It's observed that as the layers go deeper, the degree of performance degradation  $\Delta\text{Acc}$  is more consistent with the prompt embedding distance. Specifically, in the first few layers (e.g., Layer 3 & 6) the correlations between  $\Delta\text{Acc}$  and the prompt distance are irregular, while in the last several layers (e.g., Layer 9 &

12) the trends between  $\Delta\text{Acc}$  and the prompt distance become more positively correlated. This indicates the distance between the learnable prompts and the hand-crafted prompts in the pretrained CLIP can be viewed as clear signs to indicate the model generalization performance over downstream tasks. Based on preceding observations, we propose to minimize the feature discrepancy between the learnable prompts and hand-crafted prompts of the pretrained CLIP in the last several  $S$  layers, to boost the generalization performance on novel classes. Specifically, for the reciprocal  $s_{th}$  layer, we maximize the feature similarity between the learnable prompts in the text branch of COMMA and the hand-crafted prompts in the text branch of the pretrained CLIP via a cosine similarity  $\text{sim}()$  as:

$$\mathcal{L}_{\text{kd}} = \text{Sim}(P_s^l, P_s^{\text{CLIP}}) \quad (8)$$

Overall, we minimize the cross-entropy loss as well as the feature discrepancy loss with the weight  $\lambda$  over the reciprocal  $S$  layers to train our COMMA as :

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{ce}} + \lambda \sum_{i=0}^S (1 - L_{\text{kd}}^i). \quad (9)$$

## Experiments

### Benchmark Setting

**Base-to-Novel Generalization:** The datasets are split into base and novel classes to evaluate the model in a zero-shot manner. The model is trained on the base classes in a few-shot setting and evaluated on base and novel classes.

**Cross-dataset Evaluation:** To demonstrate the ability of our model in cross-dataset transfer, we train our model on the ImageNet dataset in a few-shot manner, and directly evaluate it on other datasets without further fine-tuning.

**Domain Generalization:** To evaluate the robustness of our model over out-of-distribution data, we directly test our ImageNet-trained model on four other ImageNet datasets which contain different types of domain shifts.

**Datasets** : For **base-to-novel generalization** and **cross-dataset evaluation**, we follow previous methods (Khattak et al. 2023; Yao, Zhang, and Xu 2023) to evaluate the performance of our method on 11 image classification datasets, including two generic-objects datasets, ImageNet (Deng et al. 2009) and Caltech101 (Fei-Fei, Fergus, and Perona 2004); five fine-grained datasets, OxfordPets (Parkhi et al. 2012), StanfordCars (Krause et al. 2013), Flowers102 (Nilsback and Zisserman 2008), Food101 (Bossard, Guillaumin, and Van Gool 2014), and FGVC Aircraft (Maji et al. 2013); a scene recognition dataset SUN397 (Xiao et al. 2010); an action recognition dataset UCF101 (Soomro, Zamir, and Shah 2012); a texture dataset DTD (Cimpoi et al. 2014) and a satelliteimage dataset EuroSAT (Helber et al. 2019). For **domain generalization**, we use ImageNet as the source dataset and its four variants as target datasets including ImageNetV2 (Recht et al. 2019), ImageNet-Sketch (Wang et al. 2019), ImageNet-A (Hendrycks et al. 2021b) and ImageNet-R (Hendrycks et al. 2021a).

**Implementation Details** For all experiments, we use the pretrained ViT-B/16 CLIP model by default with  $d_l = 512$ ,  $d_t = 768$ . We use a 16-shot training strategy in all experiments by default which randomly samples 16 shots for each class. Following previous methods (Khattak et al. 2023), we set prompt depth  $J$  to 9 and the language and vision prompt lengths to 2. We train our models for 5 epochs with a batch-size of 4 and a learning rate of 0.0035 with the SGD optimizer. We use the pretrained CLIP word embeddings of the template ‘a photo of a [category]’ to initialize the language prompts of the first layer  $P_0$ , and randomly initialize the prompts of the subsequent layers with a normal distribution. For **base-to-novel generalization**, we report base and novel class accuracies and their harmonic mean (HM) averaged over 3 runs. For **cross-dataset evaluation** and **domain generalization**, we train our model on the ImageNet dataset as a source model for 2 epochs with a learning rate of 0.0026, and set the prompt depth  $J$  as 3.

