DICE: Disentangled Instance-Class knowlEdge prompt tuning via SAE for Vision-Language Models

Daeun Lee[1], Seungwoo Jang[1], and Kwangsu Kim[1]

[1]Sungkyunkwan University, South Korea
9.9danilee@gmail.com, {sewo, kim.kwangsu}@skku.edu

Abstract. CLIP, known for its strong zero-shot generalization, has received significant attention in prompt tuning for its ability to adapt effectively to new tasks in few-shot settings. Recent studies have employed descriptions generated by Large Language Models (LLMs) as predefined prompts to obtain class-level semantics. However, because these predefined descriptions are not directly grounded in visual information, they often misalign with an image's visual semantics. They tend to generate generic class-level descriptions rather than instance-specific ones. As a result, LLM-based prompts are structurally biased toward class-level semantics, making it difficult to capture and reflect instance-level cues, which in turn limits generalization in few-shot settings where both levels of knowledge are crucial. To address this limitation, we propose Disentangled Instance-Class knowlEdge (DICE) prompt tuning, a framework that restructures LLM-derived class-level priors to capture instance-level semantics. Specifically, DICE decomposes LLM-derived priors into instance-level components using a Sparse Autoencoder (SAE), which selects instance-specific concept vectors. These concept vectors are then fused with class embeddings to form enriched representations. This synergy preserves class-level semantic coherence while capturing instance-level details, improving generalization to unseen classes. Our approach achieves competitive performance across 11 few-shot recognition benchmarks, while additionally offering a plug-and-play solution that enhances interpretability through SAE.

Keywords: Prompt learning · Sparse autoencoder · Plug-and-play

1 Introduction

Vision-Language Models (VLMs) show strong generalization across downstream tasks [1, 36]. CLIP [36], in particular, is widely used for few-shot learning due to its zero-shot performance. However, adapting CLIP with only a few examples remains challenging, as fine-tuning often leads to overfitting. To address this, prompt tuning has been introduced as a more efficient and robust alternative [51, 52]. More recently, researchers have leveraged Large Language Models (LLMs) to generate descriptive prompts, providing rich class-level priors that help capture each class's overall semantics [20, 35, 38, 39, 46].

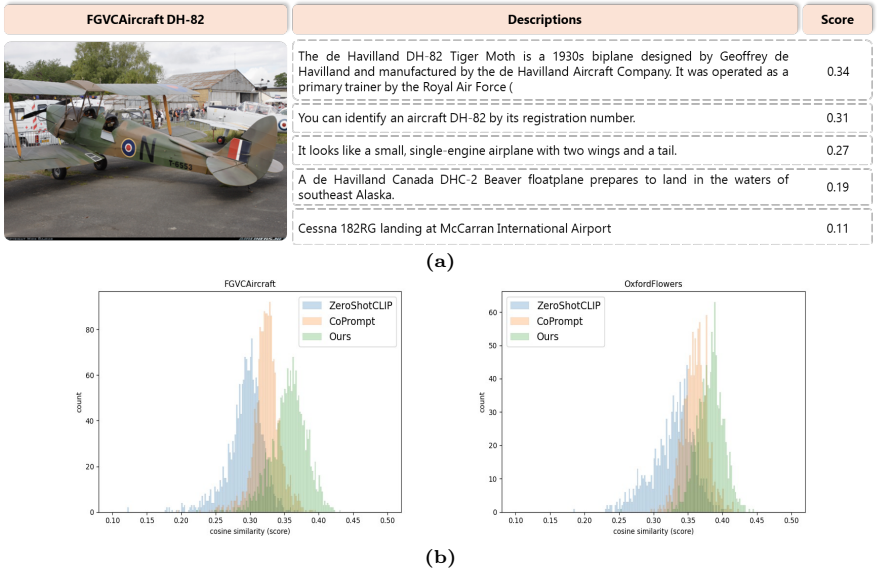


Fig. 1: Motivations. (a) Cosine similarities between LLM-generated descriptions of class DH-82 and corresponding image in FGVC-Aircraft. (b) Comparison of cosine similarity distributions among ZeroshotCLIP, CoPrompt [39], and ours.

However, existing LLM-based prompt tuning methods suffer from several limitations. First, LLM-generated descriptions that are not grounded in visual evidence are predefined and uniformly applied to all images within a class, regardless of their visual differences. In practice, a single class description is often generated from a generic query (e.g., "*What does a [class] look like?*") and then reused for all images in that class [20, 35, 38, 39, 46]. This practice prevents the prompt from capturing instance-specific variations. Second, although LLMs encode rich semantic attributes, this knowledge is typically expressed in a dense, entangled form, making it non-trivial to isolate image-relevant factors. Existing LLM-based prompt tuning lacks an explicit mechanism to decompose and select such factors conditioned on each instance. Consequently, it often falls back to coarse class-level semantics, limiting its ability to capture instance-specific variations. Third, current approaches also lack a principled way to integrate class-level priors with instance-specific visual evidence into a unified prompt representation [50, 51]. As a result, the generated prompts often capture only class-average semantics and align poorly with the actual image content.

To analyze the lack of instance-specific alignment, we measured the cosine similarity between CLIP image embeddings and CLIP text embeddings obtained from LLM-generated descriptions. As shown in Fig. 1a, a single image is compared with multiple LLM-generated descriptions. Some descriptions are inconsistent with the actual visual content, resulting in significantly lower similarity scores (e.g., the fifth description, which refers to a different scene). We further

computed the similarity distributions on the FGVC-Aircraft [30] and Oxford Flowers [32] datasets based on CLIP text embeddings derived from these descriptions, as illustrated in Fig. 1b. The distribution exhibits a pronounced left tail, indicating that a substantial portion of the descriptions have low similarity with their corresponding images. This suggests that predefined class-level descriptions often miss instance-specific visual cues, leading to suboptimal visual–textual alignment. This limitation is particularly problematic in few-shot settings, where subtle visual differences (e.g., among car or aircraft models) are essential for accurate recognition.

We argue that effective few-shot learning should integrate both class and instance-level semantics from LLMs to achieve robust generalization. Class-level semantics maintain global semantic structure and help position new categories within a coherent embedding space [50]. However, this broad generalization cannot capture the subtle visual variations that distinguish instances within a class. In contrast, instance-level cues capture such fine-grained details but lack the inter-class structure required for consistent category placement [51]. This motivates our main question: **How can we explicitly integrate class-level priors with instance-specific visual concepts within LLM-based prompt tuning?**

To answer this question, we introduce **Disentangled Instance-Class knowlEdge prompt tuning (DICE)**, a framework that jointly models instance-level visual concepts and class-level knowledge embedded in LLMs. Because vision–language embeddings are typically dense and polysemantic, it is difficult to identify which semantic factors are relevant to a particular image. To address this, we employ a Sparse Autoencoder (SAE) that decomposes embeddings into sparse and disentangled concept units, where each latent dimension represents a distinct semantic concept. DICE consists of two branches. The Class Context Module captures class-level knowledge from LLM embeddings, while the Instance Context Module extracts instance-specific semantics using an SAE-derived concept dictionary and selects discriminative concepts through a class-guided coactivation score. The selected concepts are fused with class embeddings and then fed into a transformer to generate prompts that reflect both general class knowledge and instance semantics. Figure 1b shows that our method's similarity distribution is shifted to the right compared to other models. This indicates that our method mitigates the limitations of predefined LLM prompts by effectively integrating class-level knowledge with instance-specific semantics, thus strengthening visual–textual alignment.

The main contributions of this work can be summarized as follows.

- **Concept Dictionary via Sparse Autoencoders.** We first show that sparse autoencoders can serve as a learnable concept dictionary for vision–language representations, enabling explicit concept selection and improving the interpretability of prompt tuning.
- **Class–Instance Semantic Integration.** We introduce a unified prompt formulation that integrates class-level semantic priors with instance-specific visual concepts for robust few-shot adaptation.

– **From Handcrafted Descriptions to Concept-based Prompts.** We replace predefined class descriptions with prompts constructed from disentangled concept units, enabling prompts to adapt dynamically to each input instance rather than relying on fixed class-level templates.

2 Related work

2.1 Prompt Tuning

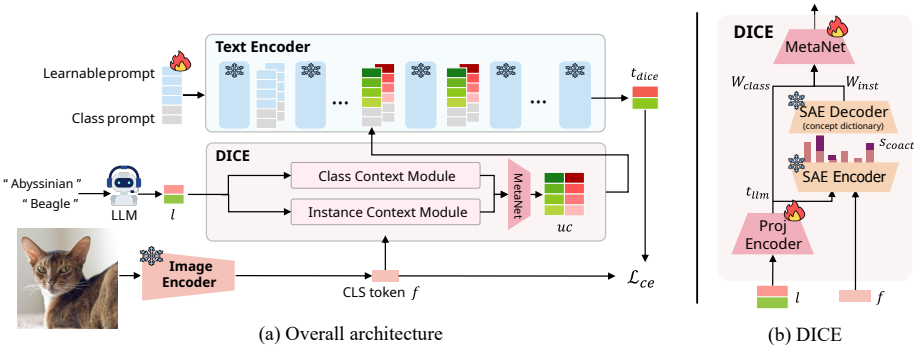
Prompt tuning was first introduced in natural language processing (NLP) as a parameter-efficient fine-tuning strategy, updating only a small number of parameters for task adaptation [26,28]. This idea has been extended to VLMs such as CLIP for few-shot learning [35, 44, 51, 52]. Among these, Context Optimization (CoOp) [52] replaces handcrafted prompts with learnable continuous vectors, mitigating sensitivity to template phrasing. However, CoOp lacks semantic grounding in the image, limiting generalization to unseen classes. To address this, CoCoOp [51] introduces instance-conditioned prompts through a lightweight meta-network, allowing the prompts to adapt to each image. KgCoOp [49] mitigates overfitting by regularizing with the pretrained CLIP model. MaPLe [22] further improves text-image alignment by applying learnable prompts to both encoders. TCP [50] enhances class discrimination by encoding class embeddings into intermediate layers of the text encoder.

