REPROGRAMMING VISION-LANGUAGE MODELS VIA STRUCTURED PROMPT REPARAMETERIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual reprogramming adapts pretrained models to downstream tasks by modifying their input and output interfaces while keeping the backbone fixed. In vision-language models, existing methods mainly rely on intra-class prompt aggregation and do not explicitly model relationships among classes. However, fine-grained categories often exhibit highly overlapping attribute descriptions and strong inter-class correlation in the text embedding space, where discriminative cues lie in subtle low-variance components. We propose Reparameterized Inter-Class Visual Reprogramming (RVP), a structured framework that aggregates multiple text prompts within each class and applies residual correction across classes. We also show that CLIP-based visual reprogramming with input-independent linear output aggregation can be expressed as a linear mapping from frozen image embeddings to downstream logits, and use this view to design a structured reparameterization that models shared semantic components and class-specific differences. RVP uses only a single visual prompt and can be reparameterized at inference into a frozen backbone followed by a linear classifier, incurring nearly zero computational overhead. Across 11 few-shot classification benchmarks and four CLIP backbones, RVP consistently improves over prior visual reprogramming methods with comparable or better inference efficiency. [1]

1 INTRODUCTION

Model reprogramming (Vinod et al., 2020; Chen, 2024; Hung et al., 2023) adapts a pretrained model to downstream tasks by modifying its input and output interfaces while keeping the pretrained parameters fixed. In vision, this is often instantiated as visual reprogramming (VR) (Cai et al., 2024b;a; Tsao et al., 2024; Chen et al., 2023; Elsayed et al., 2018; Tsai et al., 2020), where a trainable input transformation is learned while the backbone remains frozen. For vision-language models (Radford et al., 2021; Jia et al., 2021), this paradigm is particularly appealing for few-shot adaptation because it preserves pretrained representations, requires only a small number of trainable parameters, and adds little inference overhead.

In CLIP-based VR, an input image is transformed by a visual prompt and encoded by the frozen image encoder, while downstream classes are represented by text embeddings from the frozen text encoder. Classification is then performed through image-text similarity. Recent methods improve this pipeline by introducing multiple textual descriptions for each class and aggregating their similarity scores (Cai et al., 2025a;b; Wu et al., 2026). However, these methods remain limited to intra-class prompt selection and do not explicitly model relationships among classes.

This limitation becomes severe in fine-grained recognition. As shown in Fig. 1, visually similar categories often share highly overlapping attribute descriptions, and prompt groups from different classes can exhibit high cosine similarity. This suggests that attribute prompts are often highly similar and that independent intra-class prompt selection is insufficient to resolve cross-class ambiguity. Moreover, the text embedding matrix exhibits a rapidly decaying singular value spectrum, indicating strong inter-class correlation and a low effective rank. In other words, many classes share dominant semantic directions, while the truly discriminative cues lie in subtle, low-variance components. These observations motivate explicit modeling of class relationships.

[1] Code will be publicly available after the paper is accepted.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

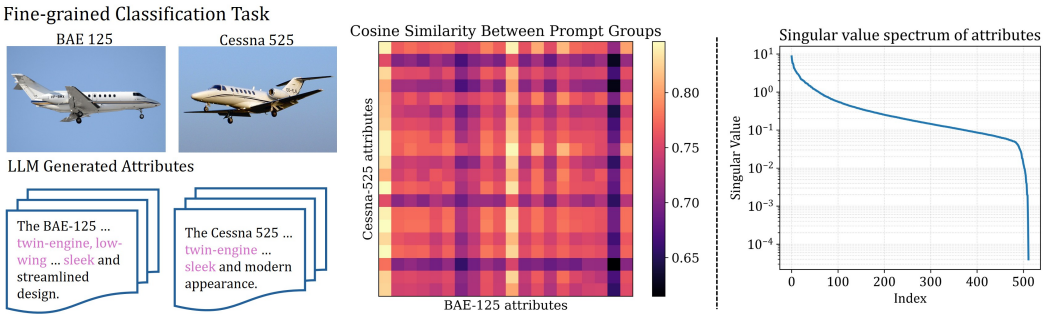


Figure 1: Fine-grained classification with attribute prompts. Prompt groups from visually similar classes show high cosine similarity, indicating that attribute prompts are highly similar. The rapidly decaying singular value spectrum of the text embedding matrix further reveals strong inter-class correlation, motivating explicit modeling of class relationships.

Based on this observation, we propose *Reparameterized Inter-Class Visual Reprogramming* (RVP), a structured framework that jointly models intra-class aggregation and inter-class residual correction. RVP first learns how to combine multiple attribute descriptions within each class, and then refines the resulting class logits through an inter-class relation matrix. This design is effective for two reasons. First, it can suppress shared semantic components and amplify subtle class-specific differences, which are exactly the cues that matter in fine-grained recognition. Second, by restricting the mapping to a structured residual form, RVP preserves the pretrained semantic subspace of CLIP and avoids the overfitting risk of dense unconstrained mappings. Unlike the prior Decoupled Visual Reprogramming (Cai et al., 2025b), which relies on multiple visual prompts and repeated backbone forward passes, RVP uses only a single visual prompt and can be reparameterized (Ding et al., 2021; Luo et al., 2023) into a frozen backbone followed by a single linear classifier, incurring nearly zero additional overhead at inference. Here, reparameterization means folding multiple parameter groups into one equivalent matrix.

In summary, our contributions are threefold. First, we show that, given a visual prompt, CLIP-based visual reprogramming reduces to a linear mapping from normalized CLIP image embeddings to downstream logits, providing a unified view of prompt aggregation and label mapping strategies. Second, we propose RVP, a reparameterizable inter-class modeling framework that introduces a structured parameterization of this mapping. Specifically, RVP first constructs class logits through text-embedding-based intra-class attribute aggregation, and then refines them with a residual inter-class correction matrix. Unlike unconstrained linear classifiers or generic logit adapters, RVP keeps the classifier anchored in the CLIP text-embedding subspace while reducing confusion among fine-grained classes. Third, through extensive experiments, we show that this structured inter-class parameterization improves few-shot visual reprogramming while preserving efficient inference.

In Section 3, we show that, given a visual prompt, CLIP-based visual reprogramming induces a linear mapping $\phi : \mathbb{R}^D \to \mathbb{R}^C$ from normalized reprogrammed-image embeddings to downstream logits, which enables reparameterizable inter-class modeling. In Section 4, we present RVP, including its training-time formulation and exact inference-time reparameterization into a single linear classifier. Sections 5 and 6 then present the experimental setup and results, showing that RVP consistently improves performance across multiple CLIP backbones, especially on fine-grained datasets such as Aircraft and Cars, while maintaining comparable or lower inference cost.

2 Related Work

Model Reprogramming. Model reprogramming (Chen, 2024) adapts pretrained models to downstream tasks by learning transformations at the input and output interfaces, without modifying internal parameters. This strategy preserves pretrained knowledge and avoids catastrophic forgetting (Kirkpatrick et al., 2017), while enabling architecture-agnostic transfer with few trainable parameters. It has been applied to vision (Chen et al., 2023; Tsai et al., 2020; Cai et al., 2024a; Jin et al., 2025), graph (Jing et al., 2023), acoustic (Yang et al., 2021; 2023; Hung et al., 2023; Yen et al., 2023), and

language models (Hambardzumyan et al., 2021; Vinod et al., 2020; Jin et al., 2024), with recent work studying its robustness (Chen et al., 2025; Zhou et al., 2025).

Input visual reprogramming (VR) (Cai et al., 2024b;a; Chen et al., 2023) is a common instantiation for image classification, where learnable patterns are injected into the input space. Typical designs include padded regions (Chen et al., 2023; Tsai et al., 2020; Tsao et al., 2024) or watermark-style perturbations (Bahng et al., 2022; Oh et al., 2023), and have been successfully extended to vision–language models (Oh et al., 2023; Zhang et al., 2024).

Prompt Learning. Prompt learning introduces trainable parameters directly into a pretrained model, often in an architecture-dependent manner. Prompts can take the form of textual tokens (Zhou et al., 2022b;a), visual prompts on images (Chen et al., 2023; Oh et al., 2023; Tsao et al., 2024), internal token prompts (Wang et al., 2023), or cross-modal mappings (Khattak et al., 2023).

Applying visual prompts to images for adapting VLMs is functionally equivalent to VR. Existing methods typically learn a single shared prompt, such as watermark overlays (Bahng et al., 2022), padded patterns (Tsai et al., 2020; Chen et al., 2023), BlackVIP (Oh et al., 2023), DAM (Huang et al., 2023). Recently, AttrVR (Cai et al., 2025a) incorporates multiple attribute prompts for each class to improve image–text alignment. DVP (Cai et al., 2025b) learns multiple prompts with distinct roles, improving learning capacity.

Feature Adapter. Few-shot CLIP adaptation can modify either features or classifiers. CLIP-Adapter (Gao et al., 2024), Tip-Adapter (Zhang et al., 2022), and Proto-CLIP (P et al., 2024) incorporate downstream visual features, while LDC (Li et al., 2025) further combines multi-level feature adaptation with sample-dependent logit correction. TaskRes (Yu et al., 2023) learns a residual directly in the classifier space, and LP++ (Huang et al., 2024) constructs text-informed classifiers from visual prototypes and text embeddings. In contrast, visual reprogramming preserves the pretrained model and adapts the task interface through input prompting and output label mapping. RVP strengthens the latter by learning structured transformations over text-derived responses.

3 Preliminaries and Insights

CLIP (Radford et al., 2021) consists of an image encoder f_{img} and a text encoder f_{txt}, which map inputs into a shared embedding space $\mathcal{Z} \subseteq \mathbb{R}^D$, where D is the embedding dimension. Let $\mathcal{X}^{\text{S}}$ denote the source image space of CLIP, $\mathcal{V}$ the text space, $x^{\text{S}} \in \mathcal{X}^{\text{S}}$ an input image, and $V \in \mathcal{V}$ a text description. The corresponding ℓ_2-normalized image and text embeddings are

$$\hat{\mathbf{v}} = \frac{f_{\text{img}}(x^{\text{S}})}{\|f_{\text{img}}(x^{\text{S}})\|_2}, \quad \hat{\mathbf{t}} = \frac{f_{\text{txt}}(V)}{\|f_{\text{txt}}(V)\|_2}. \tag{1}$$

CLIP computes the image–text similarity as $f_{\text{clip}}(x^{\text{S}}, V) = \frac{1}{\tau}\hat{\mathbf{v}}^\top \hat{\mathbf{t}}$, where τ is the temperature.

For a downstream task defined on $\mathcal{X}^{\text{T}} \times \mathcal{Y}^{\text{T}}$, let $x^{\text{T}} \in \mathcal{X}^{\text{T}}$ denote an input image and $\mathcal{Y}^{\text{T}} = \{1, \dots, C\}$ the label set with C classes. Each class $y^{\text{T}} \in \mathcal{Y}^{\text{T}}$ is represented by a set of textual descriptions $\mathcal{A}(y^{\text{T}}) \subseteq \mathcal{V}$, and let $\mathcal{A} = \bigcup_{y^{\text{T}} \in \mathcal{Y}^{\text{T}}} \mathcal{A}(y^{\text{T}})$. The class logit is computed by aggregating similarity scores:

$$\left[f_{\text{logits}}(x^{\text{T}}; \mathcal{A})\right]_{y^{\text{T}}} = \text{agg}_{a \in \mathcal{A}(y^{\text{T}})}\, f_{\text{clip}}(x^{\text{T}}, a), \tag{2}$$

where $\text{agg}(\cdot)$ is an aggregation operator (Cai et al., 2025b).

Visual reprogramming adapts downstream images to a frozen pretrained model by learning an input transformation instead of modifying model parameters. Specifically, a trainable transformation $f_{\text{in}}(\cdot \mid \delta)$, parameterized by a visual prompt δ, maps the downstream image x^{T} to a reprogrammed input that is compatible with CLIP. The reprogrammed image is then processed by the frozen CLIP encoders.

