Task-Aware Clustering for Prompting Vision-Language Models

Fusheng Hao[1,2] Fengxiang He[3] Fuxiang Wu[1,2] Tichao Wang[1,2] Chengqun Song[1,2] Jun Cheng[1,2*]
[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[2] The Chinese University of Hong Kong [3] University of Edinburgh

Abstract

Prompt learning has attracted widespread attention in adapting vision-language models to downstream tasks. Existing methods largely rely on optimization strategies to ensure the task-awareness of learnable prompts. Due to the scarcity of task-specific data, overfitting is prone to occur. The resulting prompts often do not generalize well or exhibit limited task-awareness. To address this issue, we propose a novel Task-Aware Clustering (TAC) framework for prompting vision-language models, which increases the task-awareness of learnable prompts by introducing task-aware pre-context. The key ingredients are as follows: (a) generating task-aware pre-context based on task-aware clustering that can preserve the backbone structure of a downstream task with only a few clustering centers, (b) enhancing the task-awareness of learnable prompts by enabling them to interact with task-aware pre-context via the well-pretrained encoders, and (c) preventing the visual task-aware pre-context from interfering the interaction between patch embeddings by masked attention mechanism. Extensive experiments are conducted on benchmark datasets, covering the base-to-novel, domain generalization, and cross-dataset transfer settings. Ablation studies validate the effectiveness of key ingredients. Comparative results show the superiority of our TAC over competitive counterparts. The code is available at https://github.com/FushengHao/TAC.

1. Introduction

Large-scale visual and textual data pairs have been collected and employed to pretrain high-capacity Vision-Language Models (VLMs) [1, 20, 37]. Due to the alignment of the visual and textual modalities, the resulting models have emerged with remarkable generalization ability on various downstream tasks such as zero-shot image classification [37] and open-vocabulary object detection [11]. Despite achieving decent performance, the performance is still lower than that of the customized models [61]. Considering the data-hungry nature of VLMs and the scarcity of task-specific data, it is difficult to improve performance through finetuning such models. As an alternative to finetuning, prompting has become as a powerful paradigm for adapting VLMs to downstream tasks [4, 7, 21, 22, 29, 51].

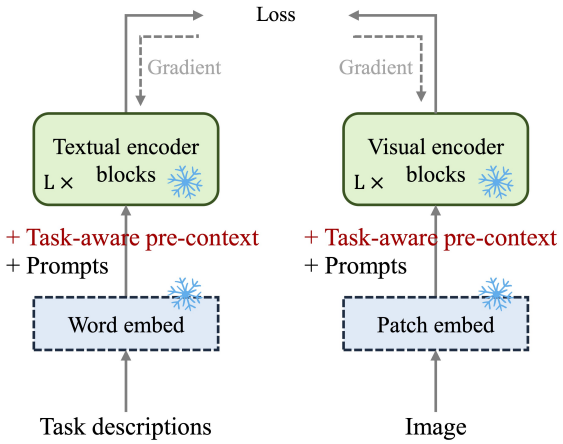

Figure 1. Motivation of our TAC. We improve the task-awareness of learnable prompts from a new perspective, i.e., introducing task-aware pre-context.

Prompting is a technique that adapts VLMs by increasing their understanding of downstream tasks. The current methodology is to provide VLMs with task-relevant context while keeping the model weights frozen. The context is also known as prompts. Many effective prompt templates have been engineered. For example, textual templates like "a photo of a [class]" [37] have been hand-crafted to provide task-relevant context for the textual encoder, and visual templates like "a red circle around an [object]" [2] have been discovered to direct the visual encoder's attention to the [object]. To avoid the need for large-scale trial and error in prompt engineering, context optimization [61] has been proposed, which treats learnable vectors as prompts and ensure their task-awareness by optimization. This strategy has attracted much attention in recent years. For example, maximizing the ground-truth class scores [61] is proposed to ensure the task-awareness of learnable prompts, and self-regulating constraints [23] are further introduced to guide

*Corresponding author: jun.cheng@siat.ac.cn

the optimization of learnable prompts for better generalization. It is worth noting that the number of labeled images per class used for prompt learning is not sufficient in the setting of context optimization[1]. Due to the scarcity of task-specific data, the resulting prompts are often overfitted [23] and do not generalize well, meaning that the task-awareness they exhibit is limited.

Instead of imposing stronger constraints [39, 45, 46], we propose to increase the task-awareness by directly introducing task-aware pre-context; see Figure 1. Considering that the class descriptions of a downstream task are given when encountering the task, the class embeddings extracted by using the well-pretrained textual encoder are good candidates for task-aware pre-context. However, one big challenge arises in this practice, i.e., the number of extra embeddings it incurs is large and equal to that of classes involved. For example, the number is 1000 for the popular dataset ImageNet [9], which is significantly larger than the typical values of the number of embeddings input to the textual and visual encoders, i.e., 77 and 197, respectively. This not only incurs large additional computational overhead, but also exceeds the maximum input limit for the textual encoder. These difficulties inspire us to keep only the backbone structure of class embeddings, which is helpful for reducing the number of embeddings incurred significantly. Motivated by these observations, we explore the prompting strategy based on clustering.

In this work, we propose a novel Task-Aware Clustering (TAC) framework for prompting VLMs. The central idea of TAC is to exploit task-aware pre-context to improve the task-awareness of learnable prompts. To generate task-aware pre-context, we conduct clustering on the class embeddings of a downstream task and perform linear transformation on the clustering centers to yield task-aware pre-context, which leads to a task-aware pre-context generalization module that can keep the backbone structure of the downstream task while being lightweight. Then, the task-aware pre-context is combined with the learnable prompts and their interaction through the well-pretrained encoders is exploited to reinforce the task-awareness of learnable prompts, which leads to a task-awareness reinforcement module. Further, a mask matrix is developed and injected into the attention mechanism to eliminate the undesired interference of visual task-aware pre-context to patch embeddings induced by the gap between textual and visual modalities, which leads to undesired interaction masking module. To suppress the overfitting induced by the scarcity of task-specific data, the prompted class and visual embeddings are encouraged to be consistent with their CLIP [37] peers, which leads to a self-regulation module. Our TAC maintains simplicity in design and improves the task-awareness of the resulting prompts, thus yielding competitive results.

[1]The typical value of this number is 16

In summary, our task-aware clustering framework for prompting has the following main contributions:

- We propose to increase the task-awareness of learnable prompts from a new perspective, i.e., introducing task-aware pre-context.
- We propose to generate task-aware pre-context based on task-aware clustering that can preserve the backbone structure of a downstream task.
- We exploit the well-pretrained encoders to reinforce the task-awareness of learnable prompts while masking out the undesired interaction.
- We demonstrate the superiority of our TAC in the base-to-novel, domain generalization, and cross-dataset transfer settings.

2. Related Works

Vision-Language Models (VLMs) integrate the textual and visual modalities, bridging the gap between them through alignment. Such models are usually pretrained on large-scale visual and textual data pairs [6, 20, 37]. They can be broadly classified into four categories according to their training paradigms [12]. The first category employs contrastive-based training [19, 40], which emphasizes reducing the distance between positive pairs and increasing the distance between negative pairs. The second category utilizes masking-based training [27, 49], which emphasizes the reconstruction of missing patches/words by exploiting the unmasked caption/image. The third category leverages generative-based training [54, 55], which emphasizes the generation of entire images or captions based on conditional captions or images. The fourth category employs LLM-based training [5, 63], which emphasizes the usage of large language models to extract supervisory information. Since the textual and visual modalities are aligned in a shared embedding space, VLMs exhibits a strong ability to understand open-vocabulary concepts, which facilitates various downstream tasks. Although VLMs have achieved decent performance, their accuracies are still far inferior to those of customized models [61]. Considering the data-hungry nature of VLMs and the scarcity of task-specific data, finetuning such models to improve their performance faces significant challenges. As an alternative to finetuning, prompting [37, 61] has attracted much attention in adapting VLMs. Orthogonal directions include the utilization of unlabeled images [43] and historical test data [57].