Recent studies have leveraged the semantic richness of LLMs to improve prompt quality. Works such as KART [20], ProText [21], HPT [46], and CoPrompt [39] generate class descriptions from LLMs, encoding class-level textual knowledge into CLIP. DeMul [25] reported that LLM-generated descriptions suffer from high variability and unreliability, and proposed using embeddings obtained directly from raw class names, which preserve rich semantic information. Similarly, we leverage the rich semantic representations embedded in class-name embeddings instead of relying on description-based approaches.

2.2 Sparse Autoencoder

Mechanistic interpretability [13] aims to understand how neural networks internally compute and produce their outputs. Dense representations entangle multiple semantic factors, making interpretability challenging. In this context, SAEs have recently gained attention as an effective technique for improving the interpretability of LLMs [6, 7, 9, 17, 43]. SAEs leverage the principle of superposition [12] to transform dense, polysemantic embeddings into a sparse latent space, where each neuron encodes a distinct, monosemantic concept. This sparse transformation facilitates the extraction of salient information and promotes disentanglement by representing semantic features as clear, interpretable concept units [5, 33].

SAEs have also been expanded beyond NLP to vision tasks [3, 15]. For instance, studies on unlearning [10] and concept bottleneck models [37,40] have employed SAEs to decompose visual embeddings into semantic concepts. Furthermore, PatchSAE [41] demonstrated that SAEs can extract class-discriminative



(a) Overall architecture (b) DICE

Fig. 2: The framework of disentangled instance-class knowledge prompt tuning

information from image embeddings. Collectively, these studies demonstrate that SAEs can yield sparse, interpretable, and effective representations for downstream tasks.

3 Methodology

Our method consists of two stages. First, we pretrain two autoencoders to prepare structured semantic representations. A projection autoencoder aligns LLM embeddings with the CLIP embedding space, enabling them to be integrated with CLIP features, while an SAE decomposes dense semantics into sparse concept bases that allow explicit concept selection. Second, during prompt tuning, we construct a unified context prompt by jointly leveraging the Class Context and Instance Context modules based on the representations obtained from the pretrained autoencoders.

3.1 Preliminary

Context optimizations. CLIP is a vision-language model that aligns images and texts in a shared embedding space via contrastive learning. It consists of an image encoder $F(\cdot)$ and a text encoder $G(\cdot)$, both projecting inputs into a d-dimensional space. Let $(x, y) \in \mathcal{D}_s$, where $\mathcal{D}_s$ denotes the labeled source dataset with image $x \in \mathbb{R}^{3\times H\times W}$ and label $y \in \{1, \ldots, C\}$. Then, the image embedding is computed as $f = F(x) \in \mathbb{R}^d$. The label is converted into a textual description using a hand-crafted template (e.g., "*a photo of a [class]*"), and embedded via the text encoder as $t^y = G(\text{template}(y)) \in \mathbb{R}^d$. Collectively, the class embeddings form $t \in \mathbb{R}^{C\times d}$.

However, such static templates lack flexibility and often fail to generalize across diverse tasks. To address this, CoOp replaces the fixed textual prefix with learnable continuous context vectors. Specifically, it learns a sequence of tokens

$\{p_1, p_2, \ldots, p_N\}$ inserted before the class token c_i to form $\{w_{\text{SOS}}, p_1, p_2, \ldots, p_N, c_i, w_{\text{EOS}}\}$.

The model is optimized using a cross-entropy loss that encourages the image embedding f to align with the correct class embedding t^y over all C candidate classes:

$$\mathcal{L}_{\text{ce}} = \frac{1}{N} \sum_{(x,y)\in\mathcal{D}_s} -\log \frac{\exp\left(\cos(f, t^y)/\tau\right)}{\sum_{c=1}^{C} \exp\left(\cos(f, t^c)/\tau\right)} \tag{1}$$

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

Sparse autoencoder. An SAE consists of a single-layer encoder and decoder with a sparsifying activation function (e.g., ReLU [9] or Top-k [31]). Given an input $x \in \mathbb{R}^d$, the encoder computes a sparse activation vector s as follows:

$$s = \text{activation}\left(E_{\text{sae}}(x - b_{\text{pre}})\right), \quad s \in \mathbb{R}^{d_{\text{hid}}} \tag{2}$$

In this formulation, E_{sae} is the encoder weight matrix, b_{pre} is a centering bias applied to the input, and $d_{\text{hid}} = \alpha \cdot d$, where α denotes the expansion ratio relative to the input dimension d. The sparsifying activation function retains only the most informative dimensions, effectively filtering out irrelevant features. In this work, we adopt a Top-k sparse autoencoder [31], which retains only the Top-k activations in s and sets the remaining values to zero. The sparse vector s is then passed through a linear decoder with weight matrix $D_{\text{sae}} \in \mathbb{R}^{d_{\text{hid}} \times d}$, reconstructing the input as a linear combination of its rows. Let $v_i \in \mathbb{R}^d$ denote the i-th row of D_{sae}. The reconstruction is given by

$$\hat{x} = D_{\text{sae}}(s) + b_{\text{pre}} = \sum_{i=0}^{d_{\text{hid}}-1} s_i v_i + b_{\text{pre}}, \quad \hat{x} \in \mathbb{R}^d. \tag{3}$$

The SAE is trained to minimize the reconstruction loss $||x - \hat{x}||_2$ along with a sparsity constraint on the latent representation s. As a result, each nonzero entry in the sparse activation vector s weights its corresponding row vector v_i in decoder weight matrix D_{sae}, and these are linearly combined to reconstruct the input. In this view, each row v_i serves as a semantically meaningful direction in the input space. This allows us to interpret D_{sae} as a learnable concept dictionary composed of d_{hid} such semantic vectors.

3.2 Pre-training

DICE pretrains two autoencoders: a projection autoencoder for aligning LLM embeddings with CLIP embeddings, and an SAE for extracting interpretable concept representations in CLIP embedding space. To this end, we cache LLM embeddings by encoding class names from 11 few-shot datasets and WordNet terms [16] using OpenAI's *text-embedding-3-large* model (3072 dimensions) [2].

Projection autoencoder. The projection autoencoder consists of a single-layer encoder $E_{\text{proj}}(\cdot)$ and decoder $D_{\text{proj}}(\cdot)$. It maps LLM embeddings $l \in \mathbb{R}^{d_{\text{llm}}}$ into the CLIP text embedding space $\mathbb{R}^d$, as follows:

$$t_{\text{llm}} = E_{\text{proj}}(l), \quad \hat{l} = D_{\text{proj}}(t_{\text{llm}}) \tag{4}$$

This module is trained to align the LLM embedding l with the corresponding CLIP embedding t_{clip} by minimizing the projection loss, defined as $\mathcal{L}_{\text{proj}} = 1 - \cos(t_{\text{llm}}, t_{\text{clip}})$. Additionally, to preserve the original semantics of the LLM embedding during projection, we apply an LLM reconstruction loss, defined as $\mathcal{L}_{\text{llm\_rec}} = 1 - \cos(l,\ \hat{l})$.

Sparse autoencoder. The SAE, as detailed in Sec. 3.1, learns structured concept representations by reconstructing CLIP embeddings $t_{\text{clip}} \in \mathbb{R}^d$ within a sparse latent space $\mathbb{R}^{d_{\text{hid}}}$. A Top-$k$ sparsity constraint activates only the most salient dimensions, while a reconstruction loss $\mathcal{L}_{\text{clip\_rec}} = \|t_{\text{clip}} - \hat{t}_{\text{clip}}\|_2$ ensures the preservation of essential semantic information. This sparsity encourages each active unit in the sparse activation vector s to correspond to a distinct and interpretable semantic factor.

3.3 DICE

Given cached LLM embeddings $l_i \in \mathbb{R}^{d_{llm}}$ for each class c_i and an input image embedding $f \in \mathbb{R}^d$, DICE constructs a unified context through two complementary modules (see Fig. 2).

Class context module. The class context module captures general class-level semantics by leveraging cached LLM embeddings. For each class c_i, we retrieve LLM embeddings $l_i \in \mathbb{R}^{d_{llm}}$, which are projected into the CLIP embedding space via a learnable encoder. These yield the class-level embeddings:

$$W_{class} = E_{proj}\left(l\right), \quad W_{class} \in \mathbb{R}^{C \times d} \tag{5}$$

Instance context module. To extract instance-level visual semantics, the instance context module builds on the SAE, which decomposes concept-like features from LLM-derived class embeddings and image features. Since CLIP aligns image and text embeddings in a joint embedding space [45, 47], we assume that the SAE trained on text embeddings can be applied to image embeddings as well.

Given an LLM-derived class embedding $l \in \mathbb{R}^{C \times d_{\text{llm}}}$ and an image embedding $f \in \mathbb{R}^d$, we first project l into the CLIP space using a learnable encoder E_{proj} to obtain t_{llm}. We then subtract a learned bias b_{pre} from each input and pass both through a shared SAE encoder E_{sae} to compute the class and instance activations $(s_{\text{class}}, s_{\text{inst}})$. The co-activation score s_{coact} is then computed via an element-wise summation (denoted by $\oplus$) of the two activations.

$$\begin{gathered} s_{\text{class}} = E_{\text{sae}}(t_{\text{llm}} - b_{\text{pre}}), \quad s_{\text{inst}} \ = E_{\text{sae}}(f - b_{\text{pre}}), \\ s_{\text{coact}} = s_{\text{class}} \oplus s_{\text{inst}}. \end{gathered} \tag{6}$$

The co-activation score highlights concept dimensions that are jointly activated by both the image and its associated class. Since sparse activation values act as concept coefficients, addition emphasizes concepts supported by both representations. Based on this score, we select the top-k activated indices from

s_{coact}, as described in Eq. (3), and retrieve the corresponding concept vectors v_i from the shared concept dictionary defined by the SAE decoder weight D_{sae}:

$$W_{\text{inst}} = \{D_{\text{sae}}[i] \mid i \in \text{Top-}k(s_{\text{coact}})\}, \quad W_{\text{inst}} \in \mathbb{R}^{C\times k\times d} \tag{7}$$

This yields the instance-level representation W_{inst}, composed of the top-k concept vectors v that are semantically aligned with both the image and its associated class. This supports more precise and interpretable prompt construction.