### Base-to-Novel Generalization

We split each dataset into two disjoint groups: base classes (Base) and new classes (New). The model is trained on the base classes and directly evaluated on the unseen new classes to validate its generalizability. We compare our COMMA with recent methods in tab. 1. Totally, COMMA achieves improved performance in 9/11 datasets upon new classes and higher harmonic mean accuracies over 10/11 datasets compared to state-of-the-art methods, demonstrating better generalizability over novel concepts. Specifically, previous methods like CoOp, CoCoOp and KgCoOp usually achieve large improvements upon base classes compared to CLIP. However, they often own worse performance on the novel classes. This is because they easily overfit the training data and lack generalization on unseen data. As a strong competitor, MaPLe introduces multi-modal prompt learning to alleviate this issue and improves a lot over new classes compared to previous methods. Our COMMA not only demonstrates much superior performance over all base classes compared to CLIP, but also shows stronger accuracy upon most (9/11) new classes with impressive generalizability, with a higher averaged harmonic mean accuracy upon all datasets. It’s worth noting that compared to previous methods, our COMMA just obtains higher performance over 3/11 datasets on base classes, but achieves better accuracy over 9/11 datasets on novel classes.

### Cross-Dataset Transfer

We test cross-dataset generalization of COMMA in tab. 2. The model is trained on the ImageNet dataset and directly evaluated on the remaining 10 datasets. It’s observed that COMMA achieves competitive performance with other methods on the source ImageNet dataset, but achieves much stronger performance over the target datasets. Especially, COMMA outperforms CoOp and MaPLe over 9/10 datasets and beats CoCoOp over all datasets. Overall, COMMA achieves the highest averaged performance over all 10 datasets, with better generalizability over downstream tasks.