Prompting adapts VLMs by providing them with contextual information or task-aware prompts while freezing their model weights, which has achieved good performance in many downstream tasks [13, 17, 24, 31, 48]. Existing methods can be broadly classified into two categories according to whether domain expertise is required. The first category relies on hand-crafted prompt templates, which emphasizes

the reliance on domain expertise. CLIP [37] crafts the textual template, i.e., "a photo of a [class]". CirclePrompt [2] engineers the visual template, i.e., "a red circle around an [object]". LaViP [26] grounds visual prompt with language. ArGue [45] crafts attribute-guided prompts with the help of large language models. The second category relies on prompt learning [61], which emphasizes the introduction of learnable prompts and the optimization of these prompts to ensure their task-awareness. PromptSRC [23] guides the optimization of learnable prompts with self-regulating constraints. GalLoP [28] learns global-local prompts to capture multi-scale visual information. CoPrompt [39] improves the generation ability of learnable prompts by encouraging the prediction consistency between the pretrained and trainable models. TCP [52] introduces class-aware prompts that are produced by performing textual knowledge embedding on class embeddings. In this work, we increase the task-awareness by introducing task-aware pre-context, which is different from the current practice that imposes extra constraints.

Clustering is a technique that can group similar data points into clusters or groups, in which K-Means is a classic example. Integrating clustering may improve the capabilities of deep models in capturing backbone structures, and their combination has attracted a lot of attention. DivClust [34] introduces a diversity controlling loss to ensure the desired degree of diversity for the produced multiple clusters. LCP-GAN [56] associates intra-class clusters with learnable visual concepts to capture the class-wise and intra-class information for image synthesis. ZeroShotULD [41] clusters random pixel locations with nearest neighbor searching for unsupervised landmarks discovery. SPC [18] replaces the deterministic prediction with semantic clustering for domain generalized semantic segmentation. MoDE [33] decides clusters for data experts by a two-step clustering on captions and uses fine-grained clusters to coarsely represent data experts. LTE [53] clusters multiple observed motions by establishing the mapping from trajectories to trajectory embeddings. In this work, we perform clustering to preserve the backbone structure of a downstream task represented by class embeddings.

3. Method

In this section, we begin by reviewing the preliminaries and then detail our Task-Aware Clustering (TAC) framework for prompting vision-language models.

3.1. Preliminaries

CLIP. CLIP [37] includes a textual encoder and a visual encoder, and is pretrained on around 400 million visual and textual data pairs with the contrastive-based training strategy. With the hand-crafted prompt templates like "a photo of a [class]", it emerges strong zero-shot prediction capabilities. Although CLIP has achieved good performance in the zero-shot image recognition task, its accuracies are still lower than those of the customized models [61]. Due to the data-hungry nature of CLIP and the scarcity of task-specific data, finetuning CLIP to improve its performance on downstream tasks is particularly challenging. Considering that prompt engineering relies on domain expertise and needs large-scale trial and error, prompt learning [61] has been proposed, in which learnable vectors are treated as prompts and their task-awareness is ensured through optimization. For ease of comparison with existing prompt learning approaches, we follow the current practice and choose CLIP as the foundation model.

Prompt learning. CoOp [61] is a typical approach for adapting CLIP in the context of prompt learning. We take a C-class image classification task as an example to show how learnable prompts are optimized in CoOp. First, a set of learnable vectors with a quantity of M^t is introduced and treated as textual prompts, i.e., $\{\boldsymbol{p}_m^t\}_{m=1}^{M^t}$. Second, these prompts are combined with the word embeddings that describe the task, which leads to a learnable version of task descriptions, i.e., $\{\boldsymbol{p}_1^t, \cdots, \boldsymbol{p}_{M^t}^t, \boldsymbol{c}_i\}_{i=1}^{C}$, where $\boldsymbol{c}_i$ is the word embedding(s) of the i-th class. Third, the learnable task descriptions are fed into the textual encoder to yield their embeddings $\{\boldsymbol{w}_c^{\text{CoOp}}\}_{c=1}^{C}$, while an image $\boldsymbol{x}$ is fed into the visual encoder to generate its embedding $\boldsymbol{f}$. Fourth, the probability of $\boldsymbol{x}$ that belongs to the c-th class can be calculated as:

$$p^{\text{CoOp}}(c|\boldsymbol{x}) = \frac{\exp(\cos(\boldsymbol{w}_c^{\text{CoOp}}, \boldsymbol{f})/\tau)}{\sum_{i=1}^{C} \exp(\cos(\boldsymbol{w}_i^{\text{CoOp}}, \boldsymbol{f})/\tau)}, \quad (1)$$

where τ is the temperature and $\cos(\cdot,\cdot)$ denotes the cosine similarity. Finally, the task-awareness of learnable textual prompts is ensured by minimizing the cross-entropy loss:

$$\mathcal{L}_{\text{CE}}{}^{\text{CoOp}} = -\sum_{c=1}^{C} \mathbb{I}(c=y) \log p^{\text{CoOp}}(c|\boldsymbol{x}), \quad (2)$$

where y is the ground-truth label of $\boldsymbol{x}$ and $\mathbb{I}(\cdot)$ denotes the indicator function. It is worth noting that (a) only the learnable prompts need to be optimized and (b) the quantity of labeled images that can be used in each class is insufficient in the setting of prompt learning, where the typical value is 16.

Afterwards, additional constraints [23, 32, 39, 59] have been imposed to improve the task-awareness of learnable prompts. Different from the current practice, we increase the task-awareness from a new perspective, i.e., introducing task-aware pre-context.

3.2. Task-aware Clustering for Prompting

The pipeline of our TAC is shown in Figure 2 and the key ingredients are detailed below.

