Advancing Textual Prompt Learning with Anchored Attributes

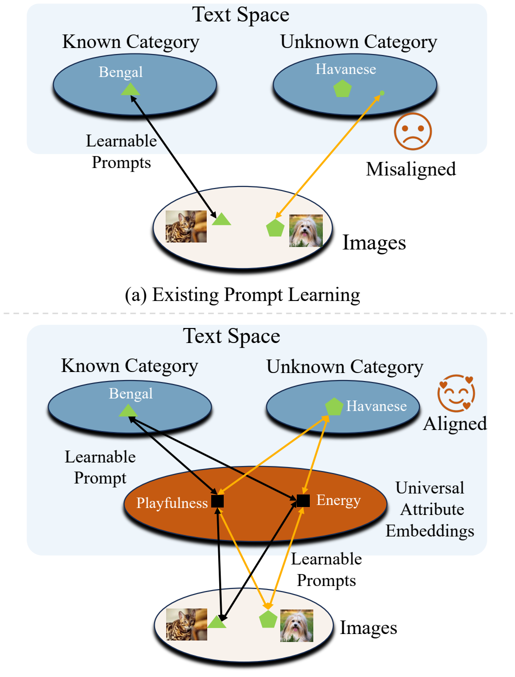
Zheng Li[1], Yibing Song[2], Ming-Ming Cheng[1], Xiang Li[1]*, Jian Yang[1]*
[1] PCA Lab, VCIP, College of Computer Science, Nankai University,
[2] DAMO Academy, Alibaba Group
zhengli97@mail.nankai.edu.cn, yibingsong.cv@gmail.com
{cmm, xiang.li.implus, csjyang}@nankai.edu.cn

Abstract

Textual-based prompt learning methods primarily employ multiple learnable soft prompts and hard class tokens in a cascading manner as text inputs, aiming to align image and text (category) spaces for downstream tasks. However, current training is restricted to aligning images with predefined known categories and cannot be associated with unknown categories. In this work, we propose utilizing universal attributes as a bridge to enhance the alignment between images and unknown categories. Specifically, we introduce an ***A****ttribute-anchored* ***T****extual* ***P****rompt learning method for vision-language models, named* ***ATPrompt****. This approach expands the learning space of soft prompts from the original one-dimensional category level into the multi-dimensional attribute level by incorporating multiple attribute tokens into the learnable soft prompts. Through this modification, we transform the text prompt from a category-centric form to an attribute-category hybrid form. Additionally, we introduce a straightforward differentiable attribute search method to identify representative and suitable attributes for downstream tasks. As an easy-to-use plug-in technique, ATPrompt can seamlessly replace the existing basic prompt format in textual-based methods, providing general improvements at a negligible computational cost. Extensive experiments across 11 datasets validate the effectiveness of our method. Code is publicly available at* `https://github.com/zhengli97/ATPrompt`*.*

1. Introduction

Vision-Language Models (VLMs) [16, 27, 28, 34, 35, 42, 46, 47], such as CLIP [4, 42] and ALIGN [16], have demonstrated exceptional performance in recent years. These models are trained with a contrastive loss to establish alignment between image and text (category) space. Inspired by the success of NLP [25, 26], prompt learning [17, 63, 64] has emerged as a parameter-efficient tool to adapt powerful VLMs to downstream tasks. Models with a few learnable soft prompt tokens can achieve performance parity with, or even outperform, fully fine-tuned ones [17]. Depending on how the soft prompt tokens are applied, existing methods can be broadly classified into textual-based [19, 20, 30, 53, 63, 64] and visual-based approaches [1, 2, 17, 23, 60]. Among these, the textual-based method is the most fundamental and straightforward, comprising the majority.



(a) Existing Prompt Learning

(b) Ours Attribute-anchored Prompt Learning

Figure 1. Comparison of image and text (category) alignment processes through learnable prompts. (a) Current prompt learning methods align images with predefined categories but fail to establish accurate associations with unknown categories. (b) ATPrompt leverages universal attributes as an intermediary to create more accurate alignments between images and unknown categories.

In typical image classification tasks, current text-based

*Corresponding author.

methods [19, 20, 63, 64] predominantly employ the traditional approach of concatenating learnable soft prompts with hard class tokens to replace handcrafted text prompts (e.g., “a photo of a {class}”) as inputs to the encoder. Although this text prompt demonstrates strong performance, it restricts image alignment during training to predefined known categories only, thereby preventing accurate associations with unknown categories, as shown in Fig. 1(a). Intuitively, when confronted with an unfamiliar category, humans often associate it with additional attributes (e.g., color, shape or texture) to increase comprehensibility and clarity, rather than merely stating the object’s name. For instance, one might describe a cheetah as: “The cheetah is a cat-like animal with a *small head*, *short yellow hair*, and *black spots*.” or refer to an apple as “That *red spherical* fruit with *orange stripes* is an apple.” instead of using a general description such as “This is a cheetah.” or “That fruit is an apple.” *Attributes can serve as bridges that connect unknown categories to our known knowledge.*

Building on these observations, we propose a novel approach that leverages attributes as a bridge to enhance the alignment between images and unknown categories. Specifically, we introduce an attribute-anchored textual prompt learning method for VLMs, named ATPrompt. This method extends the learning space of soft prompts from the original one-dimensional category level to a multi-dimensional attribute level by integrating multiple fixed universal attribute tokens into the learnable soft prompts. Guided by these anchored attributes, soft tokens acquire not only category-specific but also attribute-related general representations during training. This results in improved alignment between images and unknown categories compared to the original method, as shown in Fig. 1(b). Additionally, based on the depth at which soft prompts are applied, we propose two versions of ATPrompt: shallow and deep, respectively, ensuring compatibility with existing methods of varying depths [19, 20, 64]. To finalize these attributes, we present a simple and effective differentiable attribute search method that learns to identify suitable attributes from a candidate pool constructed by LLMs. The search operation only needs to be performed once per task, and once completed, the selected attributes can be used by ATPrompt for model training.

As an easy-to-use plug-in technique, ATPrompt can seamlessly substitute existing forms used in textual-based prompt learning methods, yielding general improvements with negligible additional computational overhead.

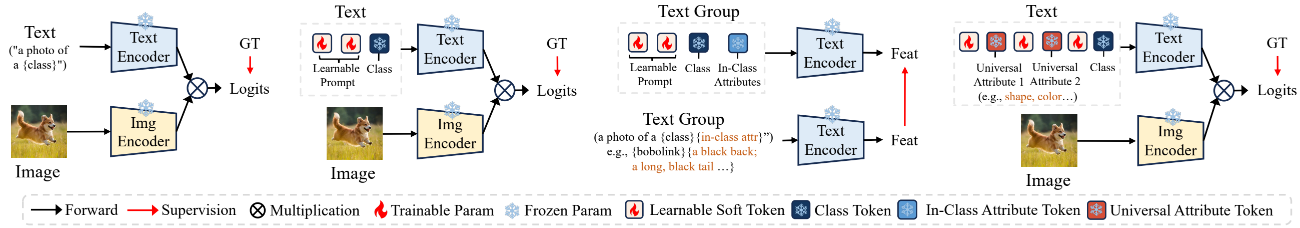
Our contributions can be summarized as follows:

- We propose an efficient attribute-anchored textual prompt learning method that expands the learning space of soft prompts from a one-dimensional class level to the multi-dimensional attribute level.
- We introduce an effective differentiable search method to select appropriate attributes for downstream tasks.
- Both shallow and deep versions of ATPrompt are introduced to ensure compatibility with existing prompt learning methods of varying depths.
- Extensive experiments demonstrate that ATPrompt can be seamlessly integrated into existing textual-based methods, resulting in consistently improved performance with negligible computational costs.

2. Related Work

Prompt Learning for VLMs. Inspired by recent advancements in NLP [25, 26], prompt learning [19, 20, 30, 44, 61, 63–65] has garnered significant interest among vision researchers aiming to apply these techniques to VLMs [16, 42, 56], such as CLIP. CoOp [64] is the pioneering text-based approach that introduced the concept of using a combination of soft textual tokens and a hard class token as input. Subsequent studies [19, 20, 24, 30, 54, 58, 63] have predominantly followed this textual prompt format. However, this form constrains the soft prompts to align with images within a one-dimensional, predefined category space, limiting their applicability to unknown categories. Therefore, training based on the current text form will be more likely to overfit to known categories, diminishing their zero-shot generalization capability for unknown categories. To address this limitation, several methods have been proposed [18, 20, 30, 58, 65]. For instance, KgCoOp [58] uses hand-crafted hard prompts to regularize learnable soft prompts during training. PromptSRC [20] utilizes CLIP’s [42] original features to regularize the learning of soft prompts for image and text branches. PromptKD [30] utilizes a pre-trained strong teacher model to guide the learning of a student model [14, 29, 31, 57] with learnable prompts. Despite these advancements, none of the above methods address the inherent limitations of the format itself. In this work, we introduce the attribute-anchored textual prompt format for VLMs, which proposes to utilize attributes as a bridge to build more accurate associations between images and unknown categories.

