On the Role of Mechanistic Interpretability for Vision-Language Prompt Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent advances in mechanistic interpretability of vision-language models (VLMs) such as CLIP propose using sparse autoencoders (SAEs) to discover monosemantic, human-understandable features that explain CLIP's internal representations. Existing work using SAEs to probe VLMs primarily focuses on post-hoc interpretability analysis. We posit that SAE-based interpretability methods are not just probing tools, but can also serve as meaningful training guides for adapting VLMs to downstream tasks. To this end, we propose IPL (Interpretability-Guided Prompt Learning), which leverages SAE decoders to extract interpretable concept directions, composes them into prompt tokens via a learnable attention selector, and injects the resulting tokens into both the vision and text encoder layers of CLIP for adaptation. We further study how prompt tokens obtained by probing vision-only, text-only, and unified concept directions from respective interpretability methods affect performance on downstream tasks. We perform extensive experiments across downstream settings such as base-to-novel generalization, domain generalization, cross-dataset transfer, and few-shot learning. While IPL using vision-only and text-only concept directions obtains decent gains, IPL with unified concept directions achieves the strongest results, outperforming prior prompt-learning methods over 15 datasets across all downstream settings. Code: https://anonymous.4open.science/r/IPL-34EF/

1 Introduction

Mechanistic interpretability (MI) research on vision-language models (VLMs) like CLIP [30] has made rapid progress in the past years. Sparse autoencoders (SAEs) [23, 32, 2] have emerged as a prominent tool in the MI toolkit and they have been shown to be able to decompose intermediate representations of VLMs into human-understandable concept directions: dictionaries of features that fire selectively on coherent semantic categories such as object parts, textures, or scene types, with explicit per-input activation magnitudes that quantify each concept's presence in any given image. These tools have given researchers a better picture of what CLIP has learned. So far, however, this growing toolbox has been used almost exclusively for post-hoc purposes like *interpreting* model behavior after the model has been trained. Our central research question is *can interpretability tools act as a training guide for adapting VLM's like CLIP to different downstream settings?*

We posit that this is exactly where mechanistic interpretability can be practically useful, and we ground the question in a concrete use case, i.e., ***prompt learning***. Prompt learning has emerged as the dominant parameter-efficient strategy for adapting CLIP to downstream tasks, with approaches like CoOp [45], MaPLe [20], and MMRL [15] introducing learnable prompt tokens at various layers of CLIP. While effective, these prompts are opaque continuous vectors with no human-interpretable semantics. Recent attempts to make prompts interpretable like CPL [42], ArGue [35], IntCoOp [14], and IPO [10], do so by importing concepts from *outside* CLIP: querying language

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

models or vision-language models for natural-language attributes and hard-coding them into the prompt. The interpretability is thus outsourced to *external* tools; it does not engage with the rich *internal* concept structure that mechanistic interpretability methods [23, 2, 32] have shown already exists *inside* CLIP itself.

This gap motivates our central proposal. We treat SAE-derived concept dictionaries not as analytical artifacts/probes but as a semantically grounded concept directions from which prompt tokens can be composed. Rather than asking a large language model (LLM) to provide semantic attributes for prompt tokens, we leverage CLIP's own interpretable features (the same features that MI researchers have already identified, labeled, and inspected) to guide the prompt-learning process. To this end, we propose **IPL** (Interpretability-Guided Prompt Learning), a prompt-learning framework to adapt to a dataset, that extracts a concept dictionary from a frozen SAE for that dataset, composes prompt tokens as learnable attention-weighted combinations of those concepts, and projects the resulting tokens into both vision and text encoder layers. A projection-preservation regularizer keeps prompt-token geometry aligned with the underlying concept directions, so that the learned prompts remain anchored to the interpretable concept directions they were composed from. The framework is agnostic to the choice of SAE based probing methods: vision-only, text-only, and unified concept dictionaries can all be plugged in.

Overall, our contributions and key findings can be summarized as follows:

- We introduce IPL, a prompt-learning framework that composes prompt tokens via attention over interpretable concept directions and preserves their geometry through a projection preservation regularizer. The framework is agnostic to the source SAE, supporting vision-only, text-only, and unified concept dictionaries.
- Through this framework, we show that mechanistic interpretability tools can be use as a training guide for vision-language prompt learning.
- We systematically study how the modality of interpretability signals (vision, text, or unified) affects downstream performance, finding that unified signals from VL-SAE [32] produce the strong gains.
- Across four standard prompt-learning downstream task settings and 15 evaluation datasets, IPL outperforms prior interpretable prompt-learning methods by up to $+2.49\%$ in harmonic mean on base-to-novel generalization while remaining competitive with strong black-box prompt-learning baselines.

2 Related Works

Mechanistic interpretability of vision-language models. Mechanistic interpretability aims to decompose neural computations into human-understandable circuits and features. Sparse autoencoders (SAEs) [7] are a promising tool for this purpose: by reconstructing intermediate activations with a sparse overcomplete dictionary, SAE decoder directions often capture monosemantic concepts. PatchSAE [23] applies SAEs to CLIP vision-transformer residual streams to reveal patch-level visual concepts. SpLiCE [2] decomposes CLIP image embeddings into sparse non-negative combinations of text-aligned concept directions. VL-SAE [32] learns a unified latent concept space for image and text inputs with modality-specific decoders. So far, these methods have mainly been used for post-hoc interpretation. In contrast, IPL uses SAE-derived concept directions as a structured basis for prompt learning, bridging mechanistic interpretability with downstream adaptation.

Prompt learning for vision-language models. CLIP [30], ALIGN [19], and related contrastive vision-language models achieve strong zero-shot recognition using hand-crafted text prompts. Prompt learning adapts these models efficiently to downstream tasks while preserving pretrained capabilities. CoOp [45] learns continuous text prompts, and CoCoOp [44] makes them image-conditional to reduce base-class overfitting. KgCoOp [40] regularizes prompts toward hand-crafted templates, while ProDA [26] models prompt distributions for diverse class semantics. Multimodal methods such as MaPLe [20] learn coupled vision-text prompts, while PromptSRC [4], MMA [39], TCP [41], 2SFS [11], and SkipT [37] improve generalization through regularization or architectural changes. More recently, MMRL [15] injects multimodal representation tokens into intermediate layers, and VaMP [5] learns sample-conditioned text prompts via variational inference. Despite their effectiveness, these methods learn opaque continuous prompt vectors without human-interpretable semantics.

Interpretable prompt learning. A separate line of work improves prompt interpretability by using human-readable concepts as prompt content. CPL [42] caches class-relevant concept descriptions

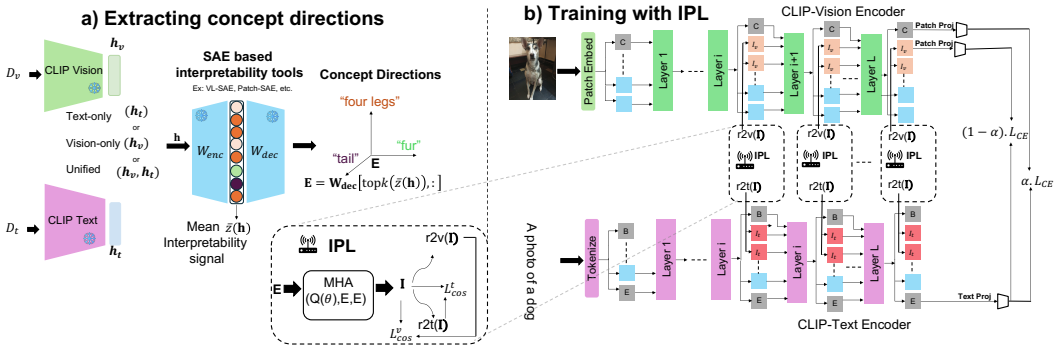

Figure 1: a) We extract concept directions from the training set of dataset to be adapted. We use these concept directions to contruct concept attention weighted prompt tokens. b) IPL prompt tokens are projected at different layers in both text and vision encoder of CLIP. A projection regularization is applied to ensure concept geometry after projection.

as context, while ArGue and ArGue-N [35] use GPT-3 to generate visual attributes per class (e.g., a long tail," black paws") and concatenate selected attributes into prompts. IntCoOp [14] extracts attribute labels with BLIP-2 and aligns them with class embeddings during prompt tuning. IPO [10] optimizes prompts in discrete token space so learned tokens remain natural-language phrases. Overall, these methods rely on *externally generated* concepts or attributes, so interpretability comes from the prompt input form rather than CLIP's internal computation. In contrast, IPL derives concept directions from *inside* CLIP using mechanistic interpretability methods, explicitly connecting prompt learning to CLIP's internal representations. Please refer to appendix Section E for detailed related works.

3 Method.

Our approach **IPL** (Interpretability-Guided Prompt Learning) leverages mechanistic interpretability to guide prompt learning for adapting VLMs such as CLIP [30] to downstream tasks like base to novel generalization, domain generalization, cross-dataset evaluation and few-shot learning. We first introduce preliminaries elucidating zero-shot classification, prompt learning with CLIP, and sparse autoencoder-based interpretability probing in Sections 3.1.1 to 3.1.3 respectively. Then, we show how our approach leverages CLIP-based mechanistic interpretability methods like VL-SAE [32], PatchSAE [23] etc, to obtain vision, language and unified interpretability signals in Section 3.2.1, and explain how we use the interpretability signals to guide construction of prompt tokens in Sections 3.2.2 and 3.2.3. In Section 3.2.4, we use the interpretability signal guided prompt tokens as learnable representations and describe the training and inference process.