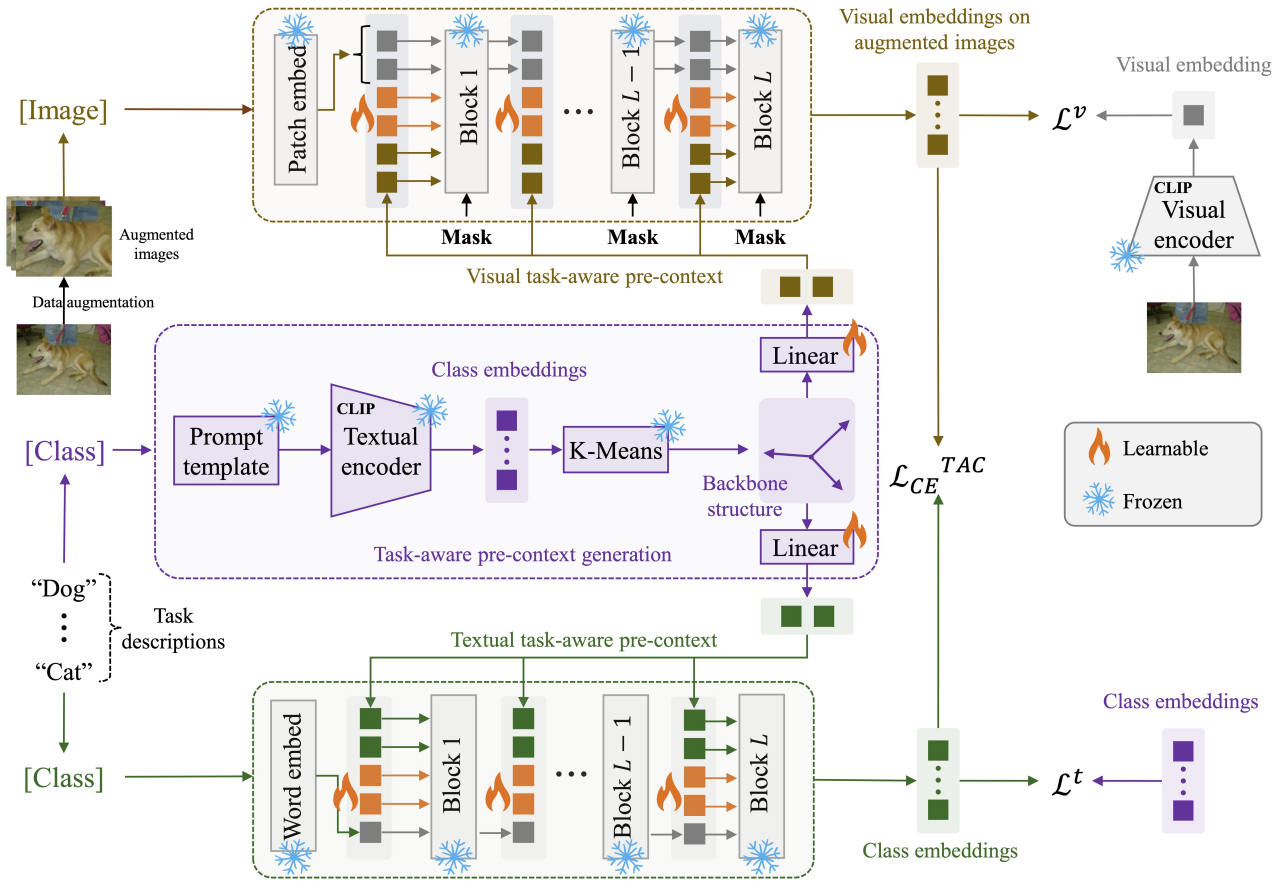

Figure 2. Pipeline of our TAC. First, the textual and visual task-aware pre-contexts are generated based on the backbone structure of a downstream task. Second, the task-aware pre-context is combined with the learnable prompts and their interaction through the well-pretrained encoders is exploited to reinforce the task-awareness of learnable prompts. Third, the undesired interaction is masked out with a customized mask matrix. Further, the textual and visual consistencies are imposed to alleviate the overfitting issue.

Task-aware pre-context generation. The class embeddings extracted by using the textual encoder of CLIP are task-aware and good candidates for the desired pre-context. Considering that the number of such embeddings is equal to that of classes involved and this number is large, applying this strategy to practice is challenging. The first challenge is that the number of extra input embeddings incurred exceeds the maximum input limit for the textual encoder, which would make the textual encoder unusable. The second challenge is that the extra computational overhead incurred is very high due to the large number of additional embeddings input into the visual encoder.

We address these challenges by yielding textual and visual task-aware pre-contexts based on clustering that can preserve the backbone structure of class embeddings. A significant additional benefit of this practice is that the number of extra embeddings incurred can be significantly reduced. Specifically, let $\{\boldsymbol{w}_c^{\text{CLIP}}\}_{c=1}^{C}$ denote the class embeddings extracted with the textual encoder of CLIP. Then, K-Means is conducted on $\{\boldsymbol{w}_c^{\text{CLIP}}\}_{c=1}^{C}$, which yields a set of clustering centers with a quantity of K, i.e., $\{\boldsymbol{\mu}_i\}_{i=1}^{K}$. Further, the textual and visual task-aware pre-contexts are generated as:

$$\boldsymbol{\mu}_i^t = \boldsymbol{W}^t \boldsymbol{\mu}_i, \tag{3}$$

$$\boldsymbol{\mu}_i^v = \boldsymbol{W}^v \boldsymbol{\mu}_i, \tag{4}$$

where $\boldsymbol{W}^t$ and $\boldsymbol{W}^v$ are the textual and visual linear transformations. It is worth noting that (a) the dimensions of $\boldsymbol{\mu}_i^t$ and $\boldsymbol{\mu}_i^v$ are set to be the internal dimensions of the textual and visual encoders, correspondingly, and (b) since only the weights of linear transformations needs to be optimized, our task-aware pre-context generation module is lightweight.

Task-awareness reinforcement. After obtaining the textual and visual task-aware pre-contexts, the task-awareness of learnable prompts is reinforced by enabling them to interact with the corresponding pre-contexts via the well-pretrained textual and visual encoders. Specifically, let L denote the quantity of transformer blocks contained in the visual/textual encoder. We construct the input embeddings to the l-th transformer block of the textual encoder as:

$$\{\boldsymbol{t}_l^{sos}, \boldsymbol{\mu}_1^t, \cdots, \boldsymbol{\mu}_K^t, \boldsymbol{p}_{l1}^t, \cdots, \boldsymbol{p}_{lM^t}^t, \boldsymbol{c}_{li}, \boldsymbol{t}_l^{eos}\}. \tag{5}$$

Here, $l = 1, \cdots, L$, $\boldsymbol{t}_l^{sos}$ is the start token input to the l-th transformer block, $\boldsymbol{t}_l^{eos}$ is the end token input to the l-th transformer block, $\{\boldsymbol{p}_{li}^t\}_{i=1}^{M^t}$ is the set of learnable textual prompts input to the l-th transformer block with a quantity of M^t, and $\boldsymbol{c}_{li}$ is the word embedding(s) input to the l-th transformer block. It is worth noting that (a) $\boldsymbol{t}_l^{sos}$, $\boldsymbol{t}_l^{eos}$, and $\boldsymbol{c}_{li}$ are the outputs of the previous transformer block and (b) $\{\boldsymbol{p}_{li}^t\}_{i=1}^{M^t}$ and $\{\boldsymbol{\mu}_i^t\}_{i=1}^{K}$ are not the outputs of the previous transformer block; see Figure 2.

Similarly, we construct the input embeddings to the l-th transformer block of the visual encoder as:

$$\{\boldsymbol{e}_l^{cls}, \boldsymbol{x}_{l1}^p, \cdots, \boldsymbol{x}_{lM}^p, \boldsymbol{\mu}_1^v, \cdots, \boldsymbol{\mu}_K^v, \boldsymbol{p}_{l1}^v, \cdots, \boldsymbol{p}_{lM^v}^v\}. \quad (6)$$

Here, $l = 1, \cdots, L$, $\boldsymbol{e}_l^{cls}$ is the class token input to the l-th transformer block, $\{\boldsymbol{x}_{lm}^p\}_{m=1}^{M}$ is the set of patch embeddings input to the l-th transformer block with a quantity of M, and $\{\boldsymbol{p}_{li}^v\}_{i=1}^{M^v}$ is the set of learnable visual prompts input to the l-th transformer block with a quantity of M^v. It is worth noting that (a) $\boldsymbol{e}_l^{cls}$ and $\{\boldsymbol{x}_{lm}^p\}_{m=1}^{M}$ are the outputs of the previous transformer block and (b) $\{\boldsymbol{p}_{li}^v\}_{i=1}^{M^v}$ and $\{\boldsymbol{\mu}_i^v\}_{i=1}^{K}$ are not the outputs of the previous transformer block; see Figure 2.