Attributes for VLMs. In practice, categories typically encompass multiple attributes. When individuals encounter an unfamiliar category, they often describe it using additional attributes to enhance the clarity of their communication, rather than merely stating its name. Inspired by this observation, numerous studies [5, 37, 49, 51, 59] have begun to leverage attributes to support their objectives. VCD [37] was the pioneering work to propose the use of LLMs to decompose class names into multiple *in-class* attributes (e.g., beak and tail for birds) for classification. AAPL [21] introduces a meta-network to extract visual attribute features based on encoded image features, facilitating image-text alignment. TAP [8] presents a structured “Tree of Attributes” approach that leverages attribute-specific knowl-



(a) Vanilla CLIP (b) Classic Prompt Learning (c) ArGue (d) ATPrompt (Ours)

Figure 2. Architectural comparison among existing methods. (a) Vanilla CLIP employs a hand-crafted text template as input to the text encoder. (b) Classical prompt learning proposes a new text form that concatenates multiple learnable soft tokens with class tokens. (c) ArGue [49] employs multiple *in-class* attributes mined by LLMs as supplementary information. These attributes are utilized to construct distinct text groups, which serve as *learning targets*, thereby regularizing the learning of soft tokens. The final prediction is achieved by ensembling all groups. (d) Our ATPrompt treats *universal attributes* as *learning components* and anchors them into existing soft prompt templates. Though this operation, we expand the learning space of soft tokens to multi-dimensional attribute levels and facilitate the alignment of images with unknown class texts.

edge graphs to enhance VLMs. ArGue [49] leverages large language models to mine multiple in-class attributes and integrates them into soft prompts and fixed templates to create multiple text groups. It employs the original text features generated by the fixed template to regularize the learning of soft tokens, as illustrated in Fig. 2(c). Most prior studies have focused on utilizing in-class attributes to enhance model performance by providing supplementary attribute information. However, when dealing with an unknown class, reacquiring the attributes of the new class becomes necessary—a process that is both complex and costly.

In this work, we believe that universal (inter-class) attributes are more efficient and robust than the in-class attributes used in previous works. Instead of taking attributes as the learning objective, we treat it as a learning component and propose to anchor universal attributes into soft prompt templates, transforming the existing class-centric form [64] into a hybrid attribute-class learnable textual prompt form. Our approach can be seamlessly integrated into existing textual-based methods, enhancing their performance without incurring additional computational costs.

3. Method

Prompt learning [19, 20, 63–65] aims to enhance the generalization ability of pre-trained VLMs like CLIP on downstream tasks by training inserted learnable soft tokens. Existing textual-based methods all follow the classic prompt paradigm, concatenating soft prompt tokens and hard class tokens as the input to the text encoder, as shown in Fig. 2(b). In this paper, we propose a simple and effective textual prompt learning method, named ATPrompt, which anchors multiple fixed universal attribute tokens into the original soft prompts, as shown in Fig. 2(d). Guided by these attributes, soft prompts can learn not only category-specific but also attribute-related general representations through training. When encountering unknown categories, these learned attribute-related tokens can provide additional information to promote better image-text alignment. Furthermore, to identify these universal attributes, we present an automated pipeline that encompasses sequential steps. Initially, we employ LLMs to synthesize the attribute pool for the current downstream category. Subsequently, we propose a differentiable attribute search method designed to identify attributes within the pool that are most suitable for our attribute-anchored prompt forms. For every task, this search operation is conducted *only once*. Once the attributes are finalized, they are integrated into our ATPrompt for specialized model fine-tuning.

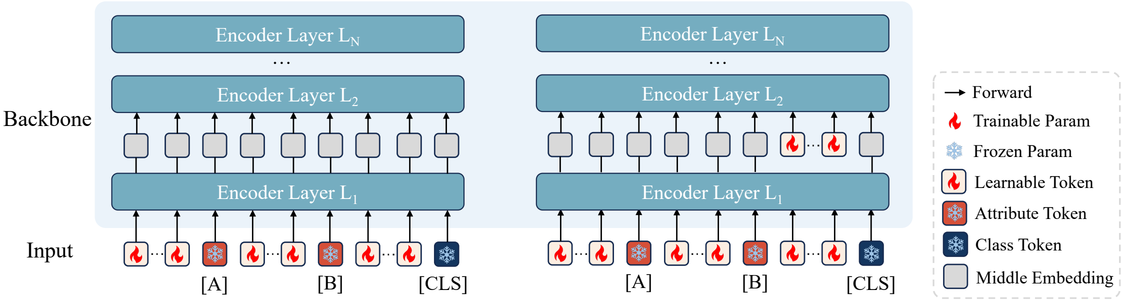
3.1. Preliminary

Vision-Language Models. Existing VLMs [16, 42], such as CLIP, have demonstrated remarkable zero-shot generalization performance after training with 400 million image-text pairs. The primary objective of these models is to learn the alignment between image and text modalities produced by each encoder. Given a labeled image classification dataset $\mathrm{D} = \{(x, c)\}$ which includes N class labels $C = \{c_i\}_{i=1}^{N}$, CLIP makes predictions by calculating the cosine similarity between image features and the text features of each class. Specifically, for each input image x, it undergoes feature extraction via the image encoder $h_I(x)$ and obtains a feature vector $u = h_I(x)$. Simultaneously, for each class, a series of textual descriptions t are generated using the hand-crafted template. Then, these text descriptions are fed into the text encoder $h_T(x)$ to obtain text features $w = h_T(t)$. Finally, the output probability for image x classified to c is calculated as follows:

$$p(c|x) = \frac{\exp(\cos(u, w_c)/\tau)}{\sum_{i=1}^{N} \exp(\cos(u, w_i)/\tau)}. \tag{1}$$

where τ is the temperature parameter and $\cos(\cdot, \cdot)$ denotes cosine similarity.

Prompt Learning for VLMs. Instead of manually designed hard prompts for image-text alignment, which is in-



(a) Shallow Version

(b) Deep Version

Figure 3. An illustration of the computation process for shallow and deep versions. Take two attributes [A] and [B] as examples. (a) The shallow version concatenates hard attribute tokens, soft prompt tokens, and class tokens and inputs them into the encoder for calculation. (b) The deep version uses the same input but discards the class-related soft prompt tokens after calculating the self-attention and introduces them again before the next layer. These two forms can be compatible with existing methods of varying prompt depths, including input-level ones like CoOp [64], CoCoOp [63] and depth-level ones like MaPLe [19].

accurate and inflexible, recent prompt learning works [19, 20, 58, 63, 64] like CoOp propose to learn appropriate soft textual prompts for downstream tasks. Concretely, M learnable soft tokens $[T_i]_{i=1}^{M}$ are concatenated with the hard class token [CLS] as the input of the text encoder, as shown in Fig. 2(b). Its form is shown as follows:

$$P_T = [T_1][T_2]...[T_M][\text{CLS}], \tag{2}$$

where M represents the length of soft tokens. For simplicity, we omit the prefix and suffix tokens in the input.

In addition to embedding soft tokens at the input level, existing studies [17, 19, 20, 30] have also explored introducing them at deeper layers. This is achieved by adding soft tokens within the Transformer blocks and subsequently removing them after the self-attention computations. For the i-th block, this process can be described as follows:

$$[\text{CLS}_i] = L_i([\text{T}_{i-1}, \text{CLS}_{i-1}]). \tag{3}$$

where L_i represents the i-th transformer block and T_i denotes the set of learnable soft tokens, defined as $\text{T}_i = \{[T_1]_i, ..., [T_M]_i\}$.

3.2. Learning soft prompts with universal attributes

Our approach introduces two variants, distinguished by the number of layers at which soft tokens are applied: a shallow version and a deep version, as shown in Fig. 3.

