Tree of Attributes Prompt Learning for Vision-Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Prompt learning has proven effective in adapting vision language models for downstream tasks. However, existing methods usually append learnable prompt tokens solely with the category names to obtain textual features, which fails to fully leverage the rich context indicated in the textual category name. To address this issue, we propose the Tree of Attributes Prompt learning (TAP), which first instructs LLMs to generate a tree of attributes with a "concept - attribute - description" structure for each associated category name, and then learn the hierarchy with vision and text prompt tokens. Unlike existing methods that merely augment category names with a set of unstructured descriptions, our approach essentially distills structured knowledge graphs associated with class names from LLMs. Furthermore, our approach introduces text and vision prompts designed to explicitly learn the corresponding visual attributes, effectively serving as domain experts. Additionally, the general and diverse descriptions generated based on the class names may be wrong or absent in the specific given images. To address this misalignment, we further introduce a vision-conditional pooling module to extract instance-specific text features. Extensive experimental results demonstrate that our approach outperforms state-of-the-art methods on the zero-shot base-to-novel generalization as well as few-shot classification across 11 diverse datasets.

1 Introduction

Recent advancements in vision-language models (VLMs) like CLIP [33] and ALIGN [13] merge the capabilities of visual perception with linguistic understanding, which have revolutionized the landscape with their zero-shot learning abilities. They proficiently handle tasks on unseen data, bypassing the conventional requirement for task-specific training. This feature has enabled a plethora of applications, ranging from content-based image retrieval to complex visual question answering, setting new benchmarks in the domain. A crucial development in this domain is the concept of prompt learning, which has significantly influenced both natural language processing (NLP) [20–22] and vision-only models [14, 43, 44, 51]. This approach leverages learnable prompts to guide model understanding, tailoring responses to specific tasks or datasets.

Prompt learning, particularly in vision-language models, has garnered considerable interest due to its parameter efficiency and rapid convergence [54, 53, 55, 8, 23]. Techniques like CoOp [54] optimize learnable continuous prompts for few-shot image recognition, enhancing model performance significantly. Recent efforts have expanded to multimodal prompt learning, optimizing prompts in both visual and language domains [15, 16, 38, 19]. Despite their success, these models rely on simplistic text prompts, typically formatted as "a photo of a {class}", illustrated in Fig. 1 (a). While functional, this approach lacks depth, failing to encapsulate the intricacies and finer details inherent in

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

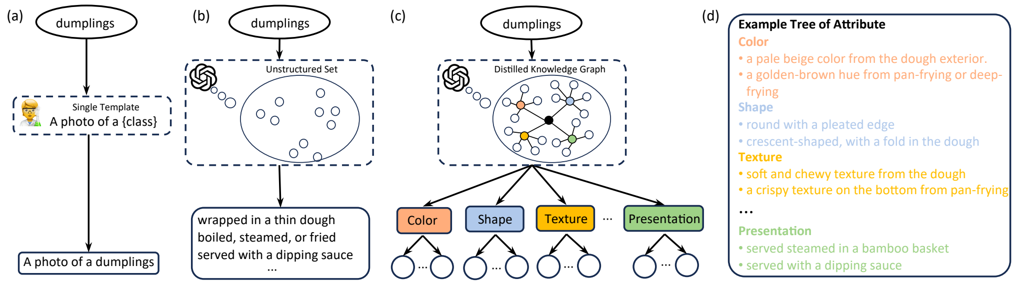

Figure 1: Illustration of the methods for CLIP text prompts formation. (a) Manually created prompt with the single "a photo of a {class}" template; (b) A unstructured set of detailed descriptions generated by LLMs; (c) The proposed Tree of Attribute that organizes the descriptions in a "concept - attribute - descriptions" structure, essentially distilling knowledge graphs from LLMs; (d) An example Tree of Attribute for "dumplings".

visual data. Such limitations hinder the model's ability to fully leverage the rich, descriptive potential offered by more detailed and contextually relevant textual information.

In parallel, another stream of research has been exploring the utilization of large language models (LLMs) to generate more elaborate and descriptive text prompts for enhancing zero-shot learning capabilities [26, 32, 35, 17, 30, 48, 49, 36, 52, 40]. These LLM-generated descriptions offer a wealth of detail and context, potentially enriching the model's interpretative capabilities. However, current methodologies in integrating these descriptions often do not exploit the full potential of this richness. As shown in Fig. 1 (b), most of these approaches lack a structured framework to organize and utilize these descriptions effectively, leading to a scattergun approach where not all generated descriptions are contextually relevant or optimally aligned with the visual content. In addition, as noted in [35], descriptions generated by such paradigms are usually diverse, which covers most possibilities of the class, but include descriptions that are either likely not co-occurring, e.g. "steamed" and "fried", or absent in the input image, e.g. "long tail" for a cat shot from the front, necessitating the need for a selective pooling mechanism for clearer image-text alignments.

In response to these challenges, our work introduces "Tree of Attribute Prompt learning (TAP)," a method that redefines the integration and utilization of detailed descriptions within VLMs. As indicated in Fig. 1 (c), unlike existing methods that merely augment category names with a set of unstructured descriptions, our approach essentially distills structured knowledge graphs associated with class names from LLMs. Specifically, we adopt a hierarchical, tree-like structure to systematically generate and integrate descriptions, ensuring a layered and comprehensive understanding of visual content. Each branch of this tree represents a specific attribute, with finer details fleshed out in the subsequent leaves, ensuring that every aspect of the visual content is captured and represented. Furthermore, we reimagine the learnable prompt tokens as "domain experts", each specializing in different aspects of the image, supplemented by the CLS token's global perspective. In addition, we introduce vision-conditional layers for each expert-attribute pair, which pool the most applicable descriptions from each of the attribute sets with condition on the input image content, ensuring optimal image-text alignment. This setup not only provides a detailed, attribute-focused analysis but also harmonizes these insights with the overall context.

Extensive experiments in both base-to-novel generalization and few-shot classification across 11 diverse datasets demonstrate the effectiveness of our method. On base-to-novel generalization, TAP achieves average performance gains of 1.07% in harmonic mean over the state-of-the-art methods, and 9.34% over the vanilla CLIP. Competitive results are also observed in few-shot classification.

