CURE: Visual Reprogramming of Vision-Language Models under Limited Supervision

Lingzhi Wang[1], Wei Wang[2,3]✉, Xinyang Chen[1]✉
[1] School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
[2] The Qingdao Research Institute, Harbin Institute of Technology (Weihai)
[3] Shandong Key Laboratory of Industrial Network Security
wanglingzhiiiiii@gmail.com,wwhit@hit.edu.cn,chenxinyang@gmail.com

Abstract

Visual reprogramming (VR) efficiently repurposes pre-trained vision-language models (VLMs) for downstream classification tasks by adding trainable patterns to inputs without modifying the backbone. Nevertheless, existing VR methods heavily depend on labeled data and thus suffer substantial performance degradation under limited supervision (i.e., scarce labeled images). We introduce semi-supervised learning (SSL) into VR to exploit abundant unlabeled images and enhance data efficiency. However, directly applying generic SSL techniques often amplifies biases in unlabeled posteriors and destabilizes VR training. To address these challenges, we propose **CURE** (**C**onsistency and **U**nlabeled **R**ecalibration with Confidence-Margin **E**nhancement), a semi-supervised VR framework comprising two key components: *(i)* Recalibrated Gaussian Confidence-Margin Soft Weighting to dynamically adjust sample importance; *(ii)* Dual-Attribute Consistency Regularization to ensure consistency across attribute prompts. Extensive experiments show that CURE consistently outperforms both existing VR methods and direct SSL extensions, improving average classification accuracy by **3.45** and **3.77** percentage points across twelve widely used benchmarks for limited supervision tasks. Our code is available at `https://github.com/iLearn-Lab/NeurIPS26-CURE/`.

1 Introduction

Pre-trained visual models, such as ResNet [59] and ViT [18], which are trained on large-scale annotated data, have achieved remarkable success in several areas, such as classification [35], segmentation [66], and retrieval [20]. These models also perform well on downstream tasks when fine-tuned to task-specific data. Prior work has investigated negative transfer in fine-tuning [12] and the trade-off between transferability and discriminability in adversarial domain adaptation [13]. Recently, a promising approach called *visual reprogramming* (VR) [9, 11], also known as *adversarial reprogramming* [21], modifies the input by injecting trainable patterns rather than fine-tuning the model weights. This allows efficient task adaptation without altering the original model. Furthermore, VR has been extended to *vision-language models* (VLMs) [2], such as CLIP [45] to take advantage of multimodal information. The existing implementation of VR for CLIP [6, 5, 7, 8] introduces additional attributes for each new class to enhance CLIP's ability.

However, current VR methods are inherently limited by their dependency on labeled data. In limited supervision settings, where only a handful of labeled examples are available, these methods often overfit to the small training set or fail to capture the diversity of the input space, leading to suboptimal performance. This limitation is clearly illustrated in the example shown in Figure 1, where the performance of VR, based on attribute descriptions, degrades as the number of labeled samples decreases across different shot settings ($[16, 4, 1]$ shots). Specifically, it can be observed that the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

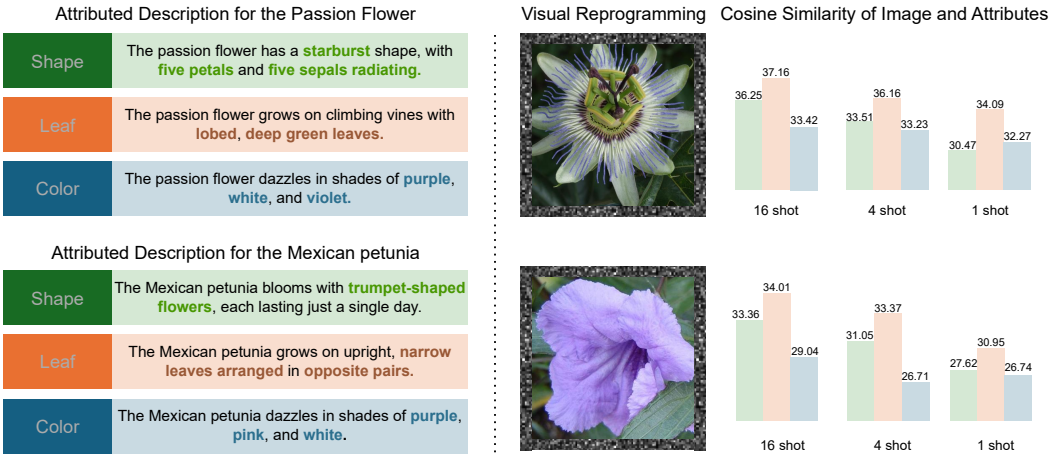


Figure 1: Attribute-level degradation of supervised VR under limited supervision. Sample descriptions of different aspects "Passion Flower" and "Mexican Petunia" are shown on the left, while recorded cosine similarities of images and descriptions are on the right under $[16, 4, 1]$ shots.

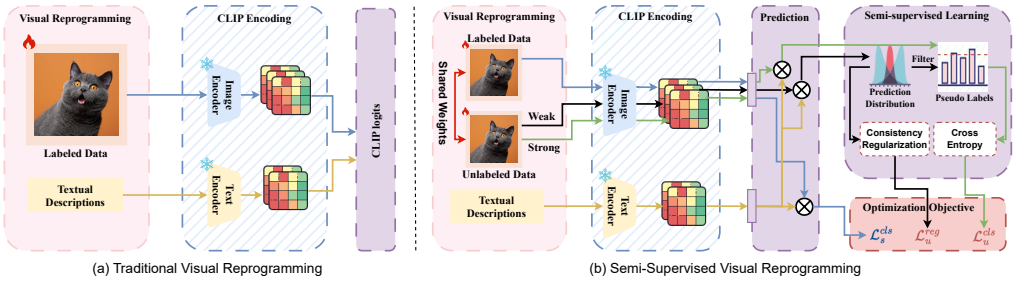


Figure 2: Semi-supervised learning for VR. On the left, conventional VR is shown, where only labeled data is used for training. On the right, SSL-VR utilizes the unlabeled data for training.

model's ability to capture the "shape" and "leaf" attributes significantly declines with fewer image samples, while the ability to capture the "color" attribute remains relatively stable. Beyond this qualitative observation, we further quantify this degradation by measuring the positive-negative attribute margin, i.e., the similarity gap between the correct class attribute and the competing incorrect class attributes. As shown in Appendix C.1, this margin consistently decreases when fewer labeled samples are available, indicating that limited supervision weakens the model's discriminative attribute alignment rather than merely reducing the absolute similarity to the correct class. These observations highlight the substantial impacts of limited supervision on different aspects of downstream tasks.

Despite the scarcity of labeled samples, unlabeled data is often readily accessible in many practical scenarios [39, 65]. This motivates us to extend VR to a semi-supervised setting, termed SSL-VR, where labeled examples provide supervised adaptation signals and unlabeled examples are leveraged through pseudo-label-based objectives, as illustrated in Figure 2. However, directly combining VR with off-the-shelf SSL algorithms [61] is suboptimal. Unlike standard semi-supervised classification, SSL-VR adapts a frozen VLM only through input-space reprogramming patterns, so biased pseudo-labels cannot be recalibrated by representation learning and may instead be amplified during pattern optimization. Meanwhile, fixed-threshold methods such as FixMatch [47] favor easy high-confidence samples and discard uncertain examples that may still contain informative signals. Soft-confidence weighting [10] mitigates this issue, but it still ignores semantic inconsistency across attribute-based prompts used for VLM classification: descriptive prompts capture common evidence, whereas distinctive prompts emphasize discriminative cues. As a result, unlabeled predictions may appear confident under one prompt but become unstable across another, suggesting that SSL-VR requires reliability estimation based on both prediction strength and cross-attribute semantic consistency.

To resolve these challenges, we propose a novel semi-supervised visual reprogramming framework, **CURE** (**C**onsistency and **U**nlabeled **R**ecalibration with Confidence-Margin **E**nhancement).

Following standard VR, CURE keeps the VLM backbone frozen and optimizes only trainable input-space reprogramming patterns, while exploiting unlabeled target-domain images through two VR-specific mechanisms: *(i) Recalibrated Gaussian Confidence–Margin Soft Weighting*, replacing fixed thresholds with variance-aware weights derived from maximum probability and top-2 margins, thus retaining informative low-confidence samples while down-weighting noisy ones; *(ii) Dual-Attribute Consistency Regularization*, enforcing agreement between descriptive and distinctive attribute views to validate pseudo-label semantics in the VLM space. Together, these components synergize to produce stable training signals from unlabeled data. Our contributions are summarized as:

1. Going beyond label-only training, we explore semi-supervised visual reprogramming for VLMs, leveraging readily available unlabeled data to alleviate the limitations of scarce labeled supervision.
2. We propose *CURE*, which enhances SSL-VR through two coordinated modules: Recalibrated Gaussian Confidence-Margin Soft Weighting to recalibrate and softly weight pseudo supervision; Dual-Attribute Consistency Regularization to encourage semantic agreement between attributes.
3. CURE consistently outperforms existing VR baselines and direct SSL extensions, with average gains of 3.45 and 3.77 percentage points across twelve widely used downstream visual datasets.

2 Related work

Prompting adaptation for classification. Prompt-based adaptation [30, 1, 43] enables large pre-trained models to specialize in downstream tasks without full fine-tuning. In vision backbones such as ViT [18], prompt learning injects trainable tokens or patterns to steer representations [30], with extensions that place prompts across layers (EEVPT [26]) or within self-attention embeddings (TransHP [54]). For vision–language models (VLMs) like CLIP [45], few-shot prompt strategies optimize the text side: CoOp learns class-conditional prompts [69], CoCoOp conditions text prompts on image features to generalize across classes [68], MaPLe introduces layer-wise mappings between modalities [32], and PromptKD distills improved prompts via knowledge transfer [36]. These approaches are parameter-efficient but remain predominantly supervised and typically require substantial architectural modifications, such as inserting learnable tokens or altering attention embeddings. In contrast, our work takes a different route by reprogramming the visual input itself and exploiting unlabeled data, avoiding heavy structural changes while improving data efficiency.

Input visual reprogramming. Model reprogramming adapts a frozen backbone by modifying the input or output spaces rather than inserting trainable modules inside the network [11]. In vision, input visual reprogramming (VR) realizes this idea by learning an input-side pattern that is optimized with the task loss while the backbone remains fixed, typically as a learnable overlay or patch [51, 9] or a padded frame around the image [6]. Beyond vision, related input-space reprogramming has also been explored in language [25, 53] and graphs [31]. Applied to vision–language models (VLMs) such as CLIP, input VR keeps the pre-trained image–text alignment intact while adapting the visual channel via a learned image pattern; the text channel supplies label prompts for supervision [9, 1, 43]. Recently, a complementary line augments VR with *visual attribute queries* to strengthen image–text alignment: large language models are queried for class-level *descriptive* and *distinctive* attributes [7, 8], often instantiated with GPT-3/3.5 [4]. Other efforts expand the candidate concept pool [62], curate concise attribute sets via sampling or ranking [50], and compute image–attribute similarities or weights to improve generalization [34, 60]. DGA [57] further enhances CLIP visual reprogramming by exploiting hierarchical label semantics and multi-scale visual representations. Despite these advances, most VR pipelines remain fully supervised and make limited use of unlabeled data. To address this limitation, we propose a semi-supervised VR framework that explicitly leverages unlabeled data.

Semi-supervised learning. Semi-supervised learning (SSL) leverages abundant unlabeled data to improve algorithm performance with scarce annotations [61]. Classic families include pseudo labeling and consistency regularization (enforcing prediction invariance). Modern SSL often centers on generating and filtering pseudo labels. FixMatch [47] combines weak and strong augmentation with hard confidence thresholds, but fixed thresholds induce a quantity-quality trade-off. Other methods address this challenge: FreeMatch [55] replaces static thresholds with self-adaptive, class-aware ones that evolve with the learning process and introduces fairness regularization to diversify predictions; SoftMatch [10] further replaces hard thresholding with truncated-Gaussian sample weighting and uniform distribution alignment to better exploit low-confidence yet informative samples. Difficulty-aware SSL methods have recently explored richer criteria beyond confidence. CGMatch [14] introduces

Count-Gap (CG), a new metric for assessing the model's learning state on unlabeled data, addressing the limitations of confidence-only selection. For VLMs, CAP [56] addresses pseudo-label imbalance via concept alignment and confusion-aware class-pair margin calibration. In the related noisy-label setting, DKAF [42] mitigates endogenous confirmation bias through cross-modal sample selection and label refinement. Our work targets semi-supervised input VR with a frozen VLM, using prediction margins for sample weighting and agreement between attributes to assess pseudo-label reliability.

3 Methodology

3.1 Framework overview

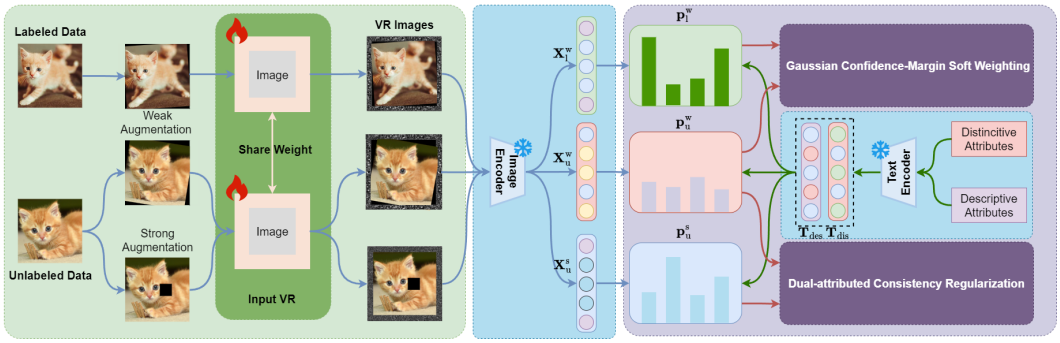


Figure 3: Overview of CURE for semi-supervised VR. CURE optimizes shared input-space VR patterns with both labeled and unlabeled data while keeping the VLM backbone frozen.

