TREE OF ATTRIBUTES PROMPT LEARNING FOR VISION-LANGUAGE MODELS

Tong Ding[1,2] **Wanhua Li**[1*] **Zhongqi Miao**[3] **Hanspeter Pfister**[1]
[1]Harvard University [2]Mass General Brigham [3]Microsoft

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 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 category, 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, cross-dataset transfer, as well as few-shot classification across 11 diverse datasets. Code is available at https://github.com/HHenryD/TAP.

1 INTRODUCTION

Recent advancements in vision-language models (VLMs) like CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) 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) (Lester et al., 2021; Li & Liang, 2021; Liu et al., 2021) and vision-only models (Jia et al., 2022; Wang et al., 2022a;b; Zhang et al., 2022). 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 (Zhou et al., 2022b;a; Zhu et al., 2023; Derakhshani et al., 2023; Lu et al., 2022). Techniques like CoOp (Zhou et al., 2022b) 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 (Khattak et al., 2023a;b; Shi & Yang, 2023; Lee et al., 2023). 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 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

*Corresponding Author

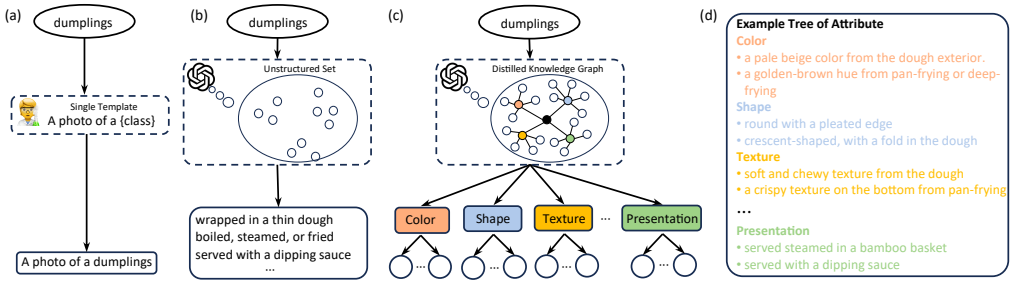

Figure 1: Illustration of the methods for CLIP text prompts formation. (a) Manually created prompt with the single "a photo of a {class}" template; (b) A unstructured set of detailed descriptions generated by LLMs; (c) The proposed Tree of Attribute distills a knowledge graph from LLMs, organizing the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

capabilities (Menon & Vondrick, 2023; Pratt et al., 2023; Roth et al., 2023; Li et al., 2024; Kim et al., 2023; Parkhi et al., 2012; Yan et al., 2023; Yang et al., 2023; Roy & Etemad, 2024; Zheng et al., 2023; Tian et al., 2023). 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 (Roth et al., 2023), 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 base-to-novel generalization, cross-dataset transfer, 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. On cross-dataset transfer, TAP outperforms existing methods on both source and target datasets by 1.03% and 0.75% in average. 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, originating in NLP (Lester et al., 2021; Li & Liang, 2021; Liu et al., 2021) and later adapted to vision-only (Jia et al., 2022; Wang et al., 2022a;b; Zhang et al., 2022) and multimodal settings (Zhou et al., 2022b;a; Khattak et al., 2023a;b; Shi & Yang, 2023; Lee et al., 2023; Tian et al., 2023; Rasheed et al., 2023; Roy & Etemad, 2024; Zheng et al., 2023; Zhu et al.,

2023; Bulat & Tzimiropoulos, 2023; Lu et al., 2022). CoOp (Zhou et al., 2022b) introduced learnable continuous prompts for few-shot image recognition but struggled to generalize to unseen classes, highlighting the challenge of base-to-novel generalization (Zhou et al., 2022a; Guo et al., 2024b; Hernandez-Hernandez et al., 2024; Guo et al., 2024a). CoCoOp (Zhou et al., 2022a) addressed this by conditioning prompts on visual features. Visual and multimodal prompt tuning methods, such as VPT (Bahng et al., 2022), DPT (Xing et al., 2023), and MaPLe (Khattak et al., 2023a), optimize prompts in pixel or text space to enhance feature alignment. Other works (Khattak et al., 2023b; Bulat & Tzimiropoulos, 2023; Li et al., 2022; Roy & Etemad, 2024) focus on regularization to improve generalization. PromptSRC introduced self-regulating prompts to better retain base knowledge, while methods like PLOT (Chen et al., 2023) and ALIGN (Wang et al., 2023) apply Optimal Transport to align prompts with local features. Our work differs by introducing a hierarchical "Tree of Attribute" framework to structure textual descriptions and guide specialized "domain expert" tokens for attribute-level understanding.