3.1 Preliminaries

3.1.1 Revisiting CLIP zero-shot classification.

CLIP consists of paired image encoder f_v and text encoder f_t trained contrastively to align image and text embeddings in a shared latent space. **Image encoding:** Given an input image $\mathbf{X} \in \mathbb{R}^{H\times W\times 3}$, the visual transformer backbone [9] first patchifies $\mathbf{X}$ into $\mathcal{M}$ patch tokens, prepends a [CLS] class token, and adds positional embeddings to obtain the input sequence $\mathbf{X^{(0)}} \in \mathbb{R}^{\mathcal{M}\times d_v}$, with $d_v = 768$ for ViT-B/16. This sequence is processed by $\mathcal{L}$ transformer layers as

$$[CLS_i, X_i] = f_v([CLS_{i-1}, X_{i-1}]), \quad \forall i \in \{1, \cdots, L\} \tag{1}$$

The final image embeddings are obtained by applying layer normalization and a patch projection layer P_v to the [CLS] token of the last layer thus projecting the output of [CLS] token into shared vision-language latent space.

$$feat_v = P_v(CLS_L) \quad feat_v \in \mathbb{R}^{d_{vl}} \tag{2}$$

Text encoding: A textual prompt like "`a photo of a` $[class]$" is tokenized and embedded, producing $\mathcal{G}$ token embeddings $w_0 = [w_0^1, w_0^2, \cdots, w_0^G] \in \mathbb{R}^{G\times d_l}$. After processing through $\mathcal{L}$ text

transformer layers $[w_i] = f_t([w_{i-1}]);\ \ \forall i \in \{1, \cdots, L\}$, the text embedding features are obtained by and passing through a final projection layer P_t

$$feat_t = P_t(w_L^G) \quad feat_t \in \mathbb{R}^{d_{vl}} \tag{3}$$

Zero-shot classification For an image $\mathbf{X}$ and text prompts with class labels $\mathcal{Y} \in \{1, 2, \cdots, C\}$. the prediction is decided by highest cosine similarity, using temperature coefficient τ:

$$\hat{y} = \arg\max_y \frac{\exp(\text{sim}(feat_v, feat_t)/\tau)}{\sum_{i=1}^{C} \exp(\text{sim}(feat_v, feat_t^i)/\tau)} \tag{4}$$

3.1.2 Understanding prompt learning for efficient adaptation.

Hand-engineered prompts like ‘‘`a photo of a [class]`’’ that are used as inputs to CLIP are often sub-optimal for downstream tasks on complex and fine-grained datasets as they may not be aligned with the semantic context of different domains. One way out is to fine-tune CLIP on prompts engineered for each domain but it can lead to catastrophic forgetting and is computationally expensive. The alternative, prompt learning, introduces continuous learnable tokens either in the input [20, 45, 44, 10, 42] or intermediate layers [15, 39, 5] for adapting CLIP to downstream tasks without hand-engineering the prompts and keeping the CLIP's encoder frozen. Existing approaches differ in the layers and encoders in which these prompt tokens are inserted. **Language prompt learning** introduces randomly initialized learnable prompt tokens $I_h^t \in \mathbb{R}^{N \times d_l}$, consisting of N tokens, into each H consecutive transformer layers $h \in \{J, J+H-1\}$ of the text encoder f_t, leaving the rest of the layers unchanged. Similarly, **vision prompt learning** introduces N learnable prompt tokens $I_h^v \in \mathbb{R}^{N \times d_v}$. In **Multimodal prompt learning** prompt tokens are added to both text and vision branches. MaPLe [20] passes language prompt tokens to a learnable coupling function to obtain vision prompts tokens, MMRL [15] proposes a learnable and shared representation space with tokens initialized by sampling from gaussian distribution and uses a linear projection layer to get vision and text branch prompt tokens, and VaMP [5] injects sample specific text prompts obtained by variation inference conditioned on image features, and also simultaneously injects vision prompts.

3.1.3 Sparse-Autoencoder based mechanistic interpretability

Recent mechanistic interpretability studies [23, 32, 7] have shown that sparse autoencoders (SAE's) find highly interpretable features in their latent space. We will first elucidate the working mechanism and terminologies for SAE's and then define what constitutes a ***"concept"*** and ***"interpretability signal"***. **Sparse Autoencoder:** Given an input image $\mathbf{X}$ to CLIP image encoder, $\mathbf{h} \in \mathbb{R}^{d_v}$ is CLIP vision encoder layer's ($l \in L$) representation. These are given as an input to SAE with encoder $\mathbf{W_{enc}} \in \mathbb{R}^{d_v \times K}$ produces concept activations

$$\mathbf{z(h)} = \sigma(\mathbf{hW_{enc}} + \mathbf{b_{enc}}) \in \mathbb{R}^K, \tag{5}$$

and decoder $\mathbf{W_{dec}} \in \mathbb{R}^{K \times d_v}$ reconstructs $\hat{\mathbf{h}}$ from $\mathbf{z}$

$$\hat{\mathbf{h}}(\mathbf{z(h)}) = \mathbf{W_{dec}z(h)} + \mathbf{b_{dec}}, \tag{6}$$

Here, σ is a non-negative activation (ReLU or TopK) inducing sparsity, $\mathbf{b_{enc}}$ and $\mathbf{b_{dec}}$ are the bias terms. The SAE is trained to minimize $\|\mathbf{h} - \hat{\mathbf{h}}(\mathbf{z(h)})\|^2 + \lambda\|\mathbf{z(h)}\|$ on a huge corpus of representations (VL-SAE [32] is trained with CC3M dataset). K is dictionary size (typically $K \gg d_v$. For instance, in PatchSAE[23] K is 64 times the dimension of d_v). Each row $\mathbf{W_{dec}}[i, :]$ is interpreted as concept i's direction in the SAE representation space. Each entry in $\mathbf{z_i}$ measures concept i's presence in $\mathbf{h}$.

Definition 1 (Concept)**.** *Each SAE decoder row can be viewed as a candidate concept feature with a unit-norm direction.*

$$\mathbf{v}_i = \frac{\mathbf{W}_{dec}[i, :]}{\|\mathbf{W}_{dec}[i, :]\|} \in \mathbb{R}^{d_v}, \qquad i \in \{1, \ldots, K\}, \tag{7}$$

Consider an image $\mathbf{X}$ of a dog passed through CLIP, yielding a residual-stream representation $\mathbf{h} \in \mathbb{R}^{d_v}$. A sparse autoencoder trained on CLIP's residuals might yield a concept dictionary containing candidate concept directions that can be empirically annotated as $\mathbf{v}_{fur}, \mathbf{v}_{four\text{-}legged}, \mathbf{v}_{tail}, \mathbf{v}_{sky}, \mathbf{v}_{indoor\ scene}, \ldots$, based on the training images on which the corresponding SAE latent activates strongly. Thus, each candidate concept feature is paired with an evidence set of training images supporting its semantic interpretation.

Definition 2 (Interpretability Signal)**.** *Given a representation* $\mathbf{h} \in \mathbb{R}^{d_v}$ *and a candidate concept feature direction* $\mathbf{v}_i$*, the* ***interpretability signal*** *for* $\mathbf{v}_i$ *on* $\mathbf{h}$ *is defined as the corresponding SAE latent activation*

$$z_i(\mathbf{h}) = \sigma\big(\mathbf{W}_{\text{enc}}[:, i]^\top \mathbf{h} + b_{\text{enc},i}\big) \in \mathbb{R}_{\geq 0}, \tag{8}$$

i.e., the i*-th entry of* $\mathbf{z}(\mathbf{h})$ *in Equation (5). The signal* $z_i(\mathbf{h})$ *serves as an input-dependent activation score for the candidate concept feature* $\mathbf{v}_i$*. Interpretability signals are therefore input-dependent: the same candidate concept feature can produce different signal magnitudes for different representations.*

Vision, text, and unified interpretability signals. The interpretability methods we use yield three modalities of signal, distinguished by which CLIP encoder produces the representation $\mathbf{h}$ that drives concept activation. **Vision interpretability signals:** are obtained by running images through CLIP's vision encoder and reading concept activations from a vision-side SAE. Methods like PatchSAE [23], VL-SAE-vision [32] operate in this mode. **Text interpretability signals:** are obtained by running per-class textual prompts through CLIP's text encoder and reading activations from a text-side SAE. Methods like VL-SAE-text [32] operate in this mode. **Unified interpretability signals:** constitute a unified concept set obtained by leveraging both image and text embeddings from CLIP. VL-SAE-unified [32] instantiates this construction.

3.2 Interpretability Guided Prompt Learning

3.2.1 Extracting concepts from interpretability methods

Given a frozen sparse autoencoder provided by any CLIP [30] based mechanistic interpretability method (we use pretrained SAE provided by VL-SAE [32] and PatchSAE [23]), we construct an interpretable concept dictionary $\mathbf{E} \in \mathbb{R}^{k \times d_v}, k \subset K$, where each row of $\mathbf{E}$ is a concept direction (Definition 1) corresponding to a human-meaningful concept in CLIP's residual-stream space. The construction of $\mathbf{E}$ has three steps.