Shallow Version. We begin by introducing the shallow version, in which hard attribute tokens are anchored solely at the input level, as illustrated in Fig. 3(a). Consider two universal attributes, A and B. According to Eqn. (2), the shallow-level text prompt P_T provided to the text encoder can be expressed as follows:

$$P_T = [T_{a_1}]...[T_{a_m}][\text{A}][T_{b_1}]...[T_{b_m}][\text{B}][T_1]...[T_M][\text{CLS}]. \tag{4}$$

where a_m and b_m are hyperparameters specifying the length of soft tokens for attributes A and B. In our method, we set these parameters to be the same by default.

While this example places the class token at the end, we also evaluate configurations where it is positioned at the front or in the middle. As shown in Tab. 5, the end position yields the best performance and is therefore adopted as our default configuration.

Deep Version. In this version, learnable soft tokens are introduced at the input of the deep layers. Previous works, such as VPT and MaPLe, discard all soft tokens and subsequently reintroduce them after the block. When this operation is applied to attribute-related words, a gap will emerge between the introduced excessive low-level tokens and the existing high-level tokens, thereby weakening the feature continuity across layers. In this study, our approach selectively discards and then re-adds only class-related soft tokens in the input, specifically $[T_1], ..., [T_M]$, as shown in Fig. 3(b). Based on Eqn. (3), the deep version of ATPrompt can be rewritten as follows:

$$[\text{F}_1, \_, \text{CLS}_1] = L_1([\text{T}_{a_0}, \text{A}, \text{T}_{b_0}, \text{B}, \text{T}_0, \text{CLS}_0]), \tag{5}$$

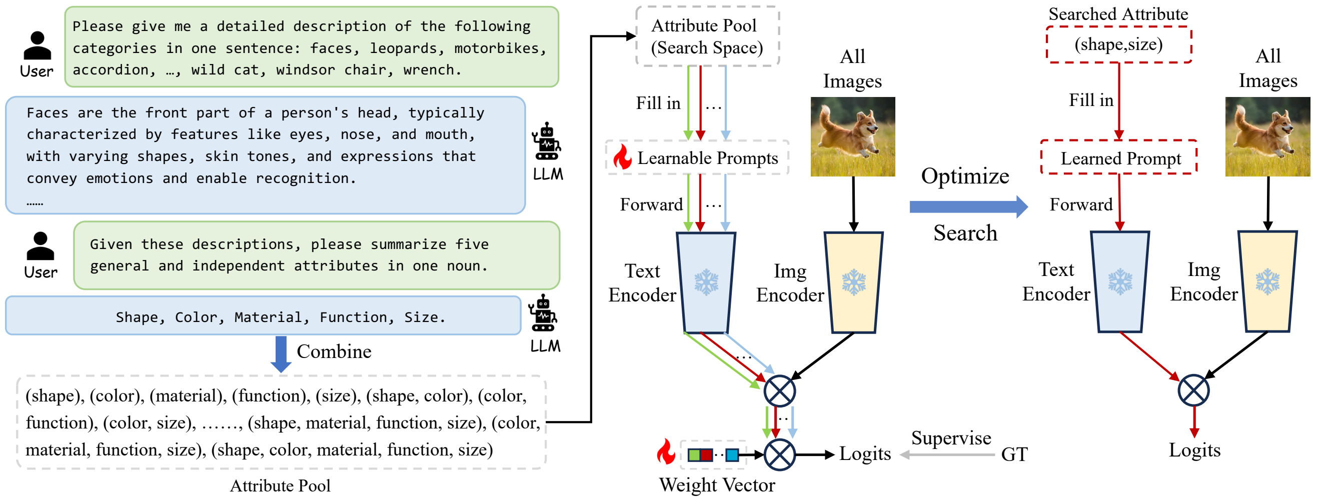
$$\begin{gathered}[\text{F}_i, \_, \text{CLS}_i] = L_i([\text{F}_{i-1}, \text{T}_{i-1}, \text{CLS}_{i-1}]). \\ i = 2, 3, ..., M.\end{gathered} \tag{6}$$

where F_i represents the features computed by the i-th Transformer layer. We demonstrate the effectiveness of this operation in Tab. 6.

Training. Let θ represent the weight of the total soft tokens and let v denote the selected fixed attribute tokens. Training is conducted on a labeled dataset $\text{D} = \{(x, c)\}$, with the objective of minimizing the cross-entropy loss between predicted values and ground truth labels. This process can be formulated as follows:

$$\min_{\theta} L_{train} = \min_{\theta} \sum_{x \in D} \text{CE}(f(x; v, \theta), c). \tag{7}$$

where $f(\cdot)$ represents the function of the CLIP model.



Please give me a detailed description of the following categories in one sentence: faces, leopards, motorbikes, accordion, …, wild cat, windsor chair, wrench.

Faces are the front part of a person's head, typically characterized by features like eyes, nose, and mouth, with varying shapes, skin tones, and expressions that convey emotions and enable recognition.
……

Given these descriptions, please summarize five general and independent attributes in one noun.

Shape, Color, Material, Function, Size.

(shape), (color), (material), (function), (size), (shape, color), (color, function), (color, size), ……, (shape, material, function, size), (color, material, function, size), (shape, color, material, function, size)

(a) Form an attribute pool.

(b) Differentiable attribute search.

Figure 4. An overview of our attribute search pipeline. (a) We first query the LLM iteratively to obtain multiple independent attributes. These are subsequently aggregated to form a pool of candidate combinations, which serves as the input for the search process. (b) The forward computation for each candidate combination is represented by a distinct colored path. To identify the optimal attributes, we employ an alternating optimization algorithm that co-optimizes the soft tokens and a corresponding path weight vector. Upon completion of training, the combination associated with the highest-weighted path is selected as the final output.

3.3. Attribute search

Selecting the attributes involves two key considerations: their content and their quantity. While directly querying a LLM is a straightforward approach, it has notable drawbacks. This method cannot determine the optimal number of attributes for a specific downstream dataset, and querying by category name alone can introduce semantic bias. To address these issues, we propose an automated pipeline that selects the appropriate content and quantity of attributes for the current downstream task, as illustrated in Fig. 4.

Attribute Pool. Inspired by CoT [52, 62], we divide the entire process into multiple steps to enhance the reasoning ability of LLMs. First, we prompt the LLM to generate descriptive sentences for each known category, thereby enriching category-related information. Using these descriptions as context, we then prompt the LLM to summarize a set of independent attribute bases that are common across these categories. An attribute pool is then formed by creating all possible combinations of these bases, as shown in Fig. 4(a). For N attribute bases, this result in a total of $w = C_N^1 + C_N^2 + ... + C_N^N$ candidates in the pool, which constitutes our search space. Note that we do not consider the order of attributes, as permutations generally do not introduce significant semantic bias or affect the final performance, a claim we validate in our experiments.

Attribute Searching. Inspired by DARTS [33], we introduce a differentiable attribute search method that learns to find representative attributes v from the search space $\mathcal{V}$, as shown in Fig. 4(b). To make the search space continuous, we relax the discrete attribute selection into a softmax-weighted sum over all w possible candidates:

$$f(x, v; \alpha, \theta) = \sum_{i \in \mathcal{V}} \frac{\exp(\alpha_i)}{\sum_{i' \in \mathcal{V}} \exp(\alpha_{i'})} f(x, v_i; \theta). \quad (8)$$

where α_i represents the weight for attribute combination v_i. The task of attribute search is thus reduced to learning the weight vector α for the candidate pool.

After relaxation, our goal is to jointly learn the attribute weight α and soft prompt tokens θ. Following standard practice [33, 41, 66], we optimize the weights α by minimizing the validation loss L_{val}, while the soft tokens are learned by minimizing the training loss L_{train}. We employ an alternating algorithm [15, 29] to solve this bi-level optimization problem, alternating between these two subpromblems:

$$\hat{\alpha} = \arg\min_{\alpha} L_{val}(f(x, v; \alpha, \hat{\theta}), c), \quad (9)$$

$$\hat{\theta} = \arg\min_{\theta} L_{train}(f(x, v; \hat{\alpha}, \theta), c). \quad (10)$$

where L_{train} and L_{val} both use the cross-entropy loss function. After the search, the attribute combination with the highest weight (α_i) is selected.