For a fixed image x^{T}, let $T \in \mathbb{R}^{|\mathcal{A}| \times D}$ be the matrix of stacked normalized text embeddings: $T = [\hat{\mathbf{t}}_{a_1}, \dots, \hat{\mathbf{t}}_{a_{|\mathcal{A}|}}]^\top$. The similarity scores over all descriptions can then be written as

$$M_a = \frac{1}{\tau} T \hat{\mathbf{v}}.$$

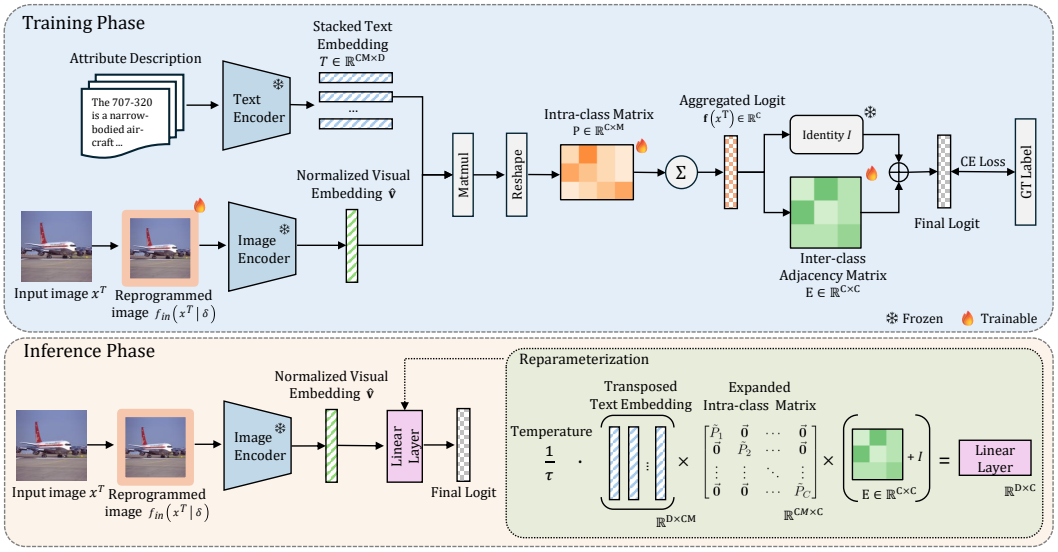


Figure 2: Overview of Reparameterized Inter-Class Visual Reprogramming (RVP). During training, a visual prompt transforms the input image, which is then encoded by a frozen image encoder. The cosine similarities between the visual embedding and text embeddings are aggregated by a learnable intra-class matrix to produce class logits, which are further refined by a residual class-relation matrix that captures inter-class dependencies. At inference, the text embeddings and label mapping are reparameterized into a single linear classifier, enabling efficient prediction with a single forward pass.

From Similarity Scores to Class Logits. For aggregation operators that can be represented by input-independent linear weights, the description-level similarity scores can be mapped to class logits through $\mathbf{M}_y = \mathbf{M}_a^\top \boldsymbol{\omega}$. where $\omega \in \mathbb{R}^{|\mathcal{A}| \times C}$ is a reweighting matrix induced by $\mathrm{agg}(\cdot)$. Substituting M_a, we obtain $M_y = \frac{1}{\tau}\hat{\mathbf{v}}^\top T^\top \omega = \phi(\hat{\mathbf{v}})$, where $\phi : \mathbb{R}^D \to \mathbb{R}^C$ denotes the mapping from the normalized image embedding to downstream class logits. The predicted class is then obtained by selecting the largest logit in M_y.

4 Reparameterized Inter-Class Visual Reprogramming

Following recent advances (Cai et al., 2025a;b), we use multiple attribute-based textual descriptions to enrich the semantic space, with M descriptions for each of the C downstream classes. However, as discussed earlier, fine-grained categories often exhibit high similarity in the text embedding space, indicating strong inter-class correlations. Methods based solely on independent intra-class prompt selection cannot disentangle these shared dominant semantic directions. To address this limitation, we propose the Reparameterized Inter-Class Visual Reprogramming (RVP) framework.

Fig. 2 illustrates the overall architecture of RVP. RVP jointly models intra-class description weighting and inter-class interactions while keeping the CLIP backbone frozen. The intra-class component refines class prototypes by learning how to aggregate attribute descriptions within each class, while the inter-class component enhances discriminative cues through message passing across classes. Instead of learning a dense $CM \times C$ mapping from all descriptions to target classes, which introduces C^2M trainable parameters and is prone to overfitting in few-shot settings, RVP adopts an explicitly structured formulation with only $C \times M$ parameters for intra-class weighting and $C \times C$ parameters for inter-class modeling. This reduced parameterization imposes a useful structural prior and improves generalization. In addition, unlike Decoupled Visual Prompting (Cai et al., 2025b), which ensembles multiple visual prompts and therefore requires multiple forward passes through the image encoder, RVP uses a single visual prompt and only one forward pass.

4.1 Training Phase: Dynamic Aggregation and Message Passing

Given a downstream image $x^{\mathrm{T}} \in \mathcal{X}^{\mathrm{T}}$, we first apply a trainable visual transformation $f_{\mathrm{in}}(\cdot \mid \delta)$ to map it into CLIP's input space, where the transformation is implemented as resizing and boundary

padding parameterized by a visual prompt δ (Cai et al., 2024b). The reprogrammed image is then fed into the frozen CLIP image encoder to obtain the ℓ_2-normalized feature vector $\hat{\mathbf{v}}$. Let $T \in \mathbb{R}^{CM \times D}$ be the stacked matrix of normalized text embeddings, assuming M textual descriptions for each of the C downstream classes. To refine intra-class prompt selection, we introduce a learnable intra-class weighting matrix $P \in \mathbb{R}^{C \times M}$. We apply a softmax over the description dimension to obtain normalized weights $\tilde{P}_c = \mathrm{softmax}(P_c)$, where P_c denotes the c-th row of P. The aggregated base logit for class c is computed as

$$f_c(x^{\mathrm{T}}) = \sum_{m=1}^{M} \tilde{P}_{c,m} \frac{1}{\tau} \hat{\mathbf{v}}^{\top} \hat{\mathbf{t}}_{c,m}, \tag{3}$$

where $\hat{\mathbf{t}}_{c,m} \in \mathbb{R}^D$ is the normalized embedding of the m-th description for class c, and $\tilde{P}_{c,m}$ is its corresponding normalized weight.

How to Model Inter-class Relationships. In fine-grained tasks, textual descriptions from different classes are often highly similar and may share dominant semantic directions. As a result, even after aggregating multiple descriptions within each class, the base logits are still formed independently across classes and can retain strong cross-class ambiguity: semantically related but incorrect classes may receive high scores because shared text semantics are not explicitly suppressed.

Let $\mathbf{f}(x^{\mathrm{T}}) = [f_1(x^{\mathrm{T}}), \ldots, f_C(x^{\mathrm{T}})] \in \mathbb{R}^C$ denote the row vector of base logits. To model these dependencies, we introduce an inter-class adjacency matrix $E \in \mathbb{R}^{C \times C}$ and formulate the final class logits $\mathbf{z} \in \mathbb{R}^C$ as a residual graph message-passing step:

$$\mathbf{z} = \mathbf{f}(x^{\mathrm{T}}) + \mathbf{f}(x^{\mathrm{T}})E = \mathbf{f}(x^{\mathrm{T}})(I + E). \tag{4}$$

Here, the term $\mathbf{f}(x^{\mathrm{T}})E$ models how evidence should be redistributed across correlated classes, allowing RVP to suppress shared semantic components and enhance subtle class-specific differences that cannot be recovered from independent intra-class aggregation alone.

The residual formulation in equation 4 provides an important structural prior. Instead of learning a dense unconstrained transformation from scratch, the identity matrix I preserves the original class logits induced by CLIP's pretrained alignment, while the matrix E, initialized to zero, learns only residual corrections between classes. This design constrains the model to refine, rather than overwrite, the pretrained semantic structure, making optimization easier and more stable in the few-shot regime. As a result, it reduces the effective complexity of the mapping and helps the model focus on the subtle discriminative differences among correlated classes.

All trainable components in RVP are learned jointly with the downstream cross-entropy loss. Specifically, the visual prompt δ, intra-class matrix P, and inter-class matrix E are optimized end-to-end, while the CLIP image and text encoders remain frozen. The final logits $\mathbf{z}_i$ are supervised by

$$\mathcal{L}_{\mathrm{CE}}(\delta, P, E) = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(z_{i,y_i^T})}{\sum_{c=1}^{C} \exp(z_{i,c})}, \tag{5}$$

where $z_{i,c}$ is the logit of class c for sample i, and z_{i,y_i^T} is the logit of its ground-truth class.

4.2 Inference Phase: Linear Reparameterization

During inference, the intra-class aggregation and inter-class message-passing can be completely absorbed into a single projection matrix. We construct a sparse routing matrix $W_1 \in \mathbb{R}^{CM \times C}$ as a block-diagonal matrix:

$$W_1 = \begin{bmatrix} \tilde{P}_1 & \vec{\mathbf{0}} & \cdots & \vec{\mathbf{0}} \\ \vec{\mathbf{0}} & \tilde{P}_2 & \cdots & \vec{\mathbf{0}} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{\mathbf{0}} & \vec{\mathbf{0}} & \cdots & \tilde{P}_C \end{bmatrix} \in \mathbb{R}^{CM \times C}, \tag{6}$$

where each $\tilde{P}_c = [\tilde{P}_{c,1}, \ldots, \tilde{P}_{c,M}]^{\top} \in \mathbb{R}^M$ is the normalized weight vector for the M descriptions of class c, and $\vec{\mathbf{0}}$ denotes a zero column vector of length M. By leveraging the associativity of

matrix multiplication, we pre-calculate a unified classifier matrix $\hat{W} \in \mathbb{R}^{D \times C}$ that absorbs the text embeddings, routing weights, and inter-class correlations:

$$\hat{W} = \frac{1}{\tau} T^\top W_1 (I + E). \tag{7}$$

The detailed structure of this reparameterized head can be unrolled as:

$$\hat{W} = \frac{1}{\tau} \underbrace{\begin{pmatrix} | & | & & | \\ \hat{\mathbf{t}}_{1,1} & \hat{\mathbf{t}}_{1,2} & \cdots & \hat{\mathbf{t}}_{C,M} \\ | & | & & | \end{pmatrix}}_{T^\top \in \mathbb{R}^{D \times CM}} \underbrace{\begin{pmatrix} \tilde{P}_1 & \vec{\mathbf{0}} & \cdots \\ \vec{\mathbf{0}} & \tilde{P}_2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}}_{W_1 \in \mathbb{R}^{CM \times C}} \underbrace{(I + E)}_{\mathbb{R}^{C \times C}}. \tag{8}$$

Consequently, the entire forward pass during inference is reduced to generating the single visual prompt, extracting the normalized image embedding via exactly one pass through the backbone, and performing a single linear projection: $\mathbf{z} = \hat{\mathbf{v}}^\top \hat{W}$.

This exact reparameterization replaces the standard aggregation weights with $\hat{W}$, guaranteeing that RVP identifies a more expressive mapping ϕ while adding nearly zero overhead during inference.