2 Related Work

Prompt Learning for Vision-Language Models. Prompt learning bridges linguistic understanding and visual perception by guiding VLMs with text prompts, a concept originated in NLP [20–22] and adapted to vision-only [14, 43, 44, 51] and multimodal contexts[54, 53, 15, 16, 38, 19, 40, 34, 36, 52, 55, 4, 23]. In the textual domain, CoOp [54] optimizes learnable continuous prompts in CLIP's language branch for few-shot image recognition, while CoCoOp [53] addresses CoOp's overfitting issues by conditioning prompts on visual features. In the visual domain, Visual Prompt Tuning (VPT) [1] and Dual-modality Prompt Tuning (DPT) [47] enhance CLIP's vision encoder by learning visual prompts in pixel space and dynamically generating prompts through cross-attention, respectively. TransHP [42] leverages category hierarchy for prompt learning to improve classification performance. LoGoPrompt [38] enhances classification by incorporating synthetic images with class name text as auxiliary visual prompts. MaPLe [15] explores multimodal prompt learning, jointly optimizing prompts in both vision and language branches. Other recent works have focused on regularizing prompt learning to leverage the knowledge from base VLMs effectively, demonstrating enhanced generalization in varied downstream visual tasks [16, 4, 36]. PromptSRC, for instance, introduced a self-regulating method that restricts both the vision and text prompt, demonstrating improved generalization. Distinct from these approaches, PLOT [5] and ALIGN [41] leverage Optimal Transport to align multiple prompts with local visual features, either from the multi-head self-attention layer or at a token level. Our work diverges from these methods by introducing a hierarchical "Tree of Attribute" framework derived from LLMs to structure textual descriptions and guide the learning of specialized "domain expert" tokens for attribute-level understanding.

Image classification by descriptions. There's a growing emphasis on using visual descriptions for zero-shot recognition, moving beyond generic prompts [54, 53]. These descriptions, like the "fur pattern" or "tail shape" of a cat, provide fine-grained and distinctive characteristics. The use of LLMs like GPT-3 [3], allows for efficient generation of a broad spectrum of class-specific descriptions, offering an advantage over manually crafted templates. While this approach has been extensively researched in zero-shot contexts [17, 26, 30, 35, 48, 49, 10, 32, 28], its application in conjunction with prompt learning for few-shot tasks remains relatively unexplored[25, 19, 40, 52, 50]. Previous methodologies, however, have largely utilized unstructured descriptions, lacking an organized framework for effective utilization. Our approach diverges by structuring these descriptions into a "Tree of Attribute" model, coupled with learnable visual prompts as domain experts. Additionally, LLM-generated descriptions often cover a wide range of potential class descriptions, of which not all may be pertinent to a given image, pointing to the need for a selective pooling mechanism to ensure optimal image-text alignment. We further introduce a vision-conditional pooling layer for refined image-text alignment. This structured approach not only enhances the interpretability of the model's learning process but also significantly improves alignment accuracy between image content and descriptive text.

3 Methodology

3.1 Preliminary

CLIP. Our approach is built on the pre-trained vision-language model, CLIP [33]. Formally, let (x, c) denote the dataset, where x is an image and $c \in \{1, \dots, C\}$ are the class labels. For an image x, the vision encoder $h_I(\cdot)$ transforms it into a feature vector $\mathbf{f}_x^v = h_I(x)$. Simultaneously, each class label c is mapped to a text prompt $t_c =$ `a photo of a {c}`, and converted into textual feature vectors $\mathbf{f}_c^t = h_T(t_c)$. The predicted class $\hat{y}$ is given by:

$$\hat{y} = \underset{c}{\operatorname{argmax}} \cos(\mathbf{f}_x^v, \mathbf{f}_c^t) \tag{1}$$

where $\cos(\cdot)$ denotes cosine similarity.

Image classification with class descriptions. To improve the model's understanding of the categories in the transfer datasets, previous works [26, 35] use more detailed descriptions from Large Language Models (LLMs) instead of the simple `"a photo of a {c}"` to prompt the CLIP text encoder. Under this approach, a convoluted set of descriptions is generated for a class c as $\mathcal{D}_c$: {`"c, which is/has/etc description."` }, e.g. `c="television"` and `description="black or grey"`. This classification is reformulated as

$$\hat{y} = \underset{c}{\operatorname{argmax}} \frac{1}{|\mathcal{D}_c|} \sum_{d \in \mathcal{D}_c} \cos(\mathbf{h_I}(x), \mathbf{h_T}(d)) \tag{2}$$

3.2 Overall Framework

We rethink the descriptions by LLM $\mathcal{D}_c$ as nodes in knowledge graphs. While previous methods generate an unstructured set of descriptions, we distill structured knowledge graphs for each class c

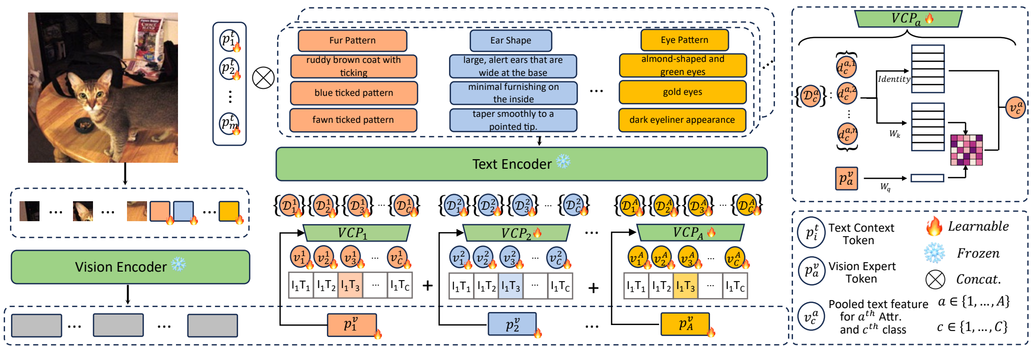

Figure 2: Overview of the proposed TAP method. TAP utilizes fine-grained descriptions from LLMs and organizes them in a Tree of Attribute. Vision expert tokens are added to the vision encoder to learn from specific attributes such as color and shape. A vision-conditional pooling layer is introduced to ensure optimal image-text alignment. Textual context tokens are also incorporated to the textual branch, shared across descriptions.

from LLM, in which the root node is the class name c, capturing the highest level semantics, and the leaf nodes are the detailed descriptions capturing fine-grained details. In this framework, previous paradigms only generate the leaf nodes of the graph, with the edges and graph structure missing, where the rich and inherent structure from the descriptions is overlooked. To address this limitation, we formulate our approach as a Tree of Attribute, which follows the "concept - attribute - description" structures, as illustrated in Fig. 1 (c).

Besides weighting the descriptions equally, previous works typically align descriptions that describe images from different aspects and at different granularities with a singular CLS token from the image encoder. However, while the use of a single CLS token is effective in certain contexts, we note that the CLS token is designed to capture the global information of an input image x [9]. As a result, even though this helps to further inform global understanding, it may fail to effectively capture the nuances and variances at the attribute level. This leads to suboptimal use of the rich descriptions. We address this by introducing a set of learnable prompt tokens that serve as domain experts in the vision branch, each of which aligns with a specific attribute-level textual embedding.