	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	69.34	74.22	71.70	CLIP	72.43	68.14	70.22	CLIP	96.84	94.00	95.40
CoOp	<b>82.63</b>	67.99	74.60	CoOp	<b>76.46</b>	66.31	71.02	CoOp	<b>98.11</b>	93.52	95.76
CoCoOp	80.47	71.69	75.83	CoCoOp	75.98	70.43	73.10	CoCoOp	97.96	93.81	95.84
KgCoOp	80.73	73.60	77.00	KgCoOp	75.83	69.96	72.78	KgCoOp	97.72	94.39	96.03
MaPLe	82.28	75.14	78.55	MaPLe	76.66	70.54	73.47	MaPLe	97.74	94.36	96.02
COMMA	82.42	<b>75.87</b>	<b>79.04</b>	COMMA	76.04	<b>70.89</b>	<b>73.86</b>	COMMA	97.94	<b>94.56</b>	<b>96.50</b>
(a) Average over 11 datasets.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	91.17	97.26	94.12	CLIP	63.37	74.89	68.65	CLIP	72.08	<b>77.80</b>	74.83
CoOp	94.24	96.66	95.43	CoOp	<b>76.20</b>	69.14	72.49	CoOp	<b>97.63</b>	69.55	81.23
CoCoOp	95.20	97.69	96.43	CoCoOp	70.49	73.59	72.01	CoCoOp	94.87	71.75	81.71
KgCoOp	94.65	97.76	96.18	KgCoOp	71.76	75.04	73.36	KgCoOp	95.00	74.73	83.65
MaPLe	95.43	97.76	96.58	MaPLe	72.94	74.00	73.47	MaPLe	95.92	72.46	82.56
COMMA	<b>95.62</b>	<b>97.84</b>	<b>96.72</b>	COMMA	73.48	<b>74.91</b>	<b>73.96</b>	COMMA	94.86	75.13	<b>83.88</b>
(b) ImageNet.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	90.10	91.22	90.66	CLIP	27.19	<b>36.29</b>	31.09	CLIP	69.36	75.35	72.23
CoOp	89.44	87.50	88.46	CoOp	<b>39.24</b>	30.49	34.30	CoOp	80.85	68.34	74.07
CoCoOp	90.70	91.29	90.99	CoCoOp	33.41	23.71	27.74	CoCoOp	79.74	76.86	78.27
KgCoOp	90.50	91.70	91.09	KgCoOp	36.21	33.55	34.83	KgCoOp	80.29	76.53	78.36
MaPLe	<b>90.71</b>	92.05	91.38	MaPLe	37.44	35.61	<b>36.50</b>	MaPLe	80.82	78.70	79.75
COMMA	90.42	<b>92.74</b>	<b>91.84</b>	COMMA	36.47	34.23	35.84	COMMA	<b>80.94</b>	<b>79.32</b>	<b>80.86</b>
(c) Caltech101.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	69.36	75.35	72.23	CLIP	70.53	77.50	73.85	CLIP	85.14	64.47	73.37
CoOp	80.85	68.34	74.07	CoOp	82.33	73.45	77.64	CoOp	82.89	76.67	79.65
CoCoOp	79.74	76.86	78.27	CoCoOp	83.00	78.66	80.77	CoCoOp	84.06	<b>80.56</b>	<b>81.84</b>
COMMA	<b>80.94</b>	<b>79.32</b>	<b>80.86</b>	COMMA	93.56	<b>74.26</b>	<b>83.42</b>	COMMA	84.06	<b>80.56</b>	<b>81.84</b>
(d) OxfordPets.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	90.10	91.22	90.66	CLIP	27.19	<b>36.29</b>	31.09	CLIP	70.53	77.50	73.85
CoOp	89.44	87.50	88.46	CoOp	<b>39.24</b>	30.49	34.30	CoOp	85.14	64.47	73.37
CoCoOp	90.70	91.29	90.99	CoCoOp	33.41	23.71	27.74	CoCoOp	82.33	73.45	77.64
KgCoOp	90.50	91.70	91.09	KgCoOp	36.21	33.55	34.83	KgCoOp	82.89	76.67	79.65
MaPLe	<b>90.71</b>	92.05	91.38	MaPLe	37.44	35.61	<b>36.50</b>	MaPLe	83.00	78.66	80.77
COMMA	90.42	<b>92.74</b>	<b>91.84</b>	COMMA	36.47	34.23	35.84	COMMA	<b>80.94</b>	<b>79.32</b>	<b>80.86</b>
(e) StanfordCars.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	27.19	<b>36.29</b>	31.09	CLIP	56.48	64.05	60.03	CLIP	70.53	77.50	73.85
CoOp	<b>39.24</b>	30.49	34.30	CoOp	91.54	54.44	68.27	CoOp	85.14	64.47	73.37
CoCoOp	33.41	23.71	27.74	CoCoOp	87.49	60.04	71.21	CoCoOp	82.33	73.45	77.64
KgCoOp	36.21	33.55	34.83	KgCoOp	85.64	64.34	73.48	KgCoOp	82.89	76.67	79.65
MaPLe	37.44	35.61	<b>36.50</b>	MaPLe	<b>94.07</b>	73.23	82.35	MaPLe	83.00	78.66	80.77
COMMA	<b>90.42</b>	58.62	<b>68.32</b>	COMMA	93.56	<b>74.26</b>	<b>83.42</b>	COMMA	84.06	<b>80.56</b>	<b>81.84</b>
(f) Flowers102.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	69.36	75.35	72.23	CLIP	70.53	77.50	73.85	CLIP	85.14	64.47	73.37
CoOp	80.85	68.34	74.07	CoOp	82.33	73.45	77.64	CoOp	82.89	76.67	79.65