Unified context injection. Finally, DICE constructs a unified context uc to integrate both general class-level semantics and instance-specific visual details. The outputs from the class context module and the instance context module are combined via element-wise summation and subsequently passed through a lightweight meta-network $\mathcal{M}$ to produce the final unified context:

$$uc = \mathcal{M}(W_{\text{class}} \oplus W_{\text{inst}}), \quad uc \in \mathbb{R}^{C\times k\times d} \tag{8}$$

We inject uc into the text encoder $G(x)$, which consists of stacked blocks $g_i = G_i(g_{i-1})$. Specifically, we replace the prompt tokens at the 8th layer, following prior work [50]. The resulting input sequence becomes $g_i = \{w_{\text{SOS}},\ uc_1,\ uc_2,\ \cdots,\ c_i,\ w_{\text{EOS}}\}$. This injection enables the model to integrate class-level and instance-level information within the prompt representation.

Training. To enable adaptation while preserving generalization from pre-trained CLIP and LLM representations, DICE is trained using a composite loss that combines the cross-entropy loss with two auxiliary terms, computed over a batch of N training examples.

The first auxiliary term, the knowledge-guided consistency loss, encourages each task-adapted DICE embedding $t^{(i)}_{\text{dice}}$ to remain close to its corresponding zero-shot CLIP embedding $t^{(i)}_{\text{clip}}$:

$$\mathcal{L}_{\text{kg}} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \cos(t^{(i)}_{\text{clip}},\ t^{(i)}_{\text{dice}})\right) \tag{9}$$

This loss serves as a regularizer to reduce overfitting and retain the generalization capability of the pretrained CLIP text encoder. The second auxiliary term, the LLM reconstruction loss (defined in Eq. (4)), ensures that the semantics of the original class embedding $l^{(i)}$ are preserved after projection and reconstruction:

$$\mathcal{L}_{\text{llm\_rec}} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \cos(l^{(i)},\ \hat{l}^{(i)})\right) \tag{10}$$

where $\hat{l}^{(i)} = D_{\text{proj}}(t^{(i)}_{\text{llm}})$ is the reconstruction of the projected LLM embedding.

The final training loss is a weighted sum of all loss components:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \lambda_{\text{kg}} \cdot \mathcal{L}_{\text{kg}} + \lambda_{\text{llm}} \cdot \mathcal{L}_{\text{llm\_rec}} \tag{11}$$

where λ_{kg} and λ_{llm} control the strength of each auxiliary term. These components work jointly to guide the model toward effective task adaptation while maintaining semantic alignment with the pre-trained representations.

4 Experiments

4.1 Experiment Setup

Datasets. We evaluate on 11 diverse datasets covering general and fine-grained visual recognition, including ImageNet [11], Caltech101 [14], SUN397 [48], UCF101 [42], EuroSAT [19], DTD [8], Stanford Cars [24], FGVC Aircraft [30], Food101 [4], Oxford Flowers [32], and Oxford Pets [34], following CoOp's train/test splits.
Baselines. We conducted comparisons with commonly used few-shot recognition methods, including CoOp [52], CoCoOp [51], MaPLe [22], TCP [50], and PSRC [23]. We further compare DICE with recent LLM-based methods, including HPT [46], CoPrompt [39], and KART [20].
Implementation details. Following the CoOp baseline, all experiments were conducted using CLIP ViT-B/16, with the number of learnable prompt tokens fixed to 4. More details are provided in Appendix.

4.2 Base-to-Novel Generalization

We evaluate base-to-novel generalization under a 16-shot setting across 11 benchmark datasets to assess a model's ability to recognize unseen categories. As shown in Tab. 1, DICE achieves the highest base/novel/H accuracies (84.45%/ 76.77%/ 80.43%) among all single-textual-prompt (tp) methods, including CoOp, CoCoOp, KART, and TCP. This establishes a new state-of-the-art performance in the single-textual-prompt setting. DICE achieves an H of 80.43%, comparable to CoPrompt at 80.48% and higher than MaPLe at 76.65%, PSRC at 79.97%, and HPT at 80.23%. Despite relying solely on textual prompts, DICE rivals or even surpasses recent multimodal prompt (mp) methods.

Our method is also implemented in a plug-and-play manner and can be easily integrated into existing prompting frameworks. When applied to CoOp, CoCoOp, and PSRC, DICE improves H scores by +5.79, +1.82, and +1.34, respectively. CoOp *w*/DICE and CoCoOp *w*/DICE even outperform the multi-prompt method MaPLe, despite using only a single textual prompt. Furthermore, PSRC *w*/DICE yields the highest overall H of 81.31%, surpassing all LLM-based and multi-modal prompt tuning methods.

DICE shows the largest gains on StanfordCars and Aircraft, where fine-grained recognition is critical. These results confirm that DICE better captures subtle visual details and aligns them with class semantics.

4.3 Cross-Dataset Evaluation

We evaluated cross-dataset generalization by training on a source dataset and testing on unseen targets without fine-tuning. According to Tab. 2, the highest average accuracy achieved on the source dataset was 72.03%, obtained by applying DICE to CoOp. Notably, both CoOp and CoCoOp equipped with DICE consistently outperform other methods on a wide range of target datasets, including those with significantly different distributions from the source. These

Table 1: Comparison of base-to-novel generalization. tp: text prompt tuning, mp: multi-modal prompt tuning. H: harmonic mean. The best overall result is in **bold**, and the best among tp-based methods is underlined.

Method	Type	**Average**			ImageNet [11]			Caltech101 [14]			OxfordPets [34]		
		Base	Novel	H	Base	Novel	H	Base	Novel	H	Base	Novel	H
CoOp [52]	tp	82.69	63.22	71.66	76.42	67.88	71.90	98.00	89.81	93.73	93.67	95.29	94.47
CoCoOp [51]	tp	80.47	71.69	75.83	75.98	70.43	73.10	97.96	93.81	95.84	95.20	97.69	96.43
KART [20]	tp	78.42	70.52	74.26	71.10	65.20	68.02	97.10	93.53	95.28	93.13	96.53	94.80
TCP [50]	tp	84.13	75.36	79.51	77.27	69.87	73.38	98.23	94.67	96.42	94.67	97.20	95.92
MaPLe [22]	mp	81.54	72.30	76.65	75.40	70.32	72.77	98.27	93.23	95.68	95.43	97.83	96.62
PSRC [23]	mp	84.26	76.10	79.97	77.60	70.73	74.01	98.10	94.03	96.02	95.33	97.30	96.30
HPT [46]	mp	84.32	76.86	80.23	**77.95**	70.74	74.17	98.37	**94.98**	**96.65**	**95.78**	97.65	96.71
CoPrompt [39]	mp	84.00	77.23	80.48	77.67	**71.27**	**74.33**	98.27	94.90	96.56	95.67	**98.10**	**96.87**
DICE	tp	84.45	76.77	80.43	76.87	70.07	73.31	**98.57**	94.33	96.40	94.80	97.37	96.07
CoOp *w*/DICE	tp	83.09	72.54	77.45	76.17	70.90	73.44	98.37	94.30	96.29	94.10	96.97	95.51
CoCoOp *w*/DICE	tp	82.70	73.19	77.65	76.30	69.23	72.59	98.10	93.87	95.94	95.27	97.47	96.36
PSRC *w*/DICE	mp	**85.76**	**77.29**	**81.31**	77.90	70.90	74.24	98.53	94.70	96.58	95.53	97.70	96.60

Method	Type	StanfordCars [24]			Flowers [32]			Food101 [4]			Aircraft [30]		
		Base	Novel	H	Base	Novel	H	Base	Novel	H	Base	Novel	H
CoOp [52]	tp	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 [51]	tp	70.49	73.59	72.01	94.87	71.75	81.71	90.70	91.29	90.99	33.41	23.71	27.74
KART [20]	tp	69.47	66.20	67.80	95.00	71.20	81.40	86.13	87.06	86.59	29.67	28.73	29.19
TCP [50]	tp	80.80	74.13	77.32	97.73	75.57	85.23	90.57	91.37	90.97	41.97	34.43	37.83
MaPLe [22]	mp	74.70	71.20	72.91	97.70	68.68	80.66	90.30	88.57	89.43	36.90	34.13	35.46
PSRC [23]	mp	78.27	74.97	76.58	98.07	76.50	85.95	90.67	91.53	91.10	42.73	37.87	40.15
HPT [46]	mp	76.95	74.23	75.57	98.17	**78.37**	**87.16**	90.46	91.57	91.01	42.68	38.13	40.28
CoPrompt [39]	mp	76.97	74.40	75.66	97.27	76.60	85.71	90.73	**92.07**	91.40	40.20	39.33	39.76
DICE	tp	80.67	**75.57**	78.04	98.07	77.00	86.27	90.83	91.63	91.23	45.77	38.07	41.56
CoOp *w*/DICE	tp	77.80	72.60	75.11	97.97	71.97	82.98	89.10	90.55	89.82	42.97	34.33	38.17
CoCoOp *w*/DICE	tp	74.57	74.00	74.28	97.27	73.53	83.75	90.40	91.20	90.80	41.77	33.97	37.47
PSRC *w*/DICE	mp	**81.93**	74.87	**78.24**	**98.40**	77.07	86.44	**93.73**	90.50	**92.09**	**48.77**	**39.37**	**43.57**