Additionally, close inspection of the LLM-generated descriptions indicates limited contextual relevance and a high degree of diversity. Previous works [35] reflect the issue of descriptions that are likely not co-occurring e.g. "steam" and "fried". We further identify cases where the descriptions are technically correct but irrelevant to certain images, such as describing "long tail" in frontal images of cats, underscoring the need for a selective pooling mechanism. Thus, we introduce a vision-conditional pooling layer to extract instance-specific text features for each attribute for selecting the most applicable descriptions.

Overall, our approach utilizes fine-grained descriptions and organizes them in a Tree of Attribute following the "concept - attributes -descriptions" structure. Learnable vision expert tokens are appended to the input image embedding to learn from specific fine-grained attributes such as color and shape. A vision-conditional pooling layer is further added for each attribute to ensure optimal image-text alignment. Inspired by CoOP [54], we also incorporate textual contextual tokens in the text encoder. The overall framework is presented in Fig. 2.

3.3 Tree of Attribute generation by LLMs

We redefine the process of integrating LLM-generated descriptions by introducing a knowledge graph $\mathcal{G}_c = \{\mathcal{V}_c, \mathcal{E}_c\}$ for each class c, where $\mathcal{V}_c$ denotes the set of nodes, and $\mathcal{E}_c$ denotes the edges that capture the semantic relationship between nodes. In previous works, $\mathcal{V}_c$ is the set of descriptions $\mathcal{D}_c$, while $\mathcal{E}_c$ is missing. We argue that such methods overlook the inherent structure among the descriptions and thus do not exploit the richness of these descriptions effectively. To better leverage knowledge from LLMs, we introduce an attribute layer to link the root node class name, and the leaf node descriptions. The attribute nodes include visual attributes generated by LLMs, such as color and shape, for systematically guiding description generation as illustrated in Fig. 1 (c). Each branch of this "tree" represents a specific attribute, with the subsequent "leaves" fleshing out the descriptions with finer details. In this framework, $\mathcal{V}_c$ includes the class name which is the root node, the set of attributes such as color and shape being the intermediate layer, and lastly the set of descriptions under each attribute node. $\mathcal{E}_c$ includes the edges that build up the hierarchy. This structure allows for a nuanced representation of class information, spanning from general concepts down to specific attributes and detailed descriptions.

To this end, we introduce the Tree of Attribute (ToA), where we use a tree structure to model the relationship and structure of the descriptions. Let $\mathcal{A}_c$ denote the set of attributes, and for each attribute $a_c \in \mathcal{A}_c$, we denote its leaf nodes as $\mathcal{D}_c^a$. Each set $\mathcal{D}_c^a$ contains descriptions that specifically pertain to attribute a for class c, which is denoted as

$$\mathcal{D}_c^a = \{d_c^{a,1}, d_c^{a,2}, \ldots, d_c^{a,n}\}, \tag{3}$$

where $d_c^{a,i}$ represents the i-th description for attribute a of class c and n is the number of descriptions per attribute.

The process of generating a Tree of Attribute (ToA) unfolds in three steps: 1) **Attribute Generation:** We first query LLMs with the dataset information and ask it to generate a set of attributes $\mathcal{A}$ which are considered relevant and characteristic of the dataset. 2) **Example Generation:** We then ask LLMs to generate descriptions for a randomly sampled class in the dataset, using the attributes $\mathcal{A}$ identified in the previous step. Each description takes the format of "class, which {is/has/etc} {description}". Human review is performed to ensure the quality of the example. 3) **Description Generation for All Classes:** Building upon the Q&A template from the previous step, the LLM is then tasked with generating descriptions for all classes in the dataset.

Additionally, we incorporate a "global context" attribute which is aligned with the CLS token in the vision encoder. The descriptions are the 7 standard templates provided in [33].

3.4 Learning TAP with Learnable Expert Tokens

To fully exploit the structured Tree of Attribute, we introduce learnable visual expert tokens $\mathbf{p}_a^v$ in the vision branch to learn from each of the attribute nodes $a \in \mathcal{A}$. Unlike traditional methods that rely on a single CLS token for alignment, these expert tokens enable focused learning on specific image attributes, such as color or shape, enhancing the model's performance and interpretability.

We denote the set of introduced visual expert tokens as $\mathcal{P}^v = \{\mathbf{p}_a^v | a \in \mathcal{A}\}$. Akin to the idea of visual prompt tuning (VPT) [14], we insert $\mathcal{P}^v$ into the input sequence of the vision encoder, forming the prompted input sequences $\tilde{\mathbf{X}}_{\mathbf{p}} = \{\mathbf{e}_{\text{CLS}}, \mathcal{P}^v, \mathbf{E}_{\text{patch}}\}$, where $\mathbf{e}_{\text{CLS}}$ is the input CLS token, and $\mathbf{E}_{\text{patch}}$ denotes the embedded patch tokens. To further boost the model's capacity for nuanced attribute representation, we employ deep prompting by introducing a zero-initialized layer residual for each prompt token across transformer layers, which provides more explicit attribute guidance across transformer layers. In parallel, we adopt a set of m learnable context tokens $\mathcal{P}^t = \{\mathbf{p}_j^t | j \in \{1, 2, ..., m\}\}$ for the text encoder shared across all descriptions, similar to [54].

3.5 Vision-Conditional Pooling

To mitigate issues of misalignment and potential misleading information from the broad spectrum of LLM-generated descriptions, we proposed an adaptive vision-conditional pooling layer, applicable to each set of attribute descriptions $\mathcal{D}_a$ shared across all classes to dynamically pool the most applicable descriptions based on the visual content of the image x using its corresponding visual expert token denoted as $\mathbf{p}_{a,x}^v$. For ease of expression, we will proceed without explicitly mentioning x, though it's important to note that both the expert token and the resulting attribute-level embeddings are dependent on the visual information. Intuitively, VCP uses attention to calculate the similarity between $\mathbf{p}_a^v$ and all embedded descriptions in attribute $\mathcal{D}_a$, which are then used as weights for a weighted sum of the original description embeddings. Formally, for each attribute a and its associated expert token $\mathbf{p}_a^v$, the pooled attribute-level embedding $\mathbf{v}_c^a$ for class c and attribute a is:

$$\begin{aligned} \text{Query} &= W_q \cdot \mathbf{p}_a^v, \\ \text{Key} &= W_k \cdot \texttt{Emb}(\mathcal{D}_c^a), \\ \text{Attention Score} &= \texttt{softmax}(\text{Query} \cdot \text{Key}^T), \\ \mathbf{v}_c^a &= \text{Attention Score} \cdot \texttt{Emb}(D_c^a), \end{aligned} \tag{4}$$

where W_q and W_k are learnable weights $\in \mathbb{R}^{d\times d}$, $\texttt{Emb}(\cdot)$ denotes the embedding function, and $\texttt{softmax}(\cdot)$ is the Softmax function. This layer mirrors cross-attention but omits W_v to maintain the output within the CLIP V-L space.