Enhancing model's understanding using visual attributes. There's a growing emphasis on the use of detailed visual descriptions for various visual understanding tasks, including more fine-grained captioning (Hsieh et al., 2024), identifying subordinate-level categories (Liu et al., 2024a), and language-guided visual classification (Menon & Vondrick, 2023). While manual creation is impractical given the large number of image classes, existing research relies either on data augmentation (Kim et al., 2024) or generation by LLMs such as GPT-3 (Brown et al., 2020), which offers an efficient generation of a broad spectrum of class-specific descriptions. These descriptions, like "fur pattern" or "tail shape" of a cat, provide fine-grained and distinctive characteristics. In an essence, such approaches can be viewed as knowledge distillation from LLMs trained on trained on vast and diverse textual corpora. However, existing studies often lack a structured methodology for distillation (Kim et al., 2023; Menon & Vondrick, 2023; Parkhi et al., 2012; Roth et al., 2023; Yan et al., 2023; Yang et al., 2023; Fabian et al., 2023; Pratt et al., 2023; Novack et al., 2023; Mao et al., 2023; Tian et al., 2023; Zheng et al., 2023; Zhang et al., 2024; Liu et al., 2024b) or fail to effectively exploit the inherent hierarchy within the knowledge (Maniparambil et al., 2023; Wang et al., 2024; Hsieh et al., 2024; Liu et al., 2024a). Our approach (TAP) addresses these limitations by introducing a novel method to distill a knowledge graph from LLMs in a top-down manner, transitioning from class names (concepts) to visual attributes (e.g., color, shape) and further to detailed descriptions of each attribute, forming a structured Tree of Attributes (ToA). To fully leverage the ToA, we propose a bottom-up integration pipeline. We introduce vision-conditional pooling (VCP) layers to aggregate descriptions into attribute-level features, effectively mitigating potential noise in the generated descriptions. The alignment between attributes and introduced visual expert tokens is then refined through this hierarchical structure. This integration enables the model to exploit structured relationships within the ToA, enhancing both the granularity and interpretability of vision-text alignment.

3 METHODOLOGY

3.1 PRELIMINARY

CLIP. Our approach is built on the pre-trained vision-language model, CLIP (Radford et al., 2021). Formally, let (x, c) denote the dataset, where x is an image and $c \in \{1, \ldots, C\}$ are the class labels. For an image x, the vision encoder $h_I(\cdot)$ transforms it into a feature vector $\boldsymbol{f}_x^v = h_I(x)$. Simultaneously, each class label c is mapped to a text prompt $t_c =$ `a photo of a {c}`, and converted into textual feature vectors $\boldsymbol{f}_c^t = h_T(t_c)$. The predicted class $\hat{y}$ is given by:

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

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 (Menon & Vondrick, 2023; Roth et al., 2023) 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 $\mathbb{D}_c$: {`"c, which is/has/etc description."` }, e.g. `c="television"`