Cost Analysis. Unlike traditional Neural Architecture Search (NAS) methods [9, 32, 48], which search computationally expensive network-level parameters, our approach focuses on a lightweight token-level search space. This design makes our method significantly more efficient than previous approaches. In practice, our search converges in approximately 5 epochs, often requiring less than 5 minutes

Method	Average			ImageNet			Caltech101			OxfordPets		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
CoOp (IJCV 22)	82.69	63.22	71.66	76.47	67.88	71.92	98.00	89.81	93.73	93.67	95.29	94.47
CoCoOp (CVPR 22)	80.47	71.69	75.83	75.98	70.43	73.10	97.96	93.81	95.84	95.20	97.69	96.43
MaPLe (CVPR 23)	82.28	75.14	78.55	76.66	70.54	73.47	97.74	94.36	96.02	95.43	97.76	96.58
PromptSRC (ICCV 23)	84.26	76.10	79.97	77.60	70.73	74.01	98.10	94.03	96.02	95.33	97.30	96.30
ArGue (CVPR 24)	83.69	78.07	80.78	76.92	72.06	74.41	98.43	95.20	96.79	95.36	97.95	96.64
DePT (CVPR 24)	83.66	71.82	77.29	77.13	70.10	73.45	98.33	94.33	96.29	94.70	97.63	96.14
CoPrompt (ICLR 24)	84.00	77.23	80.48	77.67	71.27	74.33	98.27	94.90	96.55	95.67	98.10	96.87
PromptKD (CVPR 24)	86.96	80.73	83.73	80.83	74.66	77.62	98.91	96.65	97.77	96.30	98.01	97.15
CoOp + ATPrompt	82.68	68.04	**74.65** (+2.99)	76.27	70.60	**73.33**	97.95	93.63	**95.74**	94.77	96.59	**95.67**
CoCoOp + ATPrompt	81.69	74.54	**77.95** (+2.12)	76.43	70.50	**73.35**	97.96	95.27	**96.60**	95.46	97.89	**96.66**
MaPLe + ATPrompt	82.98	75.76	**79.21** (+0.66)	76.94	70.72	**73.70**	98.32	95.09	**96.68**	95.62	97.63	**96.61**
DePT + ATPrompt	83.80	73.75	**78.45** (+1.16)	77.32	70.65	**73.83**	98.48	94.60	**96.50**	94.65	97.99	**96.29**
PromptKD + ATPrompt	87.05	81.82	**84.35** (+0.62)	80.90	74.83	**77.75**	98.90	96.52	97.70	96.92	98.27	**97.59**

Method	StanfordCars			Flowers102			Food101			FGVCAircraft		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
CoOp (IJCV 22)	78.12	60.40	68.13	97.60	59.67	74.06	88.33	82.26	85.19	40.44	22.30	28.75
CoCoOp (CVPR 22)	70.49	73.59	72.01	94.87	71.75	81.71	90.70	91.29	90.99	33.41	23.71	27.74
MaPLe (CVPR 23)	72.94	74.00	73.47	95.92	72.46	82.56	90.71	92.05	91.38	37.44	35.61	36.50
PromptSRC (ICCV 23)	78.27	74.97	76.58	98.07	76.50	85.95	90.67	91.53	91.10	42.73	37.87	40.15
ArGue (CVPR 24)	75.64	73.38	74.49	98.34	75.41	85.36	92.33	91.96	92.14	40.46	38.03	39.21
DePT (CVPR 24)	79.67	72.40	75.86	98.20	72.00	83.08	90.43	91.33	90.88	42.53	22.53	29.46
CoPrompt (ICLR 24)	76.97	74.40	75.66	97.27	76.60	85.71	90.73	92.07	91.40	40.20	39.33	39.76
PromptKD (CVPR 24)	82.80	83.37	83.13	99.42	82.62	90.24	92.43	93.68	93.05	49.12	41.81	45.17
CoOp + ATPrompt	77.43	66.55	**71.58**	97.44	67.52	**79.77**	88.74	87.44	**88.09**	40.38	27.22	**32.52**
CoCoOp + ATPrompt	74.50	73.47	**73.98**	96.52	73.59	**83.51**	90.59	91.74	**91.16**	37.30	33.15	**35.10**
MaPLe + ATPrompt	75.39	73.84	**74.61**	97.82	75.07	**84.95**	90.65	92.00	91.32	37.61	36.15	**36.87**
DePT + ATPrompt	79.29	73.47	**76.27**	98.20	73.69	**84.20**	90.42	91.69	**91.05**	43.19	33.23	**37.56**
PromptKD + ATPrompt	82.51	84.03	**83.26**	99.15	82.03	89.78	92.48	93.86	**93.22**	49.63	42.35	**45.70**

Method	SUN397			DTD			EuroSAT			UCF101		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
CoOp (IJCV 22)	80.60	65.89	72.51	79.44	41.18	54.24	92.19	54.74	68.69	84.69	56.05	67.46
CoCoOp (CVPR 22)	79.74	76.86	78.27	77.01	56.00	64.85	87.49	60.04	71.21	82.33	73.45	77.64
MaPLe (CVPR 23)	80.82	78.70	79.75	80.36	59.18	68.16	94.07	73.23	82.35	83.00	78.66	80.77
PromptSRC (ICCV 23)	82.67	78.47	80.52	83.37	62.97	71.75	92.90	73.90	82.32	87.10	78.80	82.74
ArGue (CVPR 24)	81.52	80.74	81.13	81.60	66.55	73.31	94.43	88.24	91.23	85.56	79.29	82.31
DePT (CVPR 24)	82.37	75.07	78.55	83.20	56.13	67.04	88.27	66.27	75.70	85.43	72.17	78.24
CoPrompt (ICLR 24)	82.63	80.03	81.30	83.13	64.73	72.79	94.60	78.57	85.84	86.90	79.57	83.07
PromptKD (CVPR 24)	83.69	81.54	82.60	85.84	71.37	77.94	97.54	82.08	89.14	89.71	82.27	86.10
CoOp + ATPrompt	80.84	68.64	**74.24**	80.83	45.49	**58.22**	90.34	59.79	**71.96**	84.49	64.96	**73.45**
CoCoOp + ATPrompt	80.50	76.86	**78.64**	78.63	56.89	**66.02**	87.95	74.15	**80.46**	82.74	76.40	**79.44**
MaPLe + ATPrompt	80.98	78.15	79.54	80.50	58.28	67.61	94.84	77.59	**85.35**	84.08	78.88	**81.40**
DePT + ATPrompt	82.42	76.48	**79.34**	82.64	56.77	**67.30**	89.60	69.50	**78.28**	85.60	73.15	**78.89**
PromptKD + ATPrompt	83.87	81.35	82.59	86.92	72.34	**78.96**	97.05	92.07	**94.49**	89.29	82.44	85.73

Table 1. Base-to-novel generalization experiments of five baselines with and without our ATPrompt on 11 datasets. Our method achieves consistent average performance improvement over different baselines.

Method	Source	Target Dataset										
	ImageNet	Caltech 101	Oxford Pets	Stanford Cars	Flowers 102	Food101	FGVC Aircraft	SUN397	DTD	Euro SAT	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
+ATPrompt	71.67	93.96	90.65	65.01	70.40	85.86	20.97	65.77	43.44	46.59	69.92	**65.26** (+1.38)
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
+ATPrompt	71.27	93.79	90.62	65.90	71.17	86.03	23.22	66.63	44.44	48.70	70.71	**66.59** (+0.85)
MaPLe	70.72	93.53	90.49	65.57	72.23	86.20	24.74	67.01	46.49	48.06	68.69	66.30
+ATPrompt	70.69	94.04	91.03	66.06	71.99	86.33	24.42	67.05	45.21	48.63	69.15	**66.75** (+0.45)

Table 2. Cross-dataset generalization experiments of three baselines with and without our ATPrompt on 11 datasets. ATPrompt achieves consistent average performance improvement on target datasets.

Dataset	Attribute Bases	Searched Results
ImageNet	color, size, shape, habitat, behavior	(color, shape)
Caltech101	shape, color, material, function, size	(shape,size)
OxfordPets	loyalty, affection, energy, playfulness, intelligence	(playfulness, energy)
StanfordCars	design, engine, performance, luxury, color	(luxury)
Flowers102	color, flower, habitat, growth, season	(color, habitat, growth)
Food101	flavor, texture, origin, ingredients, preparation	(flavor, preparation)

Table 3. Part of the results obtained after differentiable attribute search. Please refer to the appendix for the complete results.

on a single A800 GPU for some datasets. In addition, search efficiency can be improved by curating a smaller set of base attributes, which narrows the search space.

