Spotlighter: Revisiting Prompt Tuning from a Representative Mining View

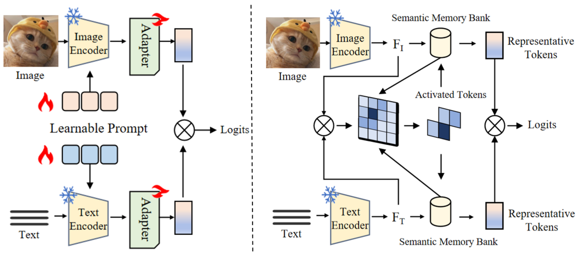
Anonymous EMNLP submission

Abstract

CLIP's success has demonstrated that prompt tuning can achieve robust cross-modal semantic alignment for tasks ranging from open-domain recognition to fine-grained classification. However, redundant or weakly relevant feature components introduce noise and incur unnecessary computational costs. In this work, we propose Spotlighter, a lightweight token-selection framework that simultaneously enhances accuracy and efficiency in prompt tuning. Spotlighter evaluates each visual token's activation from both sample-wise and semantic-wise perspectives and retains only the top-scoring tokens for downstream prediction. A class-specific semantic memory bank of learned prototypes refines this selection, ensuring semantic representativeness and compensating for discarded features. To further prioritize informative signals, we introduce a two-level ranking mechanism that dynamically weights token–prototype interactions. Across 11 few-shot benchmarks, Spotlighter outperforms CLIP by up to 11.19% in harmonic mean accuracy and achieves up to 0.8K additional FPS, with only 21 extra parameters. These results establish Spotlighter as an effective and scalable baseline for prompt tuning. Our code will be available.

1 Introduction

Recent advances in vision-language models have demonstrated remarkable capabilities in prompt tuning, particularly through approaches like CLIP (Radford et al., 2021) that achieve robust cross-modal semantic alignment. These methods have demonstrated impressive performance in tasks such as open-domain recognition (Cheng et al., 2024), fine-grained categorization (Zhu et al., 2022), and long-tailed distribution scenarios (Liu et al., 2022), leading to breakthroughs in practical applications, including intelligent surveillance and medical image analysis. The superior performance of vision-language models primarily originates from their ability to learn discriminative joint embeddings that enable precise cross-modal alignment, a fundamental driver of continual model enhancement.



(a) existing method (b) our method

Figure 1: Comparison with other methods. (a) Learnable prompts or adapters are applied to learn multimodal complex semantic information. (b) Activated and Representative tokens improve inference efficiency by mitigating noise and redundant features.

The alignment between visual and textual feature spaces enables effective classification, with ongoing research continuously enhancing representation quality through techniques like prompt learning (Zhu et al., 2023; Xu et al., 2025) and feature enhancement (Sun et al., 2023; Choi et al., 2025). However, existing methods face two primary challenges: (1) Feature noise interference: Redundant or weakly relevant components within the aligned features introduce noise, undermining the contribution of semantically critical information (Zhu et al., 2025). (2) Computational efficiency bottleneck: Full-scale feature interactions across the entire representation space result in unnecessary computational burdens and higher costs in practical applications (khattak et al., 2023).

To address these challenges, prior works (Huang et al., 2023; Yang et al., 2025) have proved that during CLIP's encoding process for effective image-text alignment, the model inherently captures a mixture of semantic signals. Since the image and text encoders operate independently, they are designed to cover a wide range of possible semantics. This results in varied importance across different parts of the feature representation concerning specific classification goals. Crucially, only a subset of these features contributes meaningfully to cross-modal alignment, while the rest may introduce redundancy. Therefore, performing the sample-level evaluation of feature importance enables us to selectively emphasize critical features and suppress irrelevant ones, enhancing accuracy and efficiency. Existing approaches (khattak et al., 2023; Li et al., 2024a; Khattak et al., 2023a) employ global learnable text-image prompts or lightweight adapters in frozen layers to capture semantic information, as shown in Fig.1(a), yet have not thoroughly explored the synergistic optimization between features representations and computational efficiency, leaving this as an open area for further research.

Based on the above analysis, we revisit cross-modal feature alignment in few-shot image classification and propose a simple yet effective model, Spotlighter, which achieves a favorable balance between accuracy and computational efficiency. The key idea is to identify and retain sparse but highly representative feature tokens while discarding redundant ones. Specifically, we evaluate each token's cross-modal semantic relevance from both sample-wise and semantic-wise perspectives, quantified as an activation score. Only a few highly activated tokens are retained for prediction, while the rest are discarded as redundant. To guide this selection, we introduce a semantic memory bank that stores a set of class-specific semantic prototypes. These prototypes help refine class boundaries during token activation, ensuring that the selected features are both semantically representative and capable of compensating for potentially missing information from discarded regions. Furthermore, recognizing the varying contributions of activated features to classification, we introduce a two-level ranking mechanism over the prototypes. This mechanism dynamically adapts to the activation distribution of each sample, allowing the model to prioritize more informative features. The final representative features for prediction are then formed by fusing features with their corresponding prototypes according to their activation levels.

Extensive experiments conducted across 11 benchmark datasets demonstrate the effectiveness of our proposed method. Compared to CLIP (Radford et al., 2021) and CLIPFit (Li et al., 2024a), our approach achieves consistent improvements in both harmonic mean accuracy (HM) and computational speed, with an improvement of 11.19% / 3.86% in HM score and 0.8K/3.8K more FPS, respectively. Remarkably, these gains come at the cost of only **21** additional parameters, highlighting the efficiency and scalability of our design.

Our main contributions are lies in:

- We investigate the role of representative feature mining in prompt tuning, highlighting its dual benefits in improving both prediction accuracy and computational efficiency.
- We propose Spotlighter, which selects the most activated tokens and enhance them via a semantic memory bank to form a compact yet informative representative feature set.
- With only 21 additional parameters, our method boosts accuracy by 11.19% and inference speed by 0.8K FPS over CLIP, establishing a strong, scalable baseline for prompt tuning.

2 Related Works

2.1 Pre-trained Vision-Language Models

Large Language Models (LLMs) like GPT-3 (Brown et al., 2020), GPT-4 (Achiam et al., 2023), LLaMA (Touvron et al., 2023), and Deepseek (Lu et al., 2024) exhibit robust zero-shot transfer capabilities in NLP tasks. Nowadays,modern vision-language models (VLMs), enhanced by natural language supervision, excel in zero-shot/few-shot learning through large-scale image-text pretraining, as seen in contrastive learning-based models like ALIGN (Li et al., 2021) and CLIP (Radford et al., 2021). Leveraging their formidable language-aligned visual representations and strong generalization, these models excel in diverse downstream tasks, such as object detection (Zhang et al., 2022; Gu et al., 2021) and semantic segmentation (Zhou et al., 2023; Li et al., 2024b). However, VLMs face significant challenges in degrading critical semantic information due to the redundant or weakly relevant components within the aligned features (Zhu et al., 2023; Khattak et al., 2023b). Spotlighter enhances semantics and boosts efficiency through the hierarchical removal of useless components.