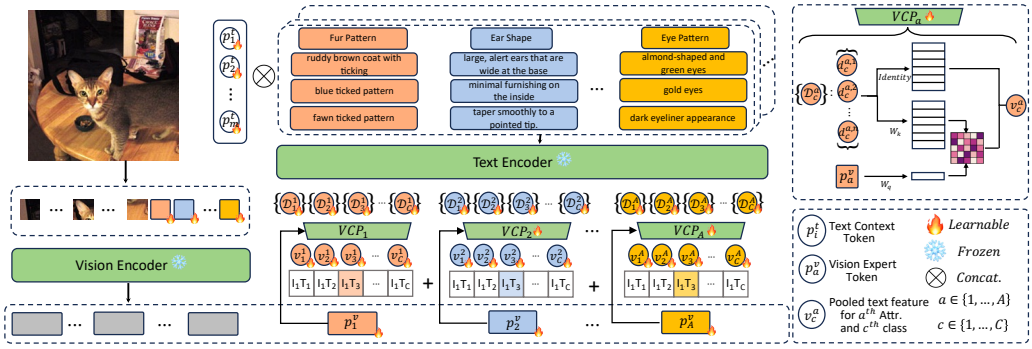

Figure 2: Overview of the proposed TAP method. TAP uses a bottom-up approach to aggregate the generated Tree of Attribute. The vision-conditional pooling (VCP) layer aggregates descriptions into attribute-level features, which are aligned with visual expert tokens focusing on specific attributes (e.g., color, texture). These attribute-level features are then combined to make class predictions via a weighted sum of logits from each attribute, fully leveraging the hierarchical structure within the tree.

and `description="black or grey"`. This classification is reformulated as

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

3.2 Overall Framework

We rethink the descriptions by LLM $\mathbb{D}_c$ as nodes in knowledge graphs. While previous methods generate an unstructured set of descriptions, we distill structured knowledge graphs for each class c 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 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 (Dosovitskiy et al., 2021). 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, which 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 (Roth et al., 2023) 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, TAP leverages the tree structure in two key ways: first, a top-down process generates attributes and corresponding descriptions for each class in a structured and contextually relevant manner. This ensures that the descriptions are structured and contextually relevant. Second, a bottom-up process aggregates information from the leaf nodes (descriptions) into attribute-level features, which are aligned with visual expert tokens. These expert tokens focus on fine-grained visual attributes, such as color or shape. Finally, the aggregated attribute-level features contribute to class predictions using a weighted sum of prediction logits, fully utilizing the hierarchical relationships within the tree. This dual approach allows TAP to capture both high-level structure and fine-grained

details, leading to enhanced alignment of visual and textual data and improved model performance and interpretability. Inspired by CoOP (Zhou et al., 2022b), 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 $\mathcal{G}_c = \{\mathbb{V}_c, \mathbb{E}_c\}$ for each class c, where $\mathbb{V}_c$ denotes the set of nodes, and $\mathbb{E}_c$ denotes the edges that capture the semantic relationship between nodes. In previous works, $\mathbb{V}_c$ is the set of descriptions $\mathbb{D}_c$, while $\mathbb{E}_c$ 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, $\mathbb{G}_c$ 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. $\mathbb{E}_c$ 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 $\mathbb{A}_c$ denote the set of attributes, and for each attribute $a_c \in \mathbb{A}_c$, we denote its leaf nodes as $\mathbb{D}_c^a$. Each set $\mathbb{D}_c^a$ contains descriptions that specifically pertain to attribute a for class c, which is denoted as

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

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 $\mathbb{A}$ 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 $\mathbb{A}$ identified in the previous step. Each description takes the format of "class, which {is/has/etc} {description}". 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 (Radford et al., 2021).

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 $a \in \mathbb{A}$. 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 $\mathbb{P}^v = \{\boldsymbol{p}_a^v | a \in \mathbb{A}\}$. Akin to the idea of visual prompt tuning (VPT) (Jia et al., 2022), we insert $\mathbb{P}^v$ into the input sequence of the vision encoder, forming the prompted input sequences $\tilde{\mathbb{X}}_p = \{\boldsymbol{e}_{\text{CLS}}, \mathbb{P}^v, \mathbb{E}_{\text{patch}}\}$, where $\boldsymbol{e}_{\text{CLS}}$ is the input CLS token, and $\mathbb{E}_{\text{patch}}$ 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 $\mathbb{P}^t = \{\boldsymbol{p}_j^t | j \in \{1, 2, ..., m\}\}$ for the text encoder shared across all descriptions, similar to (Zhou et al., 2022b).