In this work, we propose **CURE** (**C**onsistency and **U**nlabeled **R**ecalibration with Confidence-Margin **E**nhancement), a semi-supervised learning framework that leverages both labeled and unlabeled data to optimize visual reprogramming patterns. CURE is designed around two reliability questions for unlabeled data in SSL-VR: *(i)* is the pseudo-label sufficiently separated from competing classes? *(ii)* is the pseudo-label semantically stable across complementary attribute views? The posterior and margin recalibration modules address the first question by correcting class-level bias and estimating sample-level class separation. The dual-attribute consistency module addresses the second question by measuring agreement between descriptive and distinctive prompt-induced posteriors.

Specifically, we follow the standard semi-supervised adaptation protocol, where unlabeled images are sampled from the same target dataset (i.e., no additional unknown classes are introduced). As illustrated in Figure 3, the CURE pipeline consists of three main components: *(i) Task-Adaptive Image Processing*, which generates augmented and reprogrammed views of both labeled and unlabeled data; *(ii) Attribute-Guided Data Encoding*, which encodes images and attribute-enhanced textual descriptions into a shared embedding space via the frozen VLM; *(iii) Semi-Supervised Optimization*, which jointly optimizes semi-supervised objectives to update the learnable reprogramming patterns.

The *Task-Adaptive Image Processing* module, highlighted in green in Figure 3, begins with sampling labeled images $X_l \in \mathcal{D}_l$ and unlabeled images $X_u \in \mathcal{D}_u$, where each image is represented as $\mathbb{R}^{H\times W\times C}$, with H, W, and C denoting the height, width, and number of channels, respectively. For every image, we generate a weakly augmented view using the operator $\psi(\cdot)$; for unlabeled samples, an additional strongly augmented view is constructed with the operator $\Psi(\cdot)$:

$$X_l^w = \psi(X_l), \quad X_u^w = \psi(X_u), \quad X_u^s = \Psi(X_u), \tag{1}$$

where the detailed augmentation pipelines and hyperparameters are provided in Appendix B.

Next, we apply a visual reprogramming operator $\Phi_\omega(\cdot)$ that pads each view with a *learnable* border pattern while keeping the central content unchanged. Let $\mathbf{M} \in \{\mathbf{0}, \mathbf{1}\}^{\mathbf{H}'\times\mathbf{W}'\times\mathbf{C}}$ be a fixed mask that places the original image at the center, and $\mathbf{W}_\omega$ denote the learnable padding. Then Φ_ω is defined as:

$$\Phi_\omega(X) = \mathbf{M} \odot X + (1 - \mathbf{M}) \odot \mathbf{W}_\omega. \tag{2}$$

This operation preserves the original semantics while injecting a learnable pattern into each view.

In the *Attribute-Guided Data Encoding* module, illustrated in Figure 3 (highlighted in blue), we feed reprogrammed views into the image encoder $f_{\text{img}}(\cdot)$ of VLMs to obtain the visual embeddings:

$$\mathbf{X} = f_{\text{img}}(\Phi_\omega(X)). \tag{3}$$

To convert visual embeddings into class-wise predictions, we align them with attribute-guided textual embeddings produced by the text encoder $f_{\text{txt}}(\cdot)$. For each class $c \in \{1, \ldots, C\}$, there are two sets of attributes: $\mathcal{A}_{\text{des}}(c) = \{a_{\text{des},c,r}\}_{r=1}^{M_{\text{des}}(c)}$ and $\mathcal{A}_{\text{dis}}(c) = \{a_{\text{dis},c,r}\}_{r=1}^{M_{\text{dis}}(c)}$, which characterize descriptive and distinctive visual evidence, respectively. The details of the prompts and settings for generating these attributes are provided in Appendix D. The frozen VLM text encoder maps each attribute phrase into a normalized phrase-level embedding:

$$\mathbf{t}_{v,c,r} = \text{norm}(f_{\text{txt}}(a_{v,c,r})), \qquad v \in \{\text{des}, \text{dis}\},\ r = 1, \ldots, M_v(c). \tag{4}$$

For an image embedding $\mathbf{x}$, we compute view-specific score by selecting the most matched attribute:

$$s_v^{(c)}(\mathbf{x}) = \max_{1 \le r \le M_v(c)} \text{sim}(\mathbf{x}, \mathbf{t}_{v,c,r}), \qquad v \in \{\text{des}, \text{dis}\}. \tag{5}$$

The fused attribute score for class c is then defined as:

$$S_{\text{VLM}}^{(c)}(\mathbf{x}) = \alpha \cdot s_{\text{des}}^{(c)}(\mathbf{x}) + (1 - \alpha) \cdot s_{\text{dis}}^{(c)}(\mathbf{x}), \tag{6}$$

where $\alpha \in [0, 1]$ is a predefined fusion coefficient, set to 0.5 by default. Here, $\mathbf{S}_{\text{VLM}}(\mathbf{x}) = [S_{\text{VLM}}^{(1)}(\mathbf{x}), \ldots, S_{\text{VLM}}^{(C)}(\mathbf{x})]$ denotes the class-wise pre-softmax score vector. For labeled samples, we adopt the standard supervised cross-entropy objective to align the fused predictions with the ground-truth labels. Given a labeled dataset $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1}^{N_l}$, the supervised loss is defined as:

$$\mathcal{L}_s = -\frac{1}{N_l} \sum_{i=1}^{N_l} \log \left[\text{softmax}(\mathbf{S}_{\text{VLM}}(\mathbf{x}_i))\right]_{y_i}. \tag{7}$$

Since labeled samples are scarce while unlabeled samples are abundant in many downstream tasks, CURE jointly optimizes the supervised loss $\mathcal{L}_s$ and the unsupervised loss $\mathcal{L}_u$:

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_u, \tag{8}$$

where λ balances the supervised and unsupervised objectives. We set $\lambda = 1$ by default, giving both terms equal weight during optimization. The unsupervised loss $\mathcal{L}_u$ is constructed with recalibrated Gaussian confidence-margin soft weighting and dual-attribute consistency regularization.

3.2 Recalibrated Gaussian confidence-margin soft weighting

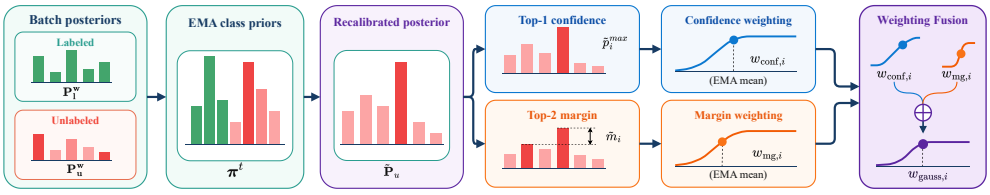


Figure 4: Overview of the Recalibrated Gaussian Confidence-Margin Module.

With limited labeled samples, the empirical class prior estimated from labeled data may deviate from the marginal distribution implicitly induced by model predictions on unlabeled data. This mismatch may cause *confirmation bias*: dominant or easier classes accumulate more high-confidence pseudo-labels, while other classes are under-represented and further suppressed during training. To mitigate this drift, we introduce a lightweight posterior recalibration that uses the labeled prediction prior as a weak anchor while preserving the prediction structure of unlabeled data, shown in Figure 4.

Class-prior estimation. We maintain Exponential Moving Average [49] (EMA) class-prior estimates, $\boldsymbol{\pi}_l, \boldsymbol{\pi}_u \in \Delta^{C-1}$, for labeled and unlabeled predictions, respectively. Both priors are initialized

as uniform distributions, i.e., $\boldsymbol{\pi} = 1/C$ for each class c. Given a batch-level prediction matrix $\mathbf{P} \in \mathbb{R}^{B\times C}$, we compute its marginal class distribution $\bar{\mathbf{p}}$ and corresponding EMA prior $\boldsymbol{\pi}^{\boldsymbol{t}}$:

$$\bar{\mathbf{p}} = \frac{1}{B}\sum_{i=1}^{B} \mathbf{P}_{i,:}, \tag{9}$$

$$\boldsymbol{\pi}^t = \alpha \boldsymbol{\pi}^{t-1} + (1-\alpha)\bar{\mathbf{p}}, \tag{10}$$

where α is the momentum parameter (default 0.99) and $\mathbf{P}$ is defined in Appendix Equation (29). Applying update separately to labeled predictions $\mathbf{P}_l$ and unlabeled predictions $\mathbf{P}_u$ yields $\boldsymbol{\pi}_l$ and $\boldsymbol{\pi}_u$.

Posterior recalibration. Given an unlabeled posterior matrix $\mathbf{P}_u \in \mathbb{R}^{B_u\times C}$, where $\mathbf{P}_{u,i,c}$ denotes the probability of class c for the i-th unlabeled sample, we recalibrate each posterior by class-wise prior correction ($\boldsymbol{\pi}_l$ and $\boldsymbol{\pi}_u$) followed by row-wise normalization:

$$\tilde{\mathbf{P}}_{u,i,c} = \frac{\mathbf{P}_{u,i,c}\cdot \frac{\boldsymbol{\pi_{l,c}}}{\boldsymbol{\pi_{u,c}}+\varepsilon}}{\sum_{j=1}^{C} \mathbf{P}_{u,i,j}\cdot \frac{\boldsymbol{\pi_{l,j}}}{\boldsymbol{\pi_{u,j}}+\varepsilon}}, \qquad i = 1,\ldots,B_u,\ c = 1,\ldots,C. \tag{11}$$

Here, ε is a small constant for numerical stability. The normalization ensures that $\tilde{\mathbf{P}}_{u,i,:} \in \Delta^{C-1}$, so the recalibrated output remains a valid posterior distribution. This operation gently corrects class-level bias in unlabeled predictions without imposing hard pseudo-label constraints.

Margin recalibration. While posterior recalibration mitigates class-level bias, pseudo-label reliability also depends on whether the predicted class is sufficiently separated from its closest competing class. For the i-th unlabeled sample, we therefore define the top-2 margin after posterior recalibration:

$$m_i = \tilde{\mathbf{P}}_{u,i,c_i^{(1)}} - \tilde{\mathbf{P}}_{u,i,c_i^{(2)}}, \tag{12}$$

where $c_i^{(1)}$ and $c_i^{(2)}$ denote the classes with the largest and second-largest probabilities in $\tilde{\mathbf{P}}_{u,i,:}$, respectively. A smaller margin indicates that the pseudo-label is less separated from its closest competing class and is therefore more ambiguous. To make the margin statistics comparable between labeled and unlabeled predictions, we maintain EMA estimates of the mean margins, denoted by μ_l^{mg} and μ_u^{mg}, for labeled and unlabeled predictions. We recalibrate each unlabeled margin by:

$$\tilde{m}_i = \frac{\mu_l^{\mathrm{mg}}}{\mu_u^{\mathrm{mg}}+\varepsilon}\, m_i, \qquad i = 1,\ldots,B_u, \tag{13}$$

where ε is a small constant for numerical stability and B_u is the number of unlabeled samples. Here, m_i and $\tilde{m}_i$ are per-sample quantities, while μ_l^{mg} and μ_u^{mg} are running batch-level statistics. This recalibration aligns the unlabeled margin scale with the labeled margin statistics and provides a sample-level class-separation signal for the subsequent soft weighting.

Gaussian confidence–margin soft weighting. Built upon posterior and margin recalibration, we replace fixed confidence thresholds and uniform sample weights with a continuous reliability estimate for each unlabeled sample. For the i-th unlabeled sample, let $\tilde{p}_i^{\max} = \max_c \tilde{\mathbf{P}}_{u,i,c}$ denote its maximum recalibrated probability. We maintain EMA estimates μ_p and σ_p^2 over $\tilde{p}_i^{\max}$, and μ_m and σ_m^2 over the recalibrated margin $\tilde{m}_i$, across unlabeled samples. We then define a lower-tail truncated Gaussian confidence weight and a lower-tail truncated Gaussian margin weight:

$$w_{\mathrm{conf},i} = \exp\left(-\frac{\max(0,\,\mu_p - \tilde{p}_i^{\max})^2}{2(\sigma_p^2+\varepsilon)}\right), \tag{14}$$

$$w_{\mathrm{mg},i} = \exp\left(-\frac{\max(0,\,\mu_m - \tilde{m}_i)^2}{2(\sigma_m^2+\varepsilon)}\right). \tag{15}$$

The final Gaussian confidence-margin weight is computed by averaging for each unlabeled sample:

$$w_{\mathrm{gauss},i} = \tfrac{1}{2}\left(w_{\mathrm{conf},i} + w_{\mathrm{mg},i}\right) \in [0,1]. \tag{16}$$

The one-sided truncation is intentional: samples whose confidence or margin is above the running mean are treated as sufficiently reliable and receive weights close to 1, while samples below the running mean are smoothly down-weighted according to their deviation from the current reliability statistics. In this way, $w_{\mathrm{gauss},i}$ avoids the brittleness of hard thresholds, preserves informative uncertain samples, and reduces the influence of highly ambiguous pseudo-labels. Potentially semantically inconsistent predictions are further handled by the dual-attribute consistency weight introduced next.