Method	Type	SUN397 [48]			DTD [8]			EuroSAT [19]			UCF101 [42]		
		Base	Novel	H	Base	Novel	H	Base	Novel	H	Base	Novel	H
CoOp [52]	tp	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 [51]	tp	79.74	76.86	78.27	77.01	56.00	64.85	87.49	66.00	75.24	85.23	71.97	78.04
KART [20]	tp	79.40	74.33	76.78	75.97	58.30	65.97	84.80	67.57	75.21	80.83	67.10	73.33
TCP [50]	tp	82.63	78.20	80.35	82.77	58.07	68.25	91.63	74.73	82.32	87.13	80.77	83.83
MaPLe [22]	mp	78.47	76.93	77.69	80.67	56.48	66.44	83.90	66.00	73.88	85.23	71.97	78.04
PSRC [23]	mp	82.67	78.47	80.52	83.37	62.97	71.75	92.90	73.90	82.32	87.10	78.80	82.74
HPT [46]	mp	82.57	**79.26**	**80.88**	**83.84**	63.33	72.16	94.24	77.12	84.82	86.52	80.06	83.16
CoPrompt [39]	mp	82.63	80.03	81.31	83.13	64.73	72.79	94.60	78.57	85.84	86.90	79.57	83.07
DICE	tp	82.73	78.23	80.42	82.30	61.53	70.42	91.83	78.53	84.66	86.53	**82.13**	**84.28**
CoOp *w*/DICE	tp	80.73	74.00	77.22	81.37	55.67	66.11	91.33	63.07	74.61	84.33	73.80	78.71
CoCoOp *w*/DICE	tp	81.53	76.17	78.76	80.13	55.80	65.79	90.20	65.03	75.57	84.13	74.77	79.17
PSRC *w*/DICE	mp	**82.95**	78.77	80.81	83.63	**65.27**	**73.32**	**94.70**	**80.57**	**87.06**	**87.30**	80.53	83.78

results indicate that DICE enhances cross-domain generalization by dynamically leveraging class- and instance-level knowledge from LLMs during inference.

4.4 Few-shot Classification

To evaluate the generalization ability of our method, we performed few-shot classification experiments on 11 benchmark datasets. As shown in Tab. 3, the proposed model, when integrated in a plug-and-play fashion, consistently enhanced performance across all datasets. These results suggest that leveraging

Table 2: Comparison of cross-dataset benchmark evaluation with existing methods. The best overall result is in **bold**.

Method	Type	ImageNet	Caltech	Pets	Cars	Flowers	Food	Aircraft	SUN397	DTD	EuroSAT	UCF	Avg
CoOp	tp	71.51	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
CoCoOp	tp	71.02	94.43	90.14	65.32	71.88	86.06	22.94	67.36	45.73	45.37	68.21	65.74
MaPLe	mp	70.72	93.53	90.49	65.57	72.23	86.20	24.74	67.01	46.49	48.06	68.69	66.30
PSRC	mp	71.27	93.60	90.25	65.70	70.25	86.15	23.90	67.10	46.87	45.50	68.75	65.81
TCP	tp	71.40	93.97	91.25	64.69	71.21	86.69	23.45	67.15	44.35	51.45	68.73	66.29
HPT	mp	71.72	94.20	**92.63**	**66.33**	**74.84**	86.21	25.68	68.75	50.87	47.36	**70.50**	67.73
CoPrompt	mp	70.80	94.50	90.73	65.67	72.30	86.43	24.00	67.57	47.07	51.90	69.73	67.00
CoOp w/DICE	tp	**72.03**	93.03	89.50	64.37	70.90	85.57	20.40	66.00	42.90	49.50	67.70	64.99
CoCoOp w/DICE	tp	71.76	**96.13**	90.23	59.57	70.73	**88.93**	**25.83**	**71.25**	**50.93**	**57.60**	70.20	**68.14**

Table 3: Comparison of few-shot learning with varying numbers of shots.

	Shots				
Method	**1**	**2**	**4**	**8**	**16**
CoOp	67.56	64.78	71.61	76.09	79.89
CoCoOp	66.79	67.65	71.21	72.96	74.90
PSRC	72.32	75.29	78.06	80.69	82.87
CoPrompt[1]	68.08	71.33	74.51	77.75	79.27
HPT[1]	69.72	72.49	76.19	78.53	81.15
CoOp w/DICE	68.93	72.33	75.18	77.70	80.96
CoCoOp w/DICE	66.92	70.52	72.68	75.25	77.97
PSRC w/DICE	**72.94**	**75.90**	**79.01**	**81.19**	**83.21**

both instance- and class-level knowledge improves generalization performance in few-shot settings.

4.5 Disentanglement

Building upon prior findings that SAEs promote semantic separation by learning a sparse activation vector s that responds to meaningful concepts, we define disentanglement as the emergence of interpretable and distinct concept units within the latent space. To examine whether such disentangled representations actually emerge, we design experiments following previous works [3, 9, 29]. For this purpose, we first collected candidate textual concepts from WordNet and matched each latent dimension of the SAE with its most strongly activated concept, thereby forming a semantic reference set that enables systematic interpretation of the learned latent space.

Disentangled Sample Analysis. This experiment aims to examine whether the SAE captures instance-specific semantic variations through disentangled latent activations. To this end, we analyzed the most activated latent dimension from each image's sparse representation s_{coact} and identified its underlying semantic concept using a WordNet-based reference. As shown in Fig. 3, meaningful and diverse visual concepts emerged (e.g., freckled for skin, mozzarella for food).

[1] Few-shot results are obtained by our re-implementation, as the original paper did not include such experiments.

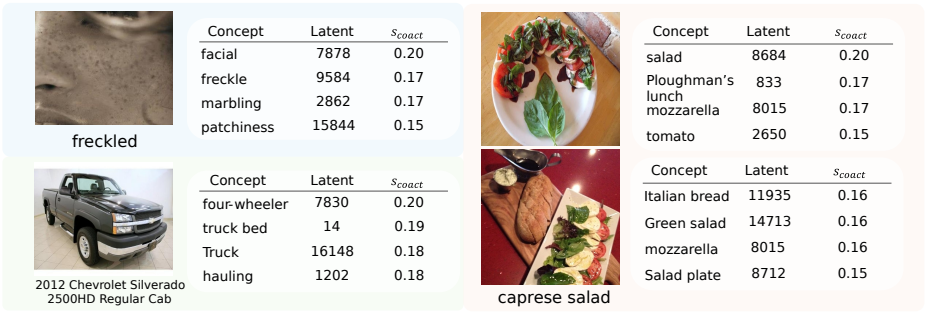


Fig. 3: Examples of disentangled latent interpretation on DTD (top left), StanfordCars (bottom left), and Food101 (right).

Table 4: Class-wise latent activation analysis.

Dataset	Class	Latent	Act.	Ref.
StanfordCars	2007 Chevrolet Corvette Ron Fellows Edition Z06	4910	0.1968	edge
		6382	0.1885	dashboard
		16351	0.1715	chair
		12769	0.1503	sports car
Caltech101	elephant	10579	0.1244	elephant
		4544	0.1187	tusker
		10851	0.1165	African elephant
		724	0.1052	big head
UCF101	Apply Eye Makeup	15299	0.1893	whiteface
		4	0.1869	eyelash
		4440	0.1848	front
		11337	0.1785	eyeliner

Table 5: SAE statistics on ImageNet.

Metric	Text	Image
MSE	0.63	0.99
Log Sparsity Avg	-0.034	-0.036
Entropy Avg	1.700	2.868

Table 6: Ablation on context modules.

Class	Inst	Base	Novel	H
✓		84.28	75.78	79.80
	✓	82.52	74.34	78.22
✓	✓	**84.45**	**76.77**	**80.43**

Even within the same class, different images emphasized distinct but related concepts, showing that the SAE captures subtle intra-class variations and produces an interpretable latent space.

Class-wise activation analysis. This experiment aims to investigate how the SAE organizes semantic concepts at the class level. To this end, we grouped images by class and identified the latent dimensions that were most frequently and strongly activated within each sparse representation s_{coact}. As shown in Tab. 4, the corresponding semantic concepts—such as *dashboard* or *sports car*—consistently aligned with their visual classes across distinct latent dimensions. Most classes selectively activated semantically related latent concepts, providing quantitative evidence that the SAE effectively captures class-level semantic structures within its latent space.

Activation Statistics of SAE. Following [5], we evaluate the Top-k SAE on ImageNet (Tab. 5). The mean squared error (MSE) reflects reconstruction fidelity, where lower values indicate better semantic preservation. The average log sparsity was –0.034 for text and –0.036 for image, suggesting that most neurons remained inactive and confirming the effectiveness of the sparsity constraint. The label entropy was 1.700 for text and 2.868 for image, indicating how selectively

Table 7: Intra/inter-class similarity.

	s_{coact}		s_{inst}	
Dataset	Intra ↑	Inter ↓	Intra ↑	Inter ↓
FGVCAircraft	0.14	0.18	0.07	0.62
Food101	0.12	0.09	0.04	0.41
OxfordPets	0.14	0.11	0.05	0.45

Table 8: Base–novel class correlation.

Dataset	Correlation
Caltech101	0.61
EuroSAT	0.40
StanfordCars	0.77

Table 9: Effect of LLM vs. CLIP.

Method	Base	Novel	H
LLM	84.45	76.77	80.43
CLIP	83.75	75.77	79.56
Δ (LLM-CLIP)	**+0.70**	**+1.00**	**+0.87**

Table 10: Effect of SAE.