2.2 Prompt Tuning

Prompt learning adapts pre-trained models to downstream few-shot tasks via prompt-based reformulation, mitigating domain gaps and leveraging prior knowledge. Early approaches (Zhou et al., 2022a,b; Yao et al., 2023) relied on manually crafting templates based on prior human knowledge. Later, MaPLe (khattak et al., 2023), PromptSRC (Khattak et al., 2023a) concentrate on aligning visual-textual prompts jointly while adapter-based approaches (Zhang et al., 2022; Farina et al., 2025; Lu et al., 2025; Kim et al., 2024; Li et al., 2024a) extend via context-aware prompt tuning using lightweight adapters in transformer layers. Despite their success, these models often optimize prompts at coarse granularity, missing subtle visual cues and limiting cross-category generalization. To solve this problem, ArGue (Tian et al., 2024), LLaMP (Chiang et al., 2024), Textrefiner (Xie et al., 2024) and SAR (Jung and Lee, 2025) fill semantic gaps caused by noise through external LLMs or internal knowledge injection. However, these methods require substantial memory consumption. We enhance semantic information through multi-level feature tokenization while reducing large-scale feature interactions.

3 Method

3.1 Overview

Vision-Language Models (VLMs), such as CLIP, leverage aligned image-text representations learned in a shared embedding space, offering advantages in few-shot image classification tasks. Building on prior work, we adopt CLIP as our foundational model, with a key overview below. CLIP consists of an image encoder, labeled as E_I, and a text encoder referred to as E_T. Let $D = \{(\boldsymbol{x}_i, \boldsymbol{t}_i)\}_{i=1}^{b}$ represents the sampled batch, where $\boldsymbol{x}_i$ denotes the image input, $\boldsymbol{t}_i$ denotes the associated caption and b is the batch size. Both encoders employ a feature extraction backbone followed by a projection layer that maps multi-modal inputs to a unified embedding space. The image encoder encodes image $\boldsymbol{x}_i$ into $\boldsymbol{F}_I$, and text $\boldsymbol{t}_i$ into $\boldsymbol{F}_T$, *i.e.*,

$$\boldsymbol{F}_I = E_{\mathrm{I}}(\boldsymbol{x}_i), \quad \boldsymbol{F}_T = E_{\mathrm{T}}(\boldsymbol{t}_i). \tag{1}$$

i During the training phase, a contrastive loss is employed to maximize the cosine similarity between them for alignment. When testing, after getting the image feature $\boldsymbol{F}_I$ for image $\boldsymbol{x}_i$, the class $\boldsymbol{c}$ it belongs to is calculated by:

$$p(c) = \frac{\exp(\cos(\boldsymbol{T}_c, \boldsymbol{F}_I)/\tau)}{\sum_{j=1}^{K} \exp(\cos(\boldsymbol{T}_j, \boldsymbol{F}_I)/\tau)}, \tag{2}$$

where τ is a temperature parameter for scaling the softmax function, $\boldsymbol{T}_j$ is text embedding of class $\boldsymbol{j}$ and $\cos(\cdot,\cdot)$ denotes the cosine similarity function. It is worth noting that CLIP aligns images and text by encoding them separately, but many features are noisy or redundant, thus extracting only the most relevant cross-modal features is necessary.

3.2 Spotlighter

To address the aforementioned challenges, we propose a plug-and-play method that selects a compact set of highly representative tokens. This strategy aims to suppress noise from redundant features and mitigate the computational overhead in the representative mining process. Our method, Spotlighter, identifies activated tokens by leveraging a well-established paradigm from classical computer vision: intermediate-layer activations in visual networks naturally encode semantically salient and fine-grained visual concepts localized in specific image regions (Zeiler and Fergus, 2014; Selvaraju et al., 2017; Kim et al., 2022). To enhance this capability, we introduce a Semantic Memory Bank, which facilitates the selection of representative tokens enriched with deeper semantic information. By integrating feature activation with representative token extraction, Spotlighter captures rich semantics in a compact representation. An overview of the proposed framework is illustrated in Figure 2, and will be discussed below in detail.

Feature Activation. To distill the most representative features across visual and textual modalities, we first evaluate each token's activation level in cross-modal semantic alignment for a given sample. These activation scores reflect the information distribution critical for prediction. To obtain reliable activation scores, we compute them at both the sample and semantic levels. The sample-level score reveals cross-modal alignment between image-text pairs (Selvaraju et al., 2017; Wang et al., 2020), derived by computing the similarity between visual features $\boldsymbol{F}_I$ and textual features $\boldsymbol{F}_T$. For semantic-level activation scores, we focus on capturing fine-grained semantic boundaries to enhance the representativeness of activated features. To achieve this, we construct a set of prototypes (Snell et al., 2017) for each semantic category, stored in a Semantic Memory Bank (SMB) $U \in \mathbb{R}^{k \times c}$, where k is the number of prototypes and c denotes the number of classes. During training, we match each image feature $\boldsymbol{F}_I$ against all semantic prototypes U in SMB

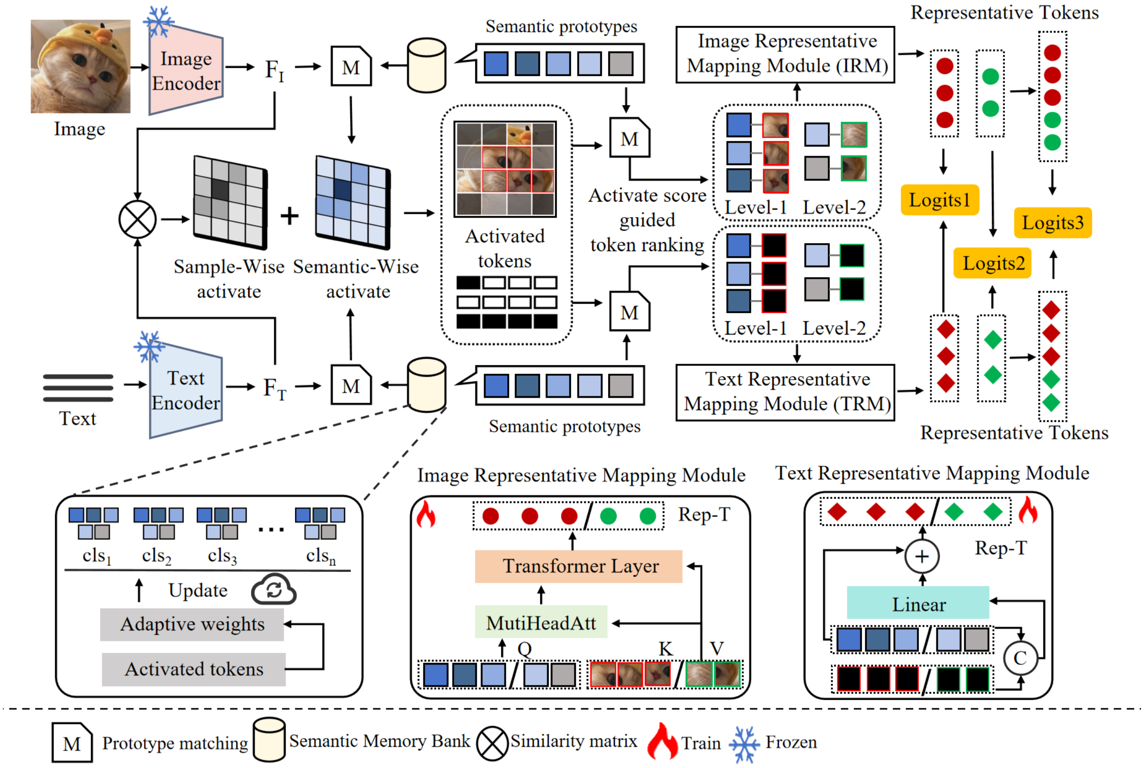


Figure 2: Overview of SpotLighter. The visual and textual features first compute sample-wise activations via a similarity matrix, which are fused with semantic-wise activations from prototype matching in the semantic memory bank. The combined activations yield k tokens with the highest scores that are further refined through score-based stratification and processed by TIRM to obtain representative tokens.