CoCoOp	79.74	76.86	78.27	CoCoOp	83.00	78.66	80.77	CoCoOp	84.06	<b>80.56</b>	<b>81.84</b>
COMMA	<b>80.94</b>	<b>79.32</b>	<b>80.86</b>	COMMA	93.56	<b>74.26</b>	<b>83.42</b>	COMMA	84.06	<b>80.56</b>	<b>81.84</b>
(g) Food101.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	53.24	<b>59.90</b>	56.37	CLIP	56.48	64.05	60.03	CLIP	70.53	77.50	73.85
CoOp	80.17	47.54	59.68	CoOp	91.54	54.44	68.27	CoOp	85.14	64.47	73.37
CoCoOp	77.01	56.00	64.85	CoCoOp	87.49	60.04	71.21	CoCoOp	82.33	73.45	77.64
KgCoOp	77.55	54.99	64.35	KgCoOp	85.64	64.34	73.48	KgCoOp	82.89	76.67	79.65
MaPLe	80.36	59.18	68.16	MaPLe	<b>94.07</b>	73.23	82.35	MaPLe	83.00	78.66	80.77
COMMA	<b>81.04</b>	58.62	<b>68.32</b>	COMMA	93.56	<b>74.26</b>	<b>83.42</b>	COMMA	84.06	<b>80.56</b>	<b>81.84</b>
(h) FGVC Aircraft.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	27.19	<b>36.29</b>	31.09	CLIP	56.48	64.05	60.03	CLIP	70.53	77.50	73.85
CoOp	<b>39.24</b>	30.49	34.30	CoOp	91.54	54.44	68.27	CoOp	85.14	64.47	73.37
CoCoOp	33.41	23.71	27.74	CoCoOp	87.49	60.04	71.21	CoCoOp	82.33	73.45	77.64
KgCoOp	36.21	33.55	34.83	KgCoOp	85.64	64.34	73.48	KgCoOp	82.89	76.67	79.65
MaPLe	37.44	35.61	<b>36.50</b>	MaPLe	<b>94.07</b>	73.23	82.35	MaPLe	83.00	78.66	80.77
COMMA	<b>90.42</b>	58.62	<b>68.32</b>	COMMA	93.56	<b>74.26</b>	<b>83.42</b>	COMMA	84.06	<b>80.56</b>	<b>81.84</b>
(i) SUN397.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	69.36	75.35	72.23	CLIP	70.53	77.50	73.85	CLIP	85.14	64.47	73.37
CoOp	80.85	68.34	74.07	CoOp	82.33	73.45	77.64	CoOp	82.89	76.67	79.65
CoCoOp	79.74	76.86	78.27	CoCoOp	83.00	78.66	80.77	CoCoOp	84.06	<b>80.56</b>	<b>81.84</b>
COMMA	<b>80.94</b>	<b>79.32</b>	<b>80.86</b>	COMMA	93.56	<b>74.26</b>	<b>83.42</b>	COMMA	84.06	<b>80.56</b>	<b>81.84</b>
(j) DTD.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	27.19	<b>36.29</b>	31.09	CLIP	56.48	64.05	60.03	CLIP	70.53	77.50	73.85
CoOp	<b>39.24</b>	30.49	34.30	CoOp	91.54	54.44	68.27	CoOp	85.14	64.47	73.37
CoCoOp	33.41	23.71	27.74	CoCoOp	87.49	60.04	71.21	CoCoOp	82.33	73.45	77.64
KgCoOp	36.21	33.55	34.83	KgCoOp	85.64	64.34	73.48	KgCoOp	82.89	76.67	79.65
MaPLe	37.44	35.61	<b>36.50</b>	MaPLe	<b>94.07</b>	73.23	82.35	MaPLe	83.00	78.66	80.77
COMMA	<b>90.42</b>	58.62	<b>68.32</b>	COMMA	93.56	<b>74.26</b>	<b>83.42</b>	COMMA	84.06	<b>80.56</b>	<b>81.84</b>
(k) EuroSAT.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	27.19	<b>36.29</b>	31.09	CLIP	56.48	64.05	60.03	CLIP	70.53	77.50	73.85
CoOp	<b>39.24</b>	30.49	34.30	CoOp	91.54	54.44	68.27	CoOp	85.14	64.47	73.37
CoCoOp	33.41	23.71	27.74	CoCoOp	87.49	60.04	71.21	CoCoOp	82.33	73.45	77.64
KgCoOp	36.21	33.55	34.83	KgCoOp	85.64	64.34	73.48	KgCoOp	82.89	76.67	79.65
MaPLe	37.44	35.61	<b>36.50</b>	MaPLe	<b>94.07</b>	73.23	82.35	MaPLe	83.00	78.66	80.77
COMMA	<b>90.42</b>	58.62	<b>68.32</b>	COMMA	93.56	<b>74.26</b>	<b>83.42</b>	COMMA	84.06	<b>80.56</b>	<b>81.84</b>
(l) UCF101.											
	Base	New	HM		Base	New	HM		Base	New	HM
CLIP	27.19	<b>36.29</b>	31.09	CLIP	56.48	64.05	60.03	CLIP	70.53	77.50	73.85
CoOp	<b>39.24</b>	30.49	34.30	CoOp	91.54	54.44	68.27	CoOp	85.14	64.47	73.37
CoCoOp	33.41	23.71	27.74	CoCoOp	87.49	60.04	71.21	CoCoOp	82.33	73.45	77.64
KgCoOp	36.21	33.55	34.83	KgCoOp	85.64	64.34	73.48	KgCoOp	82.89	76.67	79.65
MaPLe	37.44	35.61	<b>36.50</b>	MaPLe	<b>94.07</b>	73.23	82.35	MaPLe	83.00	78.66	80.77
COMMA	<b>90.42</b>	58.62	<b>68.32</b>	COMMA	93.56	<b>74.26</b>	<b>83.42</b>	COM			