3.6 Training and Inference

Training objective. During training, each visual expert token $\mathbf{p}_a^v$ is aligned with its associated attribute-level embedding $\mathbf{v}_c^a$, trained with the following contrastive objective:

$$L_{con}(\mathbf{p}_a^v, \mathbf{v}_c^a) = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\cos(\mathbf{p}_a^v, \mathbf{v}_y^a)/\tau)}{\sum_{c=1}^{C} \exp(\cos(\mathbf{p}_a^v, \mathbf{v}_c^a)/\tau)}, \tag{5}$$

where N represents the number of training samples, and τ is the learned temprature of CLIP. The total classification loss L_{class} is the average of the contrastive loss from each expert token as well as the CLS token, defined as:

$$L_{class} = \frac{1}{|\mathcal{A}|}\left(\sum_{a\in\mathcal{A}} L_{con}(\mathbf{p}_a^v, \mathbf{v}_c^a))\right), \tag{6}$$

Similar to [16] and [4], we regularize the vision CLS token, text feature, and the prediction logits from each attribute using the vanilla CLIP model. We denote the regularization loss as L_{reg}, where the details can be found in Appendix. The overall training objective is $L_{\text{total}} = L_{\text{class}} + L_{\text{reg}}$.

Prediction fusion. During inference, we integrate the prediction by each attribute expert pair by a weighted sum, formulated as follows:

$$\tilde{y} = \underset{c}{\operatorname{argmax}}\left(\alpha \cos(\mathbf{f}_{CLS}^v, \mathbf{v}_c^{CLS}) + \frac{1-\alpha}{|\mathcal{A}|-1}\sum_{a\in\mathcal{A}\setminus\{CLS\}} \cos(\mathbf{p}_a^v, \mathbf{v}_c^a)\right) \tag{7}$$

where α is a hyperparameter that signifies the weight assigned to the global context provided by the CLS token, balancing its contribution with that of the attribute-specific expert prompts.

4 Experiments

We extensively evaluate our method in two settings: 1) Base-to-novel class generalization, where the datasets are equally split into base and novel classes. We train the model on the base classes only and evaluate on both base and novel classes; and 2) Few-shot classification with 16 shots per class.

Datasets and baslines. For both base to novel class generalization and few-shot setting, we follow previous works [54, 53], using 11 image recognition datasets. The datasets span a range of recognition tasks: ImageNet [7] and Caltech101 [11] for generic object recognition; OxfordPets [30], StanfordCars [18], Flowers102 [27], Food101 [2], and FGVCAircraft [24] for fine-grained classification; SUN397 [46] for scene recognition; UCF101 [39] for action recognition; DTD [6] for texture classification; and EuroSAT [12] for satellite image analysis. We benchmark against several leading methods, including CLIP [33], CoOp [54], Co-CoOP [53], ProGrad [55], RPO [19], LoGoPrompt [38], and the state-of-the-art PromptSRC [16].

Implementation details. A pre-trained CLIP model with a ViT-B/16 vision backbone is used in all of our experiments and results are averaged over 3 runs. We use GPT-3.5-turbo [29] for attribute and description generation. We initialize the text context tokens with the word embedding of `a photo of a`. For both settings, we iteratively train the vision and text encoders with 5 epochs for vision and 1 epoch for text schedule. We set $\alpha = 0.4$, $\mu_1 = 10$, and $\mu_2 = 2.5$ for all datasets. We train the vision encoder for 50 and 100 epochs, and text encoder for 10 and 20 epochs for base-to-novel generalization and few-shot experiments, respectively. For DTD, Oxford Flowers, Stanford Cars, UCF101, and Caltech101 datasets, we use a learning rate of 0.002 for the text encoder and 0.006 for the vision encoder, with $\mu_3 = 3$. For the remaining 6 datasets, the learning rates for both text and vision encoders are set as 0.004, with $\mu_3 = 1.5$. We also use a Gaussian Prompt Weighting (GPA) following [16], with a mean of 45, std of 10 for base-to-novel generalization, and 80, 20 for few-shot experiments. Refer to the Appendix for additional implementation details.

Table 1: **Comparison of TAP in base-to-novel generalization**. HM: harmonic mean [45].

(a) **Average**

	Base	Novel	HM
CLIP	69.34	74.22	71.70
CoOp	82.69	63.22	71.66
Co-CoOp	80.47	71.69	75.83
ProGrad	82.48	70.75	76.16
RPO	81.13	75.00	77.78
LoGoPrompt	84.47	74.24	79.03
PromptSRC	84.26	76.10	79.97
TAP	**84.75**	**77.63**	**81.04**

(b) ImageNet

	Base	Novel	HM
CLIP	72.43	68.14	70.22
CoOp	76.47	67.88	71.92
Co-CoOp	75.98	70.43	73.10
ProGrad	77.02	66.66	71.46
RPO	76.60	71.57	74.00
LoGoPrompt	76.74	**70.83**	73.66
PromptSRC	77.60	70.73	**74.01**
TAP	**77.97**	70.40	73.99

(c) Caltech101

	Base	Novel	HM
CLIP	96.84	94.00	95.40
CoOp	98.00	89.81	93.73
Co-CoOp	97.96	93.81	95.84
ProGrad	98.02	93.89	95.91
RPO	97.97	94.37	96.03
LoGoPrompt	98.19	93.78	95.93
PromptSRC	98.10	94.03	96.02
TAP	**98.90**	**95.50**	**97.17**

(d) OxfordPets

	Base	Novel	HM
CLIP	91.17	97.26	94.12
CoOp	93.67	95.29	94.47
Co-CoOp	95.20	97.69	96.43
ProGrad	95.07	97.63	96.33
RPO	94.63	97.50	96.05
LoGoPrompt	**96.07**	96.31	96.18
PromptSRC	95.33	97.30	96.30
TAP	95.80	**97.73**	**96.76**