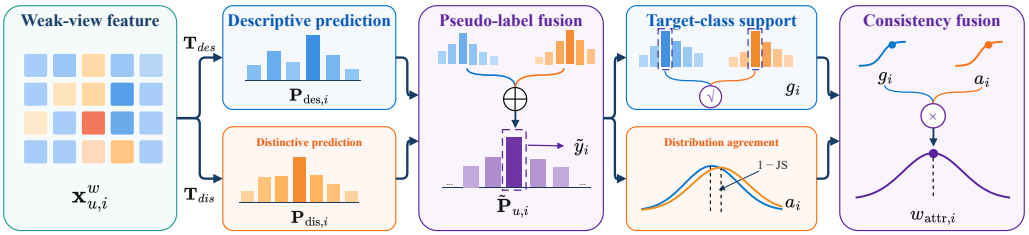


Figure 5: Overview of Dual-attribute Consistency Regularization Module.

3.3 Dual-attribute consistency regularization

Although Gaussian confidence–margin soft weighting alleviates the rigidity of fixed confidence thresholds, it only measures prediction strength through confidence and class separation. It does not assess whether the pseudo-label is semantically stable across different textual views. In attribute-based VR, a pseudo-label may appear confident under one prompt view but become unreliable under another, especially when the image matches only common class attributes. We introduce *dual-attribute consistency regularization*, which measures the agreement between predictions induced by descriptive and distinctive attribute prompts, as shown in Figure 5. This agreement serves as a semantic reliability signal for modulating the attribute-based unsupervised confidence and margin.

Formally, for each class c, the descriptive and distinctive attribute embeddings have been defined in Equation (4). For the weakly augmented unlabeled image embedding $\mathbf{x}^w_{u,i}$, following Equation (5), we obtain the view-specific scores $s_{\mathrm{des},i,c}$ and $s_{\mathrm{dis},i,c}$ by max-pooling over phrase-level attributes within the same class. The corresponding view-specific posteriors $\mathbf{P}_{\mathrm{des},i}$ and $\mathbf{P}_{\mathrm{dis},i}$ are obtained by applying softmax over classes, as detailed in Appendix A.2. Meanwhile, following Equation (6), the fused posterior used for pseudo-label generation is computed as:

$$P_{u,i,c} = \frac{\exp\left(S^{(c)}_{\mathrm{VLM}}(\mathbf{x}^w_{u,i})\right)}{\sum_{k=1}^{C}\exp\left(S^{(k)}_{\mathrm{VLM}}(\mathbf{x}^w_{u,i})\right)}. \tag{17}$$

Let $\mathbf{P}_{u,i} = [P_{u,i,1}, \ldots, P_{u,i,C}]$ denote the fused posterior distribution for the i-th unlabeled sample. The pseudo-label for the i-th unlabeled sample is obtained from the recalibrated posterior of $\mathbf{P}_{u,i}$:

$$\tilde{y}_i = \arg\max_c(\tilde{\mathbf{P}}_{u,i}), \tag{18}$$

where $\tilde{\mathbf{P}}_u$ is defined in Equation (11). The target confidence at $\tilde{y}_i$ is obtained via geometric mean:

$$g_i = \left(\mathbf{P}_{\mathrm{des},i,\tilde{y}_i} \cdot \mathbf{P}_{\mathrm{dis},i,\tilde{y}_i}\right)^{1/2}. \tag{19}$$

This term is large only when both descriptive and distinctive prompt views assign high probability to the same pseudo-label $\tilde{y}_i$. In addition to target confidence, we measure the distributional agreement between the two full attribute-view posteriors using the Jensen-Shannon divergence:

$$a_i = 1 - \frac{\mathrm{JS}\left(\mathbf{P}_{\mathrm{des},i} \| \mathbf{P}_{\mathrm{dis},i}\right)}{\log 2} \in [0,1], \tag{20}$$

where the normalization by $\log 2$ ensures that a_i lies in $[0,1]$ when natural logarithms are used for computing the JS divergence. Combining target confidence and distributional agreement, the dual-attribute consistency weight is then defined as follows:

$$w_{\mathrm{attr},i} = g_i \cdot a_i. \tag{21}$$

Thus, $w_{\mathrm{attr},i}$ increases only when both attribute views confidently support the same pseudo-label and their overall posterior distributions are consistent. A large discrepancy between the two posteriors often indicates a harder or semantically ambiguous sample, where descriptive and distinctive cues yield unstable class preferences. We therefore down-weight such unstable pseudo-labels to reduce noisy supervision, with further justification in Appendix G. Finally, for each unlabeled pair $(\mathbf{x}^w_{u,i}, \mathbf{x}^s_{u,i})$, let $q_\theta(y \mid \mathbf{x}^s_{u,i})$ denote student prediction on strongly augmented view. The unsupervised objective is:

$$\mathcal{L}_u = \frac{1}{B_u}\sum_{i=1}^{B_u} w_{\mathrm{gauss},i} \cdot w_{\mathrm{attr},i} \cdot \mathrm{H}\left(\tilde{y}_i, q_\theta(y \mid \mathbf{x}^s_{u,i})\right), \tag{22}$$

where $\mathrm{H}(\cdot,\cdot)$ denotes cross-entropy. The weights $w_{\mathrm{gauss},i}$ and $w_{\mathrm{attr},i}$ measure confidence-margin reliability and cross-attribute semantic consistency, respectively. Together, they allow CURE to exploit informative unlabeled samples while reducing the risk of semantically inconsistent pseudo-labels.

4 Experiments

4.1 Experimental setup

Datasets. We evaluate downstream classification tasks following prior works [6, 8, 7] on twelve public datasets spanning generic objects, fine-grained categories, and remote sensing: FGVC-Aircraft (Aircraft) [38], Caltech101 (Caltech) [22], Stanford Cars (Cars) [33], Describable Textures (DTD) [16], EuroSAT [27], Flowers-102 (Flowers) [41], Food-101 (Food) [3], Oxford-Pets (Pets) [44], SUN397 (SUN) [58], UCF101 (UCF) [48], ImageNet (IN) [17], and RESISC45 (Resisc) [15].

Baselines. We adopt CLIP as the pre-trained backbone and evaluate all methods in the 2-shot setting. We compare CURE with six baselines: *(i)* zero-shot CLIP (ZS); *(ii)* VP [1], which overlays learnable perturbations onto the image; *(iii)* AR [9], which places learnable patterns around the image; *(iv)* AttrVR [7], which uses descriptive and distinctive attribute prompts; *(v)* BayesVR [5], which models predictive uncertainty; and *(vi)* DAVR [8], which uses diverse attribute prompts.

Implementation details. All trainable baselines and CURE follow the same 200-epoch SGD training protocol, while all SSL-based methods share the same unlabeled pool, unlabeled-to-labeled ratio $\gamma = 7$, and data augmentation pipeline. The main experiments are conducted with three different random seeds (0, 42, 2025). Besides, we provide more details about implementation in Appendix B.

4.2 Results and analysis

Table 1: Accuracy comparison of baselines trained on 2-shot downstream tasks, where "- -" indicates that the result is unavailable: DAVR does not provide the required parameters for the dataset.

Method	Aircraft	Caltech	Cars	DTD	EuroSAT	Flowers	Food	ImageNet	Pets	RESISC	SUN	UCF	Avg.
ZS	21.36	88.44	59.57	37.41	16.80	60.54	77.97	59.40	81.49	42.33	59.92	62.54	55.65
VP	$24.30_{\pm 0.53}$	$91.56_{\pm 1.06}$	$59.17_{\pm 0.76}$	$45.43_{\pm 0.03}$	$73.54_{\pm 2.84}$	$66.48_{\pm 1.11}$	$81.32_{\pm 0.48}$	$62.20_{\pm 0.19}$	$89.01_{\pm 0.58}$	$66.91_{\pm 1.29}$	$61.88_{\pm 0.28}$	$67.24_{\pm 1.03}$	65.75
AR	$25.34_{\pm 0.51}$	$94.32_{\pm 0.35}$	$65.29_{\pm 0.21}$	$48.82_{\pm 0.46}$	$84.93_{\pm 1.17}$	$73.68_{\pm 1.57}$	$83.86_{\pm 0.24}$	$64.91_{\pm 0.24}$	$90.49_{\pm 0.55}$	$72.30_{\pm 1.93}$	$64.96_{\pm 0.25}$	$71.43_{\pm 1.09}$	70.03
AttrVR	$30.22_{\pm 0.30}$	$95.29_{\pm 0.19}$	$64.11_{\pm 0.75}$	$52.78_{\pm 0.88}$	$81.95_{\pm 3.43}$	$83.84_{\pm 1.02}$	$84.79_{\pm 0.08}$	$67.85_{\pm 0.08}$	$92.29_{\pm 0.59}$	$73.80_{\pm 1.34}$	$64.11_{\pm 0.20}$	$71.34_{\pm 0.51}$	71.86
BayesVR	$20.69_{\pm 0.62}$	$91.13_{\pm 0.71}$	$53.06_{\pm 1.56}$	$38.12_{\pm 3.04}$	$49.79_{\pm 11.37}$	$66.45_{\pm 1.69}$	$75.33_{\pm 1.61}$	$50.30_{\pm 1.41}$	$82.65_{\pm 2.60}$	$55.16_{\pm 2.81}$	$51.58_{\pm 0.16}$	$60.03_{\pm 1.81}$	57.86
DAVR	$29.61_{\pm 0.83}$	$94.52_{\pm 0.11}$	$63.78_{\pm 0.58}$	$55.24_{\pm 0.61}$	$81.56_{\pm 2.41}$	$87.94_{\pm 0.19}$	$84.28_{\pm 0.39}$	– –	$91.84_{\pm 0.63}$	$71.27_{\pm 3.85}$	$62.46_{\pm 0.43}$	$72.13_{\pm 0.84}$	– –
CURE (Ours)	$\mathbf{31.50}_{\pm 0.77}$	$\mathbf{95.82}_{\pm 0.42}$	$\mathbf{69.73}_{\pm 0.28}$	$\mathbf{59.08}_{\pm 1.45}$	$\mathbf{85.33}_{\pm 2.01}$	$\mathbf{91.96}_{\pm 0.54}$	$\mathbf{85.00}_{\pm 0.13}$	$\mathbf{68.68}_{\pm 0.17}$	$\mathbf{92.66}_{\pm 0.26}$	$\mathbf{79.06}_{\pm 1.17}$	$\mathbf{68.60}_{\pm 0.53}$	$\mathbf{76.33}_{\pm 0.52}$	**75.31**

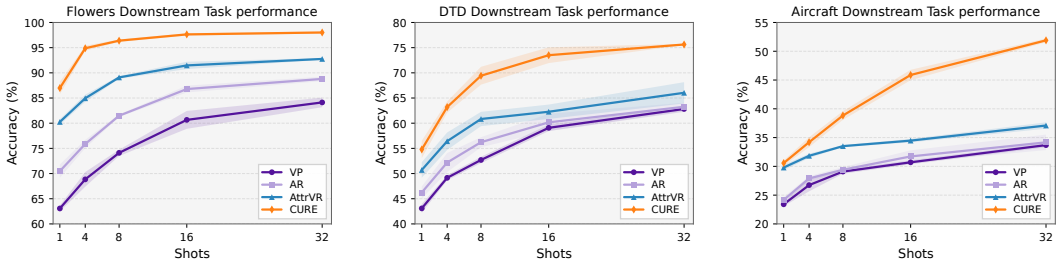


Figure 6: Accuracy comparison on different shots from $[1, 4, 8, 16, 32]$. Based on pre-trained ViT-B16 CLIP, the striped area indicates the error bars from three random seeds.

Overall performance comparison. Using CLIP with a ViT-B/16 image encoder, Table 1 shows that attribute augmentation alone is not consistently effective under limited supervision. Despite incorporating descriptive and distinctive attribute prompts, AttrVR achieves only a modest average improvement of $\mathbf{1.83}$ percentage points over AR, while underperforming on the Cars, EuroSAT, SUN, and UCF datasets. By contrast, CURE achieves the best accuracy on all twelve datasets and raises the average accuracy to $\mathbf{75.31}\%$, outperforming the strongest competing baseline by $\mathbf{3.45}$ percentage points. The improvement is especially clear on fine-grained datasets such as Cars, Flowers, and UCF datasets. Beyond CLIP, CURE also shows strong generality on SigLIP [63] and SigLIP2 [52]. As shown in Appendix C.2, CURE achieves the best average accuracy on both backbones, with gains of 5.59 and 4.95 percentage points over the baselines. These results demonstrate the effectiveness of CURE in leveraging unlabeled data and attribute cues to improve VR under limited supervision.

Evaluation of sample efficiency. To assess robustness under varying numbers of labeled samples, we evaluate the main VR variants on the Flowers, DTD, and Aircraft datasets with $K \in [1, 4, 8, 16, 32]$ shots per class. As shown in Figure 6, CURE consistently outperforms the baselines at every shot level, demonstrating strong resilience to data scarcity and effective exploitation of unlabeled data.

Table 2: Accuracy comparison of different semi-supervised learning methods for AttrVR framework.