to identify the most relevant semantic category U_c ,

$$U_c = \text{argmax}\frac{\exp(\cos(\boldsymbol{F}_I, U_j))}{\sum_{j=1}^{C}\exp(\cos(\boldsymbol{F}_I, U_j))}. \qquad (3)$$

We then compute semantic-level activation scores by comparing each prototype against both visual and textual similarity. The final activation score is obtained by aggregating both sample-level and semantic-level activation scores. Experimental evidence confirms that highly activated tokens offer more discriminative signals for sample prediction. We preserve only the most k activated tokens tok^{act} as classification evidence while treating the remaining features as redundant noise. Notably, we continuously update the Semantic Memory Bank throughout the training process. Firstly, we assign each of tok^{act} to the corresponding prototype stored in U by a softmax function to get the probability:

$$\boldsymbol{D}_{i,j} = \frac{\exp(\cos(tok_i^{act}, U_j))}{\sum_{j=1}^{K}\exp(\cos(tok_i^{act}, U_j))}. \qquad (4)$$

Then, we assign by the highest probability as:

$$\boldsymbol{U_j} = \left\{ i \,\middle|\, \underset{k}{\text{argmax}}\boldsymbol{D}_{i,k} = j \right\}. \qquad (5)$$

Later, we will update the prototype in the Bank:

$$\boldsymbol{U}_j = \beta \cdot \boldsymbol{U}_j + (1-\beta)\sum_{i \in U_j} \boldsymbol{D}_{i,j} \cdot tok_i^{act}, \qquad (6)$$

with β representing the momentum coefficient. To further ensure the effectiveness of the activated tokens, we calculate the similarity between the final U of each class and sample-wise activation tokens:

$$L_{local} = \text{CE}\left(\cos_{local}(U, tok^{act}), y\right). \qquad (7)$$

This local loss minimizes feature selection subjectivity through activation values, enhancing cross-modal knowledge transfer to compensate for limited pre-training interaction.

Extraction of Representative Tokens. To compensate for potential semantic loss from discarded inactive regions, we fuse activated features with their corresponding semantic prototypes to obtain representative features for image classification. Given varying predictive contributions among activated features, we first perform dynamic matching between semantic prototypes and the sample's activated features, then stratify them into two tiers based on activation scores, as tok^{lev1} and tok^{lev2}.

To guide the model toward category-essential features, we progressively feed the prototypes and their matched activated tokens tok^{lev1} and tok^{lev2} respectively into the Image Representative Mapping Module (IRM) and the Text Representative Mapping Module (TRM), generating two sets of discriminative cross-modal representations. In IRM, the activated tokens tok^{lev_i} $(i = 1, 2)$ serve as both the key K and the value V, while the corresponding prototypes U serve as the query Q:

$$\boldsymbol{T} = \text{MultiHead}(\text{LN}(\boldsymbol{U}), \text{LN}(tok^{lev_i}), \text{LN}(tok^{lev_i})) + \boldsymbol{U}, \tag{8}$$

$$\widehat{\boldsymbol{U}} = \text{FFN}(\text{LN}(\boldsymbol{T})) + \boldsymbol{T}, \tag{9}$$

where MultiHead $(\cdot)$ and FFN $(\cdot)$ follow the standard transformer, respectively representing multi-head attention and feed-forward neural network. Subsequently, the fused token $\widehat{\boldsymbol{U}}$ is concatenated with the tok^{act} and processed through a transformer layer to obtain representative visual tokens

$$[tok_v^{rep}, tok^{lev_i}] = \theta_i([\widehat{\boldsymbol{U}}, tok^{lev_i}], \tag{10}$$

where $[\cdot, \cdot]$ refers to the concatenation of each token and θ is the pre-trained transformer layer. Meanwhile, for TRM, we begin by matching each original text token tok_t^{ori} with corresponding activated tokens using Eq.4 to get probability $W_{i,j}$. Following this, we generate the final representative text tokens tok_t^{rep} for activated token i and utilize a residual-connected linear layer to fuse dual feature streams, where α is the coefficient hyperparameter:

$$tok_{i,t}^{rep} = \alpha \cdot \text{Linear}([tok_{t,i}^{ori}, \sum_{j=1}^{k} W_{i,j} \cdot tok_j^{lev_i}]) + tok_{t,i}^{ori}. \tag{11}$$

Notably, for text features, we employ only a simple linear layer, with detailed implementation and rationale provided in the Appendix H. Then we concatenate the tokens achieved by Level-1 and Level-2 as tok_v^{rep} and tok_t^{rep}. Moreover, we posit that the set of high-activation-score features contains more discriminative information for classification. Thus, we formulate $\mathcal{L}_{cls}^{low}$ and $\mathcal{L}_{cls}^{high}$to ensure independent classification capability for both feature sets, while reconstructing the $\mathcal{L}_{cls}^{gra} = \mathcal{L}_{cls}^{low} + \mathcal{L}_{cls}^{high}$ to prioritize high-representative features. The way to calculate loss is similar to Eq.12.

3.3 Training and Inference

Throughout the training process, we maintain the conventional CLIP architecture while employing contrastive loss as our fundamental classification objective, mathematically expressed as:

$$\mathcal{L}_{cls} = -\log \frac{\exp(\cos(tok_v^{rep}, tok_t^{rep})/\tau)}{\sum_{j=1}^{C} \exp(\cos(tok_v^{rep}, tok_{t,i}^{rep})/\tau)}. \tag{12}$$

Beyond the standard contrastive loss formulation, we augment our module with a textual regularization loss and a visual KL loss, respectively:

$$\mathcal{L}_{reg}^{text} = |tok_t^{ori} - tok_t^{rep}|, \tag{13}$$

$$L_{KL}^{visual} = \text{KL}(tok_v^{rep}, tok_v^{ori}), \tag{14}$$

where $KL(\cdot, \cdot)$ represents Kullback-Leibler divergence and $tok_{t,v}^{ori}$ is the original text and visual tokens achieved by pre-trained models. The $\mathcal{L}_{reg}^{text}$ can mitigate overfitting in VLMs fine-tuning with limited training data, while L_{KL}^{visual} ensures useful image tokens exhibiting strong alignment with the original pre-trained feature space. Then the total loss can be calculated:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 * \mathcal{L}_{cls}^{gra} + \lambda_2 * \mathcal{L}_{reg}^{text} + \lambda_3 * (\mathcal{L}_{KL}^{visual} + \mathcal{L}_{local}), \tag{15}$$

where λ_1, λ_2, λ_3 are hyper-parameters used to balance the various loss terms. In all, we only need to train **the parameters in Eq. 9 and Eq. 11**, thus improving training efficiency.