Undesired interaction masking. There is a modal gap between patch embeddings and visual task-aware pre-context. Allowing patch embeddings to interact with visual task-aware pre-context may weaken the discriminative ability of patch embeddings. To avoid this phenomenon, the information flow from visual task-aware pre-context to patch embeddings is prohibited, which is achieved by exploiting the masked attention mechanism: $\boldsymbol{A} = \text{softmax}(\boldsymbol{Q}\boldsymbol{K}^T + \boldsymbol{\mathcal{M}})$. Here, $\boldsymbol{Q}$ and $\boldsymbol{K}$ denote the query and key matrices, with the first dimension specifying the quantity of input embeddings, i.e., $N_v = 1 + M + K + M^v$. They are the linear transformations of input embeddings. $\boldsymbol{A} \in \mathbb{R}^{N_v \times N_v}$ denotes the attention matrix. $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{N_v \times N_v}$ is the customized mask matrix, which is defined as:

$$\boldsymbol{\mathcal{M}}_{ij} = \begin{cases} -\infty & \text{if } 1 < i \leq 1+M \text{ and } 1+M < j \leq 1+M+K \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

Here, $\boldsymbol{\mathcal{M}}_{ij} = -\infty$ is equal to $\boldsymbol{A}_{ij} = 0$, meaning that the i-th input embedding does not interact with the j-th peer. In effect, with $\boldsymbol{\mathcal{M}}$, patch embeddings are made to not interact with visual task-aware pre-context. It is worth noting that (a) $\boldsymbol{\mathcal{M}}$ is injected into the visual encoder in a layer-wise manner; see Figure 2, and (b) the learnable prompts acts as a glue between visual task-aware pre-context and patch embeddings.

Self-regulation. Due to the scarcity of task-specific data, overfitting can easily occur. We address this issue by exploiting the strong generation ability exhibited by the well-pretrained encoders, i.e., encouraging the prompted class and visual embeddings to be consistent with their CLIP peers. Specifically, let $\{\boldsymbol{w}_c^{\text{TAC}}\}_{c=1}^{C}$ denote the class embeddings extracted by using our TAC. The textual consistency is imposed by:

$$\mathcal{L}^t = \sum_{i=1}^{C} |\boldsymbol{w}_c^{\text{TAC}} - \boldsymbol{w}_c^{\text{CLIP}}|. \quad (8)$$

Data augmentations are applied N^d times on $\boldsymbol{x}$. Then, the augmented images are fed into our visual encoder to extract their embeddings, i.e., $\{\boldsymbol{f}_i^{\text{TAC}}\}_{i=1}^{N^d}$. Correspondingly, the visual consistency is imposed by:

$$\mathcal{L}^v = \sum_{i=1}^{N^d} |\boldsymbol{f}_i^{\text{TAC}} - \boldsymbol{f}^{\text{CLIP}}|. \quad (9)$$

Here, $\boldsymbol{f}^{\text{CLIP}}$ denotes the embedding extracted by using the visual encoder of CLIP on $\boldsymbol{x}$.

Overall loss. The probability of $\boldsymbol{f}_i^{\text{TAC}}$ that belongs to the c-th class can be calculated as:

$$p^{\text{TAC}}(c|\boldsymbol{f}_i^{\text{TAC}}) = \frac{\exp(\cos(\boldsymbol{w}_c^{\text{TAC}}, \boldsymbol{f}_i^{\text{TAC}})/\tau)}{\sum_{j=1}^{C} \exp(\cos(\boldsymbol{w}_j^{\text{TAC}}, \boldsymbol{f}_i^{\text{TAC}})/\tau)}. \quad (10)$$

Then, the following cross-entropy loss can be constructed:

$$\mathcal{L}_{\text{CE}}{}^{\text{TAC}} = -\sum_{i=1}^{N^d} \sum_{c=1}^{C} \mathbb{I}(c = y) \log p^{\text{TAC}}(c|\boldsymbol{f}_i^{\text{TAC}}). \quad (11)$$

After combining self-regulation, the overall loss becomes:

$$\mathcal{L}^{\text{Overall}} = \mathcal{L}_{\text{CE}}{}^{\text{TAC}} + \lambda\mathcal{L}^t + \beta\mathcal{L}^v. \quad (12)$$

Here, λ and β are hyper-parameters that play a balancing role.

Complexity. The discussion is three-fold. First, the clustering on $\{\boldsymbol{w}_c^{\text{CLIP}}\}_{c=1}^{C}$ and the extraction of original class and visual embeddings are pre-computed and cached. Thus, the extra computational overhead incurred by them is marginal. Second, the textual task-aware pre-context does not incur extra computational overhead since the textual encoder of CLIP only accepts a sufficiently large fixed-number of input embeddings. Third, with the visual task-aware pre-context, the quantity of embeddings input to the visual encoder only increase by a very small mount, i.e., 2.5%. Thus, the extra computational overhead incurred is very limited. In summary, the extra computational overhead incurred by our TAC is marginal.

4. Experiments

4.1. Experiment setup

Evaluation and datasets. The evaluation is conducted in the base-to-novel, domain generalization, and cross-dataset transfer settings, and the average accuracy of three

Table 1. Comparative results in the base-to-novel setting. Bold accuracies are the highest. HM denotes the harmonic mean of the base and novel accuracies.

(a) **Average**

	Base	Novel	HM
CLIP [37]	69.34	74.22	71.70
CoOp [61]	82.69	63.22	71.66
Co-CoOp [60]	80.47	71.69	75.83
ProGrad [62]	82.48	70.75	76.16
RPO [29]	81.13	75.00	77.78
MetaPrompt [58]	83.38	76.09	79.57
KAPT [21]	78.41	70.52	74.26
KgCoOp [51]	80.73	73.60	77.00
MaPLe [22]	82.28	75.14	78.55
PromptSRC [23]	84.26	76.10	79.97
TCP [52]	84.13	75.36	79.51
CoPrompt [39]	84.00	77.23	80.48
ArGue [45]	83.69	**78.07**	80.78
TAC (Ours)	**85.24**	77.60	**81.24**

(b) ImageNet

	Base	Novel	HM
CLIP [37]	72.43	68.14	70.22
CoOp [61]	76.47	67.88	71.92
Co-CoOp [60]	75.98	70.43	73.10
ProGrad [62]	77.02	66.66	71.46
RPO [29]	76.60	71.57	74.00
MetaPrompt [58]	77.39	71.06	74.09
KAPT [21]	71.10	65.20	68.02
KgCoOp [51]	75.83	69.96	72.78
MaPLe [22]	76.66	70.54	73.47
PromptSRC [23]	77.60	70.73	74.01
TCP [52]	77.27	69.87	73.38
CoPrompt [39]	77.67	71.27	74.33
ArGue [45]	76.92	**72.06**	74.41
TAC (Ours)	**78.57**	71.03	**74.61**