Computing mean interpretability signal. Let $\mathcal{D}_{\text{base}}$ denote the base-class training set, comprising images and their associated class prompts. We forward each sample through the CLIP encoder then pass the resulting representation $\mathbf{h}$ through the SAE encoder to obtain per-sample interpretability signals $z(\mathbf{h})$ over the SAE's full dictionary of K concepts (Definition 2). So for $\mathcal{S}$ samples in $\mathcal{D}_{\text{base}}$, we then compute its mean signal $\bar{z}_i$ across all samples in $\mathcal{D}_{\text{base}}$, yielding a per-concept mean signal: $\bar{z}(\mathbf{h}) = \sum_{i=1}^{S}(z_i(\mathbf{h}))$

We refer to $\bar{z}(\mathbf{h})$ as a *vision signal* when $\mathbf{h}$ is produced by the image encoder, a *text signal* when $\mathbf{h}$ is produced by the text encoder, and a *unified signal* when the two are concatenated to obtain a unified concept activation.

Top-k selection. We retain the indices $\mathcal{I}$ of the top-k, where $k \subset K$ obtained by mean signal magnitude $|\bar{z}(\mathbf{h})|$. The resulting concept dictionary is

$$\mathbf{E} = \mathbf{W}_{\text{dec}}[\mathcal{I}, :] \in \mathbb{R}^{k \times d_v}, \tag{9}$$

We instantiate three modality variants : $\mathbf{E}^{\text{vision}}$ is constructed using the SAE decoder weights ranked by mean of image interpretability signals, $\mathbf{E}^{\text{text}}$ is constructed using the SAE decoder representations ranked by mean text interpretability signals on per-class textual prompts, and $\mathbf{E}^{\text{unified}}$ is constructed using the SAE decoder representations ranked by unified interpretability signals like VL-SAE [32] encoder is trained to produce unified interpretability signals in the shared latent space from both vision and text inputs.

3.2.2 Interpretability signal guided prompt tokens

We introduce N learnable representation queries $\mathbf{Q} = [\mathbf{q}_1; \ldots; \mathbf{q}_M] \in \mathbb{R}^{M \times d_v}$, trained jointly with the rest of the prompt-learning objective. Unlike prior works like [15, 5] which uses Gaussian-initialized free-form representations, our queries select content from the frozen interpretable dictionary $\mathbf{E}$ via attention. For each rep-injection layer $\ell \in \mathcal{L}$ transformer layers, we instantiate a multi-head attention selector $\text{MHA}^{(\ell)}$ that attends from $\mathbf{Q}$ to $\mathbf{E}$:

$$\mathbf{I}^{(\ell)} = \text{MHA}^{(\ell)}(\mathbf{Q}, \mathbf{E}, \mathbf{E}) \in \mathbb{R}^{N \times d_v}, \tag{10}$$

where $\mathbf{Q}$ provides the queries and $\mathbf{E}$ provides both keys and values. Each row of $\mathbf{I}^{(\ell)}$ is a layer-specific prompt token expressed as an attention-weighted combination of interpretable concepts.

3.2.3 Projecting prompt tokens into encoder layers

The prompt tokens $\mathbf{I}^{(\ell)}$ live in CLIP's vision-encoder residual space (d_v) and must be projected into the form expected at each injection layer in both the vision and text encoders. We employ two per-layer projections:

$$\mathbf{I}_v^{(\ell)} = r2v^{(\ell)}(\mathbf{I}^{(\ell)}) \in \mathbb{R}^{N\times d_v} \quad \text{and} \quad \mathbf{I}_t^{(\ell)} = r2t^{(\ell)}(\mathbf{I}^{(\ell)}) \in \mathbb{R}^{N\times d_t}, \tag{11}$$

where $r2v^{(\ell)}$ is a linear map preserving residual width and $r2t^{(\ell)}$ projects to the text encoder width d_t. The projected tokens are concatenated to the existing token sequence at layer ℓ in their respective encoders as defined in Section 3.1.2

3.2.4 Learning with projected prompt tokens

Training. For an image-class pair $(\mathbf{x}, y)$, IPL produces two complementary classification logits. The first reflects the standard image-text similarity using the [CLS] embedding.

$$\mathbf{l}_{\text{cls}}^{(c)}(\mathbf{x}) = \frac{\exp(\text{sim}(feat_v, feat_t^{(c)})/\tau)}{\sum_{i=1}^{C}\exp(\text{sim}(feat_v, feat_t^{(i)})/\tau)}, \tag{12}$$

where $feat_v$ and $feat_t^{(c)}$ denote [CLS] token projection and CLIP text projection features computed with prompt tokens injected at layers $\mathcal{L}$, and τ is CLIP's learned temperature coefficient.

The second logit derives image features from the prompt tokens themselves: we take the mean-pooled prompt-token output at the final injection layer, denoted $\bar{\mathbf{I}}(\mathbf{x}) = mean(\mathbf{I}(\mathbf{x}))$, and use it in place of the CLS-derived image feature:

$$\mathbf{l}_{\text{rep}}^{(c)}(\mathbf{x}) = \frac{\exp(\text{sim}(\bar{\mathbf{I}}(\mathbf{x}), feat_t^{(c)})/\tau)}{\sum_{i=1}^{C}\exp(\text{sim}(\bar{\mathbf{I}}(\mathbf{x}), feat_t^{(i)})/\tau)}. \tag{13}$$

Projection-preservation regularization. A naive linear projection performed to obtain prompt tokens (as described in Section 3.2.3) can rotate concept structure arbitrarily, decoupling the prompt tokens' geometry from $\mathbf{I}$'s interpretable structure. To encourage the projections to preserve concept geometry, we introduce a projection-preservation loss:

$$\mathcal{L}_{\text{proj}} = \frac{1}{|\mathcal{L}|}\sum_{\ell\in\mathcal{L}}\Big(\underbrace{1-\cos\Big(r2v^{(\ell)}(\mathbf{I}_v^{(\ell)}), \mathbf{I}_v^{(\ell)}\Big)}_{\text{vision: cosine alignment}(\mathcal{L}_{\text{cos}}^{v})} + \underbrace{1-\cos\Big(r2t^{(\ell)}(\mathbf{I}_t^{(\ell)}), \mathbf{I}_t^{(\ell)}\Big)}_{\text{text: cosine alignment}(\mathcal{L}_{\text{cos}}^{t})}\Big), \tag{14}$$

where $\cos(\cdot,\cdot)$ averages cosine similarity row-wise. The vision term encourages $r2v^{(\ell)}$ to act near-identically on $\mathbf{I}_v^{(\ell)}$, while the text term preserves the pairwise similarity structure of $\mathbf{I}_t^{(\ell)}$ under $r2t^{(\ell)}$ even though dimensionality changes. Together, these constrain the projections to keep prompt-token geometry similar to concept directions.

Our training objective is the Lagrangian with α balancing the two cross-entropy objectives, and λ_{proj} is the projection-preservation loss weight tuned per dataset:

$$\mathcal{L} = \alpha\,\mathcal{L}_{\text{CE}}(\mathbf{l}_{\text{cls}}, y) + (1-\alpha)\;\mathcal{L}_{\text{CE}}(\mathbf{l}_{\text{rep}}, y) + \lambda_{\text{proj}}\,\mathcal{L}_{\text{proj}}. \tag{15}$$

The trainable parameters are: queries $\mathbf{Q}$ in layer-specific selectors $\{\text{MHA}^{(\ell)}\}$, and projections $\{r2v^{(\ell)}, r2t^{(\ell)}\}$. CLIP's encoders, $\mathbf{E}$, and the interpretability methods' parameters remain frozen throughout training.

Inference. For base-class evaluation, we use the fused logits $\alpha\,\mathbf{l}_{\text{cls}} + (1-\alpha)\,\mathbf{l}_{\text{rep}}$ to leverage both the standard and rep-token classification signals. For novel-class evaluation, we use only $\mathbf{l}_{\text{cls}}$, since rep tokens are trained on base classes and may bias predictions toward seen categories. This asymmetry preserves the zero-shot transfer capability of the underlying CLIP alignment for unseen classes.

4 Experiments

We evaluate IPL on four standard downstream tasks for prompt learning: base-to-novel generalization, few-shot learning, domain generalization, and cross-dataset transfer.