5 Experiments

Experimental Setup and Benchmarks. To evaluate the proposed RVP framework, we adhere to the established experimental protocol from (Cai et al., 2025b). We conduct all experiments using pretrained CLIP models across four image encoder architectures, including variants of ResNet (He et al., 2016) and Vision Transformer (ViT) (Dosovitskiy et al., 2021), under a 16-shot downstream classification setting. All reported results represent the average accuracy across three independent random seeds. Our benchmark suite comprises 11 datasets covering diverse visual domains, including textures, actions, and natural scenes. All datasets are publicly available: FGVC Aircraft (Aircraft) (Maji et al., 2013), Caltech101 (Caltech) (Fei-Fei et al., 2004), StanfordCars (Cars) (Krause et al., 2013), Describable Textures Dataset (DTD) (Cimpoi et al., 2014), EuroSAT (ESAT) (Helber et al., 2019), Flowers102 (Flowers) (Nilsback & Zisserman, 2008), Food101 (Food) (Bossard et al., 2014), OxfordPets (Pets) (Parkhi et al., 2012), SUN397 (SUN) (Xiao et al., 2010), UCF101 (UCF) (Soomro et al., 2012), and RESISC45 (Resisc) (Cheng et al., 2017). More implementation details can be found in Section D.

6 Results

Quantitative Results. We compare RVP against four prominent visual reprogramming (VR) baselines: (1) **VP** (Bahng et al., 2022), a standard VR approach that overlays learnable pixel perturbations onto rescaled downstream images; (2) **AR** (Tsai et al., 2020; Chen et al., 2023), which pads learnable noise parameters around the image boundary; (3) **AttrVR** (Cai et al., 2025a), which guides the learning of visual prompt patterns using class-specific attribute descriptions; and (4) **DVP** (Cai et al., 2025b), a decoupled visual prompting framework that ensembles multiple reprogrammed inputs. For a strictly fair comparison, we evaluate the unsupervised clustering variant of DVP (DVP-cls), which isolates the performance of the reprogramming mechanism without relying on external Large Language Models (LLMs) to generate cause-specific descriptions. RVP uses the same text prompt set as DVP.

As shown in Tables 1 to 3, RVP achieves the highest average accuracy across all evaluated backbones. With ViT-B/16 CLIP (Table 1), RVP attains the highest average accuracy of 82.7%, outperforming DVP (Cai et al., 2025b) by 3.0 points and AttrVR (Cai et al., 2025a) by 4.2 points. It achieves the best result on 9 out of 11 datasets, with especially large gains on fine-grained benchmarks such as Aircraft ($+7.4$ over DVP) and Cars ($+14.0$). These improvements are consistent with our motivation: in fine-grained recognition, many categories share highly similar semantic attributes, and the main discriminative cues arise from subtle inter-class differences. In this regime, intra-class prompt selection alone is insufficient, while RVP can explicitly suppress shared semantic components and amplify class-specific distinctions through inter-class modeling.

Table 1: Accuracy comparison of different methods trained on 16-shot downstream classification tasks, using ViT-B/16-based CLIP as the pretrained model (Mean % ± Std %, ours are highlighted and the highest result is in **bold**). Other results are taken from prior work (Cai et al., 2025b).

METHOD	AIRCRAFT	CALTECH	CARS	DTD	ESAT	FLOWERS	FOOD	PETS	SUN	UCF	RESISC	AVG.
VP	32.1	93.5	65.5	61.4	91.2	82.5	82.3	91.0	65.8	73.8	79.1	74.4
AR	31.7	95.5	68.0	62.0	93.4	85.9	85.2	92.7	67.9	78.1	81.6	76.5
ATTRVR	36.6	95.7	68.3	65.6	93.8	92.9	**85.9**	93.3	69.6	79.0	82.6	78.5
DVP	38.7	96.0	70.8	65.5	**94.1**	95.0	85.7	93.3	71.1	82.0	84.4	79.7
RVP	**46.1**±0.2	**96.5**±0.2	**84.8**±0.3	**68.7**±0.1	92.7±0.2	**96.7**±0.2	85.5±0.1	**94.0**±0.1	**73.8**±0.1	**85.1**±0.7	**85.9**±0.5	**82.7**

Table 2: Accuracy comparison of different methods trained on 16-shot downstream classification tasks, using RN50-based CLIP as the pretrained model (Mean % ± Std %, ours are highlighted and the highest result is in **bold**). Other results are taken from prior work (Cai et al., 2025b).

METHOD	AIRCRAFT	CALTECH	CARS	DTD	ESAT	FLOWERS	FOOD	PETS	SUN	UCF	RESISC	AVG.
VP	16.2	80.1	44.0	43.4	59.7	53.6	65.3	77.2	48.8	52.0	47.7	53.5
AR	18.6	86.5	53.9	46.4	66.6	60.9	74.2	82.5	56.8	59.7	58.4	60.4
ATTRVR	20.7	89.1	53.9	54.4	72.0	74.8	**75.3**	88.9	59.9	63.6	58.2	64.6
DVP	22.1	89.8	54.5	55.9	72.2	80.0	75.0	88.9	61.1	65.9	60.8	66.0
RVP	**29.1**±0.3	**92.0**±0.0	**71.0**±0.3	**62.0**±0.3	**72.9**±0.9	**91.6**±0.2	73.7±0.0	**89.7**±0.1	**66.2**±0.1	**74.3**±0.1	**72.9**±0.1	**72.3**

The advantage of RVP becomes even more pronounced with the weaker RN50 backbone. In Table 2, RVP achieves an average accuracy of 72.3%, exceeding DVP by 6.3 points and AttrVR by 7.7 points. It again shows particularly strong gains on Aircraft and Cars, indicating that the proposed structured reparameterization remains effective even when the underlying visual encoder is less expressive.

More broadly, Table 3 shows that the advantage of RVP becomes larger as the visual backbone becomes weaker. Compared with DVP, the gain is +6.3 on RN50, +4.2 on RN101, +4.8 on ViT-B/32, and +3.0 on ViT-B/16. This trend is expected, since weaker pretrained encoders produce less separable features, making downstream classification more reliant on the quality of the label mapping. In this setting, a structured mapping that explicitly models intra-class aggregation and inter-class relationships becomes more effective, as it can recover discriminative information that is not well separated in the original feature space. In contrast, stronger backbones such as ViT-B/16 already provide more discriminative and semantically aligned embeddings, leaving less room for improvement.

Table 3: Average accuracy of different VR methods on 11 datasets using different CLIP visual encoders (mean accuracy in %; ours are highlighted and the highest is in **bold**; RN denotes ResNet).

METHOD	RN50	RN101	VIT-B/32	VIT-B/16
VP	53.5	57.5	68.3	74.4
AR	60.4	62.7	66.3	76.5
ATTRVR	64.6	67.2	69.8	78.5
DVP	66.0	68.8	71.0	79.7
RVP	**72.3**	**73.0**	**75.8**	**82.7**

Few-shot Classification Performance. Fig. 3 compares few-shot performance on the Aircraft dataset under 1, 4, 8, 16, and 32 training samples per class. RVP achieves the best accuracy at every shot setting and shows a clear advantage over all prior visual reprogramming baselines. The improvement is modest in the extreme 1-shot setting, but becomes much larger as more labeled samples are available. At 32-shot, RVP attains 50.7%, substantially outperforming DVP (41.2%) and AttrVR (38.4%).

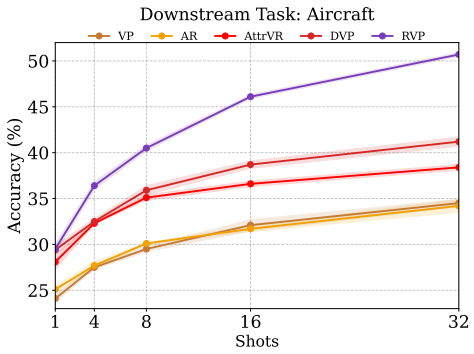


Figure 3: Accuracy comparison across different shot settings on Aircraft using ViT-B/16 CLIP. RVP consistently outperforms prior VR methods across all shot numbers. Shaded regions indicate standard deviation.

This trend suggests that the proposed structured mapping can make better use of additional supervision than existing methods. While all approaches improve as the number of shots increases, the gain of RVP is much steeper, indicating stronger scalability from low-shot to moderately supervised settings. The relatively small standard deviations across all shot numbers also show that the improvements are stable over repeated runs. Overall, the figure shows that

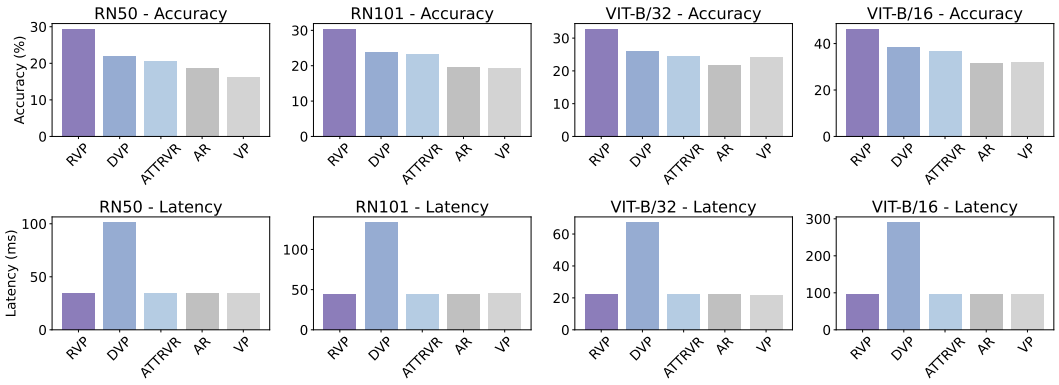


Figure 4: Accuracy and latency comparison on FGVC Aircraft across different backbones. Our method consistently achieves the highest accuracy while maintaining low latency, demonstrating a favorable trade-off between performance and efficiency.

Table 4: Ablation studies of RVP using a ViT-B/16-based CLIP backbone. The complete method is highlighted, and the best results are shown in **bold**.

METHOD	AIRCRAFT	CALTECH	CARS	DTD	ESAT	FLOWERS	FOOD	PETS	SUN	UCF	RESISC	AVG.
RVP	**46.1**	**96.5**	**84.8**	**68.7**	92.7	**96.7**	85.5	**94.0**	**73.8**	**85.1**	**85.9**	**82.7**
W/O VR	40.1	96.2	81.7	65.7	58.5	96.8	84.7	93.9	74.2	83.6	82.8	78.0
W/O INTRA-CLASS P	46.0	96.3	84.7	68.3	92.6	96.8	85.5	94.1	73.8	84.4	85.6	82.6
W/O INTER-CLASS E	35.6	96.1	68.0	63.3	**93.8**	91.7	**85.6**	93.1	67.4	78.9	83.5	77.9
LINEAR PROBE	37.2	93.9	73.5	63.5	84.2	92.2	79.5	85.6	69.1	76.8	83.5	76.3
ATTRIBUTE & LP	37.4	90.9	78.1	56.4	59.4	86.2	73.9	90.7	67.9	68.7	76.6	71.5

explicitly modeling both intra-class aggregation and inter-class relationships leads to more effective few-shot adaptation on fine-grained recognition tasks.

Accuracy and Inference Speed. RVP not only improves classification accuracy, but also retains the efficiency advantage of standard visual reprogramming. As shown in Fig. 4, RVP consistently achieves the highest accuracy on FGVC Aircraft across all four backbones, while maintaining latency close to single-prompt methods such as VP, AR, and AttrVR. In contrast, DVP relies on multiple decoupled visual prompts and therefore requires multiple forward passes through the frozen image encoder, leading to substantially higher inference latency. By using only a single visual prompt and reparameterizing the output mapping into one linear classifier, RVP avoids this overhead while still delivering stronger performance. This result shows that the gain of RVP comes from a more effective structured label mapping rather than increased inference-time computation.

Ablation Study. We conduct an ablation study in Table 4 using a ViT-B/16-based CLIP backbone. In addition to the full RVP model, we evaluate five variants: w/o VR, which removes visual reprogramming and classifies zero-padded images only; w/o intra-class P, which replaces the learnable intra-class weights with uniform averaging; w/o inter-class E, which removes inter-class modeling; Linear Probe, which trains a linear classifier on the visual embedding; and Attribute & LP, which learns a dense $CM \times C$ prompt-reweighting matrix over all text descriptions. Linear Probe and Attribute & LP are trained together with the visual prompt.