During inference, we compute the final prediction scores using the fused cross-modal representations from both visual and textual features:

$$y = \arg\max_{i} \frac{\exp(\cos(tok_t^{rep}, tok_r^{rep})/\tau)}{\sum_{j=1}^{C} \exp(\cos(tok_t^{rep}, tok_r^{rep})/\tau)}. \tag{16}$$

Unlike existing approaches that rely on redundant remaining tokens after alignment, our method simply performs inference by the most representative tokens, thus mitigating noise-induced semantic degradation while reducing high-dimensional feature interactions in the representation space.

4 Experiments

4.1 Experimental Settings

Datasets. We employ the conventional approach used in previous studies (Zhou et al., 2022a; khattak et al., 2023) to conduct the base-to-new and few-shot on 11 benchmarks, *i.e.,* ImageNet (Deng et al., 2009), Caltech (Fei-Fei et al., 2007), OxfordPets (Parkhi et al., 2012), StanfordCars (Krause et al., 2013), Flowers (Nilsback and Zisserman, 2008), Food101 (Bossard et al., 2014), FGVCAircraft (Maji et al., 2013),

Table I: Comparison with other methods on base-to-new generalization with 16-shot.

Method	Average			ImageNet			Caltech101			OxfordPets		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
CLIP	69.34	74.22	71.70	72.43	68.14	70.22	96.84	94.00	95.40	91.17	97.26	94.12
CoOp	82.69	63.22	71.66	76.47	68.78	71.92	98.00	89.81	93.73	93.67	95.29	94.47
PromptSRC	84.26	76.10	79.97	77.60	70.73	74.01	98.10	94.03	96.02	95.33	97.30	96.30
MaPLe	82.28	75.14	78.55	76.66	70.54	73.47	97.74	94.36	96.02	95.43	97.76	96.58
CLIPFit	83.72	74.84	79.03	76.20	70.17	73.06	98.30	93.70	95.94	95.23	97.13	96.17
PromptKD	84.11	78.28	81.09	**77.63**	70.96	74.15	98.31	96.29	97.29	93.42	97.44	95.39
CoOp *w/* TextRefiner	79.74	74.32	76.94	76.84	70.54	73.56	98.13	94.43	96.24	95.27	97.65	96.45
PromptKD *w/* TextRefiner	85.22	79.64	82.33	77.51	71.43	74.38	98.52	96.52	97.51	95.60	**97.90**	96.74
CoOp *w/ Spotlighter*	81.74	75.80	78.66	76.74	70.68	73.58	98.13	94.51	96.29	**97.40**	97.73	**97.56**
PromptKD *w/ Spotlighter*	**85.65**	**80.46**	**82.89**	77.62	**71.71**	**74.55**	**98.86**	**96.74**	**97.79**	96.48	97.75	97.11

Method	StanfordCars			Flowers102			Food101			FGVCAircraft		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
CLIP	63.37	74.89	68.65	72.08	77.80	74.83	90.10	91.22	90.66	27.19	36.29	31.09
CoOp	78.12	60.40	68.13	97.60	59.67	74.06	88.33	82.26	85.19	40.44	22.30	28.75
PromptSRC	78.27	74.97	76.58	98.07	76.50	85.95	90.67	91.53	91.10	42.73	37.87	40.15
MaPLe	72.94	74.00	73.47	95.92	72.46	82.56	90.71	92.05	91.38	37.44	35.61	36.50
CLIPFit	78.80	73.87	76.26	96.83	73.53	83.59	90.60	91.33	90.96	42.47	33.47	37.43
PromptKD	80.48	81.78	81.12	98.69	81.91	89.52	89.43	91.27	90.34	43.61	39.68	41.55
CoOp *w/* TextRefiner	71.40	70.90	71.15	95.92	74.33	83.76	90.88	91.43	91.15	35.35	35.87	35.61
PromptKD *w/* TextRefiner	80.91	81.83	81.37	99.30	82.91	90.37	91.42	92.71	92.06	45.01	40.12	42.42
CoOp *w/Spotlighter*	70.09	69.97	70.03	95.10	74.47	83.53	**93.63**	91.51	**92.56**	39.00	36.54	37.72
PromptKD *w/Spotlighter*	**81.62**	**82.15**	**81.88**	**99.36**	**83.47**	**90.72**	91.86	**92.93**	92.39	**46.35**	**40.68**	**43.33**

Method	SUN397			DTD			EuroSAT			UCF101		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
CLIP	69.36	75.35	72.23	53.24	59.90	56.37	56.48	64.05	60.03	70.53	77.50	73.85
CoOp	80.60	65.89	72.51	79.44	41.18	54.24	92.19	54.74	68.69	84.69	56.05	67.46
PromptSRC	82.67	78.47	80.52	83.37	62.97	71.75	92.90	73.90	82.32	87.10	78.80	82.74
MaPLe	80.82	78.70	79.75	80.36	59.18	68.16	**94.07**	73.23	82.35	83.00	78.66	80.77
CLIPFit	81.97	78.17	80.02	81.97	63.50	71.56	93.33	71.07	80.69	85.23	77.30	81.07
PromptKD	82.53	80.88	81.70	82.86	69.15	75.39	92.04	71.59	80.54	86.23	80.11	83.06
CoOp *w/* TextRefiner	80.96	76.49	78.66	75.35	58.09	65.60	74.57	72.82	73.68	82.52	75.01	78.59
PromptKD *w/* TextRefiner	83.02	80.50	81.74	83.91	71.01	76.92	92.99	79.22	85.55	89.20	81.90	85.39
CoOp *w/ Spotlighter*	81.78	75.17	78.48	76.04	58.69	66.18	85.01	82.13	83.85	86.27	**82.47**	84.34
PromptKD *w/ Spotlighter*	**83.15**	**81.06**	**82.09**	**83.94**	**71.92**	**77.47**	93.17	**84.51**	**88.63**	**89.72**	82.16	**85.77**

EuroSAT (Helber et al., 2019), UCF101 (Soomro et al., 2012), DTD (Cimpoi et al., 2014), and SUN397 (Xiao et al., 2010). For cross-dataset generalization, we experiment on ImageNet-V2 (Recht et al., 2019), ImageNet-Sketch (Wang et al., 2019), ImageNet-A (Hendrycks et al., 2021b) and ImageNet-R (Hendrycks et al., 2021a). Meanwhile, the Implementation Details will be discussed in Apeendix C.

Baselines. We compare with many state-of-the-art (SOTA) method, including CLIP (Radford et al., 2021), CoOp (Zhou et al., 2022b), PromptSRC (Zhu et al., 2023), MaPLe (khattak et al., 2023), CLIPFit (Li et al., 2024a), PromptKD (Li et al., 2024c) and TextRefiner (Xie et al., 2024).

Table II: Comparison with other methods on the few-shot learning setting with average accuracy. We plug our method in PrompKD.