Method	Aircraft	Caltech	Cars	DTD	EuroSAT	Flowers	Food	ImageNet	Pets	RESISC	SUN	UCF	Avg.
FixMatch	$29.87_{\pm 0.12}$	$94.83_{\pm 0.19}$	$63.94_{\pm 0.59}$	$54.10_{\pm 2.32}$	$80.91_{\pm 1.81}$	$80.63_{\pm 0.90}$	$84.02_{\pm 0.20}$	$67.67_{\pm 0.14}$	$92.32_{\pm 0.77}$	$71.82_{\pm 0.96}$	$63.39_{\pm 0.24}$	$70.00_{\pm 0.32}$	71.13
SoftMatch	$28.14_{\pm 0.56}$	$94.86_{\pm 0.16}$	$63.74_{\pm 0.51}$	$54.18_{\pm 1.80}$	$77.39_{\pm 2.19}$	$79.78_{\pm 0.92}$	$84.22_{\pm 0.06}$	$67.74_{\pm 0.10}$	$92.34_{\pm 0.49}$	$71.14_{\pm 1.99}$	$63.67_{\pm 0.36}$	$70.46_{\pm 0.68}$	70.64
CGMatch	$29.62_{\pm 0.11}$	$94.94_{\pm 0.35}$	$64.30_{\pm 0.61}$	$55.56_{\pm 2.46}$	$81.47_{\pm 4.54}$	$80.89_{\pm 1.33}$	$84.22_{\pm 0.07}$	$67.81_{\pm 0.13}$	$92.52_{\pm 0.55}$	$72.94_{\pm 0.98}$	$63.59_{\pm 0.28}$	$70.64_{\pm 0.35}$	71.54
FlexMatch	$28.19_{\pm 0.62}$	$94.86_{\pm 0.22}$	$62.87_{\pm 0.25}$	$54.67_{\pm 1.19}$	$77.21_{\pm 1.02}$	$79.69_{\pm 1.64}$	$84.88_{\pm 0.24}$	$67.27_{\pm 0.09}$	$92.12_{\pm 0.53}$	$70.74_{\pm 0.79}$	$63.19_{\pm 0.48}$	$69.05_{\pm 0.33}$	70.40
FreeMatch	$29.64_{\pm 0.38}$	$94.96_{\pm 0.23}$	$64.13_{\pm 0.21}$	$55.71_{\pm 1.02}$	$79.07_{\pm 2.82}$	$79.97_{\pm 1.42}$	$84.23_{\pm 0.20}$	$67.86_{\pm 0.11}$	$92.45_{\pm 0.54}$	$72.53_{\pm 1.30}$	$63.86_{\pm 0.51}$	$69.84_{\pm 1.18}$	71.19
CURE (Ours)	$\mathbf{31.50}_{\pm 0.77}$	$\mathbf{95.82}_{\pm 0.42}$	$\mathbf{69.73}_{\pm 0.28}$	$\mathbf{59.08}_{\pm 1.45}$	$\mathbf{85.33}_{\pm 2.01}$	$\mathbf{91.96}_{\pm 0.54}$	$\mathbf{85.00}_{\pm 0.13}$	$\mathbf{68.68}_{\pm 0.17}$	$\mathbf{92.66}_{\pm 0.26}$	$\mathbf{79.06}_{\pm 1.17}$	$\mathbf{68.60}_{\pm 0.53}$	$\mathbf{76.33}_{\pm 0.52}$	**75.31**

Evaluation of SSL extensions under the AttrVR setting. We further evaluate whether the improvement of CURE can be attributed simply to the use of unlabeled data. To this end, we integrate representative SSL methods into AttrVR, including FixMatch [47], SoftMatch [10], CGMatch [14], FlexMatch [64], and FreeMatch [55], and compare all methods under the same data split across 12 datasets. As reported in Table 2, these direct SSL extensions consistently underperform AttrVR, reaching 70.40%–71.54% average accuracy, whereas CURE delivers substantially better performance.

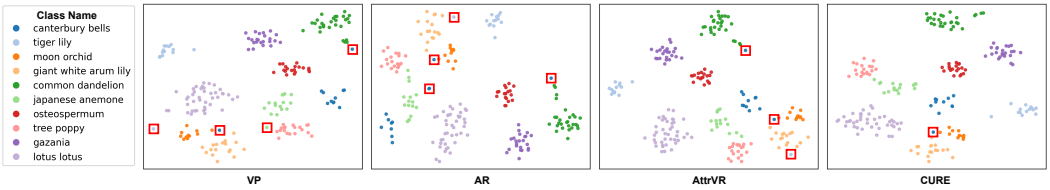


Figure 7: T-SNE of Flowers dataset embeddings. Image embeddings from ten Flowers classes are projected with t-SNE. Embeddings are extracted from VR-patterned images trained with four methods. Points are colored by class, and red boxes denote misclassified instances.

Evaluation of embedding space. Figure 7 provides a qualitative visualization of 2D t-SNE [37] embeddings for 10 *Flowers* classes under four main input VR settings, where colors denote ground-truth classes and red-boxed points denote misclassified samples. Compared with CURE, the supervised VR baselines produce more misclassified samples, which are frequently located near inter-class boundaries or closer to neighboring class clusters. This suggests that many errors arise from hard samples whose representations are insufficiently separated from semantically similar classes under scarce labeled supervision. By contrast, CURE produces fewer misclassified samples and forms more compact and better-separated clusters. Since t-SNE is a low-dimensional qualitative visualization, we further provide quantitative class-structure statistics in the original embedding space in Appendix C.4. The results show that CURE achieves the highest Fisher ratio and centroid-based silhouette score, confirming that the observed improvement is also reflected in the high-dimensional feature space.

4.3 Ablation Studies

Ablation of CURE. We ablate all proposed designs in CURE to examine their individual contributions. As shown in Table 3, removing posterior recalibration (*w/o RE*), Gaussian confidence-margin soft weighting (*w/o GA*), dual-attribute consistency regularization (*w/o CO*), EMA (*w/o EM*), or Top-2 margin (*w/o MA*) consistently degrades performance. Overall, these consistent performance drops show that each proposed component contributes effectively to CURE, with confidence-margin weighting playing the most critical role in stabilizing pseudo supervision.

Sensitivity analysis. We examine the sensitivity of CURE to automatically generated attributes by replacing the original GPT-3.5 [4] attributes with alternatives generated by GPT-4oMini-LLM [28], GPT-4oMini-MLLM [28], and LLaMA-3.1 [24], using the same fixed prompts. As shown in Table 4, CURE performs consistently with GPT-4o-generated attributes but degrades with LLaMA-3.1 and GPT-3.5, indicating slight sensitivity to attribute-generation quality. Hyperparameter sensitivity and labeled-data split sensitivity analyses in Appendix C.5 further show that CURE remains stable under the default configuration, without dataset-specific, split-specific, or attribute-source-specific tuning.

Table 3: Ablation of CURE.

	Aircraft	SAT	Food	Avg.
w/o RE	31.19	84.81	84.97	66.99
w/o GA	31.15	84.72	84.75	66.87
w/o CO	30.25	84.60	84.15	66.33
w/o EM	31.14	84.43	84.90	66.82
w/o MA	31.31	84.93	84.02	66.75
CURE	**31.50**	**85.33**	**85.00**	**67.28**

Table 4: Sensitivity results of CURE to attributes.

LLM	**DTD**	**Aircraft**	**Flowers**	**Avg.**
GPT-3.5	59.08	31.50	91.96	60.85
4o-LLM	60.99	32.62	92.73	62.11
4o-MLLM	62.81	32.38	92.85	62.68
LLaMA-3.1	55.08	31.50	88.36	58.31
$\text{Mean}_{\pm\text{Std}}$	$59.49_{\pm 2.87}$	$32.00_{\pm 0.51}$	$91.48_{\pm 1.83}$	$60.99_{\pm}1.68$

Complexity analysis. Although exploiting unlabeled data inevitably increases training complexity and runtime, the gains are not merely due to longer time for optimization, since reduced-epoch CURE still outperforms supervised AR, whereas longer-trained AR yields little additional improvement. Appendix F provides a detailed training-cost trade-off analysis and budget-controlled comparisons.

5 Conclusion

In this paper, we propose CURE, a semi-supervised visual reprogramming framework for adapting frozen VLMs to downstream classification tasks under limited supervision. CURE leverages unlabeled target-domain images through recalibrated Gaussian confidence–margin soft weighting and dual-attribute consistency regularization, which provides more reliable and semantically consistent unlabeled supervision. Extensive experiments demonstrate that CURE consistently surpasses supervised VR baselines and direct SSL extensions across twelve widely used datasets, highlighting the value of VR-specific unlabeled-data utilization. As CURE follows the standard SSL assumption that unlabeled data are primarily in-domain, its main limitation is the handling of out-of-domain unlabeled samples, which we discuss as an important direction for future work in Appendix J.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (62676119), Guangdong Provincial Key Laboratory of Artificial General Intelligence and Large Models (2026B1212010005), Guangdong Science and Technology Program (2025B0101130003), Shenzhen College Stability Support Plan (GXWD20231130151329002), and Shenzhen Science and Technology Program(KQTD20240729102207002).

References

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

[2] Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, et al. An introduction to vision-language modeling. *arXiv preprint arXiv:2405.17247*, 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] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Bayesian-guided label mapping for visual reprogramming. In *Advances in Neural Information Processing Systems*, 2024.

[6] Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Sample-specific masks for visual reprogramming-based prompting. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[7] Chengyi Cai, Zesheng Ye, Lei Feng, Jianzhong Qi, and Feng Liu. Attribute-based visual reprogramming for image classification with clip. In *International Conference on Learning Representations*, 2025.

[8] 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*, 2025.

[9] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and improving visual prompting: A label-mapping perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19133–19143, 2023.

[10] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint arXiv:2301.10921*, 2023.

[11] Pin-Yu Chen. Model reprogramming: Resource-efficient cross-domain machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22584–22591, 2024.

[12] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[13] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1081–1090, 2019.

[14] Bo Cheng, Jueqing Lu, Yuan Tian, Haifeng Zhao, Yi Chang, and Lan Du. Cgmatch: A different perspective of semi-supervised learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15381–15391, 2025.

[15] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[16] 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.

[17] 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.

[18] 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. *arXiv preprint arXiv:2010.11929*, 2020.

[19] Pan Du, Wangbo Zhao, Xinai Lu, Nian Liu, Zhikai Li, Chaoyu Gong, Suyun Zhao, Hong Chen, Cuiping Li, Kai Wang, et al. Unsupervised learning for class distribution mismatch. In *Forty-second International Conference on Machine Learning*, 2025.

[20] Shiv Ram Dubey. A decade survey of content based image retrieval using deep learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):2687–2704, 2021.

[21] Matthias Englert and Ranko Lazic. Adversarial Reprogramming Revisited. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 28588–28600. Curran Associates, Inc., 2022.

[22] 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 *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[23] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[24] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[25] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. WARP: Word-level Adversarial ReProgramming. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online, August 2021. Association for Computational Linguistics.

[26] Cheng Han, Qifan Wang, Yiming Cui, Zhiwen Cao, Wenguan Wang, Siyuan Qi, and Dongfang Liu. Eˆ 2vpt: An effective and efficient approach for visual prompt tuning. *arXiv preprint arXiv:2307.13770*, 2023.

[27] 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.

[28] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[29] Lin-Han Jia, Lan-Zhe Guo, Zhi Zhou, and Yu-Feng Li. Realistic evaluation of semi-supervised learning algorithms in open environments. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022.

[31] Yongcheng Jing, Chongbin Yuan, Li Ju, Yiding Yang, Xinchao Wang, and Dacheng Tao. Deep graph reprogramming. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24345–24354, 2023.

[32] 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.

[33] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[34] Jinhao Li, Haopeng Li, Sarah Erfani, Lei Feng, James Bailey, and Feng Liu. Visual-text cross alignment: Refining the similarity score in vision-language models. *arXiv preprint arXiv:2406.02915*, 2024.

[35] Yihao Li, Mostafa El Habib Daho, Pierre-Henri Conze, Rachid Zeghlache, Hugo Le Boité, Ramin Tadayoni, Béatrice Cochener, Mathieu Lamard, and Gwenolé Quellec. A review of deep learning-based information fusion techniques for multimodal medical image classification. *Computers in Biology and Medicine*, 177:108635, 2024.

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

[37] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

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

[39] Cristina Menghini, Andrew Delworth, and Stephen Bach. Enhancing clip with clip: Exploring pseudolabeling for limited-label prompt tuning. *Advances in Neural Information Processing Systems*, 36:60984–61007, 2023.

[40] Amir Najafi, Alireza Heidari, Mohammad Hosein Movasaghinia, Abolfazl Motahari, Babak Khalaj, et al. Out-of-domain unlabeled data improves generalization. In *The Twelfth International Conference on Learning Representations*, 2024.

[41] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[42] Feiyang Ning and Xinyang Chen. Mitigating endogenous confirmation bias in noisy label learning for vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 24576–24584, 2026.

[43] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24224–24235, 2023.

[44] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[45] 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.

[46] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[47] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[48] 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*, 2(11):1–7, 2012.

[49] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[50] 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.

[51] 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 *International Conference on Machine Learning*, pages 9614–9624. PMLR, 2020.

[52] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[53] Ria Vinod, Pin-Yu Chen, and Payel Das. Reprogramming language models for molecular representation learning. *arXiv preprint arXiv:2012.03460*, 2020.

[54] Wenhao Wang, Yifan Sun, Wei Li, and Yi Yang. Transhp: Image classification with hierarchical prompting. *Advances in Neural Information Processing Systems*, 36:28187–28200, 2023.

[55] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2022.

[56] Yuchen Wang, Xuefeng Bai, Xiucheng Li, Weili Guan, Liqiang Nie, and Xinyang Chen. Handling imbalanced pseudolabels for vision-language models with concept alignment and confusion-aware calibrated margin. In *Forty-second International Conference on Machine Learning*, 2025.

[57] 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*, pages 29347–29356, 2026.

[58] 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.

[59] Wanni Xu, You-Lei Fu, and Dongmei Zhu. Resnet and its application to medical image processing: Research progress and challenges. *Computer Methods and Programs in Biomedicine*, 240:107660, 2023.