(e) StanfordCars

	Base	Novel	HM
CLIP	63.37	74.89	68.65
CoOp	78.12	60.40	68.13
Co-CoOp	70.49	73.59	72.01
ProGrad	77.68	68.63	72.88
RPO	73.87	**75.53**	74.69
LoGoPrompt	78.36	72.39	75.26
PromptSRC	78.27	74.97	76.58
TAP	**80.70**	74.27	**77.35**

(f) Flowers102

	Base	Novel	HM
CLIP	72.08	**77.80**	74.83
CoOp	97.60	59.67	74.06
Co-CoOp	94.87	71.75	81.71
ProGrad	95.54	71.87	82.03
RPO	94.13	76.67	84.50
LoGoPrompt	**99.05**	76.52	**86.34**
PromptSRC	98.07	76.50	85.95
TAP	97.90	75.57	85.30

(g) Food101

	Base	Novel	HM
CLIP	90.10	91.22	90.66
CoOp	88.33	82.26	85.19
Co-CoOp	90.70	91.29	90.99
ProGrad	90.37	89.59	89.98
RPO	90.33	90.83	90.58
LoGoPrompt	90.82	91.41	91.11
PromptSRC	90.67	91.53	91.10
TAP	**90.97**	**91.83**	**91.40**

(h) FGVCAircraft

	Base	Novel	HM
CLIP	27.19	36.29	31.09
CoOp	40.44	22.30	28.75
Co-CoOp	33.41	23.71	27.74
ProGrad	40.54	27.57	32.82
RPO	37.33	34.20	35.70
LoGoPrompt	**45.98**	34.67	39.53
PromptSRC	42.73	**37.87**	**40.15**
TAP	44.40	36.50	40.06

(i) SUN397

	Base	Novel	HM
CLIP	69.36	75.35	72.23
CoOp	80.60	65.89	72.51
Co-CoOp	79.74	76.86	78.27
ProGrad	81.26	74.17	77.55
RPO	80.60	77.80	79.18
LoGoPrompt	81.20	78.12	79.63
PromptSRC	82.67	78.47	80.52
TAP	**82.87**	**79.53**	**81.17**

(j) DTD

	Base	Novel	HM
CLIP	53.24	59.90	56.37
CoOp	79.44	41.18	54.24
Co-CoOp	77.01	56.00	64.85
ProGrad	77.35	52.35	62.45
RPO	76.70	62.13	68.61
LoGoPrompt	82.87	60.14	69.70
PromptSRC	83.37	62.97	71.75
TAP	**84.20**	**68.00**	**75.24**

(k) EuroSAT

	Base	Novel	HM
CLIP	56.48	64.05	60.03
CoOp	92.19	54.74	68.69
Co-CoOp	87.49	60.04	71.21
ProGrad	90.11	60.89	72.67
RPO	86.63	68.97	76.79
LoGoPrompt	**93.67**	69.44	79.75
PromptSRC	92.90	73.90	82.32
TAP	90.70	**82.17**	**86.22**

(l) UCF101

	Base	Novel	HM
CLIP	70.53	77.50	73.85
CoOp	84.69	56.05	67.46
Co-CoOp	82.33	73.45	77.64
ProGrad	84.33	74.94	79.35
RPO	83.67	75.43	79.34
LoGoPrompt	86.19	73.07	79.09
PromptSRC	87.10	78.80	82.74
TAP	**87.90**	**82.43**	**85.08**

Table 2: Few shot classification results with 16 shots.

	16-Shot Classification											
	Average	ImageNet	Caltech101	Pets	Cars	Flowers	Food101	Aircraft	SUN397	DTD	EuroSAT	UCF101
CLIP	78.79	67.31	95.43	85.34	80.44	97.37	82.90	45.36	73.28	69.96	87.21	82.11
CoOp	79.89	71.87	95.57	91.87	83.07	97.07	84.20	43.40	74.67	69.87	84.93	82.23
CoCoOp	74.90	70.83	95.16	93.34	71.57	87.84	87.25	31.21	72.15	63.04	73.32	78.14
MaPLe	81.79	72.33	96.00	92.83	83.57	97.00	85.33	48.40	75.53	71.33	92.33	85.03
PSRC	82.87	73.17	96.07	93.67	83.83	97.60	87.50	**50.83**	77.23	72.73	**92.43**	86.47
TAP	**83.37**	**73.76**	**96.73**	**93.90**	**85.37**	**98.10**	**87.53**	50.43	**77.30**	**74.90**	91.90	**87.17**

4.1 Base-to-Novel Generalization

In base-to-novel generalization, we equally split the classes into base and novel classes. Initial training and evaluations are conducted on the seen base classes, followed by evaluation on the unseen novel classes in a zero-shot manner. TAP surpasses prior state-of-the-art models in terms of the base and novel class accuracy, as well as their harmonic mean across most of the 11 datasets, with an average increase of 1.53% in the zero-shot novel class prediction, and a 1.07% increase in the overall harmonic mean in average, as detailed inTable 1. Notably, our method improves unseen class prediction without compromising base class performance, exhibiting an average performance boost of 0.49%. In the challenging fine-grained tasks such as DTD, EuroSAT, and UCF101, TAP achieves significant improvements in novel class prediction by 5.03%, 8.27%, and 3.63% respectively. These results underscore the robust generalizability and efficacy of our method across diverse scenarios.

4.2 Few-Shot Classification

In few-shot classification, TAP also outperforms existing methods in 9 out of the 11 datasets. Detailed in Table 2, we achieve an average accuracy of 83.37 across the 11 datasets, surpassing the previous state-of-the-art methods by 0.5%, further demonstrating the effectiveness of our method.

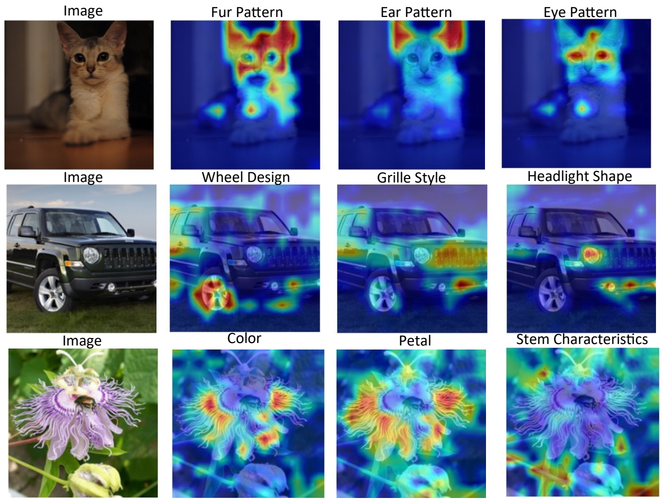

Figure 3: Visualization of the class activation maps.