4. Experiments

4.1. Settings

Base-to-Novel Generalization. Following [20, 30, 63, 64], we split the dataset into base and novel classes. The model is trained on the base class training set and evaluated on the test set. The attributes applied in this experiment are searched based on the base class. For datasets without a dedicated validation set, like ImageNet, we split the 16-shot labeled data, using half for training and the other half for attribute search.

Cross-dataset Experiments. Consistent with previous works [20, 63, 64], we first train a model on the ImageNet-1K source dataset and then evaluate its generalization performance on several out-of-distribution datasets. The attributes used are obtained from the source dataset.

Attribute Search. We select five independent attributes as the basis in the attribute pool. This results in 31 candidate attribute combinations for the search process. We use ChatGPT-4o for attribute queries. In Tab. 3 reports a subset of the queried attribute bases and the final combinations selected by our search.

Implementation Details. We evaluate the model performance on 15 datasets. We report base and novel class accuracy and their Harmonic Mean (HM) averaged over 3 runs. The details of each dataset are attached in the Appendix.

4.2. Base-to-Novel Generalization

As demonstrated in Tab. 1, we evaluate the base-to-novel generalization performance of five baseline methods, both with and without integrating ATPrompt, across 11 diverse recognition datasets. Notably, ATPrompt consistently improves the average performance of all baseline methods.

Reasons for Limited Improvement in Some Conditions. (1) Learnable text prompts are the central component in earlier works, and optimizing them through ATPrompt can obviously improve the performance. (2) Recent studies have expanded beyond learnable text prompts by introducing additional learnable modules. Since our work only involves the optimization of the learnable text prompt part, the improvement have become less obvious.

4.3. Cross-dataset Evaluation

Tab. 2 shows the cross-dataset generalization results for three baseline methods. Our method demonstrates superior performance, with improvements of 1.38%, 0.85% and 0.45% for CoOp, CoCoOp and MaPLe, respectively.

4.4. Domain Generalization

Tab. 4 shows the domain generalization results for three baseline methods. The results show that our method improves CoOp, CoCoOp and MaPLe methods by 0.90%, 0.49% and 0.33% respectively.

Method	Source	Target Dataset				Average
	ImageNet	-V2	-S	-A	-R	
CoOp	71.51	64.20	47.99	49.71	75.21	59.28
+ATPrompt	71.67	64.43	49.13	50.91	76.24	**60.18** (+0.90)
CoCoOp	71.02	64.07	48.75	50.63	76.18	59.91
+ATPrompt	71.27	64.66	49.15	51.44	76.33	**60.40** (+0.49)
MaPLe	70.72	64.07	49.15	50.90	76.98	60.27
+ATPrompt	70.69	64.40	49.10	51.77	77.11	**60.60** (+0.33)

Table 4. Domain generalization experiments of three baselines with and without our ATPrompt on four datasets. The integration of ATPrompt resulted in better generalization performance.

4.5. Further Analysis

By default, the experiments are conducted on the ImageNet. To minimize the influence of other components, we mainly adopt CoOp as the baseline method. Two attributes (color and shape) are used in our ATPrompt.

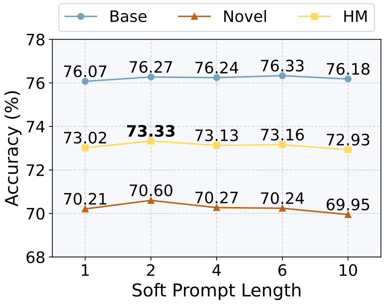


Figure 5. Illustration of varying soft token length on ImageNet. Increased tokens can lead to overfitting to the base class and weaken generalization to the novel class.

Soft Prompt Length. In Fig. 5, we examine the optimal soft token length for both attribute and class tokens. By varying the length from 1 to 10, we observe that longer prompts dilute the guiding influence of the attribute tokens, thereby reducing generalization to novel classes.

Class Token Position. The relative positioning of attribute tokens and class tokens in our method requires careful consideration. In Tab. 5, we examine configurations where the class token is positioned in the middle or on either side of two attribute tokens. The results demonstrate that optimal performance is achieved when the class token is placed at the end, which aligns with the findings of CoOp.

Position	Base	Novel	HM
Front	76.12	70.50	73.20
Middle	76.13	70.29	73.09
End	**76.27**	**70.60**	**73.33**

Table 5. Comparison of different class token positions on ImageNet. The end position works best.

Prompt Operation of Deep Version. In ATPrompt-Deep, we exclusively drop class soft tokens while retaining both hard and soft attribute tokens after they pass through the block. In this part, we compare the performance of partial drop (i.e., removing attribute soft tokens while retaining hard tokens) and full drop (i.e., removing both attribute soft and hard tokens) operations, as illustrated in Tab. 6.

Operation of Attribute Token	Base	Novel	HM
Retain all hard and soft tokens	**76.94**	**70.72**	**73.70**
Partial drop and re-add	76.87	70.44	73.51
Full drop and re-add	76.83	70.10	73.31

Table 6. Comparison of operations on deep soft and hard attribute tokens based on MaPLe+ATPrompt. Preserving hard and soft attribute tokens in deep layers performs better than other operations.

The results show that maintaining the attributes of both hard and soft tokens during the forward process results in optimal performance. Conversely, dropping and reintroducing either hard or soft attribute tokens harms performance, likely because it disrupts the continuity of attribute representations across layers and complicates optimization.

Attribute Order. In this study, we do not specifically focus on the order of attributes because varying the sequence usually does not result in semantic deviations in reality. Tab. 7 quantitatively assesses the impact of attribute order on prompt learning performance. From this table, we observe that despite variations in order, similar results are consistently produced, and the performance fluctuations across different orders remain within a reasonable range.

Attributes	Base	Novel	HM
(shape, color)	76.32	70.39	73.24
(color, shape)	76.27	70.60	**73.33**
(size, habitat)	76.44	70.23	73.20
(habitat, size)	76.46	70.16	73.14

Table 7. Comparison of different orders on ImageNet. The order of attributes does not significantly affect the model, and performance fluctuations are within a reasonable range.

Comparison to Other Attributes. In Tab. 8, we explore the effectiveness of attributes derived through alternative methods, specifically by manually selecting class-irrelevant and common attributes. It shows that manually selected irrelevant attributes exhibit comparable performance during training; however, they perform poorly when applied to new categories. This suggests that incorrect attribute tokens cause the soft tokens to develop biased representations, thereby diminishing their zero-shot generalization ability.

Type	Attributes	Base	Novel	HM
Common	(shape, size)	82.83	67.13	74.16
	(color, texture)	82.73	67.56	74.38
Irrelevant	(plane, engines)	82.81	66.22	73.59
	(football, sport)	82.77	67.14	74.14
Searched	-	82.68	**68.04**	**74.65**

Table 8. Comparison of average performance for various attribute configurations on 11 datasets. The attributes obtained by our method achieve the best performance.

5. Conclusion

In this work, we introduce ATPrompt, an attribute-anchored textual prompt learning method that uses universal attributes as a bridge to improve generalization from seen to unseen categories. Our approach expands the learning space of soft prompts from a one-dimensional, category-centric structure to a multi-dimensional attribute space by anchoring fixed attribute tokens with the prompt. To ensure the selection of optimal attributes, we propose an automated pipeline designed to identify the most suitable candidates for any given downstream task. ATPrompt is designed with both shallow and deep architectural variants, rendering it broadly compatible with existing prompt learning methods. Extensive experiments validate the effectiveness of our approach. We believe this work offers a new direction for research into the fundamental structure of learnable prompts in prompt learning area.

Limitations and future works. This work is a preliminary study of the basic prompt form, which has not been able to achieve comparable performance to regularization-based methods when working alone. Furthermore, the current selection of attribute anchors relies on manual experimentation, and automatically discovering the optimal anchor positions through a learning-based approach remains a promising direction for future research.

6. Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant Nos. 62361166670 and U24A20330. This research was also supported by the Young Scientists Fund of the National Natural Science Foundation of China (Grant No.62206134), the Fundamental Research Funds for the Central Universities 070-63253222 and the Tianjin Key Laboratory of Visual Computing and Intelligent Perception (VCIP). Computation is supported by the Supercomputing Center of Nankai University (NKSC).