[60] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Yang Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3090–3100, 2023.

[61] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE transactions on knowledge and data engineering*, 35(9):8934–8954, 2022.

[62] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19187–19197, 2023.

[63] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[64] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in neural information processing systems*, 34:18408–18419, 2021.

[65] Jiahan Zhang, Qi Wei, Feng Liu, and Lei Feng. Candidate pseudolabel learning: enhancing vision-language models by prompt tuning with unlabeled data. In *Proceedings of the 41st International Conference on Machine Learning*, pages 60004–60020, 2024.

[66] Yichi Zhang, Zhenrong Shen, and Rushi Jiao. Segment anything model for medical image segmentation: Current applications and future directions. *Computers in Biology and Medicine*, 171:108238, 2024.

[67] Yuhui Zhang, Alyssa Unell, Xiaohan Wang, Dhruba Ghosh, Yuchang Su, Ludwig Schmidt, and Serena Yeung-Levy. Why are visually-grounded language models bad at image classification? *Advances in Neural Information Processing Systems*, 37:51727–51753, 2024.

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

[69] 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.

A Preliminaries

A.1 Visual reprogramming

Visual reprogramming [6] (VR) aims to adapt a pre-trained model to a target domain with limited labeled data by introducing an additional transformation function $\phi(\cdot)$. Instead of retraining the entire backbone, VR applies $\phi(\cdot)$ to the input images, thereby reusing the knowledge embedded in the pre-trained model while reducing the need for large-scale supervision. There are two mainstream methods to process images: the first is padding learnable noise around images; the other method is covering the image rim with learnable noise. Formally, given an input $x \in \mathcal{X} \subseteq \mathcal{R}^{\mathrm{S}}$ from the target domain, the transformed sample $\phi(x) \in \mathcal{R}^{\mathrm{T}}$ is fed into the pre-trained model, which outputs predictions aligned with the new label space. This paradigm has been proven effective in transferring visual-language models to downstream tasks with limited annotated data.

A.2 Attribute-enhanced CLIP classification

CLIP [45] is a pre-trained vision-language model containing an image encoder and a text encoder, represented as $f_{\mathrm{img}} : \mathcal{X}^{\mathrm{P}} \to \mathcal{Z}$ and $f_{\mathrm{txt}} : \mathcal{V}^{\mathrm{Q}} \to \mathcal{Z}$, where $\mathcal{X}^{\mathrm{P}} \subseteq \mathbb{R}^{d_{\mathrm{P}}}$ is a d_{P}-dimensional image space, $\mathcal{V}^{\mathrm{Q}} \subseteq \mathbb{R}^{d_{\mathrm{Q}}}$ is a d_{Q}-dimensional text space, and $\mathcal{Z} \subseteq \mathbb{R}^{d}$ is the shared embedding space. After mapping an image $x \in \mathcal{X}^{\mathrm{P}}$ and a text description $v \in \mathcal{V}^{\mathrm{Q}}$ into the shared embedding space via these encoders, CLIP computes the image-text similarity score:

$$\mathrm{sim}(x, v) = \cos\left(f_{\mathrm{img}}(x), f_{\mathrm{txt}}(v)\right)/\tau, \tag{23}$$

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

For classification on a downstream task with candidate label prompts $\mathcal{V}^{\mathrm{T}}$, CLIP converts these similarity scores into a normalized predictive distribution:

$$\mathrm{P}_{\mathrm{CLIP}}(v \mid x) = \frac{\exp(\mathrm{sim}(x, v))}{\sum_{v' \in \mathcal{V}^{\mathrm{T}}} \exp(\mathrm{sim}(x, v'))}. \tag{24}$$

Conventional CLIP typically relies on simple class prompts, such as "This is a photo of <class>", to generate textual embeddings for classification. While effective in general scenarios, such basic descriptions often fail to capture fine-grained or discriminative attributes that are crucial for downstream tasks. To address this limitation, AttrVR [7] introduces additional attribute information, including **descriptive attributes** $\mathcal{A}_{\mathrm{des}}$ that characterize general class features and **distinctive attributes** $\mathcal{A}_{\mathrm{dis}}$ that emphasize class-specific differentiating cues. These attributes are generated from large language models (LLMs) to supply extra information for downstream tasks, thereby enriching the textual space with more informative representations.

For each class c, let $\mathcal{A}_{\mathrm{des}}(c) = \{a_{\mathrm{des},c,r}\}_{r=1}^{M_{\mathrm{des}}(c)}$ and $\mathcal{A}_{\mathrm{dis}}(c) = \{a_{\mathrm{dis},c,r}\}_{r=1}^{M_{\mathrm{dis}}(c)}$ denote the descriptive and distinctive attribute prompt sets, respectively. For each textual view $v \in \{\mathrm{des}, \mathrm{dis}\}$, the frozen CLIP text encoder maps each attribute phrase to a normalized phrase-level embedding:

$$\mathbf{t}_{v,c,r} = \mathrm{norm}(f_{\mathrm{txt}}(a_{v,c,r})), \qquad r = 1, \ldots, M_v(c). \tag{25}$$

Given an image x, let $\mathbf{x} = \mathrm{norm}(f_{\mathrm{img}}(x))$ denote its normalized image embedding. For each class c and textual view v, we compute the view-specific class score by selecting the most matched attribute phrase within that class:

$$s_v^{(c)}(\mathbf{x}) = \max_{1 \le r \le M_v(c)} \mathrm{sim}(\mathbf{x}, \mathbf{t}_{v,c,r}), \qquad v \in \{\mathrm{des}, \mathrm{dis}\}. \tag{26}$$

The max operation is taken over the attribute phrases of the same class and the same textual view. Thus, $s_{\mathrm{des}}^{(c)}(\mathbf{x})$ measures the strongest descriptive-attribute evidence for class c, while $s_{\mathrm{dis}}^{(c)}(\mathbf{x})$ measures the strongest distinctive-attribute evidence for class c. The view-specific predictive distribution is obtained by applying softmax over the class scores of each textual view:

$$P_v(y = c \mid x) = \frac{\exp\left(s_v^{(c)}(\mathbf{x})\right)}{\sum_{j=1}^{C} \exp\left(s_v^{(j)}(\mathbf{x})\right)}, \qquad v \in \{\mathrm{des}, \mathrm{dis}\}. \tag{27}$$

These view-specific posteriors are later used to measure descriptive–distinctive agreement in the dual-attribute consistency regularizer. For classification with both attribute views, we further fuse the descriptive and distinctive scores at the pre-softmax score level:

$$S^{(c)}_{\mathrm{CLIP}}(\mathbf{x}) = \alpha \cdot s^{(c)}_{\mathrm{des}}(\mathbf{x}) + (1-\alpha) \cdot s^{(c)}_{\mathrm{dis}}(\mathbf{x}), \tag{28}$$

where $\alpha \in [0,1]$ is the fusion coefficient. The fused predictive distribution is then obtained by applying softmax over all class-wise fused scores:

$$P'_{\mathrm{CLIP}}(y=c \mid x) = \frac{\exp\left(S^{(c)}_{\mathrm{CLIP}}(\mathbf{x})\right)}{\sum_{j=1}^{C} \exp\left(S^{(j)}_{\mathrm{CLIP}}(\mathbf{x})\right)}. \tag{29}$$

Therefore, the attribute-enhanced prediction first computes phrase-level similarities, then performs max-pooling within each class and textual view, and finally produces either view-specific posteriors P_{des}, P_{dis} or the fused posterior P'_{CLIP} depending on the training objective.

A.3 Semi-supervised learning

Although attribute augmentation enriches the textual information for downstream tasks, the vast pool of unlabeled data remains largely underutilized. To leverage the unlabeled data, we introduce the semi-supervised learning (SSL) technique for the visual reprogramming task. During semi-supervised learning, we aim to learn the features from a small amount of labeled data $\mathcal{D}_l$ and a large amount of unlabeled data $\mathcal{D}_u$. Particularly, the model F with network parameters θ is trained by minimizing a joint loss function $\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_u$, where $\mathcal{L}_s$ and $\mathcal{L}_u$ represent the supervised loss on labeled data and the unsupervised loss on unlabeled data, and λ is the weight of the unsupervised loss. In the training process, both labeled and unlabeled data are used to optimize the model. The typical supervised loss $\mathcal{L}_s$ is the cross-entropy loss between labeled data and the corresponding labels:

$$\mathcal{L}_s = \frac{1}{|\mathcal{D}_l|} \sum_{i=1}^{|\mathcal{D}_l|} \mathrm{H}(y_i, p_\theta(y_i|\psi(x_i))), \quad x_i \in \mathcal{D}_l, \tag{30}$$

where $\psi(\cdot)$ denotes the stochastic data weak augmentation function.

For unlabeled data $\mathcal{D}_u$, a classical SSL algorithm, FixMatch [47], generates pseudo labels by applying a confidence threshold on weak augmentation. Meanwhile, it measures the quality of pseudo labels by calculating the cross-entropy between the classification probabilities of strong augmentation and the pseudo labels. Its assumption is the classification consistency of weak and strong augmentations of an image. The unsupervised loss is defined as follows:

$$\mathcal{L}_u = \frac{1}{|\mathcal{D}_u|} \sum_{i=1}^{|\mathcal{D}_u|} \mathbb{I}(\max(p_i) > \tau) \mathrm{H}(\tilde{y}_i, p_\theta(\tilde{y}_i|\Psi(\tilde{x}_i))), \quad \tilde{x}_i \in \mathcal{D}_u, \tag{31}$$

where $p_i = p_\theta(\psi(x_i))$ and $p_\theta(\tilde{y}, \Psi(x_i))$ are the predicted probabilities from the weak $\psi(\cdot)$ and strong augmentation $\Psi(\cdot)$ for unlabeled data $\tilde{x}_i$; $\tilde{y}_i = \mathrm{argmax}(p_i)$ is a generated pseudo label; τ is the confidence threshold, and $\mathbb{I}$ is the indicator function used to filter samples with low prediction probabilities. Although the confidence threshold can be used to calculate the pseudo labels, a considerable portion of low-confidence samples may still yield correct pseudo labels. This defect undermines the learning efficiency of unlabeled data. Therefore, CURE introduces confidence-margin soft weighting and dual-attribute consistency to improve the reliability of pseudo supervision.

B Implementation and Experimental Details

Backbone VLM and parameter-efficient policy. We adopt CLIP as the pre-trained vision-language model and use the ViT-B/16 visual backbone unless otherwise stated. Following common parameter-efficient adaptation protocols, we keep the CLIP backbone frozen throughout training and update only lightweight task-specific parameters introduced by each adaptation method. For VR-based methods, only the visual reprogramming parameters are optimized. For prompt-learning baselines, only the prompt or context parameters are optimized. No method fine-tunes the CLIP image encoder or text encoder. This design isolates the effect of label-efficient adaptation while keeping the foundation model unchanged.

Table 5: Training protocol and hyperparameter settings for baselines and CURE. All methods use the same 2-shot labeled split, test split, and frozen CLIP ViT-B/16 backbone unless otherwise stated. Method-specific SSL hyperparameters follow the original papers or official implementations and are not tuned per dataset.

Method	Trainable parameters	Unlabeled data	γ	Augmentation	Epochs	Optimizer / schedule
ZS	None	No	–	None	–	–
VP [1]	Input overlay pattern	No	–	Weak	200	SGD, momentum 0.9, cosine LR
AR [9]	Input border pattern	No	–	Weak	200	SGD, momentum 0.9, cosine LR
AttrVR [7]	Input VR pattern	No	–	Weak	200	SGD, momentum 0.9, cosine LR
BayesVR [5]	Input VR pattern	No	–	Weak	200	SGD, momentum 0.9, cosine LR
DAVR [8]	Input VR pattern	No	–	Weak	200	SGD, momentum 0.9, cosine LR
AttrVR+FixMatch [47]	Input VR pattern	Yes	7	Weak/Strong	200	SGD, momentum 0.9, cosine LR
AttrVR+SoftMatch [10]	Input VR pattern	Yes	7	Weak/Strong	200	SGD, momentum 0.9, cosine LR
AttrVR+FlexMatch [64]	Input VR pattern	Yes	7	Weak/Strong	200	SGD, momentum 0.9, cosine LR
AttrVR+FreeMatch [55]	Input VR pattern	Yes	7	Weak/Strong	200	SGD, momentum 0.9, cosine LR
AttrVR+CGMatch [14]	Input VR pattern	Yes	7	Weak/Strong	200	SGD, momentum 0.9, cosine LR
CURE	Input VR pattern	Yes	7	Weak/Strong	200	SGD, momentum 0.9, cosine LR

Visual reprogramming setup. For VR-based methods, we attach a learnable visual prompt to the input image. Concretely, the prompt is implemented as a learnable watermark-like pattern or frame-style patch with a fixed spatial budget, which is overlaid onto the input image before being fed into CLIP's visual encoder. The prompt parameters are initialized to zeros and optimized end-to-end under the frozen-backbone policy. The prompt resolution matches the input resolution, e.g., 224×224, and the prompt budget, e.g., frame width or patch size, is fixed across datasets for consistent comparison.