Method	Shot				
	1	2	4	8	16
CLIP	45.12	54.63	65.24	66.87	71.70
CoOP	68.09	70.13	73.59	76.45	79.01
PromptSRC	72.32	75.28	78.35	80.69	82.87
MaPLe	61.79	65.28	70.66	73.82	78.55
CLIPFit	72.32	74.39	77.18	79.03	81.27
PromptKD	72.47	75.19	78.46	79.56	81.09
w/ Spotlighter	**72.53**	**75.76**	**78.80**	**81.81**	**85.65**

4.2 Comparison with State-of-art Methods

Base-to-Novel Generalization. Table I presents the quantitative results of various methods in the base-to-novel generalization setting on 11 datasets.

Table III: Comparison with other methods on cross-domain generalization with 16-shot.

Method	Source	Target			
	ImageNet	-V2	-Sketch	-A	-R
CLIP	66.73	60.83	46.15	47.77	73.96
CoOpOp	71.02	64.07	48.75	50.63	76.18
PromptSRC	71.27	64.35	49.55	50.90	77.80
CoOp	71.51	64.20	47.99	49.71	75.21
w/ Spotlighter	72.12	66.17	49.32	49.81	76.59
MaPLe	70.72	64.05	49.15	50.90	76.98
w/ Spotlighter	**72.17**	**69.62**	**50.18**	**69.83**	**83.56**

Table IV: Comparison of inference efficiency among existing methods on the ImageNet Dataset.

Method	Params	FPS	HM
CoOp	2048	9768.21	71.92
CoCoOp	35K	20.45	73.10
CLIPFit	44K	8380.91	73.06
LLaMP	5.2M	1473.46	74.48
PromptKD	2.5M	12943.34	74.15
CoOp *w/ Spotlighter*	**+21**	**+886.61**	**+1.66**
PromptKD *w/ Spotlighter*	**+21**	**+1813.52**	**+0.35**

Table V: Ablation experiments on different optimization losses on ImageNet.

L_{cls}	L_{local}	L_{cls}^{low}	L_{cls}^{high}	L_{reg}^{text}	L_{kl}^{visual}	Base	Novel	HM
✓						76.50	67.88	71.93
✓				✓		76.58	70.62	73.48
✓	✓				✓	76.16	69.75	72.81
✓	✓	✓	✓			76.24	69.88	72.92
✓	✓	✓	✓	✓		76.13	70.31	73.10
✓	✓	✓	✓		✓	76.47	70.32	73.27
✓	✓	✓		✓	✓	76.98	71.16	73.96
✓	✓		✓	✓	✓	77.25	71.34	74.18
✓	✓	✓	✓	✓	✓	**77.62**	**71.71**	**74.55**

Table VI: Effects of tok^{lev1} and tok^{lev2} in Inference.

Method	Base	Novel	HM	FPS
tok^{lev1}	75.28	70.16	72.63	221.57K
tok^{lev2}	75.47	70.29	72.79	216.32K
tok^{lev_1+2}	**77.62**	**71.71**	**74.55**	131.25K

Our method demonstrates significant capability in consistently enhancing the performance of existing approaches across all evaluation metrics (Base, New, and HM), outperforming competing methods. Notably, Spotlighter significantly boosts CoOp's generalization capability on novel classes, achieving a remarkable accuracy improvement from 63.22% to 75.80%. With PromptKD, Spotlighter achieves the best accuracy to 85.65% on the base while improving the Novel to 80.46%. This verifies that after filtering out weakly relevant tokens, our model can reduce noise introduction while enhancing relevant semantic information, improving the model's generalization capability.

Few-shot Classification. In the few-shot scenario, our method also performs well. Following CLIP, we used 1/2/4/8/16-shot settings for training and calculated the accuracy on 11 datasets. Table II shows when compared with other methods, Spotlighter displays overall consistent improvement among all settings, demonstrating robustness and efficacy even in challenging low-data regimes.

Cross-Datasets Generalization. Extending beyond standard benchmarks, we assess Spotlighter's cross-domain generalization on four established datasets. The results shown in Table III verify that in cross-data scenarios, Spotlighter can still show the best results after few-shot training on ImageNet, especially for ImageNet-A having **18.93%** improvement. This demonstrates through progressive refinement, even a limited set of representative tokens can retain sufficient semantic information.

Efficiency. We further conduct a comparative analysis of inference efficiency, benchmarked on a single NVIDIA 4090 GPU using the officially released implementation. As shown in Table IV, when plugging in Spotlighter, other methods achieve faster inference speeds. Notably, with only **21 additional parameters**, Spotlighter not only attains the best performance in HM at the fastest inference speed. This efficiency gain is primarily due to using a compact set of semantically rich representative tokens, which substantially reduces the scale of feature interactions across the representation space, leading to a notable reduction in computational overhead.

4.3 Ablation Experiments

Effects of Different Losses. In the training process, we introduced a variety of training losses, shown in Eq.15. Table V investigates the influence of these factors on the model's generalization capability. The introduced L_{local} ensures the preservation of semantic information in useful tokens, while L_{reg}^{text} and L_{kl}^{visual} incorporate knowledge from original tokens and constrain fine-grained information utilization. Additionally, $\mathcal{L}_{cls}^{low}$ and $\mathcal{L}_{cls}^{high}$ enhance cross-modal interaction. Empirical results show that combining multiple training objectives effectively balances adaptability and generalization, leading to improved overall performance.

Effects of the Activated and Representative Tokens. To boost salient tokens' information density

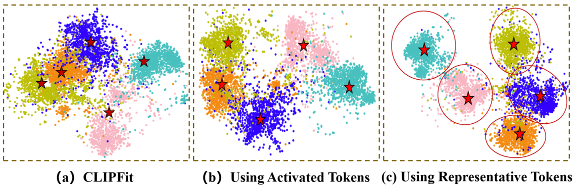

(a) CLIPFit (b) Using Activated Tokens (c) Using Representative Tokens

Figure 3: Visualization of the effect of the activated and representative tokens on the ImageNet dataset in few-shot learning via t-SNE.

Table VII: Comparison of baseline methods with and without Spotlighter in an average of 16 datasets.

Method	Base	Novel	HM
PromptKD	84.11	78.28	81.09
w/ Spotlighter	**85.65**+1.54	**80.46**+2.18	**82.89**+1.80
CLIPFit	83.72	74.84	79.03
w/ Spotlighter	**85.17**+1.45	**78.62**+3.78	**81.76**+2.73
MaPLe	82.28	75.14	78.55
w/ Spotlighter	**83.29**+1.01	**77.45**+2.31	**80.26**+1.71

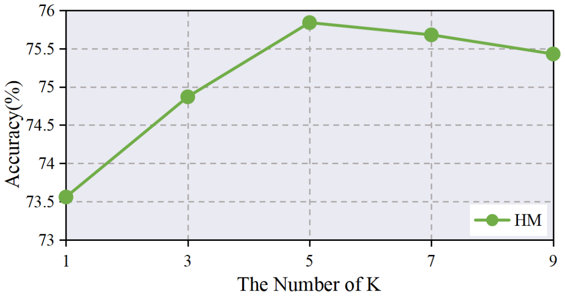


Figure 4: The impact of different activated and representative tokens.

Table VIII: Effects of the semantic action tokens.