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 $\mathbb{D}_a$ 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 $\boldsymbol{p}_{a,x}^v$. 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 $\boldsymbol{p}_a^v$ and all embedded descriptions in attribute $\mathbb{D}_a$, 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 $\boldsymbol{p}_a^v$, the pooled attribute-level embedding $\boldsymbol{v}_c^a$ for class c and attribute a is:

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

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 $\boldsymbol{W_v}$ 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:

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

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:

$$L_{class} = \frac{1}{|\mathbb{A}|}\Big(\sum_{a\in\mathbb{A}} L_{con}(\boldsymbol{p}_a^v, \boldsymbol{v}_c^a))\Big), \tag{6}$$

Similar to (Khattak et al., 2023b) and (Bulat & Tzimiropoulos, 2023), 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:

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

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 three 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; 2) Cross-dataset transfer, where we train on ImageNet with 16 shots per class, and directly evaluate on other datasets in zero-shot; and 3) Few-shot classification with 16 shots per class.

Table 1: Comparison with state-of-the-art methods in base-to-novel generalization. The model is trained on the base class, and evaluated on the unseen novel classes in zero-shot. TAP demonstrates strong generalization performance. HM: harmonic mean (Xian et al., 2017).

(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**

Datasets and baslines. For all of the three settings, we follow previous works (Zhou et al., 2022b;a), using 11 image recognition datasets, including: ImageNet (Deng et al., 2009) and Caltech101 (Fei-Fei et al., 2004) for generic object recognition; OxfordPets (Parkhi et al., 2012), StanfordCars (Krause et al., 2013), Flowers102 (Nilsback & Zisserman, 2008), Food101 (Bossard et al., 2014), and FGVCAircraft (Maji et al., 2013) for fine-grained classification; SUN397 (Xiao et al., 2010) for scene recognition; UCF101 (Soomro et al., 2012) for action recognition; DTD (Cimpoi et al., 2014) for texture classification; and EuroSAT (Helber et al., 2019) for satellite image analysis. We benchmark against several leading methods, including CLIP (Radford et al., 2021), CoOp (Zhou et al., 2022b), Co-CoOP (Zhou et al., 2022a), ProGrad (Zhu et al., 2023), RPO (Lee et al., 2023), LoGoPrompt (Shi & Yang, 2023), and the state-of-the-art PromptSRC (Khattak et al., 2023b).

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 (Ouyang et al., 2022) for attribute and description generation. We initialize the text context tokens with the word embedding of `"a photo of a."` During training, we iteratively train the vision and text encoders with 5 epochs for vision and 1 epoch for text schedule. We train a total of 60, 24, and 120 epochs for base-to-novel generalization, cross-dataset transfer, and few-shot classification respectively. We set $\alpha = 0.4$ for all datasets. We also use a Gaussian Prompt Weighting (GPA) following (Khattak et al., 2023b), with a mean of $0.9N$, std of $0.1N$, where N represents the total number of epochs, for all tasks. Refer to the Appendix for additional implementation details.

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 in Table 1. Notably, our method improves unseen class prediction without compromising base class performance, exhibiting an average performance boost

Table 2: Comparison with state-of-the-art methods in cross-dataset transfer evaluation. The model is trained on the source dataset and evaluated on the target datasets in zero-shot.

	Source	**Target**										
	ImageNet	Caltech101	Pets	Cars	Flowers	Food101	Aircraft	SUN397	DTD	EuroSAT	UCF101	Average
CoOp	71.51	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	**46.39**	66.55	63.88
CoCoOp	71.02	**94.43**	90.14	65.32	**71.88**	86.06	22.94	67.36	45.73	45.37	68.21	65.74
PSRC	71.27	93.60	90.25	**65.70**	70.25	**86.15**	23.90	67.10	46.87	45.50	68.75	65.81
TAP	**72.30**	94.30	**90.70**	65.60	70.93	86.10	**24.57**	**68.30**	**50.20**	46.00	**68.90**	**66.56**

Table 3: Comparison with state-of-the-art methods in 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
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**

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 Cross-dataset Transfer

To further investigate the generalization capability of TAP , we train on ImageNet with 16 shots per class, and directly test on the other 10 datasets under zero-shot without further tuning. As shown in Table 2, TAP demonstrates better generalizability on 8/10 target datasets compared to PromptSRC (Khattak et al., 2023b), and achieves an average performance increase of 0.75%. Additionally, while the performance increase of previous methods on target datasets come with costs on the source dataset (-0.49% for CoCoOP and -0.24% for PromptSRC) as compared to CoOP (Zhou et al., 2022b), TAP also outperform previous methods on the source dataset with 1.03% increase compared to PromptSRC (0.79% incrase compared to CoOP), demonstrating TAP 's robustness in domain generalization without sacrifice on source dataset performance.