The complete RVP achieves the best average accuracy of 82.7%, confirming that visual reprogramming, intra-class aggregation, and inter-class modeling work best together. Removing VR reduces the average accuracy to 78.0%, with large drops on Aircraft, Cars, DTD, and especially ESAT, showing that input adaptation remains important. Removing the inter-class matrix E reduces the average accuracy to 77.9%, which is the largest drop among all architectural ablations. The effect is especially strong on Aircraft and Cars, where classes share many attributes and differ only in subtle details. This shows that explicit inter-class modeling is the main source of improvement in RVP.

The comparisons with Linear Probe and Attribute & LP further highlight the value of structured reparameterization. Linear Probe reaches 76.3%, while Attribute & LP performs even worse at

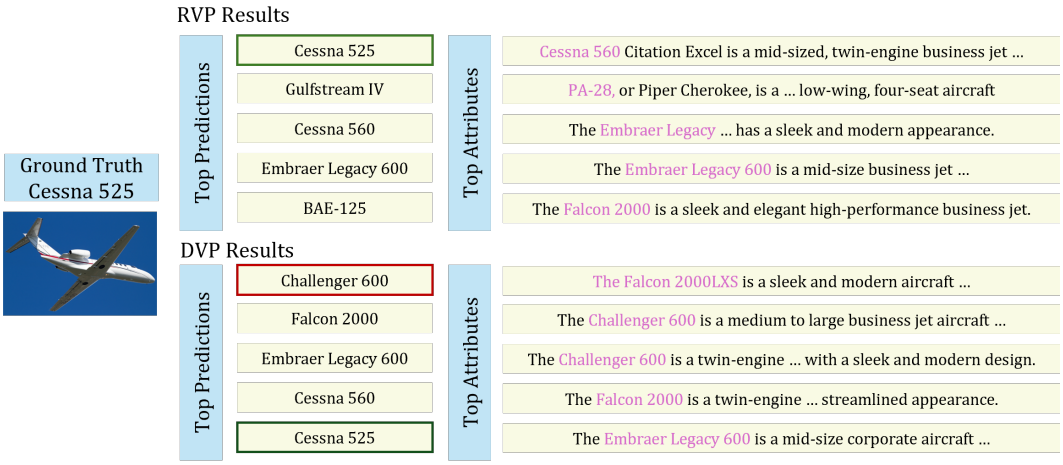


Figure 5: Top predicted classes and highest-matching attributes for a test image. For both RVP and DVP, the most similar attributes include prompts from other classes, reflecting strong semantic overlap in fine-grained recognition. However, RVP explicitly models inter-class relationships, allowing it to better resolve these cross-class ambiguities and produce the correct prediction.

71.5%, despite using a dense mapping with more parameters. This indicates that simply increasing trainable parameter size can lead to overfitting without structural constraints.

Overall, the ablation results show a clear pattern: the inter-class module E is the most important component, the visual prompt is also essential, and the intra-class matrix P provides a smaller but consistent gain. These results support the design of RVP and show that its advantage comes from structured modeling rather than a larger unconstrained classifier.

Qualitative Analysis. Fig. 5 provides a qualitative comparison between RVP and DVP on a fine-grained aircraft example. In both methods, the highest-matching attributes include prompts from semantically similar but incorrect classes. This is expected because attribute prompts are largely class-agnostic: descriptions such as *sleek*, *streamlined*, or *twin-engine* are often shared by multiple aircraft categories and therefore cannot uniquely identify a class on their own. As a result, relying only on prompt-level similarity can lead to ambiguous predictions. RVP addresses this issue by explicitly modeling inter-class relationships, allowing it to suppress misleading shared semantics and rank the ground-truth class Cessna 525 at the top, whereas DVP places it only in the fifth position.

7 Limitations

Although RVP performs strongly on most benchmarks, its performance is less pronounced on Food101. Unlike fine-grained object categories, food images often include not only the main dish but also side dishes, garnish, sauces, and other accompanying ingredients, which makes class semantics less stable across samples. This high intra-class variation weakens the consistency of text-based class relationships and reduces the advantage of our structured inter-class modeling. A more detailed discussion is provided in Section B.6.

8 Conclusion

We proposed Reparameterized Inter-Class Visual Reprogramming (RVP), which adapts frozen vision–language models through structured intra-class aggregation and inter-class modeling. RVP admits exact reparameterization into a single linear classifier at inference. Across 11 benchmarks and multiple CLIP backbones, RVP consistently improves over prior visual reprogramming methods, with the largest gains on fine-grained tasks. These results highlight the benefit of modeling class relationships in visual reprogramming and motivate extending RVP beyond CLIP. We will explore the application of RVP beyond CLIP in future work.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, LaTeX assistance, and table formatting. We have not used generative AI tools for generate or alter data, run experiments, replace author decisions about the method, or make unsupported scientific claims. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses established image-classification benchmarks and frozen pretrained models and involves no human-subject data collection, personally identifiable information, or newly collected sensitive data. As with other transfer methods, performance in deployment can inherit biases from the source models and datasets. We therefore recommend downstream validation before deployment and report the data sources and evaluation protocols used in this work.

REPRODUCIBILITY STATEMENT

The paper and appendix provide the complete method specification, pseudocode, dataset and split descriptions, preprocessing choices, model and optimization settings, validation protocol, covariance estimation details, complexity analysis, proofs, and per-dataset results. All reported results use fixed random seeds and the tables report the corresponding means and standard deviations. Upon acceptance, we will release the complete source code. configuration files, preprocessing and evaluation scripts, and the full set of trained-run metadata needed to reproduce the experiments.

REFERENCES

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv*, 2022.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014.

Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Bayesian-guided label mapping for visual reprogramming. In *NeurIPS*, 2024a.

Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Sample-specific masks for visual reprogramming-based prompting. In *ICML*, 2024b.

Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Attribute-based visual reprogramming for vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=j964C6y92q.

Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Understanding model reprogramming for clip via decoupling visual prompts. In *International Conference on Machine Learning*, 2025b.

Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and improving visual prompting: A label-mapping perspective. In *CVPR*, 2023.

Pin-Yu Chen. Model reprogramming: Resource-efficient cross-domain machine learning. In *AAAI*, 2024.

Yukun Chen, Shuo Shao, Enhao Huang, Yiming Li, Pin-Yu Chen, Zhan Qin, and Kui Ren. Refine: Inversion-free backdoor defense via model reprogramming. In *ICLR*, 2025.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 2017.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13728–13737, 2021. doi: 10.1109/CVPR46437.2021.01352.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. In *ICLR*, 2018.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR workshop*, 2004.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024. ISSN 0920-5691. doi: 10.1007/s11263-023-01891-x. URL https://doi.org/10.1007/s11263-023-01891-x.

K Hambardzumyan, H Khachatrian, and J May. Warp: Word-level adversarial reprogramming. In *ACL-IJCNLP*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

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*, 2019.

Qidong Huang, Xiaoyi Dong, Dongdong Chen, Weiming Zhang, Feifei Wang, Gang Hua, and Nenghai Yu. Diversity-aware meta visual prompting. In *CVPR*, 2023.

Yunshi Huang, Fereshteh Shakeri, Jose Dolz, Malik Boudiaf, Houda Bahig, and Ismail Ben Ayed. Lp++: A surprisingly strong linear probe for few-shot clip. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yun-Ning Hung, Chao-Han Huck Yang, Pin-Yu Chen, and Alexander Lerch. Low-resource music genre classification with cross-modal neural model reprogramming. In *ICASSP*, 2023.

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 Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4904–4916. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/jia21b.html.

Can Jin, Ying Li, Mingyu Zhao, Shiyu Zhao, Zhenting Wang, Xiaoxiao He, Ligong Han, Tong Che, and Dimitris N. Metaxas. Lor-VP: Low-rank visual prompting for efficient vision model adaptation. In *ICLR*, 2025.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *ICLR*, 2024.

Yongcheng Jing, Chongbin Yuan, Li Ju, Yiding Yang, Xinchao Wang, and Dacheng Tao. Deep graph reprogramming. In *CVPR*, 2023.

Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *CVPR*, 2023.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 2017.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshops*, 2013.

Shuo Li, Fang Liu, Zehua Hao, Xinyi Wang, Lingling Li, Xu Liu, Puhua Chen, and Wenping Ma. Logits deconfusion with clip for few-shot learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 25411–25421, June 2025.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Gen Luo, Minglang Huang, Yiyi Zhou, Xiaoshuai Sun, Guannan Jiang, Zhiyu Wang, and Rongrong Ji. Towards efficient visual adaption via structural re-parameterization, 2023. URL https://arxiv.org/abs/2302.08106.

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

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

Changdae Oh, Hyeji Hwang, Hee-young Lee, YongTaek Lim, Geunyoung Jung, Jiyoung Jung, Hosik Choi, and Kyungwoo Song. Blackvip: Black-box visual prompting for robust transfer learning. In *CVPR*, 2023.

Jishnu Jaykumar P, Kamalesh Palanisamy, Yu-Wei Chao, Xinya Du, and Yu Xiang. Proto-clip: Vision-language prototypical network for few-shot learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2594–2601, 2024. doi: 10.1109/IROS58592.2024.10801660.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. A dataset of 101 human action classes from videos in the wild. *Center for Research in Computer Vision*, 2012.

Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *ICML*, 2020.

Hsi-Ai Tsao, Lei Hsiung, Pin-Yu Chen, Sijia Liu, and Tsung-Yi Ho. Autovp: An automated visual prompting framework and benchmark. In *ICLR*, 2024.

Ria Vinod, Pin-Yu Chen, and Payel Das. Reprogramming language models for molecular representation learning. In *NeurIPS*, 2020.

Wenhao Wang, Yifan Sun, Wei Li, and Yi Yang. Transhp: Image classification with hierarchical prompting. In *NeurIPS*, 2023.

Jiayang Wu, Xinyang Chen, Ke Lv, and Weili Guan. Boosting visual reprogramming for clip with dual granularity alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29347–29356, June 2026.

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

Chao-Han Huck Yang, Yun-Yun Tsai, and Pin-Yu Chen. Voice2series: Reprogramming acoustic models for time series classification. In *ICML*, 2021.

Chao-Han Huck Yang, Bo Li, Yu Zhang, Nanxin Chen, Rohit Prabhavalkar, Tara N Sainath, and Trevor Strohman. From english to more languages: Parameter-efficient model reprogramming for cross-lingual speech recognition. In *ICASSP*, 2023.

Hao Yen, Pin-Jui Ku, Chao-Han Huck Yang, Hu Hu, Sabato Marco Siniscalchi, Pin-Yu Chen, and Yu Tsao. Neural model reprogramming with similarity based mapping for low-resource spoken command classification. In *INTERSPEECH*, 2023.

Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10899–10909, 2023.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pp. 493–510, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-19832-8. doi: 10.1007/978-3-031-19833-5_29. URL https://doi.org/10.1007/978-3-031-19833-5_29.

Yichi Zhang, Yinpeng Dong, Siyuan Zhang, Tianzan Min, Hang Su, and Jun Zhu. Exploring the transferability of visual prompting for multimodal large language models. In *CVPR*, 2024.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022b.

Shengjie Zhou, Xin Cheng, Haiyang Xu, Ming Yan, Tao Xiang, Feng Liu, and Lei Feng. Endowing visual reprogramming with adversarial robustness. In *ICLR*, 2025.

A Dataset Information

Table 5: Summary of the 11 downstream benchmark datasets used in our experiments, including task type, number of classes, and training batch size.