Method	Base	Novel	HM
Spotlighter w/o Semantic Action	77.52	71.43	74.35
Spotlighter w Semantic Action	77.62	71.71	74.55

during aggressive pruning, we enhance the activated tokens to get representative tokens with ImageNet t-SNE (Selvaraju et al., 2017) visualizations. From Fig.3, we can observe that with Spotlighter, CLIPFit can have a much clearer separation of different class image features and more correct text features embedding in high-dimensional feature space, which contingents upon more granular stratification. Additionally, representative tokens can have better distinguishing capability.

Effects of the Two-Level Feature Activation. We stratify activated tokens by activation scores into Level-1 and Level-2 subsets, yielding more representative tokens for finer alignment and richer semantics. From Table VI and Table V, we observe that using only Level-1 or Level-2 tokens improves efficiency but sacrifices semantic coverage either in loss or inference. Therefore, unifying levels for better cross-modal interaction is necessary.

Effects on Different Backbones. To systematically examine the plug-and-play functionality of Spotlighter and demonstrate broad applicability, we implement the approach across multiple representative frameworks. Shown in Table VII, all four methods exhibit significant improvement, confirming effectiveness and versatility.

Effects of Different Activated Token Numbers. The token count hyperparameter k controls the number of activated and representative tokens. In Fig.4, we analyze the impact of different numbers. The results show that when the number is small, chosen tokens can obtain limited information, but when the number increases, too many tokens decrease speed and obtain noise.

Effects of the Semantic Activation. When computing activated tokens, we add the activation scores of the sample and the semantics. We observe from Table VIII that empowered by Semantic Activation Tokens, the sampled acquire richer and more discriminative semantic representations. This is because the prototypes store the most salient information of each image category and are continuously refined through updates. Their integration with individual samples mitigates the effects of sample-level variance and information sparsity, ultimately leading to higher-quality activated tokens.

5 Conclusion

We introduce Spotlighter, a plug-and-play framework that revisits few-shot image classification from the perspective of representative token mining. By progressively selecting and categorizing informative tokens, Spotlighter effectively filters noise and reduces redundant feature interactions. Leveraging both activated tokens and representative tokens, the model enhances fine-grained cross-modal alignment with minimal parameter overhead. Extensive experiments across 11 benchmarks and diverse generalization settings show that Spotlighter consistently improves accuracy and efficiency over strong baselines. Our work highlights the importance of token-level selection and structured refinement for efficient and robust few-shot learning with vision-language models.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 - mining discriminative components with random forests. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, volume 8694 of *Lecture Notes in Computer Science*, pages 446–461. Springer.

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

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. 2024. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16901–16911.

Yuan Chiang, Elvis Hsieh, Chia-Hong Chou, and Janosh Riebesell. 2024. Llamp: Large language model made powerful for high-fidelity materials knowledge retrieval and distillation. *arXiv preprint arXiv:2401.17244*.

Hyungyu Choi, Young Kyun Jang, and Chanho Eom. 2025. Goal: Global-local object alignment learning. *arXiv preprint arXiv:2503.17782*.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 3606–3613. IEEE Computer Society.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society.

Matteo Farina, Massimiliano Mancini, Giovanni Iacca, and Elisa Ricci. 2025. Rethinking few-shot adaptation of vision-language models in two stages. *arXiv preprint arXiv:2503.11609*.

Li Fei-Fei, Robert Fergus, and Pietro Perona. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Comput. Vis. Image Underst.*, 106(1):59–70.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 12(7):2217–2226.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. 2021a. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. 2021b. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271.

Yufeng Huang, Jiji Tang, Zhuo Chen, Rongsheng Zhang, Xinfeng Zhang, Weijie Chen, Zeng Zhao, Tangjie Lv, Zhipeng Hu, and Wen Zhang. 2023. Structure-clip: Enhance multi-modal language representations with structure knowledge. *arXiv preprint arXiv:2305.06152*, 2(3).

Sehun Jung and Hyang-won Lee. 2025. Learning generalizable prompt for clip with class similarity knowledge. *arXiv preprint arXiv:2502.11969*.

Muhammad Uzair khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. 2023. Maple: Multi-modal prompt learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. 2023a. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15190–15200.

Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. 2023b. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15190–15200.

Sangwon Kim, Jaeyeal Nam, and Byoung Chul Ko. 2022. Vit-net: Interpretable vision transformers with neural tree decoder. In *International conference on machine learning*, pages 11162–11172. PMLR.

Sungyeon Kim, Boseung Jeong, Donghyun Kim, and Suha Kwak. 2024. Efficient and versatile robust fine-tuning of zero-shot models. In *European Conference on Computer Vision*, pages 440–458. Springer.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, pages 554–561. IEEE Computer Society.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.

Ming Li, Jike Zhong, Chenxin Li, Liuzhuozheng Li, Nie Lin, and Masashi Sugiyama. 2024a. Vision-language model fine-tuning via simple parameter-efficient modification. *arXiv preprint arXiv:2409.16718*.

Yunheng Li, Zhong-Yu Li, Quan-Sheng Zeng, Qibin Hou, and Ming-Ming Cheng. 2024b. Cascade-clip: Cascaded vision-language embeddings alignment for zero-shot semantic segmentation. In *International Conference on Machine Learning*.

Zheng Li, Xiang Li, Xinyi Fu, Xin Zhang, Weiqiang Wang, Shuo Chen, and Jian Yang. 2024c. Promptkd: Unsupervised prompt distillation for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26617–26626.

Xialei Liu, Yu-Song Hu, Xu-Sheng Cao, Andrew D Bagdanov, Ke Li, and Ming-Ming Cheng. 2022. Long-tailed class incremental learning. In *European Conference on Computer Vision*, pages 495–512. Springer.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. 2024. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*.

Ziqian Lu, Mushui Liu, Yunlong Yu, Zhao Wang, Xi Li, and Jungong Han. 2025. Variational adapter: Improving clip in data-imbalanced scenarios. *IEEE Transactions on Circuits and Systems for Video Technology*.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. 2013. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151.

Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *Sixth Indian Conference on Computer Vision, Graphics & Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008*, pages 722–729. IEEE Computer Society.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. 2012. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 3498–3505. IEEE Computer Society.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR.

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

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. 2024. Argue: Attribute-guided prompt tuning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28578–28587.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. 2020. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. 2019. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 32.

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

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. 2010. SUN database: Large-scale scene recognition from abbey to zoo. In

The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010, pages 3485–3492. IEEE Computer Society.

Jingjing Xie, Yuxin Zhang, Jun Peng, Zhaohong Huang, and Liujuan Cao. 2024. Textrefiner: Internal visual feature as efficient refiner for vision-language models prompt tuning. *arXiv preprint arXiv:2412.08176*.

Chen Xu, Yuhan Zhu, Haocheng Shen, Boheng Chen, Yixuan Liao, Xiaoxin Chen, and Limin Wang. 2025. Progressive visual prompt learning with contrastive feature re-formation. *International Journal of Computer Vision*, 133(2):511–526.

Kaicheng Yang, Tiancheng Gu, Xiang An, Haiqiang Jiang, Xiangzi Dai, Ziyong Feng, Weidong Cai, and Jiankang Deng. 2025. Clip-cid: Efficient clip distillation via cluster-instance discrimination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21974–21982.