References

[1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 1

[2] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *NeurIPS*, 35:25005–25017, 2022. 1

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, pages 446–461. Springer, 2014. 12

[4] Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023. 1

[5] Keyan Chen, Xiaolong Jiang, Yao Hu, Xu Tang, Yan Gao, Jianqi Chen, and Weidi Xie. Ovarnet: Towards open-vocabulary object attribute recognition. In *CVPR*, pages 23518–23527, 2023. 2

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014. 12

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 12

[8] Tong Ding, Wanhua Li, Zhongqi Miao, and Hanspeter Pfister. Tree of attributes prompt learning for vision-language models. *arXiv preprint arXiv:2410.11201*, 2024. 2

[9] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *JMLR*, 20(55):1–21, 2019. 5

[10] 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 *CVPR workshop*, pages 178–178. IEEE, 2004. 12

[11] 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. 12

[12] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pages 8340–8349, 2021. 12

[13] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pages 15262–15271, 2021. 12

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[15] Yan Huang, Shang Li, Liang Wang, Tieniu Tan, et al. Unfolding the alternating optimization for blind super resolution. *NeurIPS*, 33:5632–5643, 2020. 5

[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pages 4904–4916. PMLR, 2021. 1, 2, 3

[17] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, pages 709–727. Springer, 2022. 1, 4

[18] Baoshuo Kan, Teng Wang, Wenpeng Lu, Xiantong Zhen, Weili Guan, and Feng Zheng. Knowledge-aware prompt tuning for generalizable vision-language models. In *ICCV*, pages 15670–15680, 2023. 2

[19] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *CVPR*, pages 19113–19122, 2023. 1, 2, 3, 4, 12

[20] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *ICCV*, pages 15190–15200, 2023. 1, 2, 3, 4, 7

[21] Gahyeon Kim, Sohee Kim, and Seokju Lee. Aapl: Adding attributes to prompt learning for vision-language models. In *CVPR Workshop*, pages 1572–1582, 2024. 2

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshop*, pages 554–561, 2013. 12

[23] Nilakshan Kunananthaseelan, Jing Zhang, and Mehrtash Harandi. Lavip: Language-grounded visual prompting. In *AAAI*, pages 2840–2848, 2024. 1

[24] Dongjun Lee, Seokwon Song, Jihee Suh, Joonmyeong Choi, Sanghyeok Lee, and Hyunwoo J Kim. Read-only prompt optimization for vision-language few-shot learning. In *ICCV*, pages 1401–1411, 2023. 2

[25] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 1, 2

[26] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 1, 2

[27] Yunheng Li, Yuxuan Li, Quansheng Zeng, Wenhai Wang, Qibin Hou, and Ming-Ming Cheng. Unbiased region-language alignment for open-vocabulary dense prediction. *arXiv preprint arXiv:2412.06244*, 2024. 1

[28] Yunheng Li, Zhong-Yu Li, Quan-Sheng Zeng, Qibin Hou, and Ming-Ming Cheng. Cascade-CLIP: Cascaded vision-language embeddings alignment for zero-shot semantic segmentation. In *ICML*, pages 28243–28258. PMLR, 2024. 1

[29] Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. Curriculum temperature for knowledge distillation. In *AAAI*, pages 1504–1512, 2023. 2, 5

[30] Zheng Li, Xiang Li, Xinyi Fu, Xin Zhang, Weiqiang Wang, Shuo Chen, and Jian Yang. Promptkd: Unsupervised prompt distillation for vision-language models. In *CVPR*, pages 26617–26626, 2024. 1, 2, 4, 7, 12

[31] Zheng Li, Xiang Li, Lingfeng Yang, Renjie Song, Jian Yang, and Zhigeng Pan. Dual teachers for self-knowledge distillation. *Pattern Recognition*, 151:110422, 2024. 2

[32] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *ECCV*, pages 19–34, 2018. 5

[33] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 5, 12

[34] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*, 32, 2019. 1

[35] Hao Ma, Ming Li, Jingyuan Yang, Or Patashnik, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Clip-flow: Decoding images encoded in clip space. *CVMJ*, 10(6):1157–1168, 2024. 1

[36] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 12

[37] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*, 2022. 2

[38] 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. 12

[39] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505. IEEE, 2012. 12

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037, 2019. 12

[41] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *ICML*, pages 4095–4104. PMLR, 2018. 5

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

[43] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, pages 5389–5400. PMLR, 2019. 12

[44] Shuvendu Roy and Ali Etemad. Consistency-guided prompt learning for vision-language models. *arXiv preprint arXiv:2306.01195*, 2023. 2

[45] 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. 12

[46] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-clip: A clip model focusing on wherever you want. In *CVPR*, pages 13019–13029, 2024. 1

[47] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 1

[48] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, pages 2820–2828, 2019. 5

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

[50] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *NeurIPS*, 32, 2019. 12

[51] Yubin Wang, Xinyang Jiang, De Cheng, Dongsheng Li, and Cairong Zhao. Learning hierarchical prompt with structured linguistic knowledge for vision-language models. In *AAAI*, pages 5749–5757, 2024. 2

[52] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35:24824–24837, 2022. 5

[53] Ge Wu, Xin Zhang, Zheng Li, Zhaowei Chen, Jiajun Liang, Jian Yang, and Xiang Li. Cascade prompt learning for vision-language model adaptation. In *ECCV*, 2024. 1

[54] Shihan Wu, Ji Zhang, Pengpeng Zeng, Lianli Gao, Jingkuan Song, and Heng Tao Shen. Skip tuning: Pre-trained vision-language models are effective and efficient adapters themselves. In *CVPR*, pages 14723–14732, 2025. 2

[55] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492. IEEE, 2010. 12

[56] Chuanguang Yang, Zhulin An, Libo Huang, Junyu Bi, Xinqiang Yu, Han Yang, and Yongjun Xu. Clip-kd: An empirical study of distilling clip models. *arXiv preprint arXiv:2307.12732*, 2023. 2

[57] Chuanguang Yang, Zhulin An, Helong Zhou, Fuzhen Zhuang, Yongjun Xu, and Qian Zhang. Online knowledge distillation via mutual contrastive learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10212–10227, 2023. 2

[58] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *CVPR*, pages 6757–6767, 2023. 2, 4

[59] Yajing Zhai, Yawen Zeng, Zhiyong Huang, Zheng Qin, Xin Jin, and Da Cao. Multi-prompts learning with cross-modal alignment for attribute-based person re-identification. In *AAAI*, pages 6979–6987, 2024. 2

[60] Ji Zhang, Shihan Wu, Lianli Gao, Heng Tao Shen, and Jingkuan Song. Dept: Decoupled prompt tuning. In *CVPR*, pages 12924–12933, 2024. 1, 12

[61] Xiaoqin Zhang, Zhenni Yu, Li Zhao, Deng-Ping Fan, and Guobao Xiao. Comprompter: reconceptualized segment anything model with multiprompt network for camouflaged object detection. *Science China Information Sciences*, 68(1): 112104, 2025. 2

[62] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022. 5

[63] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825, 2022. 1, 2, 3, 4, 7, 12

[64] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. 1, 2, 3, 4, 7, 12

[65] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *ICCV*, pages 15659–15669, 2023. 2, 3

[66] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *CVPR*, pages 8697–8710, 2018. 5

Advancing Textual Prompt Learning with Anchored Attributes

Supplementary Material

S1. Implementation Details

S1.1. Dataset

We evaluate the performance of our method on 15 recognition datasets. For generalization from base-to-novel classes and cross-dataset evaluation, we evaluate the performance of our method on 11 diverse recognition datasets. Specifically, these datasets include ImageNet-1K [7] and Caltech-101 [10] for generic object classification; OxfordPets [39], StanfordCars [22], Flowers-102 [38], Food-101 [3], and FGVCAircraft [36] for fine-grained classification, SUN-397 [55] for scene recognition, UCF-101 [45] for action recognition, DTD [6] for texture classification, and EuroSAT [11] for satellite imagery recognition. For domain generalization experiments, we use ImageNet-1K [7] as the source dataset and its four variants as target datasets including ImageNet-V2 [43], ImageNet-Sketch [50], ImageNet-A [13], and ImageNet-R [12].

S1.2. Attribute Search

Inspired by DARTS [33], we employ a differentiable search method to identify the optimal content and quantity of attributes for our proposed attribute-anchored form. The search process is conducted for 10 epochs with a batch size of 32. We use SGD to optimize the soft prompts θ with an initial learning rate of 0.002. and Adam to optimize the weight vector α with an initial learning rate of 0.02. In our experiments, we use 5 attribute bases, which generate 31 (i.e., $C_5^1 + C_5^2 + C_5^3 + C_5^4 + C_5^5$) candidate combinations for the search process.