	Aircraft	Caltech	Cars	DTD	ESAT	Flowers	Food	Pets	SUN	UCF	Resisc
Task Info.	aircraft model	object	fine-grained automobile	texture	remote sensing land cover	flower	food	pet	scene	action	remote sensing scene
Class Number	100	100	196	47	10	102	101	37	397	101	45
Batch Size	64	64	64	64	64	64	64	64	64	64	64

Following prior works (Cai et al., 2025a;b), we adopt the same 16-shot benchmark protocol. The benchmark covers 11 publicly available datasets spanning diverse recognition tasks, including fine-grained object recognition, generic object classification, texture recognition, scene understanding, action recognition, and remote sensing. Specifically, we use FGVC Aircraft (Aircraft) (Maji et al., 2013), Caltech101 (Caltech) (Fei-Fei et al., 2004), StanfordCars (Cars) (Krause et al., 2013), Describable Textures Dataset (DTD) (Cimpoi et al., 2014), EuroSAT (ESAT) (Helber et al., 2019), Flowers102 (Flowers) (Nilsback & Zisserman, 2008), Food101 (Food) (Bossard et al., 2014), OxfordPets (Pets) (Parkhi et al., 2012), SUN397 (SUN) (Xiao et al., 2010), UCF101 (UCF) (Soomro et al., 2012), and RESISC45 (Resisc) (Cheng et al., 2017). As summarized in Table 5, these datasets vary substantially in semantic granularity and class cardinality, ranging from 10 classes in EuroSAT to 397 classes in SUN397. Unless otherwise specified, we use a batch size of 64 for training visual reprogramming on all datasets.

B Additional Results

B.1 Additional Results on Different Backbones

Table 6: Accuracy comparison of different methods trained on 16-shot downstream classification tasks, using RN101-based CLIP as the pretrained model (Mean %, ours are highlighted and the highest is in **bold**).

Method	Aircraft	Caltech	Cars	DTD	ESAT	Flowers	Food	Pets	SUN	UCF	Resisc	Avg.
VP	19.3	83.0	53.7	43.4	62.8	57.2	71.2	80.2	53.5	54.2	54.0	57.5
AR	19.5	89.7	62.0	46.3	70.4	60.4	78.0	84.4	58.4	60.6	60.2	62.7
AttrVR	23.3	92.0	62.2	55.6	70.3	76.2	**79.5**	89.3	62.1	64.5	64.5	67.2
DVP	23.8	92.7	62.5	58.0	**70.7**	80.6	79.1	89.5	63.7	68.1	68.4	68.8
RVP	**30.4**	**93.9**	**76.0**	**60.1**	67.6	**90.5**	76.8	**91.1**	**67.5**	**75.8**	**73.4**	**73.0**

Table 7: Accuracy comparison of different methods trained on 16-shot downstream classification tasks, using ViT-B/32-based CLIP as the pretrained model (Mean %, ours are highlighted and the highest is in **bold**).

Method	Aircraft	Caltech	Cars	DTD	ESAT	Flowers	Food	Pets	SUN	UCF	Resisc	Avg.
VP	24.3	92.3	58.6	54.9	85.9	71.2	75.0	86.8	61.0	67.3	73.9	68.3
AR	21.8	92.7	56.9	49.9	85.6	66.7	75.7	84.7	59.9	63.5	71.6	66.3
AttrVR	24.5	92.0	56.6	56.8	**88.6**	77.8	**77.2**	89.8	62.8	67.9	73.9	69.8
DVP	26.1	92.9	56.5	57.2	88.5	82.5	77.0	89.2	64.2	70.5	76.0	71.0
RVP	**32.8**	**94.1**	**74.1**	**63.4**	86.4	**93.3**	74.8	**90.3**	**68.0**	**77.8**	**79.1**	**75.8**

We further evaluate RVP on RN101- and ViT-B/32-based CLIP backbones in Tables 6 and 7. The results remain consistent with the main experiments: RVP achieves the best average accuracy on both backbones and outperforms all previous visual reprogramming baselines by a clear margin.

With the RN101 backbone, RVP reaches an average accuracy of 73.0%, improving over DVP by 4.2 points and over AttrVR by 5.8 points. It achieves the best result on 9 out of 11 datasets, with especially large gains on Aircraft (+6.6 over DVP), Cars (+13.5), Flowers (+9.9), UCF (+7.7), and Resisc (+5.0). These results again show that the proposed structured mapping is particularly effective when the downstream task requires distinguishing semantically similar categories. The only

datasets where RVP does not achieve the best performance are ESAT and Food, where the advantage of explicit inter-class modeling appears less pronounced.

A similar pattern is observed for the ViT-B/32 backbone. RVP obtains the highest average accuracy of 75.8%, surpassing DVP by 4.8 points and AttrVR by 6.0 points. It performs best on 9 out of 11 datasets and shows especially large improvements on Aircraft ($+6.7$ over DVP), Cars ($+17.6$), Flowers ($+10.8$), UCF ($+7.3$), and DTD ($+6.2$). Notably, the gain on Cars remains very large even with the stronger ViT-based encoder, further supporting our claim that explicit modeling of inter-class relationships is particularly beneficial for fine-grained recognition.

Taken together, these additional results strengthen two observations from the main paper. First, the advantage of RVP is robust across both convolutional and transformer backbones. Second, the largest improvements consistently appear on fine-grained datasets such as Aircraft and Cars, where many categories share highly similar semantic attributes and cannot be reliably separated by intra-class prompt selection alone. This further supports the central motivation of RVP: when the pretrained feature space contains strong inter-class correlation, a structured mapping that explicitly models class relationships provides a more effective adaptation mechanism than independent prompt aggregation.

B.2 Broader Comparison

Although RVP follows a different adaptation paradigm from conventional CLIP adaptation methods, we further compare it with several representative approaches under the same 16-shot setting. These methods include prompt learning, feature adaptation, task residual learning, and linear probing, while RVP performs adaptation through visual reprogramming with structured label mapping.

As shown in Table 8, RVP achieves an average accuracy of 82.7%, showing competitive performance across the 11 datasets. It performs particularly well on Aircraft and Cars, reaching 46.1% and 84.8%, respectively, while also obtaining strong results on Caltech, EuroSAT, Pets, UCF, and RESISC. Although the compared methods use different adaptation strategies, this broader comparison shows that RVP remains effective when evaluated alongside general CLIP adaptation approaches.

Table 8: Accuracy comparison of different methods trained on 16-shot downstream classification tasks, using ViT-B/16-based CLIP as the pretrained model (Mean % $\pm$ Std %, ours are highlighted and the highest result is in **bold**).

Method	Aircraft	Caltech	Cars	DTD	EuroSAT	Flowers	Food	Pets	SUN	UCF	RESISC	Avg.
CoOp	43.2	95.8	82.9	69.7	85.0	96.8	84.2	92.0	74.9	83.1	84.7	81.1
CoCoOp	33.3	95.1	72.3	63.7	73.6	89.1	**87.4**	93.4	72.6	77.2	81.6	76.3
CLIP-Adapter	34.2	94.9	74.0	59.4	71.4	92.9	87.1	92.3	74.2	80.2	85.7	76.9
Tip-Adapter-F	44.6	95.7	82.3	70.8	85.9	96.2	86.8	92.6	76.0	83.9	81.2	81.5
TaskRes	44.9	95.8	83.5	71.5	82.7	**97.5**	86.9	92.4	**76.1**	84.0	83.3	81.7
LP++	42.1	95.8	80.8	**71.9**	85.5	96.3	87.2	92.6	76.0	83.9	80.9	81.2
RVP	**46.1**$\pm$0.2	**96.5**$\pm$0.2	**84.8**$\pm$0.3	68.7$\pm$0.1	**92.7**$\pm$0.2	96.7$\pm$0.2	85.5$\pm$0.1	**94.0**$\pm$0.1	73.8$\pm$0.1	**85.1**$\pm$0.7	**85.9**$\pm$0.5	**82.7**

As shown in Table 9, RVP achieves a favorable accuracy–parameter trade-off. In particular, it slightly improves over LDC on StanfordCars while using only about 1.5% of its trainable parameters. This efficiency follows from the structured design of RVP, whose learned mapping can be exactly reparameterized into a single linear head at inference. The results therefore show that RVP can retain strong recognition performance without relying on a large adaptation module.

B.3 Computation Cost

The VP method (Bahng et al., 2022) adopts a visual noise pattern with a frame width of 30 pixels. For an input image of size 224×224, this corresponds to $224 \times 224 \times 3 - (224 - 60) \times (224 - 60) \times 3 = 69840$ trainable prompt parameters. In contrast, both AR (Tsai et al., 2020; Chen et al., 2023) and AttrVR (Cai et al., 2025a) use a narrower frame width of 16 pixels, which results in $224 \times 224 \times 3 - (224 - 32) \times (224 - 32) \times 3 = 39936$ trainable parameters. DVP employs decoupled visual prompting, typically using three visual prompts. Its total number of prompt parameters is therefore $39936 \times 3 = 119808$. Just like AR (Tsai et al., 2020; Chen et al., 2023) and AttrVR (Cai et al., 2025a), RVP uses 39936 trainable prompt parameters.

Regarding logit aggregation, VP and AR do not introduce additional trainable parameters, as they do not rely on multiple textual descriptions per class. AttrVR adopts fixed aggregation functions

Table 9: **Accuracy and parameter efficiency on StanfordCars** under the 16-shot setting with ViT-B/16. Trainable parameters count only method-specific adaptation parameters.

Method	Accuracy (%)	Trainable Params.	Inference Path
CoOp	82.9	0.008M	Fixed classifier from learned text prompts
CoCoOp	72.3	0.042M	Image-conditioned text features
CLIP-Adapter	74.0	0.131M	Nonlinear feature adapter
Tip-Adapter-F	82.3	1.606M	Cache-based adapted logits
TaskRes	83.5	0.100M	Fixed linear head
LP++	80.8	0.101M	Fixed linear head
LDC	84.2	5.336M	Multi-level adapters and adaptive fusion
RVP	**84.8**	0.082M	Prompted CLIP with exactly folded linear head

Table 10: **Relationship between text-space concentration and the benefit of inter-class correction.** r_{90} denotes the minimum number of eigenvalues explaining 90% of the spectral mass of the class-prototype Gram matrix, and Δ_E measures the accuracy gain from enabling the inter-class residual matrix $\mathbf{E}$.

Dataset	C	r_{90}	r_{90}/C	RVP	w/o $\mathbf{E}$	Δ_E
Aircraft	100	7	0.070	46.1	35.6	10.5
Caltech101	100	28	0.280	96.5	96.1	0.4
Cars	196	24	0.122	84.8	68.0	16.8
DTD	47	3	0.064	68.7	63.3	5.4
EuroSAT	10	1	0.100	92.7	93.8	-1.1
Flowers102	102	27	0.265	96.7	91.7	5.0
Food101	101	29	0.287	85.5	85.6	-0.1
Oxford Pets	37	11	0.297	94.0	93.1	0.9
SUN397	397	31	0.078	73.8	67.4	6.4
UCF101	101	22	0.218	85.1	78.9	6.2
RESISC45	45	7	0.156	85.9	83.5	2.4

(e.g., mean, average, max, or kNN), which also do not introduce learnable parameters. In contrast, DVP employs a Probability Reweighting Matrix $\boldsymbol{\omega}_{\mathrm{PRM}} \in \mathbb{R}^{CM\times C}$ for aggregating description-level logits. Although the matrix is defined over $CM \times C$ entries, only $C \times M$ mapping parameters are effectively learnable under its structured parameterization.

For RVP, the intra-class aggregation matrix $P \in \mathbb{R}^{C\times M}$ introduces $C \times M$ parameters, while the inter-class matrix $E \in \mathbb{R}^{C\times C}$ contributes an additional $C \times C$ parameters.