4.3 Few-Shot Classification

In few-shot classification, TAP also outperforms existing methods in 9 out of the 11 datasets. Detailed in Table 3, 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.

4.4 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 (Mao et al., 2023; Maniparambil et al., 2023; Liu et al., 2024b; Wang et al., 2024; Tian et al., 2023; Zheng et al., 2023), while keeping the same number of visual prompt tokens. According to Table 4, 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.

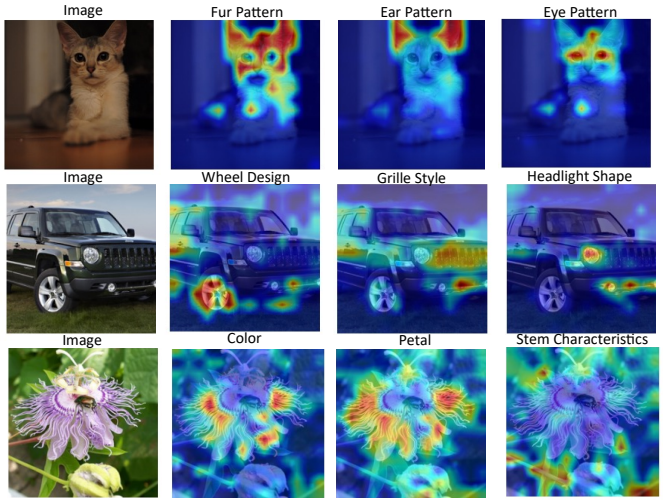

Figure 3: Visualization of the class activation maps.

Table 4: 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 5: Effects of domain experts.

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

Table 6: Effects of α

α	1.0	0.4
Base	81.54	**84.75**
Novel	73.85	**77.63**
HM	77.51	**81.04**

Table 7: Effects of the number of experts.

# Attrs.	1	2	3	4	5	6	7	8	Ours
Base	83.20	83.97	84.10	84.41	84.45	84.62	84.66	84.74	**84.75**
Novel	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**

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 (Mao et al., 2023; Lee et al., 2023; Tian et al., 2023; Zheng et al., 2023). Our findings (Table 5) 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 fusion coefficient α. α in Eq. (7) balance global and local information. We compare the performance of using CLS token only (i.e. $\alpha = 1.0$) for making the final prediction against our proposed prediction fusion with $\alpha = 0.4$. As shown in Table 6, using CLS token decreases the performance significantly on both base and novel classes. This result further demonstrates the limitations of using a singular CLS token which focuses on global information, and supports the effectiveness of the use of expert tokens which focus on local information.

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 7, 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), 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.

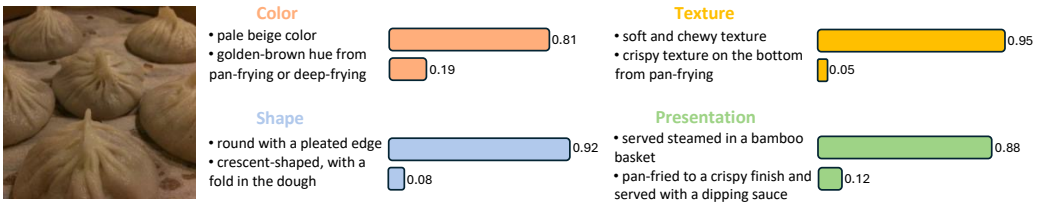

Figure 4: Visualization of the attention weights in the VCP layer for an example "dumplings" image.

Table 8: Ablation on design choice of the VCP layer. Our cross-attention based pooling mechanism demonstrates the best performance among other variants.

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**

4.5 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 (Selvaraju et al., 2017), 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 base-to-novel generalization, cross-dataset transfer, 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.

ACKNOWLEDGMENT

This work was supported in part by Microsoft's AI for Good Research Lab, the Harvard Data Science Initiative, and NIH Grant R01HD104969.