Tab. S4 presents the five attribute bases generated by the LLM, alongside the optimal attribute combination identified after the search. Furthermore, Tab. S5 displays the final weights of all candidate combinations from the search stage on the Caltech-101 dataset.

S1.3. Base-to-Novel Generalization

Baseline Methods. To evaluate ATPrompt, we integrate it with several leading textual-based prompt learning approaches, including CoOp [64], CoCoOp [63], MaPLe [19], DePT [60] and PromptKD [30]. The experimental settings are detailed below.

Settings. Our framework is implemented in PyTorch [40] and all experiments were conducted on a single NVIDIA A800 GPU. Following the baseline methods, we use a standard data augmentation scheme of random resized cropping and flipping. We employe Stochastic Gradient Descent (SGD) as the optimizer. By default, the soft token lengths for attribute and class tokens are set to be identical, as attribute and class tokens are considered equally important. The specific implementation details for each baseline method are presented as follows:

CoOp+ATPrompt: Following the baseline, we use a batch size of 32 and an initial learning rate of 0.002. The original paper reports a learnable prompt length of $M = 16$ for ResNet-50 but does not specify a length for ViT-B/16. In our setup, we set the sofo token length for both the attribute and class tokens to 2. While the baseline model is trained for 200 epochs, we reduce the training to 100 epochs while maintaining the same cosine decay schedule. Figure S1 illustrates the architectural differences between the original CoOp and CoOp+ATPrompt.

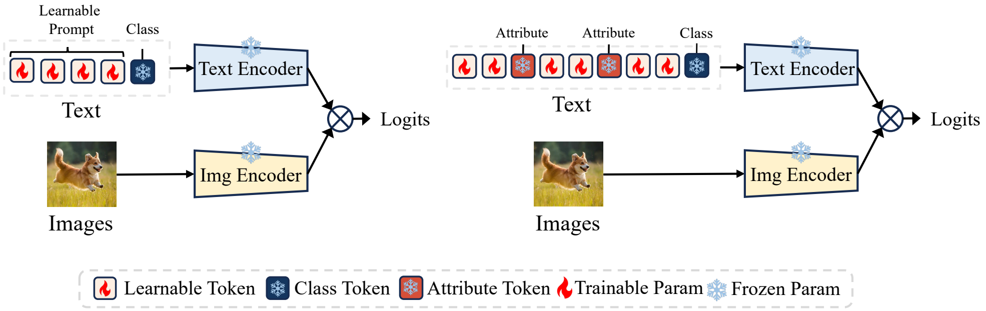
CoCoOp+ATPrompt: We adhere to the baseline's settings with a batch size of 1 and an initial learning rate of 0.002. Whereas the original paper specifies a soft class token length of 4, we set the length of our learnable tokens for the attribute and class token to 2. We adopt the same training schedule as the baseline: 10 epochs with cosine decay.

CoCoOp's original design uses a meta-network to generate offsets for all soft prompt tokens. We retain this meta-network but modify its application: the meta tokens now serve as offsets exclusively for the class soft tokens, $[T_1]$, ..., $[T_M]$, as shown in Fig. S2.

MaPLe+ATPrompt: We adhere to the baseline hyperparameters, utilizing a batch size of 4 and an initial learning rate of 0.0035. We diverge from the original prompt configuration; whereas the baseline sets the learnable prompt length to 2, our method sets the soft token lengths of both the attribute and the class token to 4. The training schedule remains consistent with the baseline.

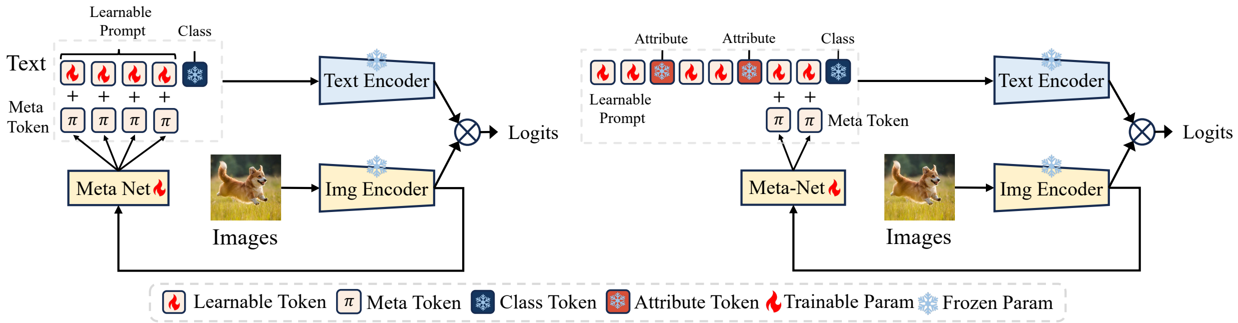
The primary architectural modification in MaPLe + ATPrompt concerns the projection mechanism. The original MaPLe framework inputs all textual soft tokens into a projection layer to generate corresponding visual tokens, which are then fused into the image encoder. Our approach, however, selectively inputs only the class soft tokens into this projection layer, while the attribute tokens are preserved without modification. This architectural difference is visualized in Fig. S3.

DePT+ATPrompt: We adopt the baseline's training configuration: a batch size of 32, a learning rate of 0.0035, a balance weight of λ=0.7, and a duration of 10 epochs. Our primary configuration for DePT+ATPrompt uses a learnable token length of 4. For datasets with lower complexity, namely Caltech-101, OxfordPets, and StanfordCars, we adjust these parameters, setting the soft token length to 2 and the balance weight to 0.6. The architectural differences between the DePT and DePT+ATPrompt models are detailed



(a) CoOp (b) CoOp + ATPrompt

Figure S1. Architectural comparison between CoOp and CoOp+ATPrompt.



(a) CoCoOp (b) CoCoOp + ATPrompt

Figure S2. Architectural comparison between CoCoOp and CoCoOp+ATPrompt. In CoCoOp+ATPrompt, meta tokens are only added as offsets to class soft tokens.

in Fig. S4.

S2. Additional Experiments

S2.1. Ablation Study

Attribute Order. In the main paper, our experiments confirm that the order of attributes does not significantly impact model performance, with results fluctuating within an acceptable range. Here we provide additional experiments in Tab. S1 to support this observation.

Attribute Position. We also investigated the impact of attribute token positioning within the prompt. Fig. S5 visualizes the positions tested, and Tab. S2 presents the results. Our findings show that the "interval" configuration, where attributes are placed between class tokens, yields the best performance.

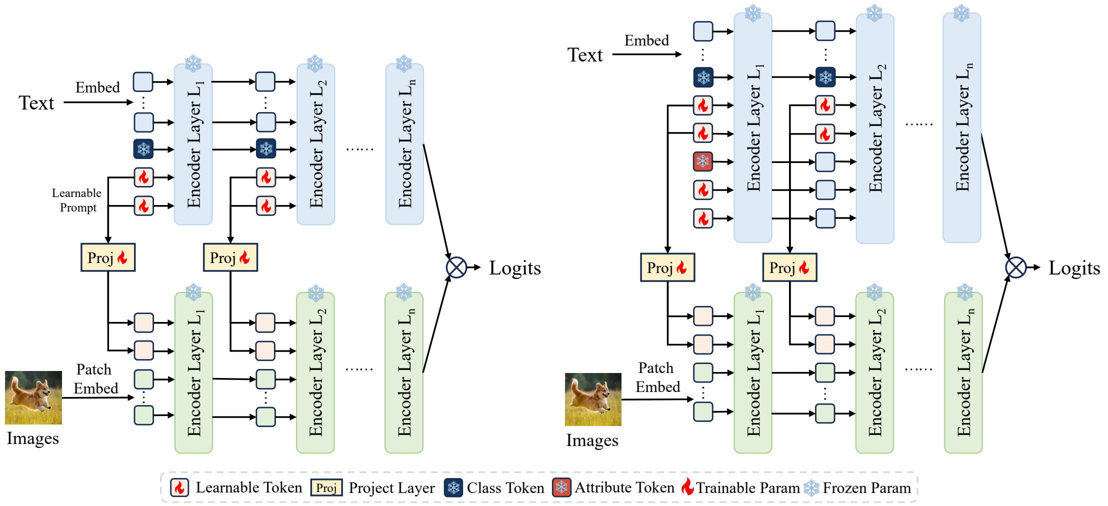
Initialization. Baseline methods typically initialize soft tokens using the embeddings of the phrase "a photo of a." The inclusion of attribute tokens makes this strategy suboptimal for our method. We instead initialize class soft tokens ($[T_1]$, ..., $[T_M]$) by sampling from a Gaussian distribution with a mean of 0 and a standard deviation of 0.02. As shown in Table S3, this random initialization provides a superior starting point for training.