Table 1: Base-to-novel generalization accuracies (B: Base, N: Novel, H: Harmonic Mean) compared to state-of-the-art interpretable and blackbox methods. Bold indicates best overall and underline indicates best interpretable method. **Note: the average reported in this table is over all 11 datasets, please refer to Section A for detailed results with on other datasets.**

	Method	Average			ImageNet			StanfordCars			SUN397			UCF101		
		B	N	H	B	N	H	B	N	H	B	N	H	B	N	H
Interpretable	CPL [42]	84.38	78.03	81.08	78.74	72.03	75.24	79.31	76.65	77.96	81.88	79.65	80.75	86.73	80.17	83.32
	IPO [10]	79.92	80.51	80.21	77.83	72.45	75.04	73.42	75.71	74.55	81.25	80.92	81.08	85.32	80.92	83.06
	IntCoOp† [14]	83.20	78.72	80.75	75.99	72.67	74.29	77.04	76.32	76.67	81.63	79.33	80.46	86.76	79.42	82.92
	ArGue-N [35]	83.77	78.74	81.18	76.95	71.86	74.32	75.06	74.18	74.62	81.89	80.48	81.18	86.00	79.43	82.58
	IPL (Vision)	85.80	77.15	81.25	78.30	70.80	74.36	81.10	75.20	78.04	83.00	79.20	81.06	88.60	79.90	84.03
	IPL (Text)	85.75	76.59	80.91	78.40	70.70	74.35	81.40	75.50	78.34	83.10	79.40	81.21	88.80	80.00	84.17
	IPL (Unified)	86.26	**81.23**	**83.67**	**79.30**	**74.50**	**76.83**	81.90	**80.40**	81.14	83.00	**82.10**	**82.55**	**89.30**	**85.80**	**87.52**
Blackbox	CLIP [30]	69.34	74.22	71.70	72.43	68.14	70.22	63.37	74.89	68.65	69.36	75.35	72.23	70.53	77.50	73.85
	CoOp [45]	82.69	63.22	71.66	76.47	67.88	71.92	78.12	60.40	68.13	80.60	65.89	72.51	84.69	56.05	67.46
	CoCoOp [44]	80.47	71.69	75.83	75.98	70.43	73.10	70.49	73.59	72.01	79.74	76.86	78.27	82.33	73.45	77.64
	ProDA [26]	81.56	72.30	76.65	75.40	70.23	72.72	74.70	71.20	72.91	78.67	76.93	77.79	85.23	71.97	78.04
	KgCoOp [40]	80.73	73.60	77.00	75.83	69.96	72.78	71.76	75.04	73.36	80.29	76.53	78.36	82.89	76.67	79.65
	MaPLe [20]	82.28	75.14	78.55	76.66	70.54	73.47	72.94	74.00	73.47	80.82	78.70	79.75	83.00	78.66	80.77
	PromptSRC [4]	84.26	76.10	79.97	77.60	70.73	74.01	78.27	74.97	76.58	82.67	78.47	80.52	87.10	78.80	82.74
	TCP [41]	84.13	75.36	79.51	77.27	69.87	73.38	80.80	74.13	77.32	82.63	78.20	80.35	87.13	80.77	83.83
	MMA [39]	83.20	76.80	79.87	77.31	71.00	74.02	78.50	73.10	75.70	82.27	78.57	80.38	86.23	80.03	82.20
	2SFS [11]	85.55	75.48	80.20	77.71	70.99	74.20	82.50	74.80	78.46	82.59	78.91	80.70	87.85	78.19	82.74
	SkipT [37]	85.04	77.53	81.11	77.73	70.40	73.89	82.93	72.50	77.37	82.40	79.03	80.68	87.30	82.47	84.81
	MMRL [15]	85.68	77.16	81.20	77.90	71.30	74.45	81.30	75.07	78.06	83.20	79.30	81.20	88.10	80.07	83.89
	VaMP [5]	**86.45**	78.67	82.37	78.98	73.45	76.11	**83.78**	80.14	**81.91**	**83.37**	78.95	81.09	88.52	78.99	83.48

Additional analyses in the appendix. Due to space constraints, several supporting analyses are deferred to the appendix: *(i)* loss-component ablations (Section C.1), *(ii)* the effect of prompt-token injection layers (Section C.2), *(iii)* the effect of concept-dictionary size (Section C.3) k, *(iv)* sensitivity to the loss weights α and λ_{proj} (Section D), *(v)* the effect of the number of prompt tokens (Section D.2), *(vi)* the effect of the layer range used for prompt-token injection (Section C.2), *(vii)* a computational complexity analysis (Section C.4), and *(viii)* IPL's performance across different SAE-based interpretability tools (Section D.1).

4.1 Experimental Settings

Datasets. We evaluate on the 15 standard image-classification datasets used by the prompt-learning literature: For Base to novel generalization, Few-shot Learning, and cross-dataset evaluation we use 11 datasets ImageNet [8], Caltech101 [22], OxfordPets [29], StanfordCars [21], OxfordFlowers [28], Food101 [3], FGVCAircraft [27], SUN397 [38], DTD [6], EuroSAT [16], and UCF101 [33]. For domain generalization we use ImageNetV2 [31], ImageNet-Sketch [36], ImageNet-A [18], and ImageNet-R [17] as target domains.

Interpretability Methods. We instantiate IPL with three interpretability tool variants. VL-SAE [32] provides three modalities: VL-SAE-vision (vision decoder), VL-SAE-text (text decoder), and VL-SAE-unified (concatenation of both decoders, see Section 3.2.1).

Hyperparameter details. All IPL runs use $N = 5$ representation tokens, $\alpha = 0.7$, soft injection mode, and inject prompt tokens at layers $\mathcal{L}_{\text{rep}} = \{6, 7, 8, 9, 10, 11, 12\}$ in both vision and text encoders. We train for 10 epochs with a learning rate of 0.0035 (cosine schedule, 1-epoch warmup), batch size 4, and SGD optimizer following the convention of MMRL [15]. All our experiments use CLIP ViT-B/16 as the frozen vision-language backbone. Image and text encoder weights from OpenAI's released checkpoints are used throughout, with **no fine-tuning of CLIP at any point.** All experiments use 16-shot training and results reported over 3 seeds.

Baselines. We compare IPL against two categories of baselines. black-box prompt-learning: which inject randomly initialized prompt tokens into CLIP encoder layer and optimize over them like zero-shot CLIP, CoOp [45], CoCoOp [44], MaPLe [20], PromptSRC [4], MMA [39], TCP [41], 2SFS [11], SkipT [37],MMRL [15], and VaMP [5]. interpretable prompt-learning: which inject the LLM queries attributes into CLIP encoder layers and optimize over them like CPL [42], ArGue-N [35], IntCoOp [14], and IPO [10]. Numbers are reported from original papers.

4.2 Results and Discussion

Base to novel generalization In this experiment, we split each dataset equally into base and novel classes across 11 datasets. During training, only the base classes are used, while evaluation is performed on both base and novel classes. This setting allows us to examine how interpretability-signal-guided prompts affect VLM generalization to both seen and unseen classes.

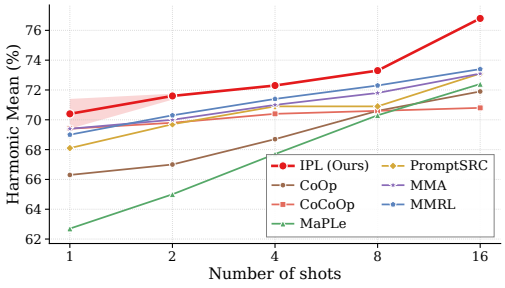

Figure 2: Few-shot performance on ImageNet under the base-to-novel protocol. IPL consistently improves harmonic mean over base and novel classes across shot counts, with the largest gain at 16 shots. Shaded bands denote ± std over 3 seeds.

In Table 1, we report the performance of IPL using vision-only interpretability signals (IPL(vision)), text-only interpretability signals (IPL (text)), and unified interpretability signals (IPL (unified)). We observe that IPL (vision) and IPL (text) achieve gains over previous interpretable prompt-learning methods on the base and novel classes for a few datasets, such as StanfordCars[21], UCF101[33], and Caltech101[22]. On average, they also improve base-class performance. However, their performance comparatively drops on several other datasets, particularly on novel classes. This is likely because interpretability signals derived from a single modality may not be sufficient to produce well-aligned logits that improve generalization.

In contrast, IPL with unified signals achieves the strongest average results on both base and novel classes. Notably, IPL (unified) also outperforms the black-box state-of-the-art method VaMP on several datasets, including ImageNet[8], Caltech101[22], OxfordPets[29], Food101[3], and UCF101[33].

Domain generalization In this experiment, we evaluate whether interpretability-guided prompt learning enables VLMs to generalize to distribution shifts during inference. We use ImageNet as the source dataset for training and evaluate the learned model on target ImageNet variants, including ImageNet-V2[31], ImageNet-Sketch[36], ImageNet-A[18], and ImageNet-R[17]. Table 2 shows that IPL (vision) and IPL (text) achieve strong source-domain accuracy on ImageNet, but show slightly weaker generalization on ImageNet-V2, ImageNet-Sketch, and ImageNet-R compared to previous interpretability-based state-of-the-art methods. IPL (unified) achieves strong results, outperforming prior interpretability-based state-of-the-art methods across all target domains and on the average target-domain accuracy.

Table 2: Domain generalization. All methods are trained on ImageNet (16-shot, ViT-B/16) and evaluated on four ImageNet variants. Results are top-1 accuracy (%). **Best** and second-best per column. The leftmost column groups methods into *Interpretable* (concept/attribute-based prompt learning) and *Blackbox* (other prompt learning).