Method	Base	Novel	H
w/ SAE	84.45	76.77	80.43
w/o SAE	81.31	73.82	77.38
Δ (*w*-*w/o*)	**+3.14**	**+2.95**	**+3.05**

each latent responds to class labels. Lower values indicate more class-specific activation, while higher values reflect broader, less discriminative responses. These results demonstrate that the SAE learns sparse, semantically preserved, and interpretable latent representations across modalities. Additional metric details can be found in Appendix.

4.6 Ablation Study

The synergistic effect of class and instance context. To assess the contribution of each component, we ablate the class and instance context modules. As shown in Tab. 6, removing either branch results in a performance drop. The full model achieves 80.43%, outperforming the class-only and instance-only variants by +0.63% and +2.21%, respectively.

Co-activation score effect. We form a joint representation by adding the image and class latent vectors, which accentuates concepts that are simultaneously activated by both. To validate that this co-activation score encourages the model to learn more discriminative concepts, we perform intra- and inter-class similarity experiments. As shown in Tab. 7, intra-class samples become closer and inter-class samples become further under s_{coact} compared to s_{inst}, indicating that the learned concepts are more discriminative and well guided.

Concept Selection Across Base and Novel Classes. To examine how generalization to novel classes arises from recombining the concept co-activation patterns learned on base classes, we measure the Pearson correlation between base- and novel-class concept-selection distributions over the frozen SAE dictionary. As shown in Tab. 8, the correlation decreases as visual diversity increases: StanfordCars shows the highest value at 0.77, Caltech101 follows with 0.61, and EuroSAT records the lowest at 0.40. These results suggest that novel class recognition emerges from recombining concepts within a fixed dictionary according to the concept selection patterns learned on base classes. Despite lower correlation, performance improves, indicating generalization through concept recombination.

Effect of LLM Embeddings. To verify that the performance improvement does not primarily depend on the use of LLM embeddings but rather arises from

Table 11: Analysis of Top-k and hidden dimension.

Top-k	Set	Hidden dimension		
		α=16	α=32	α=64
256	Base	84.36	84.32	84.48
	Novel	75.58	76.68	76.39
	H	79.73	80.36	80.23
512	Base	84.36	**84.45**	84.58
	Novel	76.36	**76.77**	76.31
	H	80.16	**80.43**	80.23
1024	Base	84.36	84.59	84.53
	Novel	76.39	76.47	76.36
	H	80.25	80.33	80.16

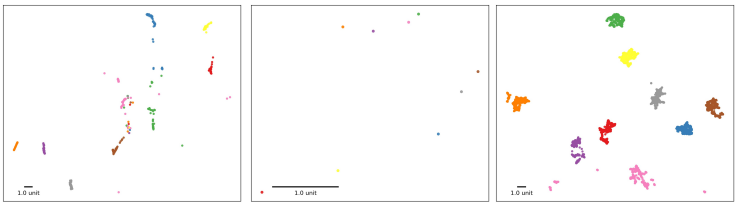


Fig. 4: Comparison of UMAP between LLM-based prompt tunings and ours.

the structural design of the proposed framework, we replace the LLM embeddings with CLIP text embeddings while keeping the overall architecture unchanged (Tab. 9). As a result, the harmonic mean (H) decreases by only 0.87%, with a 1.00% drop observed on Novel classes. This suggests that the performance gain stems from the structural properties of the proposed framework. Nevertheless, the richer semantic representations provided by LLMs offer additional benefits for generalization to unseen classes.

Effect of Sparse Autoencoder. To evaluate the structural contribution of the SAE, we replaced it with an MLP of the same parameter size and compared the performance (Tab. 10). As a result, the harmonic mean (H) drops by 3.05%. Consistent performance degradation was observed on both Base and Novel splits. This indicates that the improvement is not merely due to increased parameter capacity, but stems from the structural properties of the SAE, which enable sparse concept disentanglement and explicit concept selection.

SAE Latent Dimension Ablation Table 11 presents an ablation study on the number of top-k activations and the hidden dimension size d_{hid} of the SAE. As defined in Eq. (2), the hidden size is set by scaling the input dimension d with a factor $\alpha \in \{16, 32, 64\}$, which controls the model's representation capacity. Notably, the configuration with $k = 512$ and $\alpha = 32$ consistently yields slight improvements, indicating a favorable trade-off between sparsity and capacity.

Comparison of UMAP. UMAP visualizations on the FGVC-Aircraft are presented in Fig. 4. ZeroshotCLIP embeddings are scattered with unclear class

boundaries. CoPrompt forms partially separated clusters, but its representation space remains limited in scale with low variance. In contrast, DICE produces well-separated clusters with clear inter-class boundaries, indicating that it captures meaningful instance-level concepts and learns more discriminative and generalizable visual representations.

5 Conclusion

In this work, we show that existing LLM-based prompt tuning methods fail to capture instance-level semantics, limiting their generalization to unseen classes. To address this limitation, we propose DICE, a new framework that combines class-level priors with instance-specific semantics disentangled by a sparse autoencoder. To the best of our knowledge, we are the first to repurpose a sparse autoencoder as a concept dictionary, without relying on any hand-crafted or predefined concepts. This design builds prompts from explicit concept units, making the prompting process more interpretable and easily compatible with existing frameworks. Our results indicate that DICE improves both interpretability and generalization in VLMs, extending their capability beyond few-shot recognition.

Acknowledgements

This work was supported by Korea Internet & Security Agency(KISA) grant funded by the Korea government(PIPC) (No. RS-2026-25530348, Development of Evaluation Technologies for the Safety and Utility of Unstructured Synthetic Data). The first author would like to thank their family for their constant support, and Tori and Yeoni for their endless comfort and companionship throughout this journey.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems **35**, 23716–23736 (2022) 1
2. Balkus, S.V., Yan, D.: Improving short text classification with augmented data using gpt-3. Natural Language Engineering **30**(5), 943–972 (2024) 6, 3
3. Bhalla, U., Oesterling, A., Srinivas, S., Calmon, F., Lakkaraju, H.: Interpreting clip with sparse linear concept embeddings (splice). Advances in Neural Information Processing Systems **37**, 84298–84328 (2024) 4, 11
4. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13. pp. 446–461. Springer (2014) 9, 10, 8

5. Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J.E., Hume, T., Carter, S., Henighan, T., Olah, C.: Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread (2023), https://transformer-circuits.pub/2023/monosemantic-features/index.html 4, 12
6. Bussmann, B., Leask, P., Nanda, N.: Batchtopk sparse autoencoders. arXiv preprint arXiv:2412.06410 (2024) 4
7. Cammarata, N., Goh, G., Carter, S., Voss, C., Schubert, L., Olah, C.: Curve circuits. Distill (2021). https://doi.org/10.23915/distill.00024.006, https://distill.pub/2020/circuits/curve-circuits 4
8. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3606–3613 (2014) 9, 10
9. Cunningham, H., Ewart, A., Riggs, L., Huben, R., Sharkey, L.: Sparse autoencoders find highly interpretable features in language models. arXiv preprint arXiv:2309.08600 (2023) 4, 6, 11
10. Cywiński, B., Deja, K.: SAeuron: Interpretable concept unlearning in diffusion models with sparse autoencoders. In: ICLR 2025 Workshop: XAI4Science: From Understanding Model Behavior to Discovering New Scientific Knowledge (2025), https://openreview.net/forum?id=HFCaWGWEzi 4
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 9, 10, 8
12. Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al.: Toy models of superposition. arXiv preprint arXiv:2209.10652 (2022) 4
13. Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al.: A mathematical framework for transformer circuits. Transformer Circuits Thread **1**(1), 12 (2021) 4
14. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: 2004 conference on computer vision and pattern recognition workshop. pp. 178–178. IEEE (2004) 9, 10, 8
15. Fel, T., Boutin, V., Béthune, L., Cadène, R., Moayeri, M., Andéol, L., Chalvidal, M., Serre, T.: A holistic approach to unifying automatic concept extraction and concept importance estimation. Advances in Neural Information Processing Systems **36**, 54805–54818 (2023) 4
16. Fellbaum, C.: WordNet: An electronic lexical database. MIT press (1998) 6
17. Gao, L., la Tour, T.D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., Wu, J.: Scaling and evaluating sparse autoencoders. arXiv preprint arXiv:2406.04093 (2024) 4
18. Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., Qiao, Y.: Clip-adapter: Better vision-language models with feature adapters. International journal of computer vision **132**(2), 581–595 (2024) 5
19. Helber, P., Bischke, B., Dengel, A., Borth, D.: 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) 9, 10, 8

20. Kan, B., Wang, T., Lu, W., Zhen, X., Guan, W., Zheng, F.: Knowledge-aware prompt tuning for generalizable vision-language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15670–15680 (2023) 1, 2, 4, 9, 10
21. Khattak, M.U., Naeem, M.F., Naseer, M., Van Gool, L., Tombari, F.: Learning to prompt with text only supervision for vision-language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 4230–4238 (2025) 4
22. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19113–19122 (2023) 4, 9, 10
23. Khattak, M.U., Wasim, S.T., Naseer, M., Khan, S., Yang, M.H., Khan, F.S.: Self-regulating prompts: Foundational model adaptation without forgetting. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 15190–15200 (2023) 9, 10
24. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 554–561 (2013) 9, 10, 8
25. Lee, S., Shin, K., Park, J.H.: Weighted multi-prompt learning with description-free large language model distillation. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=NDLmZZWATc 4
26. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021) 4
27. Li, S., Liu, F., Hao, Z., Wang, X., Li, L., Liu, X., Chen, P., Ma, W.: Logits deconfusion with clip for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 25411–25421 (June 2025) 5
28. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021) 4
29. Lim, H., Choi, J., Choo, J., Schneider, S.: Sparse autoencoders reveal selective remapping of visual concepts during adaptation. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=imTO3YXlG2 11
30. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013) 3, 9, 10, 8
31. Makhzani, A., Frey, B.: K-sparse autoencoders. arXiv preprint arXiv:1312.5663 (2013) 6
32. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian conference on computer vision, graphics & image processing. pp. 722–729. IEEE (2008) 3, 9, 10, 8
33. O'Neill, C., Ye, C., Iyer, K., Wu, J.F.: Disentangling dense embeddings with sparse autoencoders. arXiv preprint arXiv:2408.00657 (2024) 4
34. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3498–3505. IEEE (2012) 9, 10, 8
35. Pratt, S., Covert, I., Liu, R., Farhadi, A.: What does a platypus look like? generating customized prompts for zero-shot image classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15691–15701 (2023) 1, 2, 4