Attributes	Base	Novel	HM
(shape, color)	76.32	70.39	73.24
(color, shape)	76.27	70.60	**73.33**
(size, habitat)	76.44	70.23	73.20
(habitat, size)	76.46	70.16	73.14
(material, function)	76.40	70.13	73.13
(function, material)	76.28	70.00	73.01
(growth, season)	76.46	70.18	73.19
(season, growth)	76.40	70.21	73.17
(color, size, shape)	76.27	69.95	72.97
(shape, size, color)	76.32	70.19	73.13
(habitat, size, shape)	76.50	70.21	73.22
(habitat, shape, size)	76.46	70.08	73.13
Searched Attributes (color, shape)	76.27	70.60	**73.33**

Table S1. Comparison of different attribute orders on ImageNet. Changes in attribute order will not significantly affect model performance.

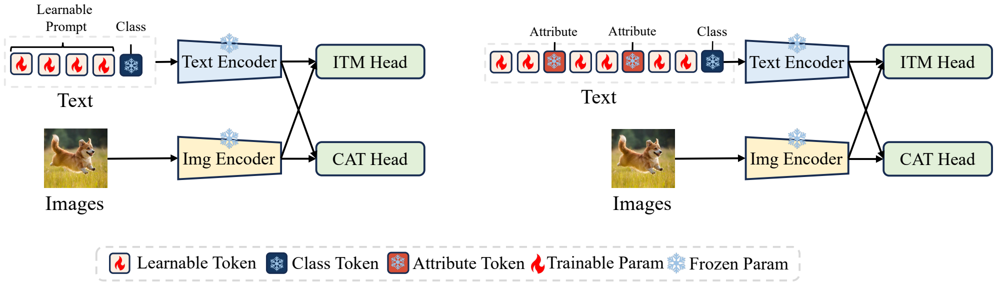
S3. Discussion

Comparison with Direct LLM Queries. Directly querying an LLM for universal attributes presents two challenges:



(a) MaPLe (b) MaPLe + ATPrompt

Figure S3. Architectural comparison between MaPLe and MaPLe+ATPrompt.



(a) DePT (b) DePT + ATPrompt

Figure S4. Architectural comparison between DePT and DePT+ATPrompt.

Version	Base	Novel	HM
Baseline (CoOp)	76.47	67.88	71.92
(a) Interval (Ours)	76.27	70.60	73.33
(b) Adjacent-front	76.39	70.22	73.18
(c) Adjacent-middle	76.46	70.11	73.15
(d) Adjacent-end	76.34	70.31	73.20
(e) Separate	76.48	70.08	73.14

Table S2. Performance results of attribute tokens at different positions in ATPrompt on ImageNet. The interval version achieves best results.

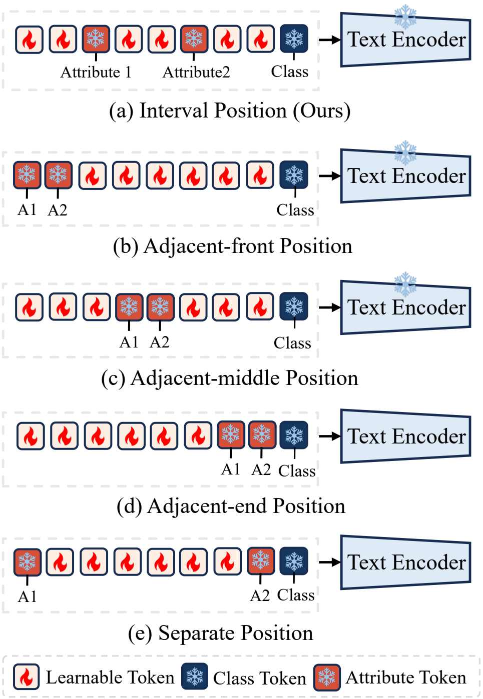
Attribute	Base	Novel	HM
“a photo of a”	76.40	70.07	73.10
Random Normal Init	76.27	70.60	73.33

Table S3. Comparison of different initialization ways on ImageNet. Random normal initialization performs better.

determining the optimal attribute content and identifying the ideal number of attributes. Our experiments suggest that two attributes are often optimal. Therefore, users can bypass our search process by directly prompting the LLM to summarize two universal attributes. This offers a simpler approach, though it may result in a slight performance trade-off.

Why do attributes searched on a source dataset (ImageNet) generalize well? The attributes identified on ImageNet (e.g., color, shape) are fundamental properties of natural objects. Representations learned under the guidance of these universal attributes are therefore inherently generalizable and transfer effectively to other datasets and classes.

Why does ATPrompt not outperform regularization-based methods in isolation? ATPrompt is a plug-in module designed to optimize the prompt’s structure. In contrast, regularization-based methods are often comprehensive frameworks that employ multiple components (e.g., learnable visual prompts, MLPs) simultaneously. While ATPrompt may not outperform these multi-faceted approaches on its own, its strength lies in its ability to be integrated into other methods, consistently improving their



(a) Interval Position (Ours)

(b) Adjacent-front Position

(c) Adjacent-middle Position

(d) Adjacent-end Position

(e) Separate Position

Figure S5. Comparison of attribute tokens at different positions, taking two attributes as an example.

performance beyond previous baselines.

S4. Limitations and Future Works.

Beyond the limitations discussed in the main paper, we identify the following directions for future research: (1) While our differentiable search method is efficient, we aim to further enhance the attribute discovery process. A promising direction is to leverage Multimodal Large Language Models (MLLMs), potentially using techniques like Chain-of-Thought (CoT), to better automate the selection of optimal attribute content and quantity. (2) Our current approach embeds fixed, explicit attributes into the prompt. In the future, we plan to explore a transition to implicit, learnable attributes. This would enable the model to discover optimal attributes in a data-driven manner during training, potentially unlocking further performance gains.

Dataset	Attribute Bases	Searched Attributes
ImageNet-1K	color, size, shape, habitat, behavior	(color, shape)
Caltech-101	shape, color, material, function, size	(shape,size)
Oxford Pets	loyalty, affection, playfulness, energy, intelligence	(playfulness, energy)
Stanford Cars	design, engine, performance, luxury, color	(luxury)
Flowers-102	color, flower, habitat, growth, season	(color, habitat, growth)
Food-101	flavor, texture, origin, ingredients, preparation	(flavor, preparation)
FGVC Aircraft	design, capacity, range, engines, liveries	(design, range)
SUN-397	architecture, environment, structure, design, function	(function)
DTD	pattern, texture, color, design, structure	(pattern, color, design)
EuroSAT	habitat, foliage, infrastructure, terrain, watercourse	(habitat)
UCF-101	precision, coordination, technique, strength, control	(precision)

Table S4. Attribute bases and searched results for each dataset.

Attribute Bases	shape, color, material, function, size
Attribute Combinations & Corresponding Weights	(shape), weight: 0.298
	(color), weight: 0.004
	(material), weight: 0.002
	(function), weight: 0.002
	(size), weight: 0.003
	(shape, color), weight: 0.003
	(shape, material), weight: 0.006
	(shape, function), weight: 0.000
	(shape, size), weight: 0.565
	(color, material), weight: 0.000
	(color, function), weight: 0.001
	(color, size), weight: 0.005
	(material, function), weight: 0.000
	(material, size), weight: 0.002
	(function, size), weight: 0.002
	(shape, color, material), weight: 0.002
	(shape, color, function), weight: 0.002
	(shape, color, size), weight: 0.000
	(shape, material, function), weight: 0.001
	(shape, material, size), weight: 0.085
	(shape, function, size), weight: 0.001
	(color, material, function), weight: 0.001
	(color, material, size), weight: 0.000
	(color, function, size), weight: 0.002
	(material, function, size), weight: 0.001
	(shape, color, material, function), weight: 0.001
	(shape, color, material, size), weight: 0.001
	(shape, color, function, size), weight: 0.001
	(shape, material, function, size), weight: 0.005
	(color, material, function, size), weight: 0.001
	(shape, color, material, function, size), weight: 0.001

Table S5. Output results after 40 epochs of attribute searching on the Caltech101 dataset.