		Source	Target				
	Method	ImageNet	-V2	-Sketch	-A	-R	**Avg.**
Interpretable	IntCoOp† [14]	71.85	65.21	49.20	51.55	76.88	60.71
	ArGue-N† [35]	71.84	65.02	49.25	51.47	76.96	60.68
	IPL (Vision)	73.63 ± 0.05	64.93 ± 0.09	48.10 ± 0.08	51.90 ± 0.14	75.90 ± 0.37	60.20 ± 0.05
	IPL (Text)	73.57 ± 0.05	64.77 ± 0.12	48.07 ± 0.05	51.73 ± 0.49	75.67 ± 0.25	60.06 ± 0.15
	IPL (Unified)	**74.50** ± 0.08	**65.83** ± 0.12	**50.17** ± 0.17	**52.13** ± 0.21	77.87 ± 0.40	**61.50** ± 0.04
Blackbox	CLIP [30]	66.73	60.83	46.15	47.77	73.96	57.18
	CoOp [45]	71.51	64.20	47.99	49.71	75.21	59.28
	CoCoOp [44]	71.02	64.07	48.75	50.63	76.18	59.91
	MaPLe [20]	70.72	64.07	49.15	50.90	76.98	60.26
	PromptSRC [4]	71.27	64.35	49.55	50.90	77.80	60.65
	MMRL [15]	72.03	64.47	49.17	51.20	77.53	60.59
	VaMP [5]	72.83	64.96	49.69	51.97	**78.01**	61.16

Few-shot learning In this experiment, we evaluate IPL's ability to adapt under different few-shot regimes using all the classes of ImageNet dataset. From Figure 2 We observe that IPL (unified) consistently outperforms previous baselines across different numbers of shots. Since only a limited number of prior methods report few-shot learning results, we compare IPL against the baselines for which such evaluations were reported. Results on other datasets in Section B.

Cross-dataset evaluation In this experiment, we evaluate the transferability of IPL to different target datasets during evaluation. Following the setting of CoCoOp [44], we train the source model on ImageNet and evaluate its performance on 10 other datasets at inference time in the few-shot setting as shown in Table 3. Similar to trends observed in base-to-novel generalization and domain generalization, IPL (unified) achieves the better overall performance across the target datasets.

Table 3: Cross-dataset transfer. All methods are trained on ImageNet (16-shot, ViT-B/16) and evaluated zero-shot on 10 downstream datasets. Top-1 accuracy (%) averaged over 3 seeds. The methods are grouped into *Interpretable* and *Blackbox*.

		Source	**Target**										
	Method	*ImageNet*	*Caltech101*	*OxfordPets*	*StanfordCars*	*Flowers102*	*Food101*	*Aircraft*	*SUN397*	*DTD*	*EuroSAT*	*UCF101*	**Avg.**
Interpretable	CPL [42]	73.53	95.52	91.64	66.17	73.35	**87.68**	27.36	68.24	48.96	51.25	70.52	68.07
	IPO [10]	72.15	94.34	90.96	66.10	72.75	86.75	25.14	67.97	47.01	48.56	69.23	67.36
	IPL (Vision)	73.70$_{\pm 0.05}$	94.80$_{\pm 0.61}$	91.60$_{\pm 0.57}$	65.20$_{\pm 0.50}$	72.20$_{\pm 0.31}$	85.20$_{\pm 0.29}$	25.30$_{\pm 0.26}$	66.60$_{\pm 0.24}$	44.90$_{\pm 1.63}$	53.80$_{\pm 3.06}$	68.60$_{\pm 0.73}$	66.82$_{\pm 0.33}$
	IPL (Text)	73.60$_{\pm 0.05}$	94.90$_{\pm 0.42}$	91.30$_{\pm 0.37}$	65.00$_{\pm 0.45}$	72.80$_{\pm 0.83}$	84.90$_{\pm 0.12}$	26.00$_{\pm 0.59}$	66.60$_{\pm 0.31}$	45.00$_{\pm 0.37}$	54.50$_{\pm 4.21}$	68.40$_{\pm 0.46}$	66.94$_{\pm 0.45}$
	IPL (Unified)	**73.70**$_{\pm 0.08}$	**95.90**$_{\pm 0.39}$	91.30$_{\pm 0.33}$	**66.80**$_{\pm 0.79}$	**73.40**$_{\pm 0.08}$	87.60$_{\pm 0.17}$	**27.40**$_{\pm 0.90}$	**68.80**$_{\pm 0.43}$	45.90$_{\pm 0.43}$	**54.50**$_{\pm 4.24}$	**70.70**$_{\pm 0.66}$	**68.23**$_{\pm 0.60}$
Blackbox	CoOp [45]	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 [44]	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 [20]	70.72	93.53	90.49	65.57	72.23	86.20	24.74	67.01	46.49	48.06	68.69	66.30
	PromptSRC [4]	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 [41]	71.40	93.97	91.25	64.69	71.21	86.69	23.45	67.15	44.35	51.45	68.73	66.29
	MMA [39]	71.00	93.80	90.30	66.13	72.07	86.12	25.33	68.17	46.57	49.24	68.32	66.61
	MMRL [15]	72.03	94.67	91.43	66.10	72.77	86.40	26.30	67.57	45.90	53.10	68.27	67.25
	VaMP [5]	72.83	94.96	**91.79**	66.10	73.18	86.97	26.76	68.04	46.82	53.82	68.93	67.74

5 Qualitative Analysis

Qualitative visualization. Figure 3 traces the interpretability chain that IPL exploits for prompt construction, illustrated on an Oxford Flowers dataset. We use VL-SAE [32] as the interpretability tool, which provides a pretrained shared latent space along with human-readable concept labels for each decoder direction. Initially we rank concepts by mean SAE activation $\bar{z}_i(\mathbf{h})$ over the dataset's base classes (Definition 2) yields the unit-norm directions $\mathbf{v}_i \in \mathbf{E}^{\text{unified}}$ that populate IPL's concept dictionary, which are visually meaningful and dataset-relevant (*pointed floret tips*, *pinched bud*). A learned prompt token $\mathbf{I}^{(\ell)}$ at injection layer ℓ then composes these directions via attention, with cosine contributions revealing that $\mathbf{I}^{(\ell)}$ re-weights concepts beyond their raw activation ranking, indicating that the multi-head attention selector learns task-discriminative combinations rather than just copying $\bar{z}(\mathbf{h})$. Comparing Grad-CAM saliency, vanilla CLIP attends to semantically irrelevant regions, while IPL concentrates on the flower head, suggesting that interpretability-guided prompts induce more semantically grounded attention.

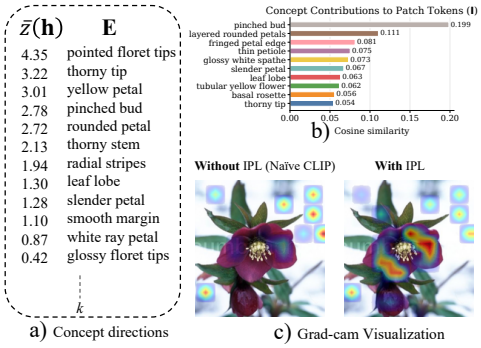

Figure 3: (a) top-ranked concept directions in $\mathbf{E}$ with mean activations $\bar{z}(\mathbf{h})$ on oxford flowers dataset, (b) contributions of these concepts to a learned prompt token $\mathbf{I}$ in terms of cosine similarity, and (c) Grad-CAM heatmaps showing that IPL focuses attention on the semantic features flower head while naive CLIP does not.

6 Conclusions

We presented IPL, a prompt-learning framework that leverages SAE based mechanistic interpretability tools as active training guides. By extracting concept directions from frozen sparse autoencoders, composing them into prompt tokens via learnable attention selector, and preserving their geometry through a projection regularizer, IPL grounds prompt learning in CLIP's own interpretable structure rather than externally generated attributes. Across base-to-novel generalization, domain generalization, cross-dataset transfer, and few-shot learning, IPL with unified concept directions performs better in average accuracy compared to most of the previous state-of-the-art methods. As a limitation we acknowledge that IPL's effectiveness depends on the quality of the underlying interpretability tool. For detailed analysis of our other limitations please refer to appendix Section C.4.

References

[1] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

[2] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *Advances in Neural Information Processing Systems*, 37:84298–84328, 2024.

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

[4] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Plot: Prompt learning with optimal transport for vision-language models. *arXiv preprint arXiv:2210.01253*, 2022.

[5] Silin Cheng and Kai Han. Vamp: Variational multi-modal prompt learning for vision-language models. *arXiv preprint arXiv:2511.22664*, 2025.

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

[7] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

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

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[10] Yingjun Du, Wenfang Sun, and Cees G Snoek. Ipo: Interpretable prompt optimization for vision-language models. *Advances in Neural Information Processing Systems*, 37:126725–126766, 2024.

[11] Matteo Farina, Massimiliano Mancini, Giovanni Iacca, and Elisa Ricci. Rethinking few-shot adaptation of vision-language models in two stages. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 29989–29998, 2025.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

[14] Soumya Suvra Ghosal, Samyadeep Basu, Soheil Feizi, and Dinesh Manocha. Intcoop: Interpretability-aware vision-language prompt tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19584–19601, 2024.

[15] Yuncheng Guo and Xiaodong Gu. Mmrl: Multi-modal representation learning for vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25015–25025, 2025.

[16] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

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

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

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

[20] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19113–19122, 2023.