(c) Caltech101

	Base	Novel	HM
CLIP [37]	96.84	94.00	95.40
CoOp [61]	98.00	89.81	93.73
Co-CoOp [60]	97.96	93.81	95.84
ProGrad [62]	98.02	93.89	95.91
RPO [29]	97.97	94.37	96.03
MetaPrompt [58]	98.28	94.58	96.39
KAPT [21]	97.10	93.53	95.28
KgCoOp [51]	97.72	94.39	96.03
MaPLe [22]	97.74	94.36	96.02
PromptSRC [23]	98.10	94.03	96.02
TCP [52]	98.23	94.67	96.42
CoPrompt [39]	98.27	94.90	96.55
ArGue [45]	98.43	95.20	96.79
TAC (Ours)	**98.57**	**95.27**	**96.89**

(d) OxfordPets

	Base	Novel	HM
CLIP [37]	91.17	97.26	94.12
CoOp [61]	93.67	95.29	94.47
Co-CoOp [60]	95.20	97.69	96.43
ProGrad [62]	95.07	97.63	96.33
RPO [29]	94.63	97.50	96.05
MetaPrompt [58]	95.71	96.98	96.34
KAPT [21]	93.13	96.53	94.80
KgCoOp [51]	94.65	97.76	96.18
MaPLe [22]	95.43	97.76	96.58
PromptSRC [23]	95.33	97.30	96.30
TCP [52]	94.67	97.20	95.92
CoPrompt [39]	95.67	98.10	96.87
ArGue [45]	95.36	97.95	96.64
TAC (Ours)	**95.93**	**98.17**	**97.04**

(e) StanfordCars

	Base	Novel	HM
CLIP [37]	63.37	74.89	68.65
CoOp [61]	78.12	60.40	68.13
Co-CoOp [60]	70.49	73.59	72.01
ProGrad [62]	77.68	68.63	72.88
RPO [29]	73.87	**75.53**	74.69
MetaPrompt [58]	75.43	74.43	74.93
KAPT [21]	69.47	66.20	67.79
KgCoOp [51]	71.76	75.04	73.36
MaPLe [22]	72.94	74.00	73.47
PromptSRC [23]	78.27	74.97	76.58
TCP [52]	80.80	74.13	77.32
CoPrompt [39]	76.97	74.40	75.66
ArGue [45]	75.64	73.38	74.49
TAC (Ours)	**81.63**	74.17	**77.72**

(f) Flowers102

	Base	Novel	HM
CLIP [37]	72.08	**77.80**	74.83
CoOp [61]	97.60	59.67	74.06
Co-CoOp [60]	94.87	71.75	81.71
ProGrad [62]	95.54	71.87	82.03
RPO [29]	94.13	76.67	84.50
MetaPrompt [58]	97.53	74.54	84.50
KAPT [21]	95.00	71.20	81.40
KgCoOp [51]	95.00	74.73	83.65
MaPLe [22]	95.92	72.46	82.56
PromptSRC [23]	98.07	76.50	85.95
TCP [52]	97.73	75.57	85.23
CoPrompt [39]	97.27	76.60	85.71
ArGue [45]	**98.34**	75.41	85.36
TAC (Ours)	97.97	76.87	**86.15**

(g) Food101

	Base	Novel	HM
CLIP [37]	90.10	91.22	90.66
CoOp [61]	88.33	82.26	85.19
Co-CoOp [60]	90.70	91.29	90.99
ProGrad [62]	90.37	89.59	89.98
RPO [29]	90.33	90.83	90.58
MetaPrompt [58]	90.76	91.77	91.26
KAPT [21]	86.13	87.06	86.59
KgCoOp [51]	90.50	91.70	91.09
MaPLe [22]	90.71	92.05	91.38
PromptSRC [23]	90.67	91.53	91.10
TCP [52]	90.57	91.37	90.97
CoPrompt [39]	90.73	**92.07**	91.40
ArGue [45]	**92.33**	91.96	**92.14**
TAC (Ours)	90.87	91.87	91.37

(h) FGVCAircraft

	Base	Novel	HM
CLIP [37]	27.19	36.29	31.09
CoOp [61]	40.44	22.30	28.75
Co-CoOp [60]	33.41	23.71	27.74
ProGrad [62]	40.54	27.57	32.82
RPO [29]	37.33	34.20	35.70
MetaPrompt [58]	39.38	37.59	38.46
KAPT [21]	29.67	28.73	29.19
KgCoOp [51]	36.21	33.55	34.83
MaPLe [22]	37.44	35.61	36.50
PromptSRC [23]	42.73	37.87	40.15
TCP [52]	41.97	34.43	37.83
CoPrompt [39]	40.20	**39.33**	39.76
ArGue [45]	40.46	38.03	39.21
TAC (Ours)	**44.60**	37.70	**40.86**

(i) SUN397

	Base	Novel	HM
CLIP [37]	69.36	75.35	72.23
CoOp [61]	80.60	65.89	72.51
Co-CoOp [60]	79.74	76.86	78.27
ProGrad [62]	81.26	74.17	77.55
RPO [29]	80.60	77.80	79.18
MetaPrompt [58]	82.10	79.01	80.53
KAPT [21]	79.40	74.33	76.78
KgCoOp [51]	80.29	76.53	78.36
MaPLe [22]	80.82	78.70	79.75
PromptSRC [23]	82.67	78.47	80.52
TCP [52]	82.63	78.20	80.35
CoPrompt [39]	82.63	80.03	81.31
ArGue [45]	81.52	**80.74**	81.13
TAC (Ours)	**83.70**	80.03	**81.82**

(j) DTD

	Base	Novel	HM
CLIP [37]	53.24	59.90	56.37
CoOp [61]	79.44	41.18	54.24
Co-CoOp [60]	77.01	56.00	64.85
ProGrad [62]	77.35	52.35	62.45
RPO [29]	76.70	62.13	68.61
MetaPrompt [58]	82.52	60.10	69.55
KAPT [21]	75.97	58.30	65.97
KgCoOp [51]	77.55	54.99	64.35
MaPLe [22]	80.36	59.18	68.16
PromptSRC [23]	**83.37**	62.97	71.75
TCP [52]	82.77	58.07	68.23
CoPrompt [39]	83.13	64.73	72.79
ArGue [45]	81.60	**66.55**	**73.31**
TAC (Ours)	**83.37**	64.27	72.58

(k) EuroSAT

	Base	Novel	HM
CLIP [37]	56.48	64.05	60.03
CoOp [61]	92.19	54.74	68.69
Co-CoOp [60]	87.49	60.04	71.21
ProGrad [62]	90.11	60.89	72.67
RPO [29]	86.63	68.97	76.79
MetaPrompt [58]	93.37	78.34	85.20
KAPT [21]	84.80	67.57	75.21
KgCoOp [51]	85.64	64.34	73.48
MaPLe [22]	94.07	73.23	82.35
PromptSRC [23]	92.90	73.90	82.32
TCP [52]	91.63	74.73	82.32
CoPrompt [39]	**94.60**	78.57	85.84
ArGue [45]	94.43	**88.24**	**91.23**
TAC (Ours)	94.37	82.60	88.10

(l) UCF101

	Base	Novel	HM
CLIP [37]	70.53	77.50	73.85
CoOp [61]	84.69	56.05	67.46
Co-CoOp [60]	82.33	73.45	77.64
ProGrad [62]	84.33	74.94	79.35
RPO [29]	83.67	75.43	79.34
MetaPrompt [58]	84.70	78.56	81.51
KAPT [21]	80.83	67.10	73.33
KgCoOp [51]	82.89	76.67	79.65
MaPLe [22]	83.00	78.66	80.77
PromptSRC [23]	87.10	78.80	82.74
TCP [52]	87.13	80.77	83.83
CoPrompt [39]	86.90	79.57	83.07
ArGue [45]	85.56	79.29	82.31
TAC (Ours)	**88.07**	**81.67**	**84.75**

Table 2. Comparative results in the cross-dataset transfer setting. Bold accuracies are the highest. Optimization is conducted on ImageNet and evaluation is performed on other datasets.