Common training protocol for baselines and CURE. To ensure a fair comparison, all methods in Tables 1 and 2 are evaluated under the same 2-shot labeled split, the same test split, and the same frozen-backbone setting. All trainable methods are trained for 200 epochs using SGD with momentum 0.9 and a cosine annealing learning-rate schedule. The initial learning rate is set to 40, and the batch size is fixed to 128 across all datasets. For purely supervised VR baselines, including VP, AR, AttrVR, BayesVR, and DAVR, only the labeled 2-shot data are used. For SSL-based baselines, including AttrVR+FixMatch, AttrVR+SoftMatch, AttrVR+FlexMatch, AttrVR+FreeMatch, AttrVR+CGMatch, and CURE, we use the same unlabeled pool, the same unlabeled-to-labeled ratio $\gamma = 7$, and the same weak/strong augmentation protocol. The method-specific hyperparameters of SSL baselines follow their original papers or official implementations and are not re-tuned for individual datasets. CURE also uses a fixed default configuration across datasets and splits, with $\lambda = 1.0$, EMA momentum $\alpha = 0.99$, and $\gamma = 7$. Thus, the comparisons mainly reflect differences in how unlabeled data are utilized, rather than differences in data splits, training budgets, augmentation policies, or optimizer settings. The detailed training protocol and hyperparameter settings for each method are summarized in Table 5.

Weak and strong augmentation setup. We use weak and strong augmentations to construct two views of each unlabeled image for consistency-based semi-supervised learning. The distinction between weak and strong augmentation is determined not only by the transformation type but also by the associated hyperparameter settings. In our implementation, the weak augmentation pipeline consists of `Resize(image-size)`, `RandomResizedCrop(image-size)`, `RandomHorizontalFlip(p=0.5)`, RGB conversion, and `ToTensor`, using the default hyperparameters unless otherwise specified. The strong augmentation pipeline follows the same transformations as the weak augmentation and further applies `RandomErasing` with `p=1.0`, `scale=(0.02, 0.4)`, and `ratio=(1, 1)`. Thus, the stronger perturbation mainly comes from the additional erasing operation and its parameterization, rather than from the transformation type alone. Both weakly and strongly augmented images are processed by the same VR module and the frozen image encoder.

Unlabeled data sampling setup. In the semi-supervised setting, the unlabeled pool is sampled from the same target dataset and the same label space as the labeled subset, i.e., no extra unknown classes are introduced. The unlabeled set contains the remaining training images with the labels removed. We control the amount of unlabeled data using an unlabeled-to-labeled ratio $\gamma = 7$, meaning that for each labeled batch, we sample seven unlabeled batches from the unlabeled pool. This unlabeled

sampling protocol is shared by all SSL-based methods, including the SSL extensions of AttrVR, AR+FlexMatch, and CURE.

Semi-supervised learning protocol. We adopt a FixMatch-style SSL protocol. For each unlabeled image, we generate a weakly augmented view and a strongly augmented view. The weakly augmented view is used to produce pseudo-labels and reliability weights, while the strongly augmented view is used for the student prediction in the unsupervised objective. Labeled images are trained with the same weak augmentation pipeline. All SSL-based methods are evaluated with the same labeled/unlabeled split, unlabeled ratio, augmentation protocol, and epoch budget.

Method-specific hyperparameters. The common training protocol is shared across methods, while method-specific hyperparameters are inherited from the corresponding original papers or official implementations. For FixMatch, SoftMatch, FlexMatch, FreeMatch, and CGMatch, we use their default thresholding, weighting, curriculum, or sample-selection settings. For AR+FlexMatch, the FlexMatch-specific settings are kept the same as those used in AttrVR+FlexMatch. For CURE, we use the fixed default setting $\lambda = 1.0$, EMA momentum $\alpha = 0.99$, and unlabeled-to-labeled ratio $\gamma = 7$ across all datasets. We do not perform dataset-specific or split-specific hyperparameter tuning for CURE or the SSL baselines.

Optimization and training details. We train all trainable methods for a fixed 200 epochs with a consistent optimizer and schedule. For optimization, we employ SGD with a momentum of 0.9 and a cosine annealing learning rate schedule, starting from an initial learning rate of 40. The batch size is fixed at 128 across all datasets, comprising both labeled and unlabeled samples for SSL-based methods. All experiments are conducted on the same workstation equipped with 8 RTX-4090 GPUs.

Evaluation protocol and error bars. We evaluate classification accuracy on the same test split of each dataset. For Oxford Flowers, we adopt the data split provided by the CoOp implementation [69]. All results in Tables 1 and 2 are reported as mean $\pm$ one standard deviation over three independent random seeds, 0, 42, and 2025, rather than the standard error.

C Additional experiments

C.1 Experiments on positive-negative attribute margin under different shot settings

Vision-language models perform classification by comparing relative image-text similarities rather than relying on the absolute similarity to the correct class alone. Therefore, to complement the qualitative visualization in Figure 1, we measure the positive-negative attribute margin:

$$\Delta_{\text{attr}} = a_c - a_i, \tag{32}$$

where a_c denotes the cosine similarity between the image feature and the attribute feature of the correct class, and a_i denotes the similarity to incorrect-class attributes. A larger margin indicates stronger discriminative image-attribute performance.

Table 6 reports the descriptive and distinctive positive-negative attribute margins under different labeled-shot settings on Oxford Flowers. For supervised VR variants such as VP and AR, the margins generally decrease as fewer labeled samples are available, indicating that limited supervision weakens the relative separation between correct and incorrect attributes. AttrVR exhibits a different but also undesirable pattern: its margins remain consistently low across shot settings, suggesting that attribute descriptions alone do not necessarily yield strong discriminative alignment under limited supervision. In contrast, CURE maintains relatively large margins, especially in low-shot settings, indicating more robust discriminative image-attribute alignment. These results support our interpretation that CURE better preserves the score-level separation needed for reliable pseudo-label learning.

C.2 Experiments on backbone generalization

To further examine the robustness and generality of CURE, we first evaluate different VR methods with various CLIP visual encoders, including both ViT-based and ResNet-based architectures. As shown in Table 7, CURE consistently achieves the best performance across all four CLIP backbones

Table 6: Positive-negative image-attribute margin under different labeled-shot settings on Oxford Flowers. The margin is defined in Equation (32).

Method	Metric ↑	16-shot	8-shot	4-shot	2-shot
VP	$\Delta_{\text{attr}}^{\text{des}}$	0.1326	0.1265	0.1215	0.1154
	$\Delta_{\text{attr}}^{\text{dis}}$	0.1316	0.1249	0.1199	0.1142
AR	$\Delta_{\text{attr}}^{\text{des}}$	0.1498	0.1435	0.1348	0.1004
	$\Delta_{\text{attr}}^{\text{dis}}$	0.1482	0.1419	0.1338	0.0993
AttrVR	$\Delta_{\text{attr}}^{\text{des}}$	0.0244	0.0254	0.0294	0.0863
	$\Delta_{\text{attr}}^{\text{dis}}$	0.0243	0.0250	0.0291	0.0852
CURE	$\Delta_{\text{attr}}^{\text{des}}$	0.1413	0.1386	0.1385	0.1302
	$\Delta_{\text{attr}}^{\text{dis}}$	0.1392	0.1369	0.1368	0.1288

Table 7: Classification accuracy (%) of different VR methods on Oxford Flowers under the 2-shot setting, using different CLIP visual encoders. RN denotes ResNet.

Method	ViT-B/32	ViT-L/14	RN50	RN101
VP	60.70	68.20	49.90	55.30
AR	60.00	73.80	57.60	60.30
AttrVR	53.80	77.10	61.90	58.20
CURE	**81.53**	**96.22**	**80.47**	**81.40**

on Oxford Flowers. The improvements are substantial for both transformer-based encoders and convolutional encoders, indicating that the effectiveness of CURE is not tied to a specific CLIP visual architecture or inductive bias. These results suggest that the proposed semi-supervised VR design can robustly benefit different backbone capacities and architectures.

Table 8: Classification accuracy (%) in the 2-shot setting using SigLIP and SigLIP2.

Backbone	Method	Pet	Caltech	SUN	RESISC	DTD	Flowers	Avg.
SigLIP	ZS	89.32	95.62	66.24	50.26	59.56	80.55	73.59
	VP	91.60	96.40	67.60	59.60	60.70	83.50	76.57
	AR	91.30	96.60	66.60	56.70	60.50	83.20	75.82
	AttrVR	84.70	88.70	56.70	62.60	57.20	80.60	71.75
	CURE	**91.71**	**97.32**	**70.60**	**68.83**	**67.61**	**96.87**	**82.16**
SigLIP2	ZS	91.96	97.12	67.67	59.14	56.50	83.03	75.90
	VP	93.00	96.90	68.80	65.70	58.10	82.90	77.57
	AR	92.50	96.80	67.80	63.50	58.00	82.40	76.83
	AttrVR	62.40	96.50	64.20	57.60	57.40	75.00	68.85
	CURE	**93.86**	**98.05**	**72.74**	**66.21**	**67.91**	**96.35**	**82.52**

Beyond the CLIP family, we further evaluate whether CURE generalizes to more recent VLM backbones, including SigLIP [63] and SigLIP2 [52], under the same 2-shot setting. All methods are evaluated with the same hyperparameter configuration as in the main experiments. As shown in Table 8, CURE consistently achieves the best average performance on both backbones. On SigLIP, CURE reaches 82.16% average accuracy, outperforming the strongest baseline by 5.59 percentage points. On SigLIP2, CURE achieves 82.52% average accuracy, exceeding the strongest baseline by 4.95 percentage points. These results further demonstrate that CURE is not limited to CLIP, and that its semi-supervised reprogramming strategy remains effective across different VLM families.

C.3 Ablation on LLM-Generated Attribute Types

To isolate the contribution of the LLM-generated attributes, we conducted a controlled prompt ablation under the same 2-shot semi-supervised setting. Specifically, we compare CURE using

only class-name prompts, **only descriptive attributes**, **only discriminative attributes**, and **both attribute types**. All other model components and training settings are kept unchanged. Since dual-attribute consistency requires predictions from both attribute types, this term is unavailable in the class-name-only and single-attribute variants.

Table 9: Ablation of LLM-generated attribute types under the 2-shot semi-supervised setting.

Method	Aircraft	Cars	DTD	EuroSAT	Flowers	UCF	Average
CURE-Classname	28.86	68.28	58.87	84.40	90.70	74.98	67.68
CURE-Descriptive	29.67	68.33	56.86	79.21	91.91	75.26	66.87
CURE-Discriminative	31.35	68.43	59.06	82.58	91.90	74.94	68.04
CURE-VR	**31.50**	**69.73**	**59.08**	**85.33**	**91.96**	**76.33**	**68.99**

As shown in Table 9, the results indicate that neither descriptive nor discriminative attributes alone consistently outperform the class-name variant across all datasets. In contrast, combining both attribute types achieves the best performance on all six datasets and improves the average accuracy from 67.68% to 68.99%. This result shows that CURE's gains cannot be attributed simply to adding LLM-generated attribute text; rather, the two attribute types provide complementary semantic information. Because the single-attribute variants also exclude dual-attribute consistency, this comparison reflects the joint benefit of complementary attributes and their consistency modeling. It complements our component ablation in Table 3, where removing the consistency term while retaining both attribute types also degrades performance. Together, these two analyses provide complementary evidence for the value of combining descriptive and discriminative attributes, and for the additional contribution of dual-attribute consistency.

C.4 Quantitative analysis of class structure in the embedding space

T-SNE visualizations provide an intuitive view of the learned representations, but they are two-dimensional projections and should not be treated as standalone evidence for the class structure in the original high-dimensional embedding space. Therefore, to complement Figure 7, we further compute quantitative class-structure statistics on Oxford Flowers using the original embedding features.

Specifically, we report the Fisher ratio and the centroid-based silhouette score to quantify the class structure in the original embedding space. Let $\mathbf{z}_i \in \mathbb{R}^d$ denote the embedding of sample i, $y_i \in \{1, \ldots, C\}$ denote its class label, N_c denote the number of samples in class c, $\boldsymbol{\mu}_c$ denote the centroid of class c, and $\boldsymbol{\mu}$ denote the global centroid. The Fisher ratio, inspired by Fisher's discriminant criterion [23], measures inter-class separation relative to intra-class compactness:

$$\mathrm{FR} = \frac{\sum_{c=1}^{C} N_c \left\| \boldsymbol{\mu}_c - \boldsymbol{\mu} \right\|_2^2}{\sum_{c=1}^{C} \sum_{i:y_i=c} \left\| \mathbf{z}_i - \boldsymbol{\mu}_c \right\|_2^2 + \epsilon}, \tag{33}$$

where ϵ is a small constant for numerical stability. A larger Fisher ratio indicates that different classes are farther apart while samples within each class are more compact.

We also compute a centroid-based silhouette score, following the intuition of the silhouette coefficient [46]. For each sample i, we define its distance to the assigned class centroid and the nearest competing class centroid as:

$$a_i = \left\| \mathbf{z}_i - \boldsymbol{\mu}_{y_i} \right\|_2, \qquad b_i = \min_{c \neq y_i} \left\| \mathbf{z}_i - \boldsymbol{\mu}_c \right\|_2. \tag{34}$$

The centroid-based silhouette score is then computed as:

$$\mathrm{Sil}_{\mathrm{centroid}} = \frac{1}{N} \sum_{i=1}^{N} \frac{b_i - a_i}{\max(a_i, b_i) + \epsilon}. \tag{35}$$

This score lies approximately in $[-1, 1]$, where a larger value indicates that samples are closer to their own class centroid than to other class centroids, reflecting clearer class grouping.

As shown in Table 10, both metrics improve from VP to AR and AttrVR, and achieve the highest values with CURE. The higher Fisher ratio indicates that CURE obtains stronger inter-class separation

relative to intra-class compactness. The higher centroid-based silhouette score further suggests that samples are more clearly grouped around their own class centers. These results confirm that CURE learns a more discriminative class structure in the original high-dimensional embedding space, which is consistent with the trend observed in the t-SNE visualization.

Table 10: Class-structure statistics on Oxford Flowers in the original embedding space.