Table 3: Effects of the Tree of Attributes.

Des. Org.	Unstructured	Ours
Base	82.89	**84.75**
Novel	75.32	**77.63**
HM	78.93	**81.04**

Table 4: Effects of domain experts.

Align. Token	CLS	Ours
Base	83.89	**84.75**
Novel	76.85	**77.63**
HM	80.22	**81.04**

Table 5: Effects of the number of experts.

Attrs. Num.	1	2	3	4	5	6	7	8	Ours
Base Acc.	83.20	83.97	84.1	84.41	84.45	84.62	84.66	84.74	**84.75**
Novel Acc.	74.90	76.20	76.35	77.06	77.13	77.17	77.35	76.67	**77.63**
HM	78.83	79.90	80.04	80.57	80.63	80.72	80.84	80.50	**81.04**

4.3 Ablation Study

Effects of Tree of Attribute. A core inquiry is whether structuring descriptions into a Tree of Attribute (ToA) offers advantages over an unstructured aggregation of LLM-generated descriptions. To evaluate, we revert to aligning a mixed, unstructured set of descriptions with the CLS token - a common practice in prior studies [25, 19, 40, 52], while keeping the same number of visual prompt tokens. According to Table 3, substituting the ToA with an unstructured set results in significant performance decreases of 1.86%, 2.31%, and 2.11% across the average base, novel, and their harmonic mean performances, respectively. This stark contrast underscores the ToA's critical role in enhancing model efficacy.

Effects of Learning through Domain Experts. Further, we examine the impact of substituting the CLS token with visual expert tokens for learning fine-grained attributes, commonly adopted in in previous works [25, 19, 40, 52]. Our findings (Table 4) reveal improvements of 0.89%, 0.78%, and 0.82% in the average base, novel, and harmonic mean accuracies, respectively, upon integrating visual expert tokens. These results support the notion that domain-specific, learnable tokens enhance the model's ability to grasp fine-grained details by focusing on distinct aspects of the image, as opposed to the CLS token's global focus.

Effects of Number of Attributes. In our framework, the selection of attributes is dynamically determined by LLMs, leading to variability across different datasets. This adaptability stands in contrast to a static approach where the number of attributes is uniformly set across all datasets. To understand the impact of this variability, we explore how altering the number of attributes from 1 to 8 influences model performance. Our findings, detailed in Table 5, reveal a performance improvement trend as the number of attributes increases, with an optimal peak at 7 attributes before a slight decline at 8. However, crucially, across all fixed-attribute scenarios, none matched the performance achieved through our method's dynamic attribute determination. These results underscore the importance of an adaptive approach to attribute selection, as opposed to a one-size-fits-all strategy.

Design choice of the vision-conditional pooling layer. Lastly, we ablate the design of the pooling layer, starting from the naive training-free average pooling, to the attention-based pooling mechanism with condition on the input image. Compared to average pooling, VCP demonstrates a performance gain of 1.08% in the average harmonic mean. Furthermore, when compared with attention-based max pooling, which selects a single description per attribute according to the attention score in Eq. (4),

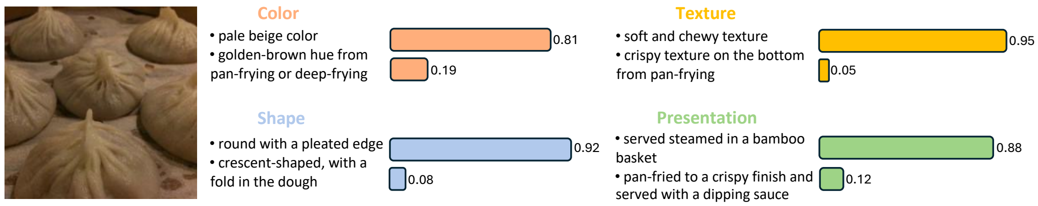

Figure 4: Visualization of the attention weights in the VCP layer for an example "dumplings" image.

Table 6: Design choice of the pooling layer.

Pooling Method	Base Acc.	Novel Acc.	HM
Attn. Max Pooling	82.90	76.36	79.49
Average Pooling	83.18	76.98	79.96
VCP (Ours)	**84.75**	**77.63**	**81.04**

VCP maintains a superior advantage of 1.55% in average harmonic mean. These outcomes attest to the VCP layer's integral role in finetuning attribute relevance to the visual context, substantiating its design and implementation within our model.

4.4 Visualization

Expert tokens focus on attribute-related regions. We further investigate the effects of vision domain experts by visualizing their class activation maps from three illustrative examples using GradCAM [37], as shown inFig. 3. These visualizations underscore the precision with which each expert token concentrates on the image regions pertinent to its designated attribute. Take the first cat image as an example. The "fur pattern" expert distinctly highlights the animal's fur texture, whereas the "ear" and "eye" experts focus precisely on the respective anatomical features. This pattern of attribute-specific attention is consistent across the evaluated examples, reinforcing the conceptualization of expert tokens as dedicated "domain experts" within the visual field.

VCP layer pools the most applicable descriptions. The inherently interpretable nature of the VCP layer, thanks to its attention mechanism, allows for insightful visualizations of its operational process. Through the examination of attention weights assigned by the VCP layer to different attributes in a given image, we elucidate the layer's capability to discern and prioritize the most applicable descriptions. As illustrated in Fig. 4 with a "dumplings" image, the VCP layer adeptly allocates higher attention weights to descriptions accurately reflecting the observed instance (e.g., assigning weights of 0.92 to "round with a pleated edge" under the "Shape" attribute and 0.95 to "soft and chewy texture" under the Texture"). In contrast, less relevant descriptions for the specific image context (e.g., "crescent-shaped" for Shape and "crispy texture from pan-frying" for Texture) receive significantly lower weights. This discernment is crucial, given the class dumplings" encompasses a broad variety of appearances based on cooking methods, yet not all descriptions are fitting for every instance. These visualizations compellingly demonstrate the VCP layer's effectiveness in refining description relevance, thereby enhancing the model's interpretative alignment with the visual content.

5 Conclusion

This paper introduces Tree of Attribute Prompt learning (TAP), a novel method that integrates detailed, LLM-generated descriptions within VLMs, achieving state-of-the-art performance in both base-to-novel generalization and few-shot image classification tasks across 11 diverse datasets. TAP leverages a hierarchical "Tree of Attribute" framework, distilling structured knowledge graphs from LLMs for nuanced representation of visual concepts, and employs learnable "domain expert" tokens and a vision-conditional pooling module for optimal image-text alignment. While promising, we note that the reliance on LLMs presents challenges in fine-grained datasets where similar classes require nuanced differentiation, in which cases LLMs generate identical descriptions for distinct classes, impacting novel class prediction performance. It highlights the current limitations of LLMs in discerning highly fine-grained distinctions. Addressing this challenge through enhanced LLM capabilities or alternative strategies will be a key focus of future research.