	Source	**Target**										
	ImageNet	Caltech101	OxfordPets	StanfordCars	Flowers102	Food101	FGVCAircraft	SUN397	DTD	EuroSAT	UCF101	*Average*
CoOp [61]	71.51	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
Co-CoOp [60]	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 [22]	70.72	93.53	90.49	65.57	72.23	86.20	24.74	67.01	46.49	48.06	68.69	66.30
Bayesian Prompt [10]	70.93	93.67	90.63	65.00	70.90	86.30	**24.93**	67.47	46.10	45.87	68.67	65.95
PromptSRC [23]	71.27	93.60	90.25	**65.70**	70.25	86.15	23.90	67.10	46.87	45.50	68.75	65.81
TCP [52]	71.40	93.97	**91.25**	64.69	71.21	**86.69**	23.45	67.15	44.35	51.45	68.73	66.29
CoPrompt [39]	70.80	94.50	90.73	65.67	**72.30**	86.43	24.00	67.57	47.07	**51.90**	69.73	**67.00**
TAC (Ours)	**72.77**	**94.53**	90.67	65.30	72.20	85.83	23.53	**67.63**	**47.57**	48.07	**70.00**	66.53

runs is reported as the final performance. The datasets involved are as follows: ImageNet [9], Caltech101 [30], OxfordPets [36], StanfordCars [25], Flowers102 [35], Food101 [3], FGVCAircraft [44], SUN397 [50], DTD [8], EuroSAT [14], UCF101 [42], ImageNet-A [16], ImageNet-R [15], ImageNet-Sketch [47], and ImageNetV2 [38].

Implementation details. Our TAC is built on top of CoOp [61] and PromptSRC [23]. The ViT-B/16 based CLIP [37] is used as the foundation model. The optimizer is SGD, with a learning rate of 2.5×10^{-3}, a weight decay of 5×10^{-4}, and a momentum of 0.9. The cosine annealing is used for learning rate scheduling. The random flip and resized crop are used for data augmentation. The batch size is 8. $K = 5$, $M^t = 4$, $M^v = 4$, $N^d = 4$, $\lambda = 20$,

Table 3. Comparative results in the domain generalization setting. Bold accuracies are the highest. Optimization is conducted on ImageNet and evaluation is performed on datasets with domain shifts.

	Source	**Target**				
	ImageNet	ImageNetV2	ImageNet-Sketch	ImageNet-A	ImageNet-R	*Average*
CLIP [37]	66.73	60.83	46.15	47.77	73.96	57.17
RPO [29]	71.67	65.13	49.27	50.13	76.57	60.27
LaViP [26]	65.95	61.60	47.23	48.91	**83.93**	60.41
TCP [52]	71.20	64.60	49.50	51.20	76.73	60.51
CoOp [61]	71.51	64.20	47.99	49.71	75.21	59.28
Co-CoOp [60]	71.02	64.07	48.75	50.63	76.18	59.90
ProGrad [62]	72.24	64.73	47.61	49.39	74.58	59.07
KgCoOp [51]	71.20	64.10	48.97	50.69	76.70	60.11
MaPLe [22]	70.72	64.07	49.15	50.90	76.98	60.26
Bayesian Prompt [10]	70.93	64.23	49.20	51.33	77.00	60.44
PromptSRC [23]	71.27	64.35	49.55	50.90	77.80	60.65
CoPrompt [39]	70.80	64.25	49.43	50.50	77.51	60.42
ArGue [45]	71.84	65.02	49.25	51.47	76.96	60.67
TAC (Ours)	**72.77**	**65.97**	**50.30**	**51.73**	78.50	**61.63**

Table 4. Ablation on the textual and visual task-aware pre-contexts. HM denotes the harmonic mean of the base and novel accuracies.

Pre-context	Base	Novel	HM
None	84.70	75.55	79.86
Textual	84.57	76.48	80.32
Visual	84.67	76.06	80.13
Textual & visual	**85.24**	**77.60**	**81.24**

Table 5. Ablation on the number of clustering centers. HM denotes the harmonic mean of the base and novel accuracies.

Number of clustering centers	Base	Novel	HM
$K = 0$	84.70	75.55	79.86
$K = 1$	84.95	77.39	80.99
$K = 2$	84.94	77.41	81.00
$K = 3$	85.30	77.32	81.14
$K = 4$	**85.40**	77.34	81.17
$K = 5$	85.24	**77.60**	**81.24**

$\beta = 25$, and $\tau = 0.07$. In the base-to-novel setting, the number of epochs is 20 and the scale of random resized crop is $(0.4, 1.0)$. In the domain generalization and cross-dataset transfer settings, the number of epochs is 5 and the scale of random resized crop is $(0.8, 1.0)$. The learning rate is warmed up for one epoch with a rate of 1.0×10^{-5}.

4.2. Comparative results

Base-to-novel. In this setting, optimization is conducted on the base set and evaluation is performed on the base and novel sets. Table 1 shows the comparative results. Our TAC achieves the best average base accuracy, the second-best average novel accuracy, and the best harmonic mean. These results demonstrate the generalizability of our task-aware clustering prompting framework.

Cross-dataset transfer. In this setting, optimization is conducted on ImageNet and evaluation is performed on other datasets. Table 2 shows the comparative results. Our TAC achieves the best accuracy on ImageNet. Also, our TAC yields the second-best average accuracy on other datasets, which is slightly worse than CoPrompt [39]. It is worth noting that our TAC beats CoPrompt in the base-to-novel and domain generalization settings. These results demonstrate the transferability of our task-aware clustering prompting framework.

Domain generalization. In this setting, optimization is conducted on ImageNet and evaluation is performed on datasets with domain shifts. Table 3 shows the comparative results. Our TAC achieves the best accuracy on ImageNet and the best average accuracy on the datasets with domain shifts. These results demonstrate the domain generalizability of our task-aware clustering prompting framework.

4.3. Ablation studies

Ablation studies are conducted in the base-to-novel setting. The average base accuracy, the average novel accuracy, and their harmonic mean are reported. *More is shown in the appendix*.

Task-aware pre-context. We ablate the effect of the task-aware pre-context by removing the textual pre-context, the visual pre-context, or both. It is worth noting that the other ingredients are kept in this experiment. Table 4 shows the results. Our observations are as follows: (a) both the textual and visual task-aware pre-contexts can improve the harmonic mean and (b) their combination achieves the greatest performance gains. These observations demonstrate the effectiveness of introducing task-aware pre-context.

Number of clustering centers. We ablate the effect of the number of clustering centers by setting K to different values. It is worth noting that (a) the maximum value of K is 5 since the dataset EuroSAT [14] only has 5 classes and (b) the other ingredients are kept in this experiment. Table 5 shows the results. Our observations are as follows: (a) $K = 0$ means that the task-aware pre-context is

Table 6. Ablation on the self-regulation. HM denotes the harmonic mean of the base and novel accuracies.