Metric	**VP**	**AR**	**AttrVR**	**CURE**
FR↑	31.62	33.42	33.65	**35.99**
$\mathrm{Sil}_{\mathrm{centroid}}$ ↑	0.4533	0.4552	0.4569	**0.4683**

C.5 Hyperparameter sensitivity and split robustness

We first evaluate the sensitivity of CURE to its main hyperparameters, including the unsupervised loss weight λ and the prior momentum α. The default setting is $\lambda = 1.0$ and $\alpha = 0.99$, and we vary one hyperparameter at a time while keeping the others fixed. As shown in Table 11, CURE maintains stable performance across different hyperparameter choices on DTD, Pets, and Caltech. The standard deviations are small, with 0.07, 0.22, and 0.21 for the three datasets, respectively. These results indicate that CURE does not require fine-grained hyperparameter tuning for downstream tasks.

Table 11: Hyperparameter sensitivity of CURE with respect to the unsupervised loss weight λ and prior momentum α.

Dataset	λ=0.8	λ=1.0	λ=1.2	α=0.95	α=0.97	α=0.99	**Mean±Std**
DTD	59.06	59.08	59.17	59.02	59.23	59.08	59.11±0.07
Pets	92.18	92.66	92.53	92.42	92.87	92.66	92.55±0.22
Caltech	95.70	95.82	95.82	95.26	95.47	95.82	95.65±0.21

Table 12: Split robustness of CURE under the fixed default hyperparameter configuration.

Dataset	**Split1**	**Split2**	**Split3**	**Split4**	**Split5**	**Mean±Std**
Caltech	95.98	95.46	95.50	95.42	95.74	95.62±0.21
Aircraft	32.04	31.71	32.34	31.06	31.96	31.82±0.43
Flowers	92.69	91.96	93.03	92.29	93.95	92.78±0.69

We further examine whether CURE is robust to different labeled-data partitions under the default hyperparameter configuration. Following common practice in semi-supervised learning, we use $\lambda = 1.0$ for the unsupervised loss. These values are fixed across datasets and splits without split-specific tuning. As shown in Table 12, CURE achieves stable performance across five labeled-data splits on Caltech, FGVC Aircraft, and Flowers. The small standard deviations suggest that the selected default hyperparameters are not tailored to a particular split, and that CURE remains robust under different limited-supervision partitions.

C.6 Experiments on applying CURE to visual prompt tuning

To examine whether CURE's benefits extend beyond visual reprogramming, we implemented CURE-VPT by replacing visual input reprogramming with trainable visual prompt tokens while retaining the proposed Gaussian confidence-margin weighting and dual-attribute consistency. We also adapted CPL [65] to the limited SSL settings for a complementary cross-paradigm comparison. All variants use ViT-B/16, the same 2-shot labeled split, an unlabeled-to-labeled ratio of 7, and 200 epochs.

CPL-VPT outperforms CPL-SSL on all 6 datasets, with the average accuracy dropping from 63.38% (CPL-VPT) to 59.59% (CPL-SSL) when switching from VPT to limited SSL, as shown in Table 13. This result is consistent with the fact that VR constitutes a more constrained adaptation setting, since it modifies the input while keeping the pre-trained model architecture backbone unchanged. In contrast, VPT introduces trainable tokens into the encoder and alters its internal feature interactions, providing greater adaptability at the visual representation level. Although CURE was originally

Table 13: The results of applying CURE to visual prompt tuning.

Method	DTD	EuroSAT	Aircraft	Flowers	Cars	UCF	Avg.
CPL-VPT	54.73	82.48	25.68	77.34	66.78	73.27	63.38
CPL-SSL	46.10	75.11	25.47	72.88	66.42	71.53	59.59
CURE-SSL	**59.08**	**85.33**	31.50	91.96	69.73	76.33	**68.99**
CURE-VPT	59.04	73.23	**32.91**	**93.71**	**70.53**	**77.82**	67.87

designed and tuned for this stricter VR setting, directly transferring it to VPT still yields an average accuracy of 67.87% and outperforms CURE-VR on Aircraft, Flowers, Cars, and UCF. CURE-VPT also surpasses CPL-VPT on five of the six datasets. These results indicate that the benefits of CURE are not restricted to visual reprogramming and can extend to visual prompt tuning. However, the exception on SAT suggests that achieving consistent gains under VPT may require more careful adaptation-specific design and optimization rather than a direct transfer of the VR configuration.

D LLM to generate attributes

We used GPT-3.5 [4] to generate class attributes (both descriptive and distinctive). Following [7], the generation hyperparameters are as follows: the temperature is 0.99, the maximum token size is 50, and generating 25 ($M_{\text{des}}(c) = M_{\text{dis}}(c) = 25$) candidates per class. Decoding stops at the first period '.'; candidates shorter than 20 characters are discarded, and the remaining ones are retained.

Table 14: Dataset Information

	Pets	Caltech	Aircraft	Cars	SUN	UCF	EuroSAT	RESISC	DTD	Food	IN	Flowers
Task Info.	pet	object	fine-grained aircraft model	fine-grained automobile	scene	action	remote sensing land cover	remote sensing scene	texture	food	object	flower
Class Num.	37	100	100	196	397	101	10	45	47	101	1000	102

To generate descriptive attributes, we queried each class in every task with the following prompt:

Prompt 1. Describe the appearance of the [`Task Information`] [`Class Name`].

To generate distinctive attributes, we use the following prompt:

Prompt 2. Describe the unique appearance of a/an [`Class Name`] from other [`Task Information`].

The [`Task Information`] represents the description of downstream tasks, as shown in Table 14, and the [`Class Name`] denotes the name of the label $y \in Y$.

E Preliminary Extension to Multimodal Large Language Models

Our main experiments focus on contrastive vision-language models such as CLIP and SigLIP. We further investigate whether the core idea of CURE can be transferred to a generative multimodal large language model (MLLM) while preserving the frozen-backbone property of visual reprogramming. As a preliminary study, we adapt CURE to LLaVA-1.5-7B by treating the **input-space visual reprogramming (VR) pattern as the only trainable component**, while keeping the vision encoder, vision-language projector, and language model entirely frozen.

Likelihood-based classification with MLLMs. Unlike contrastive VLMs, LLaVA does not directly provide a class-level similarity distribution. We formulate classification as likelihood-based scoring over the predefined candidate classes rather than free-form generation or answer matching. Given an image x, a classification instruction q, and the tokenized class name $\mathbf{w}_c = \{w_{c,1}, \ldots, w_{c,T_c}\}$ for candidate class c, we teacher-force the response and compute its conditional log-likelihood:

$$s_c(x) = \frac{1}{T_c} \sum_{t=1}^{T_c} \log p_\theta \left(w_{c,t} \mid x, q, w_{c,<t} \right), \tag{36}$$

where θ denotes the frozen parameters of the MLLM. The scores over all candidate classes are normalized to obtain the class prediction distribution:

$$p_{\text{cls}}(c \mid x) = \frac{\exp(s_c(x))}{\sum_{c'} \exp(s_{c'}(x))}. \tag{37}$$

Only the tokens corresponding to the class name are included in the likelihood computation; answer templates, punctuation, and the EOS token are excluded. Length normalization prevents the scoring procedure from systematically favoring classes with shorter names.

Attribute-based predictions. We retain the same class-specific **descriptive** and **discriminative** attributes used in the contrastive-VLM experiments. For the descriptive branch, the MLLM is prompted to describe the visible appearance of the main object, whereas the discriminative branch asks for the visible feature that best distinguishes the object. The corresponding predefined attribute sentences are treated as fixed candidate responses and evaluated using the same teacher-forced conditional-likelihood scoring rather than free-form generation. Candidate scores are aggregated at the class level and normalized to produce descriptive and discriminative class distributions.

These two attribute distributions provide the attribute-agreement signal used by CURE, while the primary class-name distribution provides the confidence and prediction margin for Gaussian confidence-margin soft weighting. Therefore, the overall learning principle of CURE remains unchanged; only the mechanism for obtaining class-level prediction distributions is adapted from contrastive similarity scores to conditional language-model likelihoods.

Experimental setting. We follow the same 2-shot semi-supervised protocol as in the main experiments, using two labeled and fourteen unlabeled images per class with the same labeled/unlabeled splits. Because inference and optimization with LLaVA-1.5-7B are substantially more computationally expensive than with contrastive VLMs, we reduce the training schedule from 200 to 50 epochs. Importantly, **only the input-space VR pattern is optimized throughout training**; the vision encoder, vision-language projector, and language model remain frozen.

Table 15: Preliminary extension of CURE to the frozen MLLM under the 2-shot setting.

Method	EuroSAT	DTD	Aircraft	Flowers	Pets	Caltech	Avg.
CLIP Zero-shot	16.80	**37.41**	**21.36**	**60.54**	**81.49**	**88.44**	**51.01**
LLaVA-1.5-7B Zero-shot	27.30	13.83	7.50	8.50	37.86	70.60	27.60
LLaVA-1.5-7B + CURE	**35.11**	35.93	12.84	47.13	42.87	71.48	40.89

As shown in Table 15, zero-shot LLaVA-1.5-7B exhibits considerably weaker classification performance than zero-shot CLIP on five of the six datasets. This observation is consistent with prior findings that generative visually grounded language models can underperform contrastive models such as CLIP on discriminative image classification, despite retaining useful visual information in their internal representations [67]. Nevertheless, applying CURE consistently improves LLaVA-1.5-7B across all six datasets, increasing its average accuracy from 27.60% to 40.89%, corresponding to an absolute gain of 13.29 percentage points. The improvement is particularly pronounced on DTD and Flowers, where CURE raises the accuracy from 13.83% to 35.93% and from 8.50% to 47.13%, respectively. These results suggest that the proposed adaptation principle is not inherently tied to contrastive similarity-based VLMs. By replacing contrastive class similarities with likelihood-based class distributions, CURE can optimize an input-space visual pattern for a generative MLLM without modifying its internal parameters. This provides a parameter-preserving adaptation route for MLLMs under limited supervision: the learned VR pattern is detachable, and removing it restores the original model without overwriting the vision encoder, projector, and language model parameters.

We emphasize, however, that this experiment is intended as a **proof-of-concept extension** rather than a comprehensive evaluation of CURE on MLLMs. The current study considers only LLaVA-1.5-7B and image classification, and substantial performance gaps to contrastive CLIP remain on several datasets. Therefore, these results provide initial evidence that CURE can be instantiated on a frozen generative MLLM, but do not establish its general effectiveness across different MLLM architectures or downstream tasks. A broader investigation of model architectures, prediction tasks, and alternative adaptation modules is left for future work.

F Training computational complexity and budget analysis

Consider one training iteration with B_l labeled images and B_u unlabeled images, and define the unlabeled-to-labeled ratio as $\gamma := B_u/B_l$. Let c_s denote the per-image cost of a strong view that requires forward and backward propagation through the frozen image encoder with respect to the input-space pattern, and let c_w denote the per-image cost of a weak view that is used only for pseudo-label generation with forward propagation. We denote $\rho := c_w/c_s \in (0, 1)$. Text prototypes are cached, and posterior recalibration, margin estimation, EMA updates, and JS-based agreement weighting involve only lightweight vector operations, whose costs are negligible compared with encoder passes.

For a supervised VR method that processes B_l labeled images per iteration, the encoder-pass-equivalent cost is computed as follows:

$$\mathcal{T}_{\mathrm{Sup}} \approx B_l c_s. \tag{38}$$

CURE processes one weak view for labeled images, one strong view for unlabeled images, and one additional weak view for unlabeled images. Therefore, we obtain the CURE cost as follows:

$$\mathcal{T}_{\mathrm{CURE}} \approx B_l c_s + B_u c_s + B_u c_w, \tag{39}$$

$$= \big(B_l + B_u(1+\rho)\big)c_s. \tag{40}$$

The theoretical encoder-pass-equivalent factor is thus calculated as follows:

$$\frac{\mathcal{T}_{\mathrm{CURE}}}{\mathcal{T}_{\mathrm{Sup}}} \approx 1 + (1+\rho)\gamma. \tag{41}$$

With the common SSL setting $\gamma = 7$ and an empirical estimate $\rho \approx 0.25$ for the weak forward-only view, the theoretical encoder-pass-equivalent factor estimate is as follows:

$$\frac{\mathcal{T}_{\mathrm{CURE}}}{\mathcal{T}_{\mathrm{Sup}}} \approx 1 + (1 + 0.25) \times 7 = 9.75. \tag{42}$$

This value should be interpreted as a conservative sequential encoder-pass-equivalent estimate, rather than the actual wall-clock training ratio. In practice, the wall-clock overhead is noticeably smaller because the weak and strong views are processed in batched GPU computation, whose runtime grows sublinearly with the number of views. To reflect this, we can write an effective wall-clock approximation as follows:

$$\frac{\mathcal{T}^{\mathrm{wall}}_{\mathrm{CURE}}}{\mathcal{T}^{\mathrm{wall}}_{\mathrm{Sup}}} \approx 1 + \eta(1+\rho)\gamma, \qquad 0 < \eta \leq 1, \tag{43}$$

where η captures the effective marginal cost reduction due to batch parallelism and hardware utilization. In our implementation, the observed wall-clock training time is about $5\times$ that of purely supervised VR, corresponding to $\eta \approx 0.46$ under the above setting. Thus, the theoretical factor in Equation (41) should be viewed as a conservative pass-equivalent proxy, while the practical wall-clock overhead can be substantially lower under batched GPU execution.