[21] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[22] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, Apr 2022.

[23] Hyesu Lim, Jinho Choi, Jaegul Choo, and Steffen Schneider. Sparse autoencoders reveal selective remapping of visual concepts during adaptation. *arXiv preprint arXiv:2412.05276*, 2024.

[24] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021.

[25] Wang Lu, Jindong Wang, Han Yu, Lei Huang, Xiang Zhang, Yiqiang Chen, and Xing Xie. Fixed: Frustratingly easy domain generalization with mixup. In *Conference on Parsimony and Learning*, pages 159–178. PMLR, 2024.

[26] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5206–5215, 2022.

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

[28] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[29] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

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

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

[32] Shufan Shen, Junshu Sun, Qingming Huang, and Shuhui Wang. Vl-sae: Interpreting and enhancing vision-language alignment with a unified concept set. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[33] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[34] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain adaptation in computer vision applications*, pages 153–171. Springer, 2017.

[35] Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. 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, 2024.

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

[37] Shihan Wu, Ji Zhang, Pengpeng Zeng, Lianli Gao, Jingkuan Song, and Heng Tao Shen. Skip tuning: Pre-trained vision-language models are effective and efficient adapters themselves. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14723–14732, 2025.

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

[39] Lingxiao Yang, Ru-Yuan Zhang, Yanchen Wang, and Xiaohua Xie. Mma: Multi-modal adapter for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23826–23837, 2024.

[40] Hantao Yao, Rui Zhang, and Changsheng Xu. 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, 2023.

[41] Hantao Yao, Rui Zhang, and Changsheng Xu. Tcp: Textual-based class-aware prompt tuning for visual-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23438–23448, 2024.

[42] Yi Zhang, Ce Zhang, Ke Yu, Yushun Tang, and Zhihai He. Concept-guided prompt learning for generalization in vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7377–7386, 2024.

[43] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International conference on machine learning*, pages 7523–7532. PMLR, 2019.

[44] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022.

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

Appendix

A Base to novel generalization results

Table 4: Comparison with state-of-the-art methods on base-to-novel generalization across 11 datasets Methods are grouped into *Interpretable* and *Blackbox*. Our method (IPL) is reported using vision-only, text-only and unified interpretability signals using VL-SAE[32]. Bold indicates best per column and underline indicates second-best per column. Delta rows show absolute accuracy-point change over ArGue-N, with positive gains in green and negative changes in red. †IntCoOp does not report DTD; its average in the source paper is over 10 datasets.

	Method	Average			Caltech101			OxfordPets			Flowers102		
		B	N	H	B	N	H	B	N	H	B	N	H
Interpretable	CPL [42]	84.38	78.03	81.08	98.35	95.13	96.71	95.86	98.21	97.02	98.07	80.43	88.38
	IPO [10]	79.92	80.51	80.21	97.32	95.23	96.26	95.21	98.23	96.70	96.78	78.32	86.58
	IntCoOp† [14]	83.20	78.72	80.75	97.80	94.76	96.25	95.92	98.20	97.04	97.82	75.54	85.24
	ArGue-N [35]	83.77	78.74	81.18	98.63	94.70	96.63	96.23	98.59	97.40	98.62	77.96	87.08
	IPL (Vision)	85.80±0.42	77.15±1.09	81.25	99.10±0.19	95.50±0.75	97.27	95.90±0.24	97.70±0.60	96.79	98.70±0.09	76.60±0.45	86.26
	Δ vs ArGue-N	+2.03	-1.59	+0.07	+0.47	+0.80	+0.64	-0.33	-0.89	-0.61	+0.08	-1.36	-0.82
	IPL (Text)	85.75±0.36	76.59±0.83	80.91	99.30±0.21	94.50±0.38	96.84	96.30±0.33	97.10±0.59	96.70	98.60±0.16	77.10±0.57	86.53
	Δ vs ArGue-N	+1.98	-2.15	-0.27	+0.67	-0.20	+0.21	+0.07	-1.49	-0.70	-0.02	-0.86	-0.55
	IPL (Unified)	86.26±0.40	**81.23±1.05**	**83.67**	**99.50±0.24**	**96.40±0.75**	**97.93**	96.60±0.62	**98.60±0.29**	**97.59**	98.80±0.37	81.80±0.38	89.50
	Δ vs ArGue-N	+2.49	+2.49	+2.49	+0.87	+1.70	+1.30	+0.37	+0.01	+0.19	+0.18	+3.84	+2.42
Blackbox	CLIP [30]	69.34	74.22	71.70	96.84	94.00	95.40	91.17	97.26	94.12	72.08	77.80	74.83
	CoOp [45]	82.69	63.22	71.66	98.00	89.81	93.73	93.67	95.29	94.47	97.60	59.67	74.06
	CoCoOp [44]	80.47	71.69	75.83	97.96	93.81	95.84	95.20	97.69	96.43	94.87	71.75	81.71
	ProDA [26]	81.56	72.30	76.65	98.27	93.23	95.68	95.43	97.83	96.62	97.70	68.68	80.66
	KgCoOp [40]	80.73	73.60	77.00	97.72	94.39	96.03	94.65	97.76	96.18	95.00	74.73	83.65
	MaPLe [20]	82.28	75.14	78.55	97.74	94.36	96.02	95.43	97.76	96.58	95.92	72.46	82.56
	PromptSRC [4]	84.26	76.10	79.97	98.10	94.03	96.02	95.33	97.30	96.30	98.07	76.50	85.95
	TCP [41]	84.13	75.36	79.51	98.23	94.67	96.42	94.67	97.20	95.92	97.73	75.57	85.23
	MMA [39]	83.20	76.80	79.87	98.40	94.00	96.15	95.40	98.07	96.72	97.77	75.93	85.48
	2SFS [11]	85.55	75.48	80.20	98.71	94.43	96.52	95.32	97.82	96.55	98.29	76.17	85.83
	SkipT [37]	85.04	77.53	81.11	98.50	95.33	96.89	95.70	97.87	96.77	98.57	75.80	85.70
	MMRL [15]	85.68	77.16	81.20	98.97	94.50	96.68	95.90	97.60	96.74	**98.97**	77.27	86.78
	VaMP [5]	**86.45**	78.67	82.37	98.95	95.96	97.43	**96.95**	95.24	96.08	98.96	**83.97**	**90.85**

	Method	Food101			FGVCAircraft			DTD			EuroSAT		
		B	N	H	B	N	H	B	N	H	B	N	H
Interpretable	CPL [42]	91.92	93.87	92.88	42.27	38.85	40.49	80.92	62.27	70.38	94.18	81.05	87.12
	IPO [10]	90.92	93.08	91.99	41.21	41.42	41.31	82.14	66.81	73.69	94.25	80.11	86.61
	IntCoOp† [14]	91.45	91.99	91.72	38.55	35.90	37.17	–	–	–	95.26	78.01	85.77
	ArGue-N [35]	91.42	92.40	91.91	41.29	38.80	40.01	80.33	67.03	73.08	95.10	**90.68**	**92.84**
	IPL (Vision)	90.70±0.12	91.50±0.05	91.10	46.30±0.66	38.20±1.02	41.86	86.10±0.61	64.50±1.55	73.75	96.00±0.53	79.60±6.25	87.03
	Δ vs ArGue-N	-0.72	-0.90	-0.81	+5.01	-0.60	+1.85	+5.77	-2.53	+0.67	+0.90	-11.08	-5.81
	IPL (Text)	90.90±0.22	91.50±0.00	91.20	45.90±1.25	37.90±0.62	41.52	85.00±0.39	64.70±0.95	73.47	95.60±0.42	74.10±4.78	83.49
	Δ vs ArGue-N	-0.52	-0.90	-0.71	+4.61	-0.90	+1.51	+4.67	-2.33	+0.39	+0.50	-16.58	-9.35
	IPL (Unified)	90.60±0.05	**97.70±0.05**	**94.02**	46.50±0.57	42.20±0.70	**44.25**	**86.30±0.29**	66.50±1.87	75.12	**97.10±0.48**	87.50±5.34	92.05
	Δ vs ArGue-N	-0.82	+5.30	+2.11	+5.21	+3.40	+4.24	+5.97	-0.53	+2.04	+2.00	-3.18	-0.79
Blackbox	CLIP [30]	90.10	91.22	90.66	27.19	36.29	31.09	53.24	59.90	56.37	56.48	64.05	60.03
	CoOp [45]	88.33	82.26	85.19	40.44	22.30	28.75	79.44	41.18	54.24	92.19	54.74	68.69
	CoCoOp [44]	90.70	91.29	90.99	33.41	23.71	27.74	77.01	56.00	64.85	87.49	60.04	71.21
	ProDA [26]	90.30	88.57	89.43	36.90	34.13	35.46	80.67	56.48	66.44	83.90	66.00	73.88
	KgCoOp [40]	90.50	91.70	91.09	36.21	33.55	34.83	77.55	54.99	64.35	85.64	64.34	73.48
	MaPLe [20]	90.71	92.05	91.38	37.44	35.61	36.50	80.36	59.18	68.16	94.07	73.23	82.35
	PromptSRC [4]	90.67	91.53	91.10	42.73	37.87	40.15	83.37	62.97	71.75	92.90	73.90	82.32
	TCP [41]	90.57	91.37	90.97	41.97	34.43	37.83	82.77	58.07	68.25	91.63	74.73	82.32
	MMA [39]	90.13	91.30	90.71	40.57	36.33	38.33	83.20	65.63	73.38	85.46	82.34	83.87
	2SFS [11]	90.71	91.34	91.02	**47.48**	35.51	40.63	84.60	65.01	73.52	96.91	67.09	79.29
	SkipT [37]	90.67	92.03	91.34	45.37	37.13	40.84	83.77	**67.23**	74.59	92.47	83.00	87.48
	MMRL [15]	90.57	91.50	91.03	46.30	37.03	41.15	85.67	65.00	73.82	95.60	80.17	87.21
	VaMP [5]	**92.77**	93.16	92.96	46.77	41.13	43.76	86.14	67.20	**75.50**	95.78	77.21	85.49

Table 1 reports IPL's base-to-novel results on the remaining datasets, complementing the subset shown in the main paper (Table 4). The trends mirror those in the main results: IPL with unified concept directions consistently achieves the strongest harmonic mean, while the vision-only and text-only variants remain competitive with prior interpretable prompt-learning methods.