Self-regulation	Base	Novel	HM
$\mathcal{L}_{\text{CE}}^{\text{TAC}}$	**85.29**	74.18	79.34
$\mathcal{L}_{\text{CE}}^{\text{TAC}} + \lambda\mathcal{L}^t$	83.52	73.73	78.32
$\mathcal{L}_{\text{CE}}^{\text{TAC}} + \beta\mathcal{L}^v$	84.48	73.89	78.83
$\mathcal{L}_{\text{CE}}^{\text{TAC}} + \lambda\mathcal{L}^t + \beta\mathcal{L}^v$	85.24	**77.60**	**81.24**

Table 7. Ablation on the undesired interaction masking. HM denotes the harmonic mean of the base and novel accuracies.

Undesired interaction masking	Base	Novel	HM
✗	85.00	77.05	80.83
✓	**85.24**	**77.60**	**81.24**

not exploited, (b) the introduction of task-aware pre-context can improve performance by a noticeable margin, and (c) the biggest performance gains are achieved when $K = 5$. These observations demonstrate the effectiveness of introducing task-aware pre-context and lead us to set the default value of K to 5.

Self-regulation. We ablate the effect of the self-regulation by removing the textual consistency, the visual consistency, or both. It is worth noting that the other ingredients are kept in this experiment. Table 6 shows the results. Our observations are as follows: (a) without imposing self-regulation, the accuracy on the novel set is very poor, which also leads to a low harmonic mean, (b) imposing the textual consistency or visual consistency alone brings a noticeable performance drop, and (c) imposing the two consistencies simultaneously leads to a significant performance gain. The reason for these observations might be that the alignment between textual and visual modalities is well preserved by imposing the two consistencies, while it is difficult to achieve this with only one consistency. These observations demonstrate the necessity of self-regulation.

Undesired interaction masking. We ablate the effect of the undesired interaction masking by removing it. It is worth noting that (a) the mask matrix is inserted into the visual encoder in a layer-wise manner and (b) the other ingredients are kept in this experiment. Table 7 shows the results. With the undesired interaction masking, the harmonic mean is improved by a margin of 0.41%. Considering the performance gain is achieved in a experiment setting involving 11 datasets, this observation demonstrates the effectiveness of undesired interaction masking.

Complexity. We evaluate the complexity by comparing our TAC with the competitive counterparts. It is worth noting that (a) PromptSRC [23] inserts learnable prompts into the textual and visual encoders in a layer-wise manner and (b) our TAC inserts the learnable prompts, along with the task-aware pre-contexts, into the textual and visual encoders in the same way. Table 8 shows the results. Since the task-aware pre-context generation is lightweight and the quantity of additional input embeddings incurred by the task-aware pre-context is small, the extra computational overhead incurred by our TAC is marginal compared to PromptSRC [23]. Moreover, it is worth noting our TAC outperforms PromptSRC with noticeable performance gains in all experiment settings.

Table 8. Ablation on the complexity. Evaluation is performed on a RTX 3090 GPU.

Method	Optimization (ms/image)	Inference (ms/image)
CLIP [37]	-	1.7
CoOp [61]	68.1	3.0
CoCoOp [60]	269.0	126.6
MaPLe [22]	70.2	1.7
PromptSRC [23]	71.5	3.8
TAC (Ours)	73.2	4.1

Table 9. Ablation on the number of learnable prompts. HM denotes the harmonic mean of the base and novel accuracies.

Number of learnable prompts	Base	Novel	HM
1	84.15	75.63	79.66
2	84.67	76.61	80.44
4	**85.24**	**77.60**	**81.24**
6	85.02	76.85	80.73

Number of learnable prompts. We ablate the effect of the number of learnable prompts by setting M^t and M^v to different values. It is worth noting that (a) M^t and M^v are set to be equal by default and (b) the other ingredients are kept in this experiment. Table 9 shows the results. Our observations are as follows: (a) increasing the number of learnable prompts can improve the harmonic mean and (b) the greatest performance gain is achieved when $M^t = 4$ and $M^v = 4$. These observations lead us to set the default value of M^t and M^v to 4.

5. Conclusion

Prompt learning has emerged as a powerful tool for adapting vision-language models. However, due to the scarcity of task-specific data, the resulting prompts often exhibit limited task-awareness. In this work, we propose to increase the task-awareness by introducing task-aware pre-context, which results in a novel task-aware clustering framework for prompting. The key ingredients are three-fold. First, the backbone structure of a downstream task is preserved by task-aware clustering, based on which the task-aware pre-context is generated. Second, the task-awareness of learnable prompts is enhanced by allowing them to interact with task-aware pre-context via the well-pretrained encoders. Finally, the undesired interference of visual task-aware pre-context to patch embeddings is eliminated by masked attention mechanism. Extensive ablation studies confirm the effectiveness of key ingredients. Comparative results in the base-to-novel, domain generalization, and cross-dataset transfer settings show the advantage of our task-aware clustering for prompt learning.

Acknowledgements

This work was supported by Guangdong Major Project of Basic and Applied Basic Research (2023B0303000016), National Natural Science Foundation of China (62206268, U21A20487, 62372440), Shenzhen Technology Project (JCYJ20240813154723031), Shenzhen High-tech Zone Development Special Plan Innovation Platform Construction Project, the proof of concept center for high precision and high resolution 4D imaging, Guangdong Technology Project (2023TX07Z126), and Yunnan Science & Technology Project (202302AD080008, 202305AF150152).

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 1

[2] Shtedritski Aleksandar, Rupprecht Christian, and Vedaldi Andrea. What does clip know about a red circle? visual prompt engineering for vlms. In *ICCV*, pages 11987–11997, 2023. 1, 3

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, pages 446–461, 2014. 6

[4] Adrian Bulat and Georgios Tzimiropoulos. Lasp: Text-to-text optimization for language-aware soft prompting of vision & language models. In *CVPR*, pages 23232–23241, 2023. 1

[5] Yun-Hao Cao, Kaixiang Ji, Ziyuan Huang, Chuanyang Zheng, Jiajia Liu, Jian Wang, Jingdong Chen, and Ming Yang. Towards better vision-inspired vision-language models. In *CVPR*, pages 13537–13547, 2024. 2

[6] Delong Chen, Zhao Wu, Fan Liu, Zaiquan Yang, Shaoqiu Zheng, Ying Tan, and Erjin Zhou. Protoclip: Prototypical contrastive language image pretraining. *IEEE TNNLS (Early Access)*, pages 1–15, 2023. 2

[7] Eulrang Cho, Jooyeon Kim, and Hyunwoo J. Kim. Distribution-aware prompt tuning for vision-language models. In *ICCV*, pages 22004–22013, 2023. 1

[8] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014. 6

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2, 6

[10] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor G. Turrisi da Costa, Cees G.M. Snoek, Georgios Tzimiropoulos, and Brais Martinez. Bayesian prompt learning for image-language model generalization. In *ICCV*, pages 15237–15246, 2023. 6, 7

[11] Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. Promptdet: Towards open-vocabulary detection using uncurated images. In *ECCV*, 2022. 1

[12] Bordes Florian, Pang Richard Yuanzhe, Ajay Anurag, Li Alexander C, Bardes Adrien, Petryk Suzanne, Mañas Oscar, Lin Zhiqiu, Mahmoud Anas, Jayaraman Bargav, et al. An introduction to vision-language modeling. *arXiv preprint arXiv:2405.17247*, 2024. 2

[13] Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. Domain adaptation via prompt learning. *IEEE TNNLS (Early Access)*, pages 1–11, 2023. 2

[14] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *J-STARS*, 12(7):2217–2226, 2019. 6, 7