Source	Target											
	ImageNet	Caltech101	OxfordPets	S-Cars	Flowers102	Food101	Aircraft	SUN397	DTD	EuroSAT	UCF101	Average
CoOp	<b>71.51</b>	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
CoCoOp	71.02	94.43	90.14	65.32	71.88	86.06	22.94	67.36	45.73	45.37	68.21	65.74
MaPLe	70.72	93.53	90.49	65.57	72.23	<b>86.20</b>	24.74	67.01	46.49	48.06	68.69	66.30
COMMA	71.22	<b>93.84</b>	<b>90.78</b>	<b>66.36</b>	<b>73.14</b>	85.87	<b>25.14</b>	<b>67.56</b>	<b>46.52</b>	<b>48.85</b>	<b>68.71</b>	<b>66.84</b>

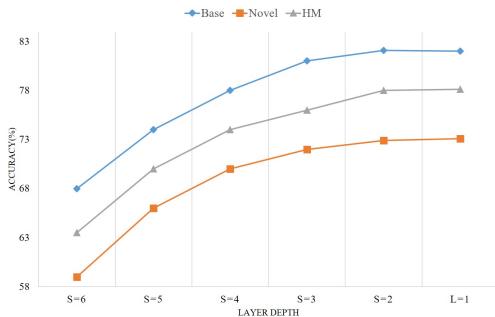
Table 2: Comparison of COMMA with recent methods on the cross-dataset evaluation setting.

Source	Target				
	ImgNet	ImgNetV2	ImgNet-S	ImgNet-A	ImgNet-R
CLIP	66.73	60.83	46.15	47.77	73.96
CoOp	<b>71.51</b>	64.20	47.99	49.71	75.21
CoCoOp	71.02	64.07	48.75	50.63	76.18
MaPLe	70.72	64.07	49.15	50.90	76.98
KgCoOp	71.20	64.10	48.97	50.69	76.70
COMMA	71.22	<b>64.84</b>	<b>49.65</b>	<b>51.64</b>	<b>77.56</b>

Table 3: Comparison of COMMA with existing approaches in the domain generalization setting.

Configurations	Accuracy(%)
-	75.86
w/ correlated prompts	78.02
w/ knowledge transfer	76.84
COMMA	<b>79.04</b>

Table 4: Effectiveness of each component in COMMA.

Figure 4: Relationships concerning the base-class, novel-class and harmonic mean accuracy with the number of reciprocal layers ( $S$ ). Accuracies are averaged over 11 datasets.

frames per second (FPS) in tab. 5. It's noticed the CoCoOp has the lowest FPS due to its instance-conditioned design. Its FPS decreases as the batch size rises, which greatly hinders its application in real life. With improved accuracy compared to previous methods (CoOp, CoCoOp and KgCoOp), MaPLe increases the required parameters from 2048 to 3.55M. Compared to other methods, our COMMA owns slightly increased parameters with comparable FPS

Method	Params	FPS (with BS)		HM
		1	16	
CoOp	2048	9.4	147.6	71.66
CoCoOp	35360	45.2	726.2	75.83
KgCoOp	2048	40.1	642.3	77.00
MaPLe	3.55M	39.8	639.8	78.55
COMMA	4.87M	39.6	637.8	<b>79.04</b>

Table 5: Comparison of prompting efficiency with other methods over 11 datasets. 'BS' denotes 'Batch Size'.

$\lambda$	0.5	1.0	1.5	2.0	4.0	8.0
Accuracy	78.68	<b>79.04</b>	78.74	78.62	78.21	78.02

Table 6: Effects for the weight  $\lambda$  over 11 datasets.

$J=3$	$J=6$	$J=9$	$J=12$	$P=1$	$P=2$	$P=3$	$P=4$
77.98	78.56	<b>79.04</b>	78.74	78.64	<b>79.04</b>	78.75	78.54

Table 7: Ablations for prompt depth  $J$  and prompt length  $P$ .

and the highest averaged performance over all 11 datasets, demonstrating better accuracy with competitive computational costs.

**The choice of  $\lambda$ .** We test the choice for the weight  $\lambda$  of the knowledge transfer loss  $\mathcal{L}_{kg}$  in tab. 6. As  $\lambda$  rises, the accuracy increases and reaches the peak when  $\lambda=1.0$ . We set  $\lambda=1.0$  by default.

**The choices of prompt depth  $J$  and prompt length  $P$ .** We ablate the choices of prompt depth  $J$  and prompt length  $P$  in tab. 7. It's observed that  $J=9$  and  $P=2$  could offer the best results.