36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021) 1
37. Rao, S., Mahajan, S., Böhle, M., Schiele, B.: Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In: European Conference on Computer Vision. pp. 444–461. Springer (2024) 4
38. Roth, K., Kim, J.M., Koepke, A., Vinyals, O., Schmid, C., Akata, Z.: Waffling around for performance: Visual classification with random words and broad concepts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15746–15757 (2023) 1, 2
39. Roy, S., Etemad, A.: Consistency-guided prompt learning for vision-language models. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=wsRXwlwx4w 1, 2, 4, 9, 10
40. Schrodi, S., Schur, J., Argus, M., Brox, T.: Concept bottleneck models without predefined concepts. arXiv preprint arXiv:2407.03921 (2024) 4
41. Sharkey, L., Braun, D., beren: Taking features out of superposition with sparse autoencoders. https://www.lesswrong.com/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition (December 2022), aI Alignment Forum. Accessed: June 26, 2026 4, 1
42. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012) 9, 10, 8
43. Tack, J., Lanchantin, J., Yu, J., Cohen, A., Kulikov, I., Lan, J., Hao, S., Tian, Y., Weston, J., Li, X.: Llm pretraining with continuous concepts. arXiv preprint arXiv:2502.08524 (2025) 4
44. Tian, X., Zou, S., Yang, Z., Zhang, J.: Argue: Attribute-guided prompt tuning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28578–28587 (2024) 4
45. Wang, J., Shao, W., Chen, M., Wu, C., Liu, Y., Wu, T., Zhang, K., Zhang, S., Chen, K., Luo, P.: Adapting llama decoder to vision transformer. arXiv preprint arXiv:2404.06773 (2024) 7
46. Wang, Y., Jiang, X., Cheng, D., Li, D., Zhao, C.: Learning hierarchical prompt with structured linguistic knowledge for vision-language models. In: Proceedings of the AAAI conference on artificial intelligence. vol. 38, pp. 5749–5757 (2024) 1, 2, 4, 9, 10
47. Wang, Z., Liang, J., He, R., Xu, N., Wang, Z., Tan, T.: Improving zero-shot generalization for clip with synthesized prompts. arXiv preprint arXiv:2307.07397 (2023) 7
48. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 3485–3492. IEEE (2010) 9, 10, 8
49. Yao, H., Zhang, R., Xu, C.: Visual-language prompt tuning with knowledge-guided context optimization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6757–6767 (2023) 4
50. Yao, H., Zhang, R., Xu, C.: Tcp: Textual-based class-aware prompt tuning for visual-language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23438–23448 (2024) 2, 3, 4, 8, 9, 10
51. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16816–16825 (2022) 1, 2, 3, 4, 9, 10

52. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022) 1, 4, 9, 10

Supplementary Material

A Pseudocode of DICE

In this section, we present the pseudocode of DICE (Algorithm 1), which can be seamlessly integrated into existing prompt modules.

B Implementation details

To ensure reproducibility, we fix the random seed and report the average performance over three runs with seeds $\{1, 2, 3\}$. During pretraining, the SAE, projection encoder, and decoder are jointly optimized for 30 epochs using the Adam optimizer with a learning rate of 0.002 and a batch size of 8. The hidden dimension of the SAE is set to $32 \times d$, and the sparsity constraint retains the top-$k = 512$ activations, following [41]. During prompt tuning, we select $k = 4$ concepts from the learned concept dictionary based on activation scores. The same optimizer and training schedule used in pretraining are applied. Most experiments use a batch size of 8, except for ImageNet and SUN397 where a batch size of 6 is used. The weighting coefficient λ_{llm} is fixed to 4.0 for all datasets. The coefficient λ_{kg} is tuned per dataset: 5.0 for StanfordCars, 6.0 for DTD, 10.0 for EuroSAT, 9.0 for SUN397, Flowers, and Food101, 12.0 for Caltech101, and 8.0 for the remaining datasets. In the plug-and-play setting, the knowledge-guided consistency loss is not applied. For all baseline models, we replace the original optimizer with Adam while keeping the learning rate, backbone architecture, prompt length, and number of training epochs identical to their original settings to ensure fair comparison.

C Limitations

DICE preserves the generalization ability of LLM-based class knowledge while leveraging image-specific details through prompt tuning. However, the method has one limitation. To generate instance-conditioned prompts, DICE requires a separate forward pass through the text encoder for each image. Similar to CoCoOp, this design can increase memory usage and slow down training when the batch size becomes large. Reducing this computational overhead and enabling more efficient generation of instance-conditioned prompts remains an important direction for future research.

D LLM Utilization Details

In this section, we describe how LLM embeddings are obtained. We use the embedding API provided by OpenAI to extract semantic representations for each

Algorithm 1: DICE

```python
class SparseAutoencoder(nn.Module):
    def __init__(self, clip_dim, hidden_dim, topk):
        super(SparseAutoencoder, self).__init__()
        self.encoder = nn.Linear(clip_dim, hidden_dim, bias=False)
        self.decoder = nn.Linear(hidden_dim, clip_dim, bias=False)
        self.pre_bias = nn.Parameter(torch.zeros(clip_dim))
        self.latent_bias = nn.Parameter(torch.zeros(hidden_dim))
        self.concept_dict = self.decoder.weight.t()
        self.topk = topk

    def concept_selection(self, pre_x):
        values, indices = torch.topk(pre_x, self.topk, dim=-1)
        return self.concept_dict[indices]

    def forward(self, f, t_llm):
        f = f - self.pre_bias
        t_llm = t_llm - self.pre_bias
        s_inst = self.encoder(f) + self.latent_bias
        s_class = self.encoder(t_llm) + self.latent_bias
        s_coact = s_inst.unsqueeze(1) + s_class.unsqueeze(0)
        return self.concept_selection(s_coact)

class ProjectAutoencoder(nn.Module):
    def __init__(self, llm_dim, clip_dim):
        super(ProjectAutoencoder, self).__init__()
        self.encoder = nn.Linear(llm_dim, clip_dim)
        self.decoder = nn.Linear(clip_dim, llm_dim)

    def forward(self, l):
        t_llm = self.encoder(l)
        l_hat = self.decoder(t_llm)
        return t_llm, l_hat

class DICE(nn.Module):
    def __init__(self, clip_dim, llm_dim, hidden_dim, topk):
        super(DICE, self).__init__()
        self.projector = ProjectAutoencoder(llm_dim, clip_dim)
        self.sae = SparseAutoencoder(clip_dim, hidden_dim, topk)
        self.meta_net = nn.Sequential(
            nn.Linear(4 * clip_dim, clip_dim),
            QuickGELU(),
            nn.Linear(clip_dim, 4 * clip_dim)
        )

    def forward(self, f, l):
        t_llm, l_hat = self.projector(l)
        concepts = self.sae(f, t_llm)
        uc = self.meta_net(t_llm.unsqueeze(0).unsqueeze(2) + concepts)
        return uc, l_hat
```

class name, specifically using the text-embedding-3 series models [2]. We use the class name itself ("class") as the textual input. The resulting embeddings are cached before training. During all experiments, we reuse the same embeddings to avoid the cost and variability of repeated LLM calls. This approach helps maintain consistency and improves the efficiency and reproducibility of our experimental setup.

E LLM Model Ablation

We conducted an ablation study on various LLM embedding models under a 16-shot setting across 11 benchmark datasets. For comparison, we selected three representative text embedding models provided by OpenAI: text-embedding-ada-002, text-embedding-3-small, and text-embedding-3-large.

Among them, text-embedding-ada-002 is the oldest version. It offers higher efficiency by producing 1536-dimensional embeddings, which lowers FLOPs compared to larger models. However, due to its older architecture, it may have limitations in terms of expressiveness and semantic precision compared to more recent models.

text-embedding-3-small is a lightweight model from the newer text-embedding-3 series, released in early 2024. It also produces 1536-dimensional vectors but shows significantly improved performance over ada-002 due to architectural and training improvements. This model is designed as a smaller and faster alternative to text-embedding-3-large, while still providing strong semantic representations at lower cost.

On the other hand, text-embedding-3-large is the most powerful model in the series. It generates 3072-dimensional embeddings and provides richer semantic information and more precise contextual understanding.

Table 12: Comparison of OpenAI Embedding Models

Models	Base	Novel	H
`text-embedding-ada-002`	83.11	75.14	78.92
`text-embedding-3-small`	84.52	75.86	79.95
`text-embedding-3-large`	84.45	76.77	80.43

As shown in Tab. 12, text-embedding-3-large achieved the best overall performance among the three embedding models, with a harmonic mean (H) score of 80.43.