Hantao Yao, Rui Zhang, and Changsheng Xu. 2023. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6757–6767.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer.

Haotian* Zhang, Pengchuan* Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. 2022. Glipv2: Unifying localization and vision-language understanding. *arXiv preprint arXiv:2206.05836*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022a. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*.

Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. 2023. Zegclip: Towards adapting clip for zero-shot semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. 2023. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15659–15669.

Haowei Zhu, Wenjing Ke, Dong Li, Ji Liu, Lu Tian, and Yi Shan. 2022. Dual cross-attention learning for fine-grained visual categorization and object re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4692–4702.

Lianghui Zhu, Xinggang Wang, Jiapei Feng, Tianheng Cheng, Yingyue Li, Bo Jiang, Dingwen Zhang, and Junwei Han. 2025. Weakclip: Adapting clip for weakly-supervised semantic segmentation. *International Journal of Computer Vision*, 133(3):1085–1105.

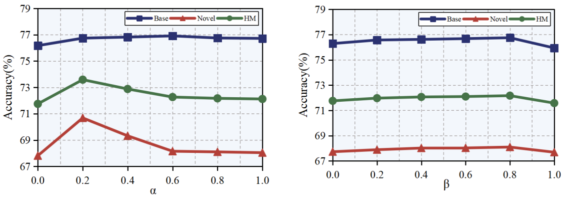
Supplementary Material of *Spotlighter: Revisiting Prompt Tuning from a Representative Mining View*

A Limitations

Spotlighter is primarily designed for fine-tuning vision-language models in image classification and may not generalize well to other vision tasks such as object detection or image segmentation, where dense or spatially localized predictions are required. This limitation partly stems from the reduced number of final tokens, which may omit fine-grained spatial details essential for those tasks. Moreover, the effectiveness of our method relies on the presence of sufficiently discriminative representative tokens; performance degrades when such tokens are sparse or class boundaries are highly entangled, particularly in ultra-fine-grained settings. In future work, we plan to extend Spotlighter to dense prediction tasks by incorporating spatial-aware token selection and hierarchical refinement. We also aim to investigate adaptive token filtering strategies that dynamically adjust to data complexity and class granularity.

B Dataset Statistics

To rigorously assess the effectiveness and generalization ability of our method, we performed extensive experiments on 11 standard benchmark datasets spanning multiple visual domains (Table IX). The selected datasets cover diverse recognition tasks including: ImageNet (Deng et al., 2009) for object classification; Caltech (Fei-Fei et al., 2007) for natural object recognition; OxfordPets (Parkhi et al., 2012) for fine-grained pet classification; StanfordCars (Krause et al., 2013) for vehicle categorization; Flowers (Nilsback and Zisserman, 2008) for flower species identification; Food101 (Bossard et al., 2014) for food classification; FGVCAircraft (Maji et al., 2013) for aircraft recognition; EuroSAT (Helber et al., 2019) for satellite imagery analysis; UCF101 (Soomro et al., 2012) for action recognition; DTD (Cimpoi et al., 2014) for texture classification; and SUN397 (Xiao et al., 2010) for scene understanding.In distribution shift experiments,we also introduce ImageNet-V2 (Recht et al., 2019), ImageNet-Sketch (Wang et al., 2019), ImageNet-A (Hendrycks et al., 2021b) and ImageNet-R (Hendrycks et al., 2021a).These datasets are all to improve ImageNet test reliability. This comprehensive evaluation across multiple domains effectively demonstrates our approach's robustness and versatility in various scenarios.



(a) Aggregation Coefficient . (b) Momentum Coefficient.

Figure 5: The impact of different aggregation coefficient and momentum coefficient.

C Implementation Details

We adopt ViT-B/16 CLIP model to conduct all of our experiments. We report both base and novel class accuracies along with their harmonic mean (HM) (Xian et al., 2017), with all metrics averaged across three independent runs. To ensure a fair comparison, final performance metrics are computed as the mean over three different random seeds. The experimental settings remain consistent with the original papers while the only modification is in the number of training epochs where CoOp is reduced to 15 epochs, while ClipFit and PromptKD are reduced to 30 epochs. The number of fusion coefficient α is 0.2 and momentum coefficient β is 0.8 respectively. For the hyper-parameters in the loss, we set λ_1, λ_2, λ_3 to 0.02, 20, 0.1 supported by empirical findings and fixed in different datasets to facilitate downstream tasks.

D Effects of Coefficient α and β

Hyperparameters α and β control original information retention and filtered knowledge preservation, respectively. In Fig.5a, accuracy on base classes remains stable with increasing α, while novel classes peak then decline, suggesting overfitting from fine-grained feature dependence. Meanwhile, increasing β yields gentle rise-then-fall trends for both Base and Novel, confirming the discriminative token selection.

E Hyperparameter Analysis of Optimization Objectives.

Our systematic investigation of the balancing hyperparameters in Eq.15 reveals important insights into the method's behavior. Through controlled experiments where we varied individual parameters while

Table IX: The detailed statistics of datasets used in our work.

Datasets	Classes	Training Size	Validation Size	Testing Size	Tasks	Hand-crafted Prompt
ImageNet	1,000	1.28M	N/A	50,000	General object recognition	"a photo of a [CLASS]."
Caltech	100	4,128	1,649	2,465	General object recognition	"a photo of a [CLASS]."
EuroSAT	10	13,500	5,400	8,100	Satellite image recognition	"a centered satellite photo of [CLASS]."
SUN397	397	15,880	3,970	19,850	Scene recognition	"a photo of a [CLASS]."
DTD	47	2,820	1,128	1,692	Texture recognition	"[CLASS] texture."
UCF101	101	7,639	1,808	3,783	Action recognition	"a photo of a person doing [CLASS]."
FGVCAircraft	100	3,334	3,333	3,333	Fine-grained aircraft recognition	"a photo of a [CLASS], a type of aircraft."
OxfordPets	37	2,944	736	3,669	Fine-grained pets recognition	"a photo of a [CLASS], a type of pet."
StanfordCars	196	6,509	1,635	8,041	Fine-grained car recognition	"a photo of a [CLASS], a type of flowers."
Flowers	102	4,093	1,633	2,463	Fine-grained flowers recognition	"a photo of a [CLASS]."
Food101	101	50,500	20,200	30,300	Fine-grained food recognition	"a photo of a [CLASS], a type of food."
ImageNetV2	1000	N/A	N/A	10,000	Improve ImageNet test reliability	"a photo of a [CLASS]."
ImageNet-Sketch	1000	N/A	N/A	50,899	Improve ImageNet test reliability	"a photo of a [CLASS]."
ImageNet-A	1000	N/A	N/A	7,500	Improve ImageNet test reliability	"a photo of a [CLASS]."
ImageNet-R	1000	N/A	N/A	30,000	Improve ImageNet test reliability	"a photo of a [CLASS]."

Table X: Ablation experiments on different backbones.

Backbone	Parameters	Base	Novel	HM
ViT-B/16	151M	85.64	80.37	83.01
ViT-L/14	427M	85.68	81.29	83.43

Table XI: The method chosen for Image/Text Representative Mapping Module.