B.4 Additional Analysis

Inter-class structure and the benefit of E. To examine when inter-class correction is most beneficial, we characterize the class structure induced by the text embeddings. For each dataset, we uniformly aggregate the 20 normalized attribute embeddings of each class, compute the eigenspectrum of the resulting class-prototype Gram matrix, and define r_{90} as the minimum number of eigenvalues required to explain 90% of the spectral mass. Since the number of classes C varies substantially across datasets, from 10 to 397, we use the normalized quantity r_{90}/C as the primary statistic. We measure the benefit of inter-class modeling as $\Delta_E = \mathrm{Acc}(\mathrm{RVP}) - \mathrm{Acc}(\mathrm{w/o}\ E)$.

Across the 11 datasets, r_{90}/C is negatively associated with Δ_E (Spearman $\rho = -0.527$). Since r_{90}/C can also depend on the number of classes, we additionally compute a partial Spearman correlation while controlling for C, which yields $\rho = -0.625$ with $p = 0.040$. The relationship is also stable under leave-one-dataset-out analysis: all partial correlations remain negative, ranging from -0.772 to -0.510. These observations are consistent with the hypothesis that when class prototypes occupy a more concentrated shared text subspace, there is more room for inter-class correction to improve class discrimination.

Large gains and the learned structure of E. The large improvement on StanfordCars is not explained by $\mathbf{E}$ acting as a negligible residual. On this dataset, $\mathbf{E}$ contains 38,416 parameters, while the 16-shot training set contains 3,136 images. On held-out data, the mean relative correction

induced by $\mathbf{E}$ is 0.456, and enabling $\mathbf{E}$ changes 36.2% of predictions. At the same time, the learned matrix exhibits clear structure: its stable rank is only 15.6, compared with 50.1 ± 1.0 under an entry-permutation null. Moreover, $\|\operatorname{diag}(\mathbf{E})\|_F / \|\mathbf{E}\|_F = 0.075$, close to the null value of 0.072, indicating that the learned correction is predominantly off-diagonal and therefore genuinely inter-class rather than a simple per-class rescaling.

Table 11: Number of trainable parameters for different methods on the Aircraft dataset ($C = 100$, $M = 20$).

Method	**Prompt Parameters**	**Total**	Accuracy
VP	69840	69840	32.1
AR	39936	39936	31.7
AttrVR	39936	39936	36.6
DVP	119808	121808	38.7
RVP	39936	51936	46.1

As shown in Table 11, RVP achieves the best accuracy on Aircraft while remaining parameter-efficient. Although it uses the same number of prompt parameters as AR and AttrVR, its additional structured aggregation introduces only a modest overhead, resulting in a total of 51936 parameters. In contrast, DVP uses substantially more parameters due to multiple visual prompts, yet still underperforms RVP. This shows that the gain of RVP comes from a more effective parameterization of class relationships rather than from simply increasing model size.

Table 12: Accuracy comparison of RVP, AttrVR, and DVP under the same text prompt setting, where DVP is restricted to a single group of trainable visual prompts, using ViT-B/16 CLIP as the pretrained model (mean %; ours are highlighted and the best results are shown in **bold**).

	AIRCRAFT	CALTECH	CARS	DTD	ESAT	FLOWERS	FOOD	PETS	SUN	UCF	RESISC	AVG.
ATTRVR (DESATTR)	35.9	95.6	68.2	64.4	93.8	92.4	**85.7**	93.0	67.7	78.6	81.8	77.9
DVP (NUM=1)	36.4	95.8	69.1	65.3	**94.1**	93.6	**85.7**	93.1	70.0	80.2	82.8	78.7
RVP	**46.1**	**96.5**	**84.8**	**68.7**	92.7	**96.7**	85.5	**94.0**	**73.8**	**85.1**	**85.9**	**82.7**

Table 12 compares RVP, AttrVR, and DVP under a controlled setting where all methods use the same text prompts and DVP is restricted to a single group of trainable visual prompts. Specifically, since AttrVR originally uses two groups of text prompts, namely *Descriptive Attributes* and *Distinctive Attributes*, we retain only *Descriptive Attributes* here to ensure a fair comparison across methods. Under this setting, RVP still achieves clear and consistent improvements over both AttrVR and DVP. In particular, RVP attains the best average accuracy of 82.7%, outperforming DVP by 4.0 points and AttrVR by 4.8 points. The improvement is especially pronounced on fine-grained datasets such as Aircraft and Cars, where RVP surpasses DVP by 9.7 and 15.7 points, respectively. These results indicate that the advantage of RVP does not rely on using more diverse text prompts or multiple prompt groups. Instead, the gain comes from its more effective modeling of intra-class aggregation and inter-class relationships, which allows it to better suppress shared semantics and enhance subtle class-specific differences.

Table 13: Accuracy comparison of our RVP and DVPlite trained on 16-shot downstream classification task, using ViT-B/16-based CLIP as the pretrained model (Mean %, ours is highlighted and the highest is in **bold**).

	AIRCRAFT	CALTECH	CARS	DTD	ESAT	FLOWERS	FOOD	PETS	SUN	UCF	RESISC	AVG.
ATTRVR	36.6	95.7	68.3	65.6	**93.8**	92.9	**85.9**	93.3	69.6	79.0	82.6	78.5
DVPLITE	39.3	95.9	71.4	66.5	**93.8**	95.2	85.8	93.4	71.6	81.0	83.6	79.8
RVP	**46.1**	**96.5**	**84.8**	**68.7**	92.7	**96.7**	85.5	**94.0**	**73.8**	**85.1**	**85.9**	**82.7**

Table 13 compares our method with DVPlite, an efficient variant of DVP proposed by Cai et al. (Cai et al., 2025b). DVPlite decomposes the visual prompt into four directional components, namely up, down, left, and right, and assigns them to different cause groups generated by an LLM. Unlike standard DVP, this design avoids multiple forward passes through the image encoder. However, it still relies on substantially more text prompts than our method, since each direction is associated

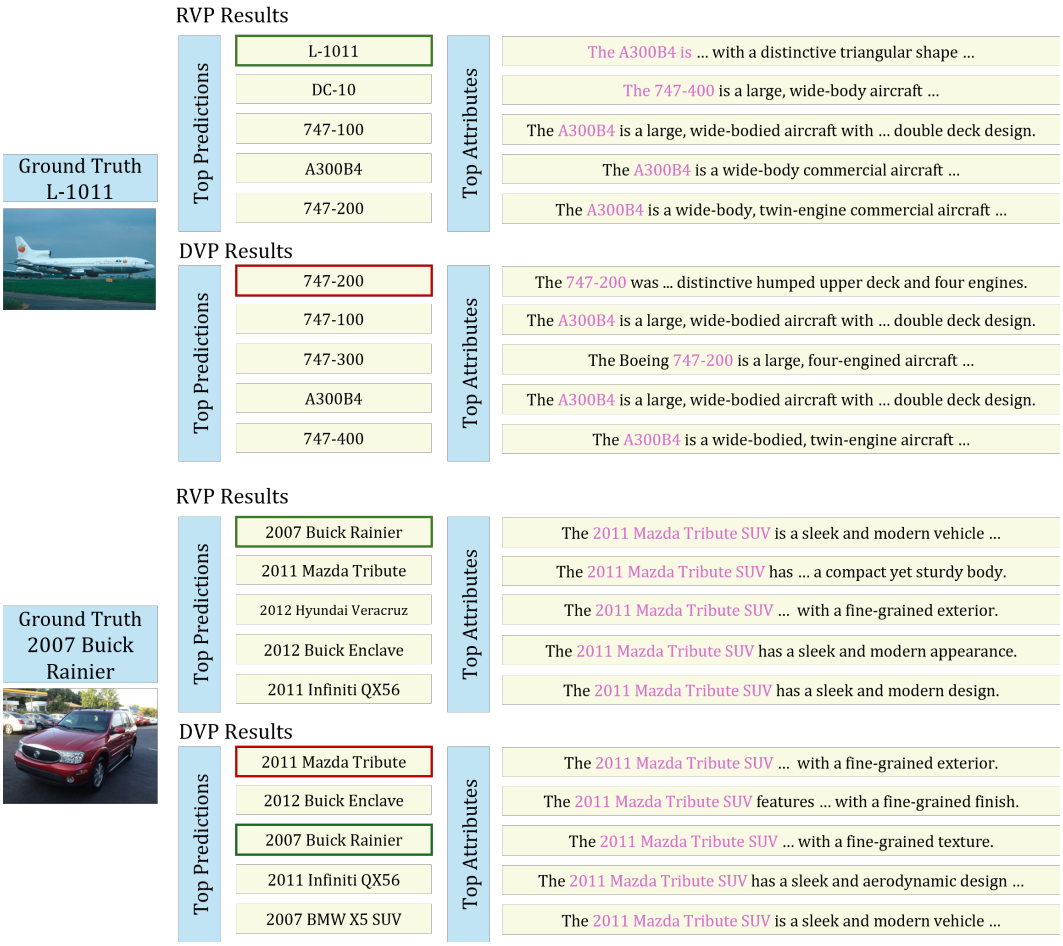


Figure 6: Additional visualization of the top predicted classes and highest-matching attributes for a test image. For both RVP and DVP, the most similar attributes include prompts from other classes, reflecting strong semantic overlap in fine-grained recognition. However, RVP explicitly models inter-class relationships, allowing it to better resolve these cross-class ambiguities and produce the correct prediction.

with its own set of $C \times M$ prompts. Despite this more complex prompt design, our method achieves the best overall performance. As shown in Table 13, RVP attains an average accuracy of 82.7%, outperforming DVPlite by 2.9 points. These results show that RVP is not only more accurate, but also simpler to use, as it avoids directional prompt decomposition and additional LLM-based cause grouping while still delivering stronger performance.

B.5 Additional Visualization

Fig. 6 further shows that attribute matching alone is not sufficient for fine-grained recognition. In both the aircraft and car examples, the top-matched attributes are dominated by semantically similar but incorrect classes, indicating that these attributes largely overlap and cannot reliably determine the final label on their own. Despite receiving similarly misleading attribute evidence, RVP still predicts the correct class, whereas DVP fails. This is because RVP does not rely only on prompt-level similarity; instead, it explicitly captures inter-class relationships, allowing it to suppress confusing evidence from correlated classes and produce better predictions.



Figure 7: Qualitative examples on Food101. Food categories often exhibit large intra-class variation and strong cross-class visual overlap due to differences in plating, viewpoint, garnish, and accompanying side dishes. As shown here, classes such as Apple Pie and Waffles can be confused when the main dish is partially visible or co-occurs with similar desserts, while Beet Salad and Tuna Tartare may share similar fine-grained presentation and ingredients. Correct predictions are shown in green and incorrect predictions in red.

B.6 Additional Error Analysis

Food101 is particularly challenging because its class semantics are often compositional rather than visually stable. Unlike fine-grained object categories, a food image may contain the main dish together with side dishes, garnish, sauces, or additional ingredients, so the same class can vary substantially across samples. As illustrated in Fig. 7, Apple Pie can co-occur with cream or be presented in ways that resemble other desserts, while Beet Salad and Tuna Tartare may share similar plating style, color, and ingredient structure. In such cases, the ambiguity is driven not only by inter-class similarity, but also by high intra-class variation and unstable visual cues, which reduces the benefit of structured inter-class modeling.



Figure 8: Qualitative examples on EuroSAT. Several classes, especially Sea_or_lake, River, and Highway/Road, exhibit strong visual similarity in satellite crops due to elongated shapes, curved boundaries, and limited scene context. As a result, some samples remain ambiguous even under the proposed method, suggesting that EuroSAT is less dominated by inter-class semantic ambiguity than fine-grained recognition benchmarks. Correct predictions are shown in green and incorrect predictions in red.