B Results on few shot learning other datasets.

In Table 5, we study performance of few-shot learning with (1, 2, 4, 8, and 16) shots for different datasets. We observe that IPL performs better than previous blackbox based prompt learning methods. Note that interpretability based prompt learning methods do not provide results for few-shot learning.

Table 5: Few-shot classification accuracies across selected datasets under 1, 2, 4, 8, and 16 shots.

Method	Average					Caltech101					OxfordPets					Flowers102				
	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16
IPL (Vision)	69.51	72.54	77.06	81.15	84.62	93.93	94.63	95.57	96.37	96.63	88.10	91.33	92.70	93.20	93.77	76.50	83.23	88.90	94.57	96.93
IPL (Text)	69.47	72.82	76.15	80.75	84.95	93.97	94.43	95.77	96.13	96.63	88.30	90.97	93.03	92.53	93.80	76.40	83.53	89.97	94.50	96.97
IPL (Unified)	**74.03**	77.29	80.47	82.54	**87.07**	93.73	94.53	**96.80**	**96.53**	**97.73**	90.60	91.83	92.83	93.13	**94.80**	**86.37**	**91.33**	**95.77**	96.50	**98.90**
Linear probe CLIP [30]	48.72	60.44	70.51	76.41	80.51	79.88	89.01	92.05	93.41	95.43	44.06	58.37	71.17	78.36	85.34	69.74	85.07	92.02	96.10	97.37
CoOp [45]	67.34	71.37	74.85	77.87	80.99	92.60	93.07	94.40	94.37	95.57	90.37	89.80	92.57	91.27	91.87	77.53	87.33	92.17	94.97	97.07
CoCoOp [44]	65.63	66.21	71.21	73.35	75.88	93.83	94.82	94.98	95.04	95.16	91.27	**92.64**	92.81	93.45	93.34	72.08	75.79	78.40	84.30	87.84
MaPLe [20]	70.88	74.28	77.31	80.49	83.32	92.57	93.97	94.43	95.20	96.00	89.10	90.87	91.90	92.57	92.83	83.30	88.93	92.67	95.80	97.00
PromptSRC [4]	73.36	76.42	79.72	82.04	84.40	93.67	94.53	95.27	95.67	96.07	**92.00**	92.50	**93.43**	**93.50**	93.67	85.93	91.17	93.87	96.27	97.60
MMA [39]	69.55	72.48	77.51	80.91	84.38	92.90	94.00	94.33	95.37	96.33	91.23	91.97	92.23	92.77	93.23	83.60	90.30	93.00	95.97	97.97
MMRL [15]	73.83	**77.37**	**80.71**	**82.95**	86.09	**94.17**	**94.83**	96.03	96.27	97.13	90.87	91.57	92.57	93.03	93.83	85.97	91.20	94.60	**96.60**	98.40

Method	Food101					FGVCAircraft					DTD					EuroSAT				
	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16	1	2	4	8	16
IPL (Vision)	85.13	85.47	86.23	86.97	**87.50**	29.10	34.40	37.47	44.17	52.70	52.43	56.93	64.93	69.58	73.87	61.40	61.77	73.63	83.20	90.93
IPL (Text)	85.07	85.47	86.27	**87.03**	87.43	28.80	34.43	37.40	44.27	54.33	50.07	56.90	64.63	69.87	74.23	63.67	64.03	65.97	80.93	91.27
IPL (Unified)	85.10	85.40	86.17	86.87	87.47	**29.17**	**35.30**	39.37	46.93	**59.60**	**56.50**	60.03	66.07	69.60	75.03	**76.73**	82.60	86.30	88.23	**95.97**
Linear probe CLIP [30]	43.96	61.51	73.19	79.79	82.90	19.61	26.41	32.33	39.35	45.36	34.59	40.76	55.71	63.46	69.96	49.23	61.98	77.09	84.43	87.21
CoOp [45]	84.33	84.40	84.47	82.67	84.20	21.37	26.20	30.83	39.00	43.40	50.23	53.60	58.70	64.77	69.87	54.93	65.17	70.80	78.07	84.93
CoCoOp [44]	**85.65**	**86.22**	**86.88**	86.97	87.25	12.68	15.06	24.79	26.61	31.21	48.54	52.17	55.04	58.89	63.04	55.33	46.74	65.56	68.21	73.32
MaPLe [20]	80.50	81.47	81.77	83.60	85.33	26.73	30.90	34.87	42.00	48.40	52.13	55.50	61.00	66.50	71.33	71.80	78.30	84.50	87.73	92.33
PromptSRC [4]	84.87	85.70	86.17	86.90	**87.50**	27.67	31.70	37.47	43.27	50.83	56.23	59.97	65.53	69.87	72.73	73.13	79.37	86.30	**88.80**	92.43
MMA [39]	83.03	82.50	82.13	83.00	84.57	28.73	31.90	37.57	44.83	52.70	52.27	56.90	63.93	67.97	73.47	55.07	59.80	79.40	86.47	92.37
MMRL [15]	84.87	85.53	85.77	86.33	87.03	28.53	34.23	**40.47**	**48.07**	57.60	56.37	**61.37**	**67.87**	**71.60**	**75.30**	76.00	**82.87**	**87.67**	88.73	93.37

C Ablation Studies

C.1 Losses.

Table 6 studies the effect of each loss component in IPL using the unified variant on ImageNet[8]. The results show that the projection-preservation regularization losses, given by the vision- and text-side cosine losses $\mathcal{L}_{\text{cos}}^{v}$ and $\mathcal{L}_{\text{cos}}^{t}$, are important for improving generalization. The cross-entropy losses provide task-level supervision, with $\mathcal{L}_{\text{CE-rep}}$ further strengthening representation learning. The best performance is obtained when all losses are used together, indicating that task discrimination and preservation of interpretable concept directions are complementary for balancing base-class adaptation and novel-class generalization.

Table 6: Loss-component ablation on ImageNet for IPL (unified). ✓indicates the loss is active during training.

$\mathcal{L}_{\text{CE}}$	$\mathcal{L}_{\text{CE-rep}}$	$\mathcal{L}_{\text{cos}}^{v}$	$\mathcal{L}_{\text{cos}}^{t}$	Base	Novel	HM
✓	✗	✗	✗	72.43 ± 0.90	68.70 ± 0.10	70.52
✓	✓	✗	✗	77.93 ± 0.29	69.40 ± 0.44	73.42
✓	✗	✓	✗	72.27 ± 0.71	68.77 ± 0.29	70.47
✓	✗	✗	✓	72.47 ± 0.67	70.30 ± 0.26	71.37
✓	✓	✓	✗	77.93 ± 0.15	69.27 ± 0.15	73.34
✓	✓	✗	✓	78.27 ± 0.25	70.60 ± 0.20	74.24
✓	✗	✓	✓	72.50 ± 0.66	70.30 ± 0.26	71.38
✓	✓	✓	✓	**79.30** ± 0.20	**74.50** ± 0.10	**76.83**

C.2 Layers for prompt token insertion.

Figure 4 analyzes the effect of inserting IPL prompt tokens at different CLIP encoder layers on ImageNet. We observe that mid-to-late layer insertion performs best, with layers 6-12 achieving the strongest harmonic mean. Layers 3-12 show a similar trend, suggesting that interpretability-guided prompts are most effective when injected into semantic representations. In contrast, early-layer insertion leads to weaker generalization, likely because early layers encode lower-level visual features less suited for concept-guided adaptation.

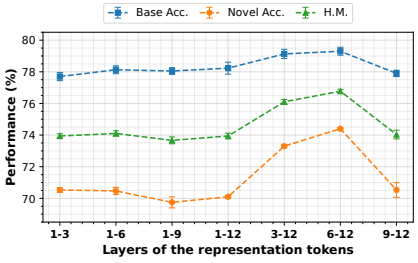

Figure 4: Effect of representation-token insertion layers on ImageNet base-to-novel generalization.

C.3 Different sizes of $k \in K$ for E concepts.