Method	Base	Novel	HM	FPS
liner+liner	76.27	70.98	73.53	135.19K
trans+trans	77.64	71.75	74.58	86.89K
original	77.62	71.71	74.50	131.25K

fixing others, we observe that the method demonstrates consistent performance across a wide range of configurations, highlighting its robustness and broad applicability to different pre-trained models, shown in Fig.6. However, the performance analysis also identifies critical limitations that tremendous values of $\lambda_{1,2,3}$ lead to noticeable degradation in model performance. This suggests that while the balancing terms are essential for proper alignment, pushing them too far can be counterproductive. The performance drop likely stems from two interrelated factors: first, excessive alignment may force the model to capture artifactual correlations in the training data, leading to overfitting; second, overly strong regularization can constrain the model's capacity to learn meaningful feature representations. These findings emphasize the importance of finding an appropriate balance in parameter settings, where sufficient alignment is achieved without compromising the model's learning capability. The demonstrated robustness across parameter variations further confirms the method's reliability for practical deployment scenarios.

F More Few-shot Learning Results

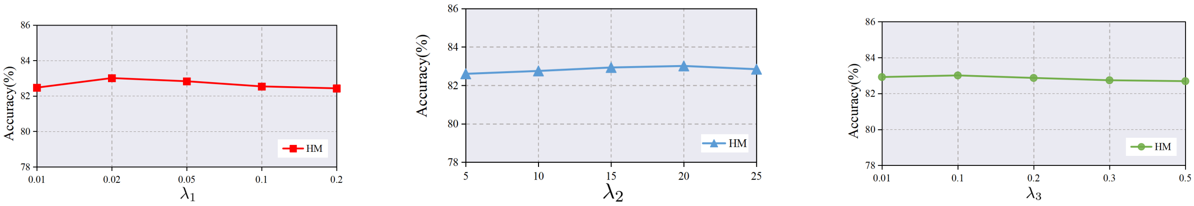
We adopted the few-shot evaluation protocol from (Radford et al., 2021), evaluating our method's ability to acquire task-specific knowledge through 1,2,4,8 and 16-shot learning scenarios while measuring classification accuracy. In Fig.7, we further conducted a visual comparison between our method and CLIPFit, demonstrating superior performance across all 11 datasets.

G Effects of Different Backbones of CLIP

The choice of backbone networks with varying parameter sizes significantly influences model performance. To systematically evaluate our method's compatibility with different architectures, we conduct extensive experiments across multiple backbone networks (Table X). The results reveal a consistent trend: model performance scales positively with increasing network capacity, demonstrating our approach's strong adaptability to different architectural scales. Notably, performance gains exhibit diminishing returns while smaller networks show limited capability due to constrained feature extraction capacity; the performance improvement becomes more pronounced as network size increases. This pattern suggests that our method effectively leverages the enhanced representational power of larger networks to capture richer feature hierarchies while maintaining stable performance across different architectural scales.

H Design of Image/Text Representative Mapping Module.

We derive the final representative tokens through the Image/Text Mapping Module. In Table XI, we contrast the methodological designs employed for alignment. We observe that while employing



(a) (b) (c)

Figure 6: The effect of different loss balance parameters λ_1, λ_2, and λ_3 on the model classification accuracy.

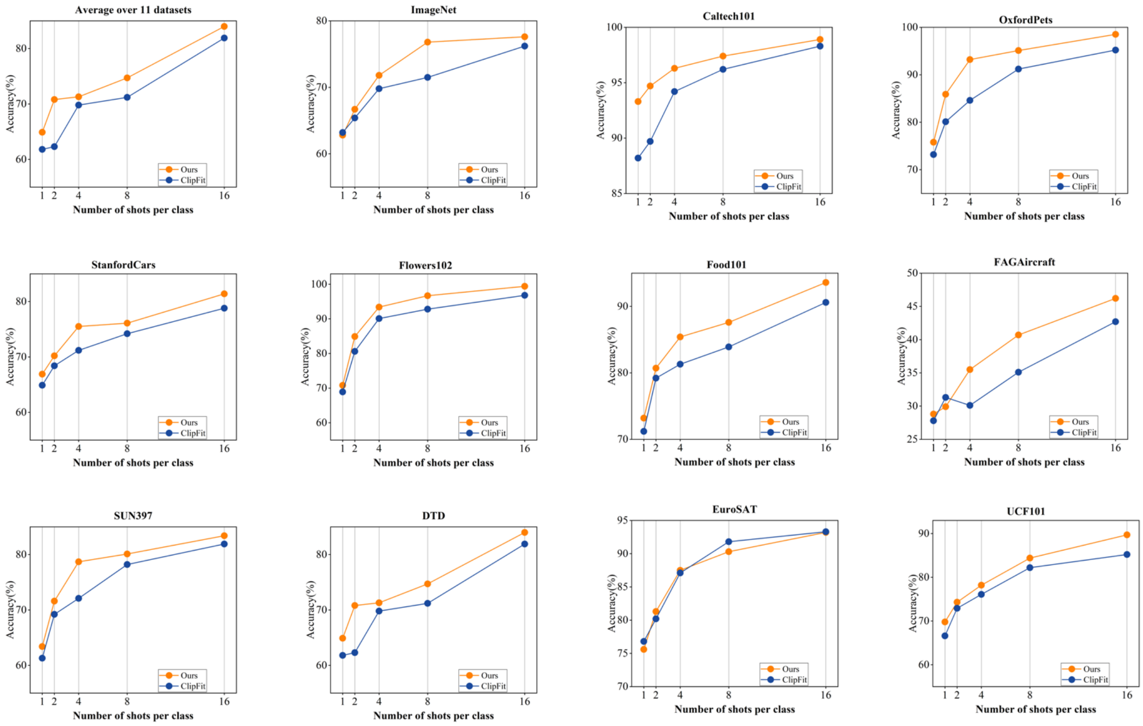


Figure 7: Performance of few-shot learning across 11 datasets compared with CLIPFit (Li et al., 2024a). The result demonstrates that our method shows better performance than CLIPFit, even with fewer parameters and fewer tokens.

simple linear layers for multimodal processing improves computational efficiency, it leads to noticeable accuracy degradation. Conversely, adopting full transformer architectures yields marginal accuracy gains over current methods while significantly compromising computational efficiency. This occurs because text tokens inherently encode simpler information compared to visual tokens. Overly complex architectures (e.g., transformers) prove less effective for processing such straightforward patterns, where lightweight linear layers suffice.

Table XII: Effects of whether to recalculate score.

Method	Base	Novel	HM
Spotlighter w/o recaculate	77.53	71.67	74.48
Spotlighter w recaculate	77.62	71.71	74.55

I Effects of whether to recalculate activation score.

During the secondary classification of activated tokens, we rematch them with prototypes and recompute their activation scores to choose the new top-k. Alternatively, one could directly stratify the top-k activated tokens without recalibration. The Table XII demonstrates that recalibration yields superior performance compared to direct selection. This improvement stems from the progressively enriched semantic information encapsulated in the updated prototypes. By rematching and recomputing activated tokens against these refined prototypes, we more accurately identify tokens with the highest semantic density.