References

[1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 3:11–12, 2022.

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 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*, pages 446–461. Springer, 2014.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Adrian Bulat and Georgios Tzimiropoulos. Lasp: Text-to-text optimization for language-aware soft prompting of vision & language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23232–23241, 2023.

[5] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Prompt learning with optimal transport for vision-language models. In *ICLR*, 2023.

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor G Turrisi da Costa, Cees GM Snoek, Georgios Tzimiropoulos, and Brais Martinez. Bayesian prompt learning for image-language model generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15237–15246, 2023.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

[10] Zalan Fabian, Zhongqi Miao, Chunyuan Li, Yuanhan Zhang, Ziwei Liu, Andrés Hernández, Andrés Montes-Rojas, Rafael Escucha, Laura Siabatto, Andrés Link, et al. Multimodal foundation models for zero-shot animal species recognition in camera trap images. *arXiv preprint arXiv:2311.01064*, 2023.

[11] Li Fei-Fei, Rob Fergus, and Pietro Perona. 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*, pages 178–178. IEEE, 2004.

[12] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: 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, 2019.

[13] 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 *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[14] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022.

[15] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023.

[16] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15190–15200, October 2023.

[17] Jae Myung Kim, A Koepke, Cordelia Schmid, and Zeynep Akata. Exposing and mitigating spurious correlations for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2584–2594, 2023.

[18] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[19] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1401–1411, 2023.

[20] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021.

[21] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation, 2021.

[22] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602, 2021. URL `https://arxiv.org/abs/2110.07602`.

[23] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5206–5215, 2022.

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

[25] Chengzhi Mao, Revant Teotia, Amrutha Sundar, Sachit Menon, Junfeng Yang, Xin Wang, and Carl Vondrick. Doubly right object recognition: A why prompt for visual rationales. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2722–2732, 2023.

[26] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *ICLR*, 2023.

[27] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[28] Zachary Novack, Julian McAuley, Zachary Lipton, and Saurabh Garg. Chils: Zero-shot image classification with hierarchical label sets. In *International Conference on Machine Learning (ICML)*, 2023.

[29] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[30] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[31] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*, 2017.

[32] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701, 2023.

[33] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[34] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6545–6554, 2023.

[35] Karsten Roth, Jae Myung Kim, A. Sophia Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Waffling around for performance: Visual classification with random words and broad concepts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15746–15757, October 2023.

[36] Shuvendu Roy and Ali Etemad. Consistency-guided prompt learning for vision-language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=wsRXwlwx4w.

[37] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[38] Cheng Shi and Sibei Yang. Logoprompt: Synthetic text images can be good visual prompts for vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2932–2941, 2023.

[39] 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.

[40] Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. Argue: Attribute-guided prompt tuning for vision-language models. *arXiv preprint arXiv:2311.16494*, 2023.

[41] Dongsheng Wang, Miaoge Li, Xinyang Liu, MingSheng Xu, Bo Chen, and Hanwang Zhang. Tuning multi-mode token-level prompt alignment across modalities. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=A253n2EXCd.

[42] Wenhao Wang, Yifan Sun, Wei Li, and Yi Yang. TransHP: Image classification with hierarchical prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=vpQuCsZXz2.

[43] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer, 2022.

[44] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022.

[45] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4582–4591, 2017.

[46] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[47] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, Peng Wang, and Yanning Zhang. Dual modality prompt tuning for vision-language pre-trained model. *IEEE Transactions on Multimedia*, pages 1–13, 2023. doi: 10.1109/TMM.2023.3291588.

[48] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3090–3100, 2023.

[49] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197, 2023.

[50] Yi Zhang, Ce Zhang, Ke Yu, Yushun Tang, and Zhihai He. Concept-guided prompt learning for generalization in vision-language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38 (7):7377–7386, Mar. 2024. doi: 10.1609/aaai.v38i7.28568. URL https://ojs.aaai.org/index.php/AAAI/article/view/28568.

[51] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Neural prompt search. *arXiv preprint arXiv:2206.04673*, 2022.

[52] Zhaoheng Zheng, Jingmin Wei, Xuefeng Hu, Haidong Zhu, and Ram Nevatia. Large language models are good prompt learners for low-shot image classification. *arXiv preprint arXiv:2312.04076*, 2023.

[53] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022.

[54] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[55] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023.

A Appendix

A.1 Model regularization

Denote the frozen image feature from CLIP vision encoder as $\mathbf{f}^v$, the frozen text feature for description d from CLIP text encoder as $\mathbf{f}_d^t$, and the zero-shot logit prediction from CLIP as $\hat{y}$. Additionally, denote the trained image feature as $\tilde{\mathbf{f}}^v$, the trained text feature for description d as $\tilde{\mathbf{f}}_d^t$, and the logit prediction from attribute a after training as $\tilde{y}_a$. The losses are as follows:

$$L_{L_1-V} = ||\mathbf{f}^v - \tilde{\mathbf{f}}^v||_1 \tag{8}$$

$$L_{con-T} = -\sum_{d\in\mathcal{D}} \left(\frac{1}{2} \log \frac{\exp(cos(\mathbf{f}_d^t, \tilde{\mathbf{f}}_d^t))}{\sum_{k\in\mathcal{D}_s} \exp(cos(\mathbf{f}_d^t, \tilde{\mathbf{f}}_k^t))} + \frac{1}{2} \log \frac{\exp(cos(\mathbf{f}_d^t, \tilde{\mathbf{f}}_d^t))}{\sum_{k\in\mathcal{D}_s} \exp(cos(\mathbf{f}_k^t, \tilde{\mathbf{f}}_d^t))} \right) \tag{9}$$

$$L_{KL-attr} = \frac{1}{|\mathcal{A}|} \left(\sum_{a\in\mathcal{A}} \mathcal{D_{KL}}(\hat{y}, \tilde{y}_a) \right) \tag{10}$$