In this ablation, we analyze the impact of number of concept directions **E** used to contruct prompt tokens. From Figure 5, we observe that $E = 256$ achieves comparitively better harmonic mean, indicating that a compact set of high-ranked interpretable concept directions is sufficient for IPL's prompt token construction. performance comparitively degrades with larger **E**, suggesting that lower ranked concepts add noise rather than useful structure. We use $E = 256$ as the default in all main experiments.

C.4 Limitations analysis and future scope

IPL introduces additional compute relative to existing prompt-learning baselines. As reported in Table 7, IPL requires approximately 23.1 GMAC per forward pass compared to 20.9 GMAC for MMRL [15] and 20.5 GMAC for ArGue [35], a $\sim$10–13% increase, attributable entirely to the per-layer multi-head attention selectors and $r2v/r2t$ projection layers introduced for interpretability-guided prompt construction. We note that this overhead is a deliberate consequence of IPL's per-layer composition: the selector $\mathrm{MHA}^{(\ell)}$ and projections are instantiated independently at each rep-injection layer ℓ, which provides per-layer flexibility in how concepts are composed at different depths but multiplicatively scales prompt-side FLOPs with $|\mathcal{L}_{\text{rep}}|$. Several straightforward extensions can mitigate this cost: (i) sharing MHA and projection parameters across injection layers, which would reduce prompt-side FLOPs by roughly $7\times$ in our default configuration at the cost of less per-layer specialization; (ii) injecting at fewer layers, which our ablation (Section C.2) shows incurs only a small accuracy drop; or (iii) lower-rank parameterizations of the projection layers. A systematic study of these efficiency-accuracy trade-offs is left to future work.

Table 7: Computational cost on CLIP ViT-B/16 (single forward pass). *Prompt* columns report FLOPs introduced by each method's prompt-token machinery. Δ is the percentage increase in total GMAC over ArGue.

Method	**Vision** (GMAC)	**Text** (GMAC)	**Prompt** (GMAC)	**Total** (GMAC)	**Δ** (%)
ArGue [35]	17.563	2.980	0.000	20.543	—
MMRL [15]	17.833	3.096	0.023	20.951	+1.99
IPL (Vision)	17.833	3.096	2.203	23.132	+12.60
IPL (Text)	17.833	3.096	2.203	23.132	+12.60
IPL (Unified)	17.833	3.096	2.203	23.132	+12.60

D Hyperparameter details

Table 8: Ablation analysis of α and λ on base-to-new generalization.

α	Base	New	HM
0.3	76.30 ± 0.20	73.96 ± 0.16	75.11
0.5	77.63 ± 0.30	73.84 ± 0.08	75.69
0.7	$\mathbf{79.30 \pm 0.29}$	$\mathbf{74.50 \pm 0.06}$	**76.83**
0.9	77.41 ± 0.35	73.29 ± 0.10	75.29
1.0	76.73 ± 0.25	69.78 ± 0.15	73.09

λ	α	Base	New	H
0.01	0.7	77.70	71.10	74.25
0.1	0.7	77.70	73.03	75.29
0.2	0.7	77.67	71.07	74.22
0.5	0.7	77.70	73.00	75.28
3.0	0.7	77.60	**74.50**	**76.02**
4.0	0.7	78.67	71.90	75.13
5.0	0.7	77.67	73.10	75.32
6.0	0.7	**79.30**	70.83	74.83
7.0	0.7	77.53	71.83	74.57

IPL is trained for 10 epochs with batch size 4, SGD, and a cosine schedule peaking at lr 0.0035 after a one-epoch warmup. We use N=5 rep tokens injected at layers $\{6, ..., 12\}$, α=0.7, and a concept budget k of 256. Results are averaged over 3 seeds at 16 shots. From Table 8, we observe that IPL is most effective at $\alpha = 0.7$, which balances the standard and prompt-token cross-entropy objectives, and at $\lambda_{\text{proj}} = 3.0$, where the projection-preservation loss provides sufficient geometric regularization without overconstraining the projection layers.

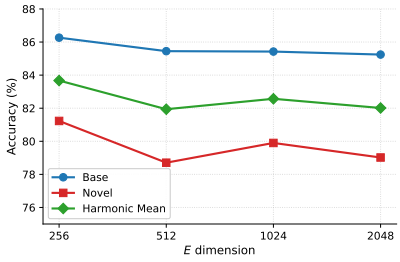

Figure 5: Effect of concept direction size $\mathbf{E}$ averaged across 11 datasets for base to novel generalization.

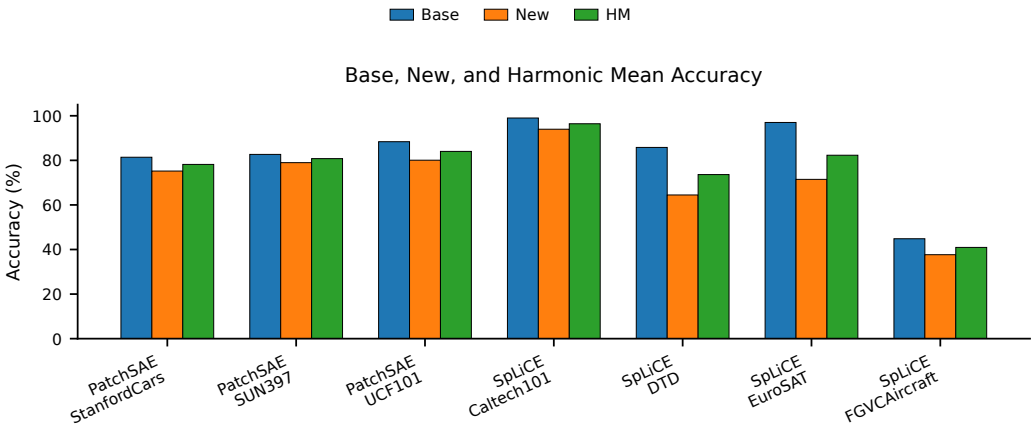

Figure 6: Performace of other interpretability tools like PatchSAE [23] and SpLiCE [2] on base to novel generalization for different datasets.

D.1 Results on other Interpretability Methods

In Figure 6, we study how different interpretability tools like PatchSAE[23] and splice [2] perform on base to novel generalization for different dataset. We observe that these tools perform on par with VL-SAE (vision).

D.2 Representation-token count.

Figure 7 shows that performance improves rapidly as the number of representation tokens increases from 1 to 5 and then saturates. We therefore use $N_{\text{rep}} = 5$ throughout the main experiments, since larger values provide only marginal gains while increasing prompt length.

E Detailed literature survey

Distribution shifts and dealing with novelty. Adapting models under distribution shift has a long history in classical machine learning, predating vision-language models. *Domain adaptation* (DA) methods minimize feature-distribution divergence between a labeled source and unlabeled target domain via discrepancy measures [13], adversarial alignment [34], or self-training on pseudo-labels [24]. *Domain generalization* (DG) extends this to the harder setting where the target domain is unseen at training time, with approaches based on domain-invariant feature learning [43], meta-learning across source domains [12], and feature augmentation or mixup [25]. *Few-shot learning* similarly has a rich literature on metric learning [1] and meta-learning [12] for adaptation under limited supervision.

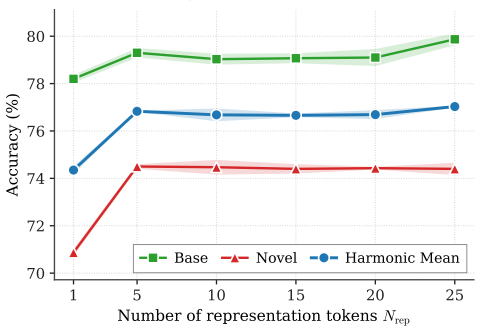

Figure 7: Effect of the number of representation tokens N_{rep} on ImageNet base-to-novel generalization. Performance saturates at $N_{\text{rep}} = 5$, which we use in all main experiments. Shaded bands denote $\pm$ std over 3 seeds.

These classical formulations target a fundamentally different setting from ours: they assume full access to model weights and adapt task-specific networks, often trained from scratch, by reshaping their feature space to bridge source-target gaps. In contrast, prompt-learning for vision-language models, the regime IPL operates in and adapts a single *frozen* foundation model (CLIP) by learning a small set of continuous prompt tokens, without modifying any backbone parameters. The challenge is therefore not to learn a domain-invariant

representation from data, but to steer a pretrained representation that is already broadly capable toward task-, domain-, or class-specific behavior without overwriting its zero-shot generalization. Recent work formalizes the various distributional axes along which this steering must succeed: base-to-novel generalization [44, 20, 4] (unseen classes within a dataset), domain generalization [31, 36, 18, 17] (input shifts at fixed label space), cross-dataset transfer [44] (zero-shot transfer to disjoint label spaces), and few-shot learning under limited supervision. Existing prompt-learning methods often optimize design choices for a single protocol [40, 4, 41, 35, 14]. IPL treats all four protocols as instances of a single underlying problem, adapting prompts under limited supervision while preserving CLIP's pretrained generalization, and addresses them with one mechanism: anchoring prompt tokens in interpretable concept directions discovered inside CLIP itself. This unified framing avoids per-protocol architectural changes, and is empirically borne out by IPL's consistent gains across all four evaluation regimes (Section 4.2).