F Detailed Analysis of SAE Statistics

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\|x_i - \hat{x}_i\|_2^2, \tag{12}$$

$$\text{LogSparsity} = \frac{1}{D}\sum_{j=1}^{D}\log\left(1 - s_j\right), \tag{13}$$

$$\begin{aligned}\text{LabelEntropy} &= -\sum_{c\in\mathcal{C}}\left(\text{prob}_c \log \text{prob}_c\right), \\ \text{prob}_c &= \frac{\text{sum}_c}{\sum_{c\in\mathcal{C}}\text{sum}_c}\end{aligned} \tag{14}$$

- **Mean Squared Error (MSE).** MSE measures the reconstruction fidelity between the input x_i and its reconstruction $\hat{x}_i$. A lower value indicates that the autoencoder preserves semantic information more accurately, while a higher value suggests loss of detail or incomplete reconstruction.
- **Log Sparsity.** LogSparsity quantifies the sparsity of latent activations using the mean neuron activation s_j. By computing $\log(1 - s_j)$, it emphasizes inactive neurons—values close to zero indicate that most neurons remain silent, while large negative values imply dense activations.
- **Label Entropy.** LabelEntropy captures how distinctly each neuron responds to different categories, offering a quantitative signal related to class-level separability. When the label entropy approaches zero, a neuron activates almost exclusively for a single or very small set of classes, indicating a highly class-specific and interpretable response pattern. As the entropy increases, the neuron's responses become distributed across a broader range of categories, suggesting that multiple labels contribute to its activation and that the neuron is less specialized.

G Layer-wise Ablation Study

Figure 5 illustrates the variation of the harmonic mean (H) across layers. The performance gradually improves from lower layers and reaches its peak at layer L8 (80.43), then slightly declines toward higher layers. This trend indicates that DICE learns the most generalizable representations at intermediate feature levels within the network.

H Additional UMAP Visualizations

In Fig. 4 of the main paper, we showed UMAP visualizations only for the FGVC-Aircraft dataset. To provide a broader comparison, we selected 9 random classes from each dataset and visualized the UMAP of text embeddings from different methods. The results are shown in Fig. 6.

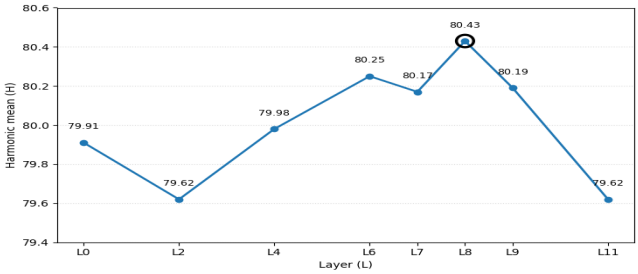


Fig. 5: Layer-wise performance

Table 13: Effect of the number of selected concepts k.

Topk	Base	Novel	H
2	84.48	75.84	79.93
8	84.57	76.12	80.12
4	**84.45**	**76.77**	**80.43**

Table 14: Comparison with adapter-based methods under few-shot learning.

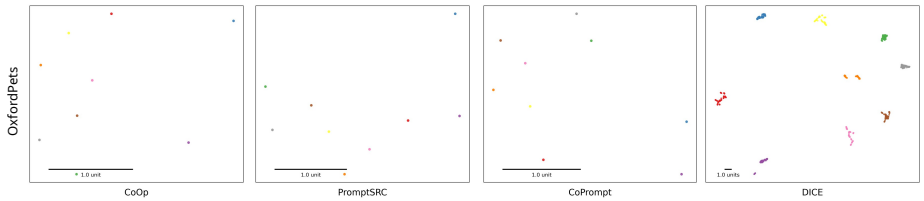
Methods	2-shot	4-shot	16-shot
CLIP-Adapter [18]	66.92	67.80	71.01
LDC [27]	74.17	77.12	82.17
PSRC w/DICE	**75.90**	**79.01**	**83.21**

Furthermore, when comparing the scale bars across methods, our model exhibits a noticeably wider spread in the embedding space. Because DICE incorporates instance-level information, embeddings from the same class also show meaningful intra-class variance. These observations indicate that DICE constructs a richer and more discriminative representation space than existing approaches.

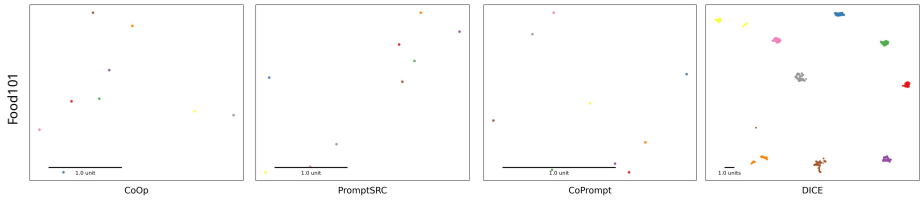
I Visualizations of Learned Concepts

This section examines whether each latent s_i of the activation score s captures a different semantic factor. To evaluate this, we collect the images and the WordNet-derived texts that most strongly activate each latent and check whether they form a coherent semantic concept.

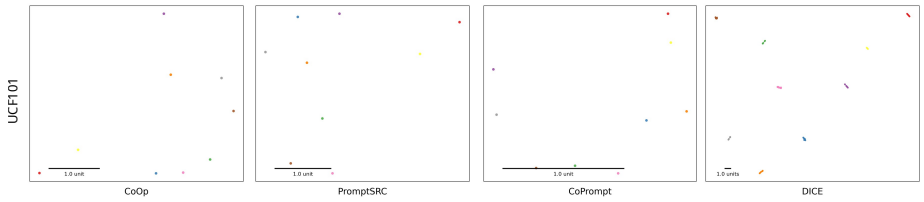
Figure 7 illustrates this process. It visualizes our definition of disentanglement as the restructuring of polysemantic embeddings into distinct and interpretable concept units. Each latent s_i selects a consistent group of images—such as horse-racing scenes or food items. These observations indicate that each latent functions as a meaningful concept axis, supporting the claim that the model has learned a disentangled representation.



(a) Pets dataset.



(b) Food dataset.



(c) UCF dataset.

Fig. 6: UMAP comparison of text embeddings from diverse methods

Table 15: Ablation on the weighting coefficient λ_{kg} with λ_{llm} fixed to 4.0.

λ_{kg}	ImageNet	Caltech	Pets	Cars	Flowers	Food	Aircraft	SUN397	DTD	EuroSAT	UCF	Avg
6	72.98	96.22	95.89	77.44	83.73	92.00	40.57	80.12	70.42	80.72	83.03	79.37
8	73.31	96.20	96.07	77.79	83.65	92.13	41.56	80.14	66.61	82.31	84.28	79.46
10	72.76	96.19	96.10	76.92	83.86	91.98	40.53	80.26	65.53	84.66	82.78	79.23

J Ablation on the Knowledge-Guided Consistency Loss Weight

We analyze the effect of the knowledge-guided consistency loss weight λ_{kg} defined in Eq. (11), while fixing the LLM reconstruction loss weight to $\lambda_{\text{llm}} = 4.0$. Specifically, λ_{kg} is varied among $\{6, 8, 10\}$. As shown in Tab. 15, the performance remains stable across different values of λ_{kg}, indicating that our method is not highly sensitive to this hyperparameter.

K Ablation on the Number of Selected Concepts

In Eq. (7), we select the top-k concept vectors v_i according to the co-activation score s_{coact}. We evaluate different values of k and report the average performance

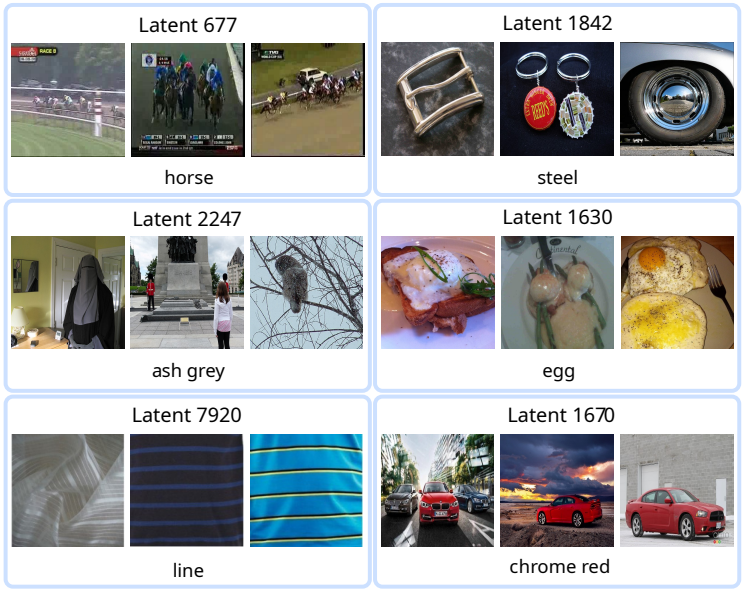


Fig. 7: Qualitative evaluation of the learned concept set.

across 11 datasets to study the impact of concept selection. The results are shown in Table 13.

L Comparison with Adapter-Based Methods

Table 14 reports few-shot learning results compared with existing adapter-based methods. We follow the experimental protocols of the corresponding papers and evaluate performance under the 2-shot, 4-shot, and 16-shot settings on 11 benchmark datasets. For a fair comparison, the backbone architecture, training schedule, and evaluation protocol are kept identical to the original implementations.

M Additional UMAP Visualizations

In Fig. 4 of the main paper, we showed UMAP visualizations only for the FGVC-Aircraft dataset. To provide a broader comparison, we selected 9 random classes from each dataset and visualized the UMAP of text embeddings from different methods. The results are shown in Fig. 6.

N Detailed Few-shot Learning Results

This table reports few-shot classification accuracy across 11 datasets under 1, 2, 4, 8 and 16-shot settings. Each baseline method (CoOp, CoCoOp, and PSRC) is

compared with its DICE-augmented counterpart to evaluate the contribution of instance-aware context modeling. The values in parentheses indicate the performance change relative to the corresponding baseline, with green denoting gains and red indicating drops. The results show that DICE generally improves performance across many settings, particularly on fine-grained datasets, as shown in Tab. 16.