The regularization loss is then:

$$L_{reg} = \mu_1 L_{L_1-V} + \mu_2 L_{KL-attr} + \mu_3 L_{con-T}, \tag{11}$$

Our overall training objective is thus given by:

$$L_{\text{total}} = L_{\text{class}} + L_{\text{reg}} \tag{12}$$

A.2 Additional implementation details

We use PyTorch [31] to implement all experiments on a single NVIDIA A100-80GB GPU. Our code is developed based on the implementation of CoOp [54], which is available at https://github.com/KaiyangZhou/CoOp and released under the MIT license. Our code is also released under the MIT license. Baseline results for both base-to-novel generalization and few-shot classification are taken from their respective publications. For the "global context" attribute which is aligned with the CLS token in the vision encoder, we use the following 7 selected templates provided in [33].

```
"itap of a {class}."
"a bad photo of the {class}."
"a origami {class}."
"a photo of the large {class}."
"a {class} in a video game."
"art of the {class}."
"a photo of the small {class}."
```

A.3 Prompts for Tree-of-Attribute generation

As introduced in Section 3.3, we generate the Tree-of-Attribute with the following three steps: 1) Attribute Generation, 2) In-Context Example Generation, and 3) Description Generation for All Classes. The prompts for each step are as follows:

1) Attribute Generation:

{Dataset Description.}

Visual attributes refer to observable, describable features of the images that can include color, shape, size, texture, and any specific patterns or markings, which can help differentiate between classes for the dataset. They *should be consistently observable across multiple images of the same class. Your task is to generate a list of visual attributes (less than 10) for the {Dataset Name} dataset. Ensure this list is clear, concise, and specific to the dataset's needs. Avoid generic attributes that do not contribute to distinguishing between classes.*

2) In-Context Example Generation

Describe describe what a "{Random Class Name}" class in the {Dataset Name} dataset look like using the generated visual attributes.

You must follow the following rules:

1. For each visual attribute, describe all possible variations as separate sentences. This approach allows for a detailed and clear presentation of each attribute's range.

2. Provide a maximum of five descriptions for each visual attribute to maintain focus and relevance. Also, aim to provide at least two descriptions to ensure a comprehensive overview of the attribute.

3. The descriptions should provide clear, distinguishable features of each class to support image classification tasks.

4. Descriptions for each attribute are independent from each other, and they should not serve as context for each other.

5. Each description describes an image independetly. If certain description is possible for a class, please just list that description, and do not use words like "may have" or "sometimes have".

6. Reply descriptions only. Do not include any explanation before and after the description.

7. The descriptions should follow the format of "classname, which ...", where "..." is the description of the visual attribute.

3) Description Generation for All Classes

{Dataset Description.}

Your task is to write detailed descriptions for various classes within the {Dataset Name} dataset, using the provided visual attributes such as color and shape. These descriptions will help in accurately classifying and understanding the unique features of each class.

You must follow the following rules:

1. For each visual attribute, describe all possible variations as separate sentences. This approach allows for a detailed and clear presentation of each attribute's range.

2. Provide a maximum of five descriptions for each visual attribute to maintain focus and relevance. Also, aim to provide at least two descriptions to ensure a comprehensive overview of the attribute.

3. The descriptions should provide clear, distinguishable features of each class to support image classification tasks.

4. Descriptions for each attribute are independent from each other, and they should not serve as context for each other.

5. Each description describes an image independetly. If certain description is possible for a class, please just list that description, and do not use words like "may have" or "sometimes have".

6. Reply descriptions only. Do not include any explanation before and after the description.

7. The descriptions should follow the format of "classname, which ...", where "..." is the description of the visual attribute.

Q: Describe what a "{Random Class Name}" in the {Dataset Name} look like using the following visual attributes: {Visual Attributes from Step 1.}

A: {Answer from Step 2.}

Q: Describe what a "{Target Class Name}" in the {Dataset Name} look like using the following visual attributes: {Visual Attributes from Step 1.}

A:

In the prompt templates, *"Dataset Description"* is the description of the dataset from their official website, *"Random Class Name"* is a randomly sampled class name in the dataset for in-context example generation, and *"Target Class Name"* is the class name of interest for the current query. While step 1 and 2 are made in two consecutive calls to provide contexts which are queried once per dataset, step 3 is queried independently for each of the remaining classes in the dataset. Human review is performed after step 2 to ensure a high-quality set of attributes and in-context example.

A.4 Potential societal impacts

While our work primarily focuses on advancing prompt learning in vision-language models, it's crucial to acknowledge the potential broader societal implications of such advancements. On the positive side, TAP could lead to more efficient and accurate image understanding systems, benefiting various domains. For instance, it could enhance accessibility for visually impaired individuals by providing more detailed descriptions of visual content. Furthermore, improved visual understanding could contribute to more effective content moderation, mitigating the spread of harmful online materials. However, these advancements also present potential risks. LLMs used for description generation can perpetuate existing societal biases present in their training data, leading to biased outcomes in image recognition. Moreover, sophisticated VLMs could be misused to create misleading visual content, contributing to misinformation and manipulation. The enhanced ability to analyze and understand images also raises privacy concerns, particularly in surveillance contexts where personal information could be extracted from visual data. Addressing these potential negative impacts necessitates careful consideration of bias mitigation techniques during LLM training, promoting transparency and explainability in VLM decision-making, and establishing ethical guidelines for responsible development and deployment of such technologies.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the problem of limited context in existing prompt learning methods, propose TAP as a solution using structured knowledge graphs and domain experts, and highlight the strong experimental results in both base-to-novel generalization and few-shot classification. This accurately reflects the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a discussion of the limitations associated with relying on LLMs for generating descriptions, particularly in fine-grained datasets where similar classes require nuanced differentiation. This discussion can be found in "Conclusion".

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper focuses on proposing a novel method for prompt learning in VLMs and evaluating its empirical performance. It doesn't introduce any new theoretical results or theorems requiring formal proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides all the necessary information for reproducing the experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Our codebase is built based on the CoOP and CoCoOP [54, 53], and can be reproduced based on our Methods, Implementation details in main text and appendix. Our code will be released upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The training and test details can be found in section Experiments and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow previous works [54, 53] to report results averaged over 3 runs. Error bars are not reported.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The type of compute used is provided in Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We conform with the NeurIPS Code of Ethics in every aspect.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The potential societal impacts are discussed in Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper primarily focuses on a novel prompt learning method and doesn't involve the release of a new pre-trained LLM, image generator, or scraped dataset. Therefore, this question doesn't directly apply in this context. We leverage an existing pre-trained LLM (GPT-3.5-turbo), and any ethical considerations regarding its release and potential misuse fall under the responsibility of its creators.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We credited the creators of the CoOp codebase [54] by including the attribution statement in appendix.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: Code will be realseased upon acceptance.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our paper doesn't involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper focuses on developing a novel prompt learning method and evaluating its performance on established image recognition datasets. It doesn't involve any form of crowdsourcing, human subject research, or data collection that would necessitate IRB approval or ethical considerations related to study participants. Therefore, this question doesn't apply to our research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.