REFERENCES

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.

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*, pp. 446–461. Springer, 2014.

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.

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*, pp. 23232–23241, 2023.

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.

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*, pp. 3606–3613, 2014.

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*, pp. 248–255. Ieee, 2009.

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*, pp. 15237–15246, 2023.

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`.

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.

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*, pp. 178–178. IEEE, 2004.

Qianrong Guo, Saiveth Hernandez, and Pedro Ballester. Umap-clustering split for rigorous evaluation of ai models for virtual screening on cancer cell lines. 2024a.

Qianrong Guo, Saiveth Hernandez-Hernandez, and Pedro J Ballester. Scaffold splits overestimate virtual screening performance. In *International Conference on Artificial Neural Networks*, pp. 58–72. Springer, 2024b.

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.

Saiveth Hernandez-Hernandez, Qianrong Guo, and Pedro J Ballester. Conformal prediction of molecule-induced cancer cell growth inhibition challenged by strong distribution shifts. *bioRxiv*, pp. 2024–03, 2024.

Yu-Guan Hsieh, Cheng-Yu Hsieh, Shih-Ying Yeh, Louis Béthune, Hadi Pouransari, Pavan Kumar Anasosalu Vasu, Chun-Liang Li, Ranjay Krishna, Oncel Tuzel, and Marco Cuturi. Graph-based captioning: Enhancing visual descriptions by interconnecting region captions. *arXiv preprint arXiv:2407.06723*, 2024.

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*, pp. 4904–4916. PMLR, 2021.

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*, pp. 709–727. Springer, 2022.

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*, pp. 19113–19122, 2023a.

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)*, pp. 15190–15200, October 2023b.

Gahyeon Kim, Sohee Kim, and Seokju Lee. Aapl: Adding attributes to prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1572–1582, 2024.

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*, pp. 2584–2594, 2023.

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*, pp. 554–561, 2013.

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*, pp. 1401–1411, 2023.

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

Wanhua Li, Xiaoke Huang, Zheng Zhu, Yansong Tang, Xiu Li, Jie Zhou, and Jiwen Lu. Ordinalclip: Learning rank prompts for language-guided ordinal regression. *NeurIPS*, 35:35313–35325, 2022.

Wanhua Li, Zibin Meng, Jiawei Zhou, Donglai Wei, Chuang Gan, and Hanspeter Pfister. Socialgpt: Prompting llms for social relation reasoning via greedy segment optimization. *NeurIPS*, 2024.

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

Mingxuan Liu, Subhankar Roy, Wenjing Li, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Democratizing fine-grained visual recognition with large language models. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=c7DND1iIgb.

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.

Xin Liu, Jiamin Wu, Wenfei Yang, Xu Zhou, and Tianzhu Zhang. Multi-modal attribute prompting for vision-language models. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024b.

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*, pp. 5206–5215, 2022.

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

Mayug Maniparambil, Chris Vorster, Derek Molloy, Noel Murphy, Kevin McGuinness, and Noel E O'Connor. Enhancing clip with gpt-4: Harnessing visual descriptions as prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 262–271, 2023.

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*, pp. 2722–2732, 2023.

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

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*, pp. 722–729. IEEE, 2008.

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.

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.

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

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.

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*, pp. 15691–15701, 2023.

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*, pp. 8748–8763. PMLR, 2021.

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*, pp. 6545–6554, 2023.

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)*, pp. 15746–15757, October 2023.

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`.

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*, pp. 618–626, 2017.

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*, pp. 2932–2941, 2023.

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.

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

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.

Yubin Wang, Xinyang Jiang, De Cheng, Dongsheng Li, and Cairong Zhao. Learning hierarchical prompt with structured linguistic knowledge for vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5749–5757, 2024.

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*, pp. 631–648. Springer, 2022a.

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*, pp. 139–149, 2022b.

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*, pp. 4582–4591, 2017.

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*, pp. 3485–3492. IEEE, 2010.

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*, pp. 1–13, 2023. doi: 10.1109/TMM.2023.3291588.

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*, pp. 3090–3100, 2023.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

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*, pp. 19187–19197, 2023.