O Overview of Datasets

In this section, we briefly describe the datasets used in our experiments. The selected datasets cover a diverse range of domains, including objects, scenes, fine-grained categories, and satellite imagery.

- **ImageNet** [11]: A large-scale dataset containing over 1 million images across 1,000 object categories. It is widely used for pretraining and benchmarking general visual recognition models.
- **Caltech101** [14]: A dataset of 9,146 images grouped into 101 object categories and a background class. It contains a moderate number of samples per class and is commonly used for fine-grained classification.
- **Oxford Pets** [34]: This dataset consists of 7,349 images of 37 breeds of cats and dogs, with roughly balanced class sizes. It includes both breed and segmentation labels.
- **Stanford Cars** [24]: A fine-grained dataset containing 16,185 images of 196 classes of cars, defined by make, model, and year.
- **Oxford Flowers 102** [32]: Comprising 8,189 images of 102 flower categories, this dataset presents a fine-grained classification challenge with significant intra-class variation.
- **Food101** [4]: A dataset of 101 food categories, each with 1,000 images. It includes real-world food photos with noisy labels and high visual diversity.
- **FGVC Aircraft** [30]: Contains 10,000 images across 100 aircraft variants, focusing on fine-grained recognition of visually similar classes.
- **SUN397** [48]: A large-scale scene classification dataset consisting of 397 scene categories and over 100,000 images, covering indoor, outdoor, and natural environments.
- **EuroSAT** [19]: A dataset of 27,000 labeled satellite images from Sentinel-2, spanning 10 land use and land cover classes. It is used for remote sensing and geospatial analysis.
- **UCF101** [42]: A video action recognition dataset with 13,320 videos across 101 human action categories. In our work, we used extracted frame-level visual concepts.

Table 16: Few-shot accuracy on 11 datasets under 1, 2, 4, 8 and 16-shot settings. Parentheses show differences from the corresponding baselines.

Shot	Dataset	CoOp	CoOp *w*/DICE	CoCoOp	CoCoOp *w*/DICE	PSRC	PSRC *w*/DICE
1	ImageNet	66.33	69.70 (+3.37)	69.43	69.70 (+0.27)	68.13	66.90 (-1.23)
	SUN397	66.77	68.61 (+1.84)	68.33	68.50 (+0.17)	69.67	70.10 (+0.43)
	Cars	67.43	69.28 (+1.85)	67.22	65.17 (-2.05)	69.40	70.33 (+0.93)
	UCF101	71.23	72.30 (+1.07)	70.30	71.33 (+1.03)	74.80	75.87 (+1.07)
	Caltech	92.60	94.40 (+1.80)	93.83	86.50 (-7.33)	93.67	94.23 (+0.56)
	EuroSAT	54.93	44.60 (-10.33)	55.33	44.00 (-11.33)	73.13	75.53 (+2.40)
	Aircraft	21.37	29.40 (+8.03)	12.68	28.27 (+15.59)	27.67	30.03 (+2.36)
	Food	84.33	86.10 (+1.77)	85.65	86.33 (+0.68)	84.87	85.57 (+0.70)
	DTD	50.23	51.27 (+1.04)	48.54	50.23 (+1.69)	56.23	55.63 (-0.60)
	Flowers	77.53	82.37 (+4.84)	72.08	76.07 (+3.99)	85.93	86.60 (+0.67)
	Pets	90.37	90.20 (-0.17)	91.27	89.97 (-1.30)	92.00	91.53 (-0.47)
	Average	**67.56**	**68.93 (+1.37)**	**66.79**	**66.92 (+0.13)**	**72.32**	**72.94 (+0.62)**
2	ImageNet	67.1	70.0 (+2.9)	69.8	70.4 (+0.6)	69.8	70.5 (+0.7)
	SUN397	66.5	70.0 (+3.5)	69.0	70.7 (+1.7)	71.6	71.8 (+0.2)
	Cars	70.5	70.0 (-0.5)	68.4	68.7 (+0.3)	73.4	74.7 (+1.3)
	UCF101	73.4	75.8 (+2.4)	73.5	73.2 (-0.3)	78.5	78.6 (+0.1)
	Caltech	89.0	94.6 (+5.6)	94.8	93.9 (-0.9)	94.5	94.9 (+0.4)
	EuroSAT	62.0	64.6 (+2.6)	46.7	57.8 (+11.0)	79.4	82.1 (+2.7)
	Aircraft	26.4	31.1 (+4.7)	15.1	27.0 (+11.9)	31.7	32.1 (+0.4)
	Food	73.4	86.4 (+13.0)	86.2	86.5 (+0.3)	85.7	84.4 (-1.3)
	DTD	40.8	54.3 (+13.5)	52.2	52.2 (+0.0)	60.0	61.0 (+1.0)
	Flowers	85.1	87.9 (+2.8)	75.8	83.6 (+7.8)	91.2	92.2 (+1.0)
	Pets	58.4	90.9 (+32.5)	92.6	91.9 (-0.7)	92.5	92.6 (+0.1)
	Average	**64.8**	**72.3 (+7.5)**	**67.7**	**70.5 (+2.9)**	**75.3**	**75.9 (+0.6)**
4	ImageNet	68.7	70.7 (+2.0)	70.4	70.6 (+0.2)	71.1	71.5 (+0.4)
	SUN397	69.9	71.7 (+1.8)	70.2	72.0 (+1.8)	74.0	74.4 (+0.4)
	Cars	74.5	73.2 (-1.3)	69.4	70.9 (+1.5)	74.0	78.3 (+4.3)
	UCF101	77.1	78.7 (+1.6)	74.8	76.8 (+2.0)	81.6	81.4 (-0.2)
	Caltech	92.1	95.2 (+3.1)	95.0	94.4 (-0.6)	95.3	94.8 (-0.5)
	EuroSAT	77.1	71.3 (-5.8)	65.6	61.6 (-4.0)	86.3	88.7 (+2.4)
	Aircraft	32.3	34.3 (+2.0)	24.8	31.3 (+6.5)	37.5	38.3 (+0.8)
	Food	77.1	86.6 (+9.5)	86.9	86.7 (-0.2)	86.2	87.0 (+0.8)
	DTD	55.7	60.3 (+4.6)	55.0	56.5 (+1.5)	65.5	66.7 (+1.2)
	Flowers	92.4	91.5 (-0.9)	78.4	85.6 (+7.2)	93.9	94.6 (+0.7)
	Pets	71.2	93.2 (+22.0)	92.8	93.2 (+0.4)	93.4	93.5 (+0.1)
	Average	**71.6**	**75.2 (+3.6)**	**71.2**	**72.7 (+1.5)**	**78.1**	**79.0 (+0.9)**
8	ImageNet	70.6	71.2 (+0.6)	70.6	71.3 (+0.7)	72.3	72.1 (-0.2)
	SUN397	71.5	73.6 (+2.1)	70.8	73.1 (+2.3)	75.7	76.0 (+0.3)
	Cars	79.3	76.7 (-2.6)	70.4	73.1 (+2.7)	81.0	81.3 (+0.3)
	UCF101	80.2	80.6 (+0.4)	77.1	78.6 (+1.5)	84.3	84.2 (-0.1)
	Caltech	93.4	95.3 (+1.9)	95.0	94.9 (-0.1)	95.7	96.0 (+0.4)
	EuroSAT	84.4	77.6 (-6.8)	68.2	67.3 (-0.9)	88.8	91.7 (+2.9)
	Aircraft	39.4	38.1 (-1.3)	26.6	34.5 (+7.9)	43.3	44.2 (+0.9)
	Food	80.2	86.9 (+6.7)	87.0	87.2 (+0.2)	86.9	87.1 (+0.2)
	DTD	63.5	66.2 (+2.7)	58.9	63.6 (+4.7)	69.9	69.9 (+0.0)
	Flowers	96.1	95.3 (-0.8)	84.3	91.2 (+6.9)	96.3	96.7 (+0.4)
	Pets	78.4	93.4 (+15.0)	93.5	92.8 (-0.7)	93.5	93.8 (+0.3)
	Average	**76.1**	**77.7 (+1.6)**	**73.0**	**75.3 (+2.3)**	**80.7**	**81.2 (+0.5)**
16	ImageNet	71.87	72.13 (+0.26)	70.83	72.13 (+1.30)	73.17	73.40 (+0.23)
	SUN397	74.67	79.76 (+5.09)	72.15	75.00 (+2.85)	77.23	77.07 (-0.16)
	Cars	83.07	82.20 (-0.87)	71.57	76.60 (+5.03)	83.83	83.10 (-0.73)
	UCF101	82.23	83.57 (+1.34)	78.14	80.70 (+2.56)	86.47	87.07 (+0.60)
	Caltech	95.57	95.70 (+0.13)	95.16	93.50 (-1.66)	96.07	96.27 (+0.20)
	EuroSAT	84.93	84.70 (-0.23)	73.32	73.40 (+0.08)	92.43	93.37 (+0.94)
	Aircraft	43.40	44.00 (+0.60)	31.21	41.10 (+9.89)	50.83	51.77 (+0.94)
	Food	84.20	87.47 (+3.27)	87.25	87.40 (+0.15)	87.50	87.67 (+0.17)
	DTD	69.87	70.73 (+0.86)	63.04	68.30 (+5.26)	72.73	73.57 (+0.84)
	Flowers	97.07	96.83 (-0.24)	87.84	95.70 (+7.86)	97.60	97.83 (+0.23)
	Pets	91.87	93.47 (+1.60)	93.34	93.80 (+0.46)	93.67	94.20 (+0.53)
	Average	**79.89**	**80.96 (+1.07)**	**74.90**	**77.97 (+3.07)**	**82.87**	**83.21 (+0.34)**