A possible reason why RVP is less advantageous on EuroSAT is that this benchmark does not primarily require the kind of inter-class semantic disambiguation that RVP is designed to address. Unlike fine-grained tasks such as Aircraft and Cars, where many classes share highly similar semantic attributes, EuroSAT contains only 10 classes, and many errors arise from coarse visual ambiguity in satellite crops rather than from strong overlap in text semantics. As illustrated in Fig. 8, categories such as Sea_or_lake, River, and Highway/Road can appear visually similar due to limited resolution (64×64 pixels), elongated structures, and missing global scene context. In such cases, the main difficulty lies in ambiguous visual evidence and spatial layout, rather than in class relationships within the text embedding space. As a result, explicit inter-class modeling provides less benefit on EuroSAT than on more fine-grained benchmarks.

C PROPOSITIONS AND PROOF

C.1 THEORETICAL JUSTIFICATION OF RVP

In this section, we provide a formal justification for why RVP is well-suited to few-shot visual reprogramming, especially for fine-grained recognition. We do not claim that RVP is universally optimal for all data distributions. Rather, the results below show that under a natural low-rank shared-semantic assumption, RVP has three desirable properties: (i) it preserves the pretrained CLIP semantic subspace, (ii) its inter-class residual can explicitly suppress shared semantic components, and (iii) it refines decision margins in a stable manner.

Setup. Recall that the inference rule of RVP is

$$\mathbf{z} = \hat{\mathbf{v}}^\top \hat{W}, \qquad \hat{W} = \frac{1}{\tau} T^\top W_1 (I + E), \tag{9}$$

where $T \in \mathbb{R}^{CM \times D}$ is the stacked text embedding matrix, $W_1 \in \mathbb{R}^{CM \times C}$ is the block-diagonal intra-class aggregation matrix, and $E \in \mathbb{R}^{C \times C}$ is the inter-class residual matrix.

Proposition C.1 (Exact linear reparameterization)**.** *For any learned intra-class weights W_1 and inter-class matrix E, there exists a single matrix $\hat{W} \in \mathbb{R}^{D \times C}$ such that the training-time classifier and the inference-time classifier are identical:*

$$\mathbf{z} = \frac{1}{\tau} \hat{\mathbf{v}}^\top T^\top W_1 (I + E) = \hat{\mathbf{v}}^\top \hat{W}. \tag{10}$$

Proof. This follows directly from the associativity of matrix multiplication by defining

$$\hat{W} = \frac{1}{\tau} T^\top W_1 (I + E).$$

Substituting this definition into the classifier gives the desired result. □

Proposition C.2 (Text-span preservation)**.** *Every column of the reparameterized classifier $\hat{W}$ lies in the column space of $T^\top$. Equivalently,*

$$\mathrm{col}(\hat{W}) \subseteq \mathrm{col}(T^\top). \tag{11}$$

Proof. By Proposition C.1,

$$\hat{W} = \frac{1}{\tau} T^\top W_1 (I + E).$$

Hence, each column of $\hat{W}$ is a linear combination of the columns of $T^\top$. Therefore,

$$\mathrm{col}(\hat{W}) \subseteq \mathrm{col}(T^\top).$$

□

Proposition C.2 shows that RVP constructs its classifier entirely from the span of the pretrained CLIP text embeddings, rather than introducing classifier directions outside this text-induced space. Thus, the inter-class correction recombines existing text-derived directions instead of learning an unconstrained classifier directly in $\mathbb{R}^D$. When the text embeddings occupy a lower-dimensional subspace, this additionally restricts the effective classifier space and provides a structured inductive bias for few-shot adaptation.

Assumption C.3 (Shared-semantic decomposition)**.** Let the base class-logit vector before inter-class correction be $\mathbf{f}(x) \in \mathbb{R}^C$. Assume that there exists an r-dimensional subspace $\mathcal{S} \subset \mathbb{R}^C$, with orthonormal basis $U \in \mathbb{R}^{C \times r}$, such that

$$\mathbf{f}(x) = \mathbf{s}(x) + \mathbf{d}(x), \tag{12}$$

where $\mathbf{s}(x) \in \mathcal{S}$ is a shared semantic component and $\mathbf{d}(x) \in \mathcal{S}^\perp$ is a class-discriminative component.

Assumption C.3 formalizes the empirical observation that fine-grained classes often share dominant semantic directions, while useful class-specific information resides in weaker contrastive components.

Theorem C.4 (Suppression of shared semantic components)**.** *Under Assumption C.3, there exists a residual matrix E such that the inter-class correction removes the shared semantic component exactly. In particular, if we choose*

$$E = -UU^\top, \tag{13}$$

then

$$\mathbf{z} = \mathbf{f}(x)(I + E) = \mathbf{d}(x). \tag{14}$$

Proof. Substituting $E = -UU^\top$ yields

$$I + E = I - UU^\top,$$

which is the orthogonal projector onto $\mathcal{S}^\perp$. Since $\mathbf{s}(x) \in \mathcal{S}$, we have

$$\mathbf{s}(x)(I - UU^\top) = \mathbf{0}.$$

Since $\mathbf{d}(x) \in \mathcal{S}^\perp$, we have

$$\mathbf{d}(x)(I - UU^\top) = \mathbf{d}(x).$$

Therefore,

$$\mathbf{z} = \big(\mathbf{s}(x) + \mathbf{d}(x)\big)(I - UU^\top) = \mathbf{d}(x).$$

□

Theorem C.4 provides a formal explanation for why inter-class modeling is useful. If different classes share a low-rank semantic component, then a suitable residual class-relation matrix can cancel that shared component and retain only the discriminative part.

Corollary C.5 (Margin recovery under the projected classifier)**.** *Under the conditions of Theorem C.4, suppose the true label is y and the discriminative component satisfies*

$$d_y(x) - \max_{j \neq y} d_j(x) > 0. \tag{15}$$

Then the RVP classifier with $E = -UU^\top$ predicts the correct class:

$$y = \arg\max_c z_c. \tag{16}$$

Proof. By Theorem C.4, $\mathbf{z} = \mathbf{d}(x)$. Hence

$$z_y - \max_{j \neq y} z_j = d_y(x) - \max_{j \neq y} d_j(x) > 0,$$

which implies $y = \arg\max_c z_c$. □

Proposition C.6 (Bounded margin degradation under residual correction)**.** *Let*

$$\mathbf{z} = \mathbf{f} + \Delta, \qquad \Delta = \mathbf{f}E. \tag{17}$$

For any class y, define the multiclass margin

$$m_y(\mathbf{a}) = a_y - \max_{j \neq y} a_j. \tag{18}$$

Then

$$m_y(\mathbf{z}) \geq m_y(\mathbf{f}) - 2\|\Delta\|_\infty. \tag{19}$$

Proof. Since $\mathbf{z} = \mathbf{f} + \Delta$,

$$z_y \geq f_y - \|\Delta\|_\infty,$$

and

$$\max_{j \neq y} z_j \leq \max_{j \neq y} f_j + \|\Delta\|_\infty.$$

Subtracting the second inequality from the first gives

$$m_y(\mathbf{z}) = z_y - \max_{j \neq y} z_j \geq \big(f_y - \|\Delta\|_\infty\big) - \big(\max_{j \neq y} f_j + \|\Delta\|_\infty\big),$$

which simplifies to

$$m_y(\mathbf{z}) \geq m_y(\mathbf{f}) - 2\|\Delta\|_\infty.$$

□

Proposition C.6 bounds the possible degradation of the classification margin under residual correction: the margin can decrease from the base margin by at most $2\|\mathbf{f}E\|_\infty$. In particular, if

$$m_y(\mathbf{f}) > 2\|\mathbf{f}E\|_\infty,$$

then $m_y(\mathbf{z}) > 0$, and the original prediction for class y is preserved.

Proposition C.7 (Structured restriction of the hypothesis class)**.** *Let*

$$\mathcal{H}_{\text{dense}} = \left\{ \hat{\mathbf{v}} \mapsto \frac{1}{\tau}\hat{\mathbf{v}}^\top T^\top \Omega \ : \ \Omega \in \mathbb{R}^{CM\times C} \right\} \tag{20}$$

be the class of dense description-to-class mappings, and let

$$\mathcal{H}_{\text{RVP}} = \left\{ \hat{\mathbf{v}} \mapsto \frac{1}{\tau}\hat{\mathbf{v}}^\top T^\top W_1(I+E) \ : \ W_1 \textit{ is block diagonal}, \ E \in \mathbb{R}^{C\times C} \right\}. \tag{21}$$

Then

$$\mathcal{H}_{\text{RVP}} \subseteq \mathcal{H}_{\text{dense}}. \tag{22}$$

Moreover, RVP contains $CM + C^2$ trainable parameters in its structured output mapping, whereas the dense mapping contains C^2M trainable parameters.

Proof. For any W_1 and E, define

$$\Omega = W_1(I+E).$$

Since

$$\Omega \in \mathbb{R}^{CM\times C},$$

every function in $\mathcal{H}_{\text{RVP}}$ is also an element of $\mathcal{H}_{\text{dense}}$. Hence,

$$\mathcal{H}_{\text{RVP}} \subseteq \mathcal{H}_{\text{dense}}.$$

For the parameter count, W_1 is determined by C groups of M intra-class weights, corresponding to CM trainable parameters, while E contributes C^2 trainable parameters. Therefore, the structured RVP output mapping contains $CM + C^2$ trainable parameters. In contrast, the dense matrix $\Omega \in \mathbb{R}^{CM\times C}$ contains

$$CM \times C = C^2M$$

trainable parameters. □

Proposition C.7 formalizes the regularization effect of RVP. It does not enlarge the dense hypothesis class. Instead, it restricts it to a structured subset that first aggregates prompts within each class and then applies a residual inter-class correction. This is particularly desirable in the few-shot regime, where unrestricted dense mappings are more likely to overfit.

Discussion. Taken together, the results above explain why RVP is effective. Proposition C.2 shows that RVP preserves the pretrained CLIP semantic subspace. Theorem C.4 and Corollary C.5 show that its inter-class residual can explicitly remove low-rank shared semantic components that obscure fine-grained discrimination. Proposition C.6 shows that this correction is stable because it acts in residual form. Finally, Proposition C.7 shows that RVP achieves these benefits while restricting the classifier to a structured low-complexity family. Together, these properties provide a principled explanation for why RVP works well in few-shot fine-grained visual reprogramming.

C.2 Proof of Removing Visual Embedding Normalization at Inference

In this section, we show that the ℓ_2-normalization of the visual embedding can be omitted at inference without changing the final predicted class, provided that only classification decisions are of interest.

Recall that the inference-time logits of RVP are given by

$$\mathbf{z} = \hat{\mathbf{v}}^\top \hat{W}, \tag{23}$$

where $\hat{\mathbf{v}} \in \mathbb{R}^D$ is the normalized visual embedding and $\hat{W} \in \mathbb{R}^{D\times C}$ is the reparameterized classifier matrix. Let $\mathbf{v} \in \mathbb{R}^D$ denote the corresponding unnormalized visual embedding produced by the frozen image encoder. By definition,

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}. \tag{24}$$

Substituting this into the inference equation gives

$$\mathbf{z} = \left(\frac{\mathbf{v}}{\|\mathbf{v}\|_2}\right)^\top \hat{W} = \frac{1}{\|\mathbf{v}\|_2}\mathbf{v}^\top \hat{W}. \tag{25}$$

Now define the logits computed without visual normalization as

$$\tilde{\mathbf{z}} = \mathbf{v}^\top \hat{W}. \tag{26}$$

Then we have

$$\mathbf{z} = \frac{1}{\|\mathbf{v}\|_2}\tilde{\mathbf{z}}. \tag{27}$$

That is, the normalized and unnormalized logits differ only by the multiplicative factor $1/\|\mathbf{v}\|_2$, which is a positive scalar shared by all classes for the same sample.