[15] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pages 8340–8349, 2021. 6

[16] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pages 15262–15271, 2021. 6

[17] Nisha Huang, Yuxin Zhang, Fan Tang, Chongyang Ma, Haibin Huang, Weiming Dong, and Changsheng Xu. Diffstyler: Controllable dual diffusion for text-driven image stylization. *IEEE TNNLS (Early Access)*, pages 1–14, 2024. 2

[18] Wei Huang, Chang Chen, Yong Li, Jiacheng Li, Cheng Li, Fenglong Song, Youliang Yan, and Zhiwei Xiong. Style projected clustering for domain generalized semantic segmentation. In *CVPR*, pages 3061–3071, 2023. 3

[19] Ahmet Iscen, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. Retrieval-enhanced contrastive vision-text models. In *ICLR*, 2024. 2

[20] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pages 4904–4916, 2021. 1, 2

[21] Baoshuo Kan, Teng Wang, Wenpeng Lu, Xiantong Zhen, Weili Guan, and Feng Zheng. Knowledge-aware prompt tuning for generalizable vision-language models. In *ICCV*, pages 15670–15680, 2023. 1, 6

[22] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *CVPR*, pages 19113–19122, 2023. 1, 6, 7, 8

[23] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *ICCV*, pages 15190–15200, 2023. 1, 2, 3, 6, 7, 8

[24] Konwoo Kim, Michael Laskin, Igor Mordatch, and Deepak Pathak. How to adapt your large-scale vision-and-language model, 2022. 2

[25] Jonathan Krause, Michael Stark, Jia Deng, and Fei-Fei Li. 3d object representations for fine-grained categorization. In *ICCV*, pages 554–561, 2013. 6

[26] Nilakshan Kunananthaseelan, Jing Zhang, and Mehrtash Harandi. Lavip: Language-grounded visual prompting. In *AAAI*, pages 2840–2848, 2024. 3, 7

[27] Gukyeong Kwon, Zhaowei Cai, Avinash Ravichandran, Erhan Bas, Rahul Bhotika, and Stefano Soatto. Masked vision and language modeling for multi-modal representation learning. In *ICLR*, 2023. 2

[28] Marc Lafon, Elias Ramzi, Clément Rambour, Nicolas Audebert, and Nicolas Thome. Gallop: Learning global and local prompts for vision-language models. In *ECCV*, 2024. 3

[29] Dongjun Lee, Seokwon Song, Jihee Suh, Joonmyeong Choi, Sanghyeok Lee, and Hyunwoo J. Kim. Read-only prompt optimization for vision-language few-shot learning. In *ICCV*, pages 1401–1411, 2023. 1, 6, 7

[30] Fei-Fei Li, Fergus Rob, and Perona Pietro. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshop*, pages 178–178, 2004. 6

[31] Yanxin Long, Jianhua Han, Runhui Huang, Hang Xu, Yi Zhu, Chunjing Xu, and Xiaodan Liang. Fine-grained visual–text prompt-driven self-training for open-vocabulary object detection. *IEEE TNNLS (Early Access)*, pages 1–11, 2023. 2

[32] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, pages 5206–5215, 2022. 3

[33] Jiawei Ma, Po-Yao Huang, Saining Xie, Shang-Wen Li, Luke Zettlemoyer, Shih-Fu Chang, Wen-Tau Yih, and Hu Xu. Mode: Clip data experts via clustering. In *CVPR*, pages 26354–26363, 2024. 3

[34] Metaxas Ioannis Maniadis, Tzimiropoulos Georgios, and Patras Ioannis. Divclust: Controlling diversity in deep clustering. In *CVPR*, pages 3418–3428, 2023. 3

[35] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pages 722–729. IEEE, 2008. 6

[36] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505, 2012. 6

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 1, 2, 3, 6, 7, 8

[38] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, pages 5389–5400, 2019. 6

[39] Shuvendu Roy and Ali Etemad. Consistency-guided prompt learning for vision-language models. In *ICLR*, 2024. 2, 3, 6, 7

[40] Lavoie Samuel, Kirichenko Polina, Ibrahim Mark, Assran Mido, Wilson Andrew Gordon, Courville Aaron, and Ballas Nicolas. Modeling caption diversity in contrastive vision-language pretraining. In *ICML*, pages 26070–26084, 2024. 2

[41] Tourani Siddharth, Alwheibi Ahmed, Mahmood Arif, and Khan Muhammad Haris. Pose-guided self-training with two-stage clustering for unsupervised landmark discovery. In *CVPR*, pages 23041–23051, 2024. 3

[42] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6

[43] Vladan Stojnić, Yannis Kalantidis, and Giorgos Tolias. Label propagation for zero-shot classification with vision-language models. In *CVPR*, pages 23209–23218, 2024. 2

[44] Maji Subhransu, Rahtu Esa, Kannala Juho, Blaschko Matthew, and Vedaldi Andrea. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 6

[45] Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. Argue: Attribute-guided prompt tuning for vision-language models. In *CVPR*, pages 28578–28587, 2024. 2, 3, 6, 7

[46] Yujun Tong, Da Li, Dongliang Chang, Tianwei Cao, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Vision-language subspace prompting, 2024. 2

[47] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019. 6

[48] Mengmeng Wang, Jiazheng Xing, Jianbiao Mei, Yong Liu, and Yunliang Jiang. Actionclip: Adapting language-image pretrained models for video action recognition. *IEEE TNNLS (Early Access)*, pages 1–13, 2023. 2

[49] Zihao Wei, Zixuan Pan, and Andrew Owens. Efficient vision-language pre-training by cluster masking. In *CVPR*, pages 26815–26825, 2024. 2

[50] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492, 2010. 6

[51] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *CVPR*, pages 6757–6767, 2023. 1, 6, 7

[52] Hantao Yao, Rui Zhang, and Changsheng Xu. Tcp: Textual-based class-aware prompt tuning for visual-language model. In *CVPR*, pages 23438–23448, 2024. 3, 6, 7

[53] Lochman Yaroslava, Olsson Carl, and Zach Christopher. Learned trajectory embedding for subspace clustering. In *CVPR*, pages 19092–19102, 2024. 3

[54] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *TMLR*, 2022. 2

[55] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023. 2

[56] Yunfei Zhang, Xiaoyang Huo, Tianyi Chen, Si Wu, and Hau San Wong. Exploring intra-class variation factors with learnable cluster prompts for semi-supervised image synthesis. In *CVPR*, pages 7392–7401, 2023. 3

[57] Yabin Zhang, Wenjie Zhu, Hui Tang, Zhiyuan Ma, Kaiyang Zhou, and Lei Zhang. Dual memory networks: A versatile adaptation approach for vision-language models. In *CVPR*, pages 28718–28728, 2024. 2

[58] Cairong Zhao, Yubin Wang, Xinyang Jiang, Yifei Shen, Kaitao Song, Dongsheng Li, and Duoqian Miao. Learning domain invariant prompt for vision-language models. *IEEE TIP*, 33:1348–1360, 2024. 6

[59] Zhaoheng Zheng, Jingmin Wei, Xuefeng Hu, Haidong Zhu, and Ram Nevatia. Large language models are good prompt learners for low-shot image classification. In *CVPR*, pages 28453–28462, 2024. 3

[60] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825, 2022. 6, 7, 8

[61] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. 1, 2, 3, 6, 7, 8

[62] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *CVPR*, pages 15659–15669, 2023. 6, 7

[63] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 2