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.

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

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.

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*, pp. 16816–16825, 2022a.

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, 2022b.

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*, pp. 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:

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

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

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

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:

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

To investigate the effectiveness of model regularization, we compare TAP against existing methods with and without regularization. As evidenced in Table 9, the proposed model regularization helps in both base and novel performance, with an increase of 1.62% in average harmonic mean. Comparing to existing methods, TAP is consistently better than other baselines in both settings, demonstrating the robustness of our method.

Table 9: Effectiveness of model regularization. TAP achieves favorable results under both settings.

	Regularization	Base	Novel	HM
PSRC-reg	×	84.21	71.79	77.51
MaPLe	×	82.28	75.14	78.55
TAP-reg	×	**83.37**	**75.82**	**79.42**
PSRC	✓	84.26	76.10	79.97
TAP	✓	**84.75**	**77.63**	**81.04**

A.2 ADDITIONAL IMPLEMENTATION DETAILS

Because the number of attributes vary across the 11 datasets which results in different number of learnable parameters, we group the datasets into two and apply two sets of learning rates to balance generalizability and performance. For DTD, Oxford Flowers, Stanford Cars, UCF101, and Caltech101 datasets, which have fewer attributes, we use a low learning rate of 0.002 for the text encoder to avoid overfitting and a high learning rate of 0.006 for the vision encoder to facilitate the learning process. A high $\mu_3 = 3$ is also used to regularize the text encoder for preventing overfitting. For the remaining 6 datasets, which have more attributes, the learning rates for both text and vision encoders are set as 0.004, with $\mu_3 = 1.5$. $\mu_1 = 10$, and $\mu_2 = 2.5$ are used for all datasets.

For base-to-novel generalization and few-shot classification evaluations, we use an adaptive approach for generating the attributes, in which the attributes vary across datasets. Although it turns out to be better than using a fixed set of attributes as shown in Table 7, it is not applicable to the cross-dataset transfer experiment as both VCP layers and visual expert tokens are specific to their corresponding attributes. Therefore, for cross-dataset transfer, we use the following fixed set of 4 attributes that are applicable to all 11 datasets: Pattern, Texture, Shape, and Context.

We use PyTorch Paszke et al. (2017) to implement all experiments on a single NVIDIA A100-80GB GPU. Our code is developed based on the implementation of CoOp Zhou et al. (2022b), 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 the three tasks 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 Radford et al. (2021).

```
"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 Robustness of LLMs

To investigate the robustness of our methods against different LLMs, we additionally generate the descriptions using a locally-served small LLM - Qwen-2-7B-Instruct (Yang et al., 2024), in which the results are comparable.

Table 10: Robustness against different LLMs.

LLMs	Base Acc.	Novel Acc.	HM
Qwen-2-7B-Instruct	84.68	77.31	80.83
GPT-3.5-Turbo	84.75	77.63	81.04

A.4 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. Our carefully designed prompts for step 1 and 2 guide the LLM in generating high-quality examples. Human review further confirms that the generated in-context examples from these prompts are of high quality even without any manual intervention.

A.5 ATTRIBUTE SETS

The attribute sets generated by LLMs are shown in Table 11 - 12.

Table 11: Attribute sets generated by LLMs for the 11 datasets.

Dataset	Attributes
ImageNet	Orientation Shape Pattern Texture Pose Context
Caltech101	Dominant Feature Shape Texture Color Size
StanfordCars	Body Type Wheel Design Grille Style Headlight Shape Rear Taillight Design Roof Style
Flowers102	Color Petal Center structure Stem characteristics
Food101	Color Shape Texture Ingredients Presentation Style

Table 12: Attribute sets generated by LLMs for the 11 datasets. Cont.

Dataset	Attributes
FGVCAircraft	Wing Configuration Winglet Presence Engine Configuration Number of Engines Fuselage Length Fuselage shape Wingspan
SUN397	Indoor/Outdoor Color Dominant elements Environment Architectural style Patterns
DTD	Texture Pattern Repetition Contrast
EuroSAT	Contrast Texture Orientation Edge Size Color Symmetry
UCF101	Action Pose Number of People Background Setting Objects Present