Let z_c and $\tilde{z}_c$ denote the c-th entries of $\mathbf{z}$ and $\tilde{\mathbf{z}}$, respectively. Then for any two classes i and j,

$$z_i - z_j = \frac{1}{\|\mathbf{v}\|_2}\left(\tilde{z}_i - \tilde{z}_j\right). \tag{28}$$

Since $\|\mathbf{v}\|_2 > 0$, multiplication by $1/\|\mathbf{v}\|_2$ preserves the sign of every pairwise logit difference. Therefore,

$$z_i > z_j \quad \Longleftrightarrow \quad \tilde{z}_i > \tilde{z}_j. \tag{29}$$

This implies that the ordering of class logits is unchanged, and hence

$$\arg\max_{c\in\{1,\dots,C\}} z_c = \arg\max_{c\in\{1,\dots,C\}} \tilde{z}_c. \tag{30}$$

Therefore, omitting the ℓ_2-normalization of the visual embedding does not affect the final predicted class. In other words, if only top-1 classification is required, the inference rule

$$\mathbf{z} = \hat{\mathbf{v}}^\top \hat{W} \tag{31}$$

is equivalent to

$$\tilde{\mathbf{z}} = \mathbf{v}^\top \hat{W}. \tag{32}$$

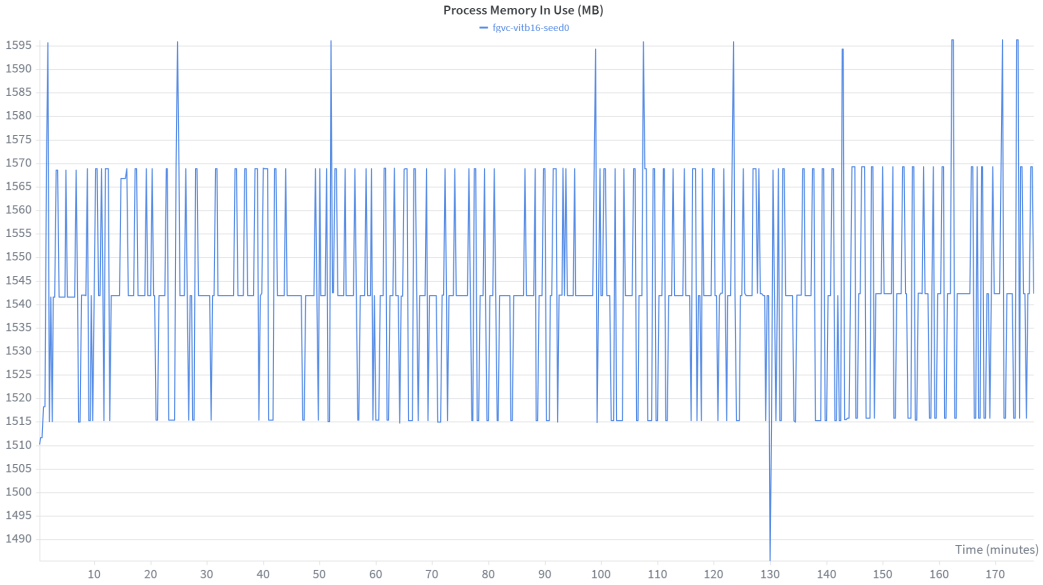
The latter has exactly the same form as a standard linear classifier applied to backbone features.

We emphasize that this equivalence holds for classification decisions, but not for the absolute scale of the logits or calibrated confidence scores. Indeed, removing the normalization changes the magnitude of the logits by a sample-dependent factor $\|\mathbf{v}\|_2^{-1}$, which may affect softmax probabilities, confidence calibration, or any downstream procedure that depends on logit scale.

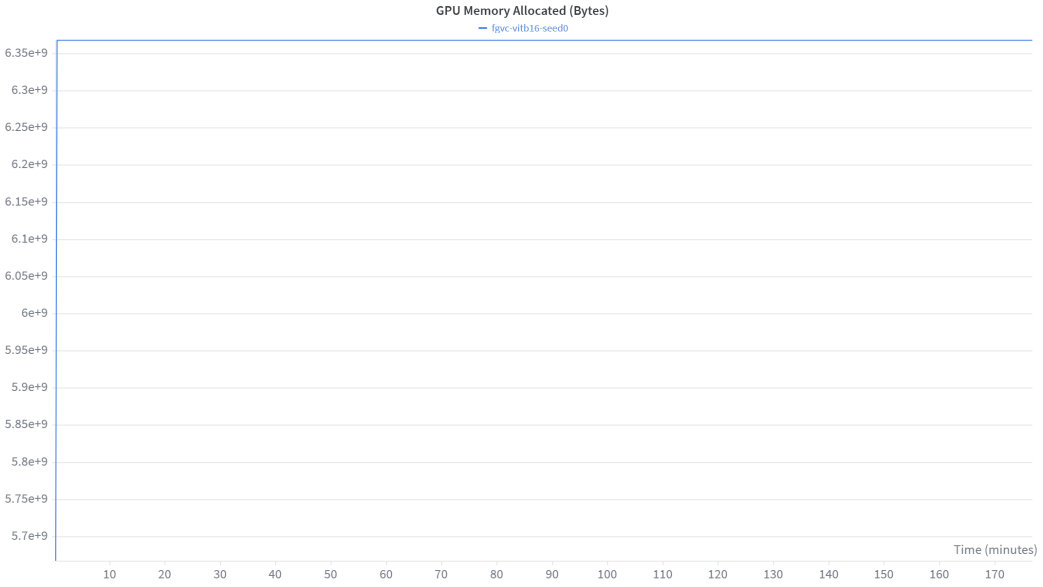
D Implementation Details.

Following prior work (Cai et al., 2025a;b), we train the visual prompt with a learning rate of 40, momentum 0.9, using stochastic gradient descent (SGD), and a cosine annealing scheduler (Loshchilov & Hutter, 2017) for 200 epochs. For all datasets, we use a batch size of 64. For the intra-class matrix P and inter-class matrix E, we use a learning rate of 10^{-3}. To ensure a fair comparison, we adopt the same text descriptions as in (Cai et al., 2025b), using $M = 20$ prompts per class. Our method does not introduce additional hyperparameters. The temperature τ is inherited from the pretrained CLIP model and kept fixed during training.

All experiments are conducted on a single NVIDIA L40S GPU with 48 GB of memory. The full set of experiments across 11 datasets requires approximately 47.5 hours. As shown in Fig. 9, the training process uses about 6.37 GB of GPU memory.



(a) Host (CPU) memory usage during training



(b) GPU memory usage during training

Figure 9: Memory consumption when training RVP with a ViT-B/16-based CLIP backbone on the FGVC Aircraft dataset. Left: host memory. Right: GPU memory.

E NOTATIONS

In this section, we summarize the abbreviations and key mathematical notations used in this paper to improve clarity.

E.1 ABBREVIATIONS

Table 14: Abbreviations used in the paper

Abbreviation	Description
RVP	Reparameterized Inter-Class Visual Reprogramming.
VR	Visual Reprogramming.
VLM	Vision-Language Model.
CLIP	Contrastive Language-Image Pre-training.
VP	Visual Prompting / standard visual reprogramming baseline.
AR	Adversarial Reprogramming baseline.
AttrVR	Attribute-based Visual Reprogramming.
DVP	Decoupled Visual Prompting.
DVP-cls	DVP with partitions formed by unsupervised clustering of description embeddings.
LLM	Large Language Model.
PRM	Probability Reweighting Matrix used in DVP.
CE Loss	Cross-Entropy Loss.
SGD	Stochastic Gradient Descent.

E.2 NOTATION

Table 15: Generic notation in visual reprogramming

Symbol	Description
f_{img}	CLIP image encoder.
f_{txt}	CLIP text encoder.
$\mathcal{X}^{\text{S}}$	Source image space of the pretrained CLIP model, with $\mathcal{X}^{\text{S}} \subseteq \mathbb{R}^{d_{\text{S}}}$.
$\mathcal{X}^{\text{T}}$	Target image space for the downstream task, with $\mathcal{X}^{\text{T}} \subseteq \mathbb{R}^{d_{\text{T}}}$.
$\mathcal{Y}^{\text{T}}$	Label space of the downstream task, with $\mathcal{Y}^{\text{T}} = \{1, \dots, C\}$.
$\mathcal{V}$	Text space containing textual descriptions.
$\mathcal{Z}$	Shared embedding space for image and text features, with $\mathcal{Z} \subseteq \mathbb{R}^{D}$.
x^{S}	Source-domain image.
x^{T}	Target-domain image.
V	A text description in $\mathcal{V}$.
y^{T}	A downstream class label.
C	Number of downstream classes.
D	Embedding dimension of CLIP.
d_{S}	Input dimensionality of source-domain images.
d_{T}	Input dimensionality of target-domain images.
$\hat{v}$	ℓ_2-normalized visual embedding produced by the CLIP image encoder.
$\hat{t}$	ℓ_2-normalized text embedding produced by the CLIP text encoder.
$f_{\text{clip}}(x^{\text{S}}, V)$	CLIP similarity score between image x^{S} and text V.
τ	Temperature parameter in CLIP similarity computation.
$\mathcal{A}$	Full set of textual descriptions used for the downstream task.
$\mathcal{A}(y^{\text{T}})$	Set of textual descriptions associated with class y^{T}.
M	Number of textual descriptions per class.
a	A description element from $\mathcal{A}$.
$\text{agg}(\cdot)$	Aggregation operator over description-level similarities.
$[f_{\text{logits}}(x^{\text{T}}; \mathcal{A})]_{y^{\text{T}}}$	Logit of class y^{T} computed from aggregated image-text similarities.
$\lvert\mathcal{A}\rvert$	Number of textual descriptions in $\mathcal{A}$.
$f_{\text{in}}(x^{\text{T}} \mid \delta)$	Input transformation to map a target-domain image into CLIP input space.
δ	Trainable visual prompt.
$\mathcal{D}$	Downstream training set.
ω	Reweighting matrix that maps description-level similarities to class logits.
ϕ	Overall linear mapping from the normalized image embedding to downstream class logits.
N	Number of training samples in $\mathcal{D}$.

Table 16: Notation for visual reprogramming and the linear mapping view

Symbol	Description
T	Stacked matrix of normalized text embeddings. In RVP with C classes and M descriptions per class, $T \in \mathbb{R}^{CM \times D}$.
M_a	Vector of similarity scores over all textual descriptions.
M_y	Vector of downstream class logits.
$P \in \mathbb{R}^{C \times M}$	Learnable intra-class weighting matrix for aggregating attribute descriptions within each class.
$\tilde{P}_c$	Softmax-normalized weight vector for the c-th class, obtained from the c-th row of P.
$\tilde{P}_{c,m}$	Normalized weight assigned to the m-th textual description of class c.
$\hat{t}_{c,m}$	Normalized embedding of the m-th textual description for class c.
$f_c(x^{\mathrm{T}})$	Aggregated base logit for class c before inter-class refinement.
$f(x^{\mathrm{T}})$	Row vector of base logits for all classes, $f(x^{\mathrm{T}}) = [f_1(x^{\mathrm{T}}), \ldots, f_C(x^{\mathrm{T}})] \in \mathbb{R}^C$.
$E \in \mathbb{R}^{C \times C}$	Learnable inter-class adjacency matrix for modeling class relationships.
I	Identity matrix used in residual message passing.
$z \in \mathbb{R}^C$	Final class logit vector after inter-class refinement.
$W_1 \in \mathbb{R}^{CM \times C}$	Sparse routing matrix constructed from the normalized intra-class weights.
$\hat{W} \in \mathbb{R}^{D \times C}$	Reparameterized linear classifier that absorbs text embeddings, intra-class aggregation, and inter-class refinement.
$\hat{v}^\top \hat{W}$	Final inference form of RVP as a single linear projection on the normalized visual embedding.
$v^\top \hat{W}$	Equivalent inference form for top-1 prediction when visual feature normalization is omitted.