## Conclusion

Adapting large VLMs for downstream tasks is challenging due to the large scale of optimized parameters and limited size of downstream data. Prompt learning is an efficient promising approach to adapt VLMs to novel concepts. To increase the generalization performance of VLMs, we propose to enhance the correlations of multi-modal prompts and preserve generic knowledge during the fine-tuning process. Experimental results show that our model is both parameter-efficient and robust across a series of downstream tasks.

## Acknowledgments

This work is supported by National Key Research and Development Program of China (2020YFC1522700) and National Natural Science Foundation of China (Project No. 62072334).

## References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6077–6086.

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101—mining discriminative components with random forests. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VI 13*, 446–461. Springer.

Cheng, X.; Lin, H.; Wu, X.; Yang, F.; and Shen, D. 2021. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *arXiv preprint arXiv:2109.04290*.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3606–3613.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Ding, J.; Xue, N.; Xia, G.-S.; and Dai, D. 2022. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11583–11592.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fang, H.; Xiong, P.; Xu, L.; and Chen, Y. 2021. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, 178–178. IEEE.

Feng, C.; Zhong, Y.; Jie, Z.; Chu, X.; Ren, H.; Wei, X.; Xie, W.; and Ma, L. 2022. Promptdet: Towards open-vocabulary detection using uncurated images. In *European Conference on Computer Vision*, 701–717. Springer.

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. *arXiv preprint arXiv:2110.04544*.

Gu, X.; Lin, T.-Y.; Kuo, W.; and Cui, Y. 2021. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*.

Helber, P.; Bischke, B.; Dengel, A.; and Borth, D. 2019. Eurusat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7): 2217–2226.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; et al. 2021a. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8340–8349.

Hendrycks, D.; Zhao, K.; Basart, S.; Steinhardt, J.; and Song, D. 2021b. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15262–15271.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, 2790–2799. PMLR.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, 4904–4916. PMLR.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *European Conference on Computer Vision*, 709–727. Springer.

Jiang, H.; Misra, I.; Rohrbach, M.; Learned-Miller, E.; and Chen, X. 2020. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10267–10276.

Jin, W.; Cheng, Y.; Shen, Y.; Chen, W.; and Ren, X. 2021. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. *arXiv preprint arXiv:2110.08484*.

Khattak, M. U.; Rasheed, H.; Maaz, M.; Khan, S.; and Khan, F. S. 2023. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19113–19122.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 554–561.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 12888–12900. PMLR.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35.

Liu, X.; Ji, K.; Fu, Y.; Tam, W. L.; Du, Z.; Yang, Z.; and Tang, J. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Lüddecke, T.; and Ecker, A. 2022. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7086–7096.

Maaz, M.; Rasheed, H.; Khan, S.; Khan, F. S.; Anwer, R. M.; and Yang, M.-H. 2022. Class-agnostic object detection with multi-modal transformer. In *European Conference on Computer Vision*, 512–531. Springer.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

Nguyen, D.-K.; Goswami, V.; and Chen, X. 2020. Movie: Revisiting modulated convolutions for visual counting and beyond. *arXiv preprint arXiv:2004.11883*.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, 722–729. IEEE.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, 3498–3505. IEEE.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2019. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, 5389–5400. PMLR.

Shu, M.; Nie, W.; Huang, D.-A.; Yu, Z.; Goldstein, T.; Anandkumar, A.; and Xiao, C. 2022. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35: 14274–14289.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sung, Y.-L.; Cho, J.; and Bansal, M. 2022. Vi-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5227–5237.

Wang, H.; Ge, S.; Lipton, Z.; and Xing, E. P. 2019. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32.

Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J.; et al. 2022. Dual-prompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, 631–648. Springer.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, 3485–3492. IEEE.

Yao, H.; Zhang, R.; and Xu, C. 2023. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6757–6767.

Yao, L.; Huang, R.; Hou, L.; Lu, G.; Niu, M.; Xu, H.; Liang, X.; Li, Z.; Jiang, X.; and Xu, C. 2021. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*.

Zhang, R.; Fang, R.; Zhang, W.; Gao, P.; Li, K.; Dai, J.; Qiao, Y.; and Li, H. 2021. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16816–16825.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348.

Zhu, B.; Niu, Y.; Han, Y.; Wu, Y.; and Zhang, H. 2022. Prompt-aligned gradient for prompt tuning. *arXiv preprint arXiv:2205.14865*.