To further evaluate the performance-efficiency trade-off, we conduct a budget-controlled comparison. We compare CURE trained with the original schedule and with substantially reduced epochs against supervised VR baselines trained with the original schedule and with much longer schedules. This analysis examines whether the improvement of CURE comes merely from using more training computation or from the proposed semi-supervised design.

As shown in Table 16, CURE remains competitive even with substantially fewer epochs. With only 20 epochs, CURE achieves 70.79% average accuracy on these 11 datasets, slightly higher than AR trained for 200 epochs. With 40 epochs, CURE reaches 73.86%, exceeding AR-200 by 3.37 percentage points. The full schedule further improves the average accuracy to 75.92%. These results suggest that longer training is not the primary source of CURE's improvement; rather, CURE already matches or surpasses a fully trained supervised AR baseline with far fewer epochs.

Table 17 further shows that simply extending the training schedule of supervised VR baselines by $5\times$ or $10\times$ does not yield meaningful additional gains. Together, these results indicate that CURE provides a more favorable performance–efficiency trade-off than supervised VR baselines. Although

Table 16: CURE under reduced training epochs compared with the supervised AR baseline on the 11 datasets used for budget analysis.

Method	Aircraft	Caltech	Cars	DTD	EuroSAT	Flowers	Food	Pet	RESISC	SUN	UCF	Avg.
CURE-20	30.51	95.25	67.55	53.84	56.62	85.59	82.81	92.18	73.11	67.58	73.70	70.79
CURE-40	31.29	95.42	68.21	55.79	76.95	89.61	84.56	92.42	75.04	68.21	74.95	73.86
CURE-200	**31.50**	**95.82**	**69.73**	**59.08**	**85.33**	**91.96**	**85.00**	**92.66**	**79.06**	**68.60**	**76.33**	**75.92**
AR-200	25.34	94.32	65.29	48.82	84.93	73.68	83.86	90.49	72.30	64.96	71.43	70.49

Table 17: Supervised VR methods with extended training schedules.

Method	Caltech	EuroSAT	DTD
AttrVR-1000	95.53	83.10	52.70
AttrVR-2000	95.53	83.10	52.70
VP-1000	94.30	59.70	47.16
VP-2000	94.30	59.70	47.16
AR-1000	95.10	85.50	48.60
AR-2000	95.10	85.50	48.60

exploiting unlabeled data inevitably increases training computation, the resulting improvements are not merely a consequence of longer optimization. Instead, they come from more effective unlabeled supervision. Importantly, this additional cost is incurred only during training. At inference time, CURE follows standard visual reprogramming and requires only one forward pass through the frozen VLM, so deployment latency remains unchanged.

G Justification for dual-attribute consistency

We justify the dual-attribute consistency regularizer under a multimodal class-conditional feature model with unequal projected variances. We model the visual features of each class as a finite Gaussian mixture and study the one-dimensional score distributions induced by the descriptive and distinctive attribute directions. Because the *ordinary* Jensen–Shannon divergence between Gaussian distributions does not admit a closed form in general, we avoid exact closed-form calculations and instead derive an explicit upper bound. The resulting bound shows that the discrepancy between the two induced score distributions is controlled by the mismatch between their attribute directions.

Setup. Let X be random images with the corresponding label Y, and let $Z = f(X) \in \mathbb{R}^d$ denote the image feature extracted by the frozen visual encoder. For a fixed class y, we model the conditional distribution of Z given $Y = y$ as a finite Gaussian mixture:

$$Z \mid Y = y \sim \sum_{k=1}^{K_y} \pi_{y,k}\, \mathcal{N}(\mu_{y,k}, \Sigma_{y,k}), \qquad \pi_{y,k} \geq 0, \quad \sum_{k=1}^{K_y} \pi_{y,k} = 1, \tag{44}$$

where $\Sigma_{y,k} \succeq 0$. Let $t_y^{\mathrm{des}}, t_y^{\mathrm{dis}} \in \mathbb{S}^{d-1}$ denote the normalized descriptive and distinctive text directions, respectively. We then define the projected scores:

$$S_y^{\mathrm{des}} := \langle t_y^{\mathrm{des}}, Z \rangle, \qquad S_y^{\mathrm{dis}} := \langle t_y^{\mathrm{dis}}, Z \rangle, \tag{45}$$

and let $\mathrm{P}_y^{\mathrm{des}}$ and $\mathrm{P}_y^{\mathrm{dis}}$ denote their probability laws. We also define the difference of $t_y^{\mathrm{des}}, t_y^{\mathrm{dis}}$:

$$\delta_y := t_y^{\mathrm{des}} - t_y^{\mathrm{dis}}, \qquad \theta_y := \arccos\big(\langle t_y^{\mathrm{des}}, t_y^{\mathrm{dis}} \rangle\big) \in [0, \pi]. \tag{46}$$

Since both directions are unit vectors, their Euclidean distance can be expressed in terms of the angle:

$$\|\delta_y\|_2^2 = \|t_y^{\mathrm{des}} - t_y^{\mathrm{dis}}\|_2^2 = 2 - 2\cos\theta_y = 4\sin^2(\theta_y/2). \tag{47}$$

Lemma 1 (Projected score laws are one-dimensional GMMs). We use $v \in \{\mathrm{des}, \mathrm{dis}\}$ to index the two attribute views, where t_y^{des} and t_y^{dis} denote the descriptive and distinctive text directions for class y, respectively. For each view v, define the scalar projected score:

$$S_y^v := \langle t_y^v, Z \rangle, \tag{48}$$

and let P_y^v denote the probability law of S_y^v under the conditional feature distribution $Z \mid Y = y$. Then P_y^v is a one-dimensional Gaussian mixture:

$$\mathrm{P}_y^v = \sum_{k=1}^{K_y} \pi_{y,k} \mathcal{N}\left(m_{y,k}^v, \nu_{y,k}^v\right), \tag{49}$$

where $m_{y,k}^v := \langle t_y^v, \mu_{y,k} \rangle$ and $\nu_{y,k}^v := (t_y^v)^\top \Sigma_{y,k} t_y^v \geq 0$ denote the mean and variance of the projected score for the k-th mixture component under attribute view v, respectively.

Lemma 2 (Component-wise mean and variance perturbation bounds). Lemma 2 quantifies how the mismatch between the descriptive and distinctive attribute directions affects the projected Gaussian components. For each mixture component k, the projected means and variance satisfy:

$$\left|m_{y,k}^{\mathrm{des}} - m_{y,k}^{\mathrm{dis}}\right| = \left|\langle \delta_y, \mu_{y,k} \rangle\right| \leq \|\mu_{y,k}\|_2 \, \|\delta_y\|_2, \tag{50}$$

$$\left|\nu_{y,k}^{\mathrm{des}} - \nu_{y,k}^{\mathrm{dis}}\right| \leq 2\|\Sigma_{y,k}\|_{\mathrm{op}} \, \|\delta_y\|_2, \tag{51}$$

where $\|\Sigma_{y,k}\|_{\mathrm{op}}$ denotes the spectral/operator norm of the covariance matrix.

Proof. For the mean term, by the definitions in Lemma 1, the difference between attributes is:

$$m_{y,k}^{\mathrm{des}} - m_{y,k}^{\mathrm{dis}} = \langle t_y^{\mathrm{des}}, \mu_{y,k} \rangle - \langle t_y^{\mathrm{dis}}, \mu_{y,k} \rangle \tag{52}$$

$$= \langle t_y^{\mathrm{des}} - t_y^{\mathrm{dis}}, \mu_{y,k} \rangle = \langle \delta_y, \mu_{y,k} \rangle. \tag{53}$$

Applying the Cauchy–Schwarz inequality gives $\left|m_{y,k}^{\mathrm{des}} - m_{y,k}^{\mathrm{dis}}\right| \leq \|\delta_y\|_2 \|\mu_{y,k}\|_2$.

For the variance term, let $a := t_y^{\mathrm{des}},\ b := t_y^{\mathrm{dis}},\ \Sigma := \Sigma_{y,k}$. Since $\nu_{y,k}^{\mathrm{des}} = a^\top \Sigma a$ and $\nu_{y,k}^{\mathrm{dis}} = b^\top \Sigma b$:

$$\nu_{y,k}^{\mathrm{des}} - \nu_{y,k}^{\mathrm{dis}} = a^\top \Sigma a - b^\top \Sigma b \tag{54}$$

$$= (a-b)^\top \Sigma a + b^\top \Sigma (a-b). \tag{55}$$

Taking the absolute value and using the operator norm inequality, we can obtain:

$$\left|\nu_{y,k}^{\mathrm{des}} - \nu_{y,k}^{\mathrm{dis}}\right| \leq \left|(a-b)^\top \Sigma a\right| + \left|b^\top \Sigma (a-b)\right| \tag{56}$$

$$\leq \|\Sigma\|_{\mathrm{op}} \|a-b\|_2 \|a\|_2 + \|\Sigma\|_{\mathrm{op}} \|b\|_2 \|a-b\|_2. \tag{57}$$

Because a and b are unit vectors, $\|a\|_2 = \|b\|_2 = 1$, and $a - b = \delta_y$. Consequently, we have:

$$\left|\nu_{y,k}^{\mathrm{des}} - \nu_{y,k}^{\mathrm{dis}}\right| \leq 2\|\Sigma_{y,k}\|_{\mathrm{op}} \|\delta_y\|_2. \tag{58}$$

The above derivation completes the proof of Lemma 2. □

Moreover, θ_y denotes the angle between t_y^{des} and t_y^{dis}, then $\|\delta_y\|_2 = \|t_y^{\mathrm{des}} - t_y^{\mathrm{dis}}\|_2 = 2\sin(\theta_y/2)$. Substituting this into Equation (50) and (51) yields:

$$\left|m_{y,k}^{\mathrm{des}} - m_{y,k}^{\mathrm{dis}}\right| \leq 2\|\mu_{y,k}\|_2 \sin(\theta_y/2), \tag{59}$$

$$\left|\nu_{y,k}^{\mathrm{des}} - \nu_{y,k}^{\mathrm{dis}}\right| \leq 4\|\Sigma_{y,k}\|_{\mathrm{op}} \sin(\theta_y/2). \tag{60}$$

Thus, as the two attribute directions become more aligned, both the component-wise projected mean gap and variance gap shrink.

Lemma 3 (A JS upper bound for two univariate Gaussians). Let $G_1 = \mathcal{N}(m_1, \nu_1)$ and $G_2 = \mathcal{N}(m_2, \nu_2)$ with $\nu_1, \nu_2 > 0$. If $\underline{\nu} \leq \min\{\nu_1, \nu_2\}$, then:

$$\mathrm{JS}(G_1 \| G_2) \leq \frac{(m_1 - m_2)^2}{4\underline{\nu}} + \frac{(\nu_1 - \nu_2)^2}{8\underline{\nu}^2}. \tag{61}$$

Proof. Let $M = (G_1 + G_2)/2$. By the convexity of KL divergence in its second argument:

$$\mathrm{KL}(G_1 \| M) \leq \frac{1}{2}\mathrm{KL}(G_1 \| G_1) + \frac{1}{2}\mathrm{KL}(G_1 \| G_2) = \frac{1}{2}\mathrm{KL}(G_1 \| G_2), \tag{62}$$

and similarly $\mathrm{KL}(G_2\|M) \le \frac{1}{2}\,\mathrm{KL}(G_2\|G_1)$. From the above, we derive:

$$\mathrm{JS}(G_1\|G_2) = \frac{1}{2}\,\mathrm{KL}(G_1\|M) + \frac{1}{2}\,\mathrm{KL}(G_2\|M) \le \frac{1}{4}\Big(\,\mathrm{KL}(G_1\|G_2) + \mathrm{KL}(G_2\|G_1)\Big). \tag{63}$$

For univariate Gaussians, the KL divergence admits the simplified form:

$$\mathrm{KL}(G_1\|G_2) = \frac{1}{2}\left[\log\frac{\nu_2}{\nu_1} + \frac{\nu_1 + (m_1 - m_2)^2}{\nu_2} - 1\right]. \tag{64}$$

Summing the two KL terms yields:

$$\mathrm{KL}(G_1\|G_2) + \mathrm{KL}(G_2\|G_1) = \frac{1}{2}\left[\frac{(\nu_1 - \nu_2)^2}{\nu_1\nu_2} + (m_1 - m_2)^2\left(\frac{1}{\nu_1} + \frac{1}{\nu_2}\right)\right]. \tag{65}$$

Combining this with Equation (63) and using $\nu_1, \nu_2 \ge \underline{\nu}$ gives Equation (61). □

Proposition 1 (Population JS bound under multimodal unequal-variance features). Assume that for every mixture component k, let $\underline{\nu}_{y,k} := \min\{\nu^{\mathrm{des}}_{y,k}, \nu^{\mathrm{dis}}_{y,k}\} > 0$. Hence, we can obtain:

$$\mathrm{JS}\big(\mathrm{P}^{\mathrm{des}}_y\|\mathrm{P}^{\mathrm{dis}}_y\big) \le \left[\sum_{k=1}^{K_y}\pi_{y,k}\left(\frac{\|\mu_{y,k}\|_2^2}{4\underline{\nu}_{y,k}} + \frac{\|\Sigma_{y,k}\|_{\mathrm{op}}^2}{2\underline{\nu}_{y,k}^2}\right)\right]\|\delta_y\|_2^2. \tag{66}$$

Equivalently, we define the aggregated coefficient as:

$$C_y := \sum_{k=1}^{K_y}\pi_{y,k}\left(\frac{\|\mu_{y,k}\|_2^2}{4\underline{\nu}_{y,k}} + \frac{\|\Sigma_{y,k}\|_{\mathrm{op}}^2}{2\underline{\nu}_{y,k}^2}\right). \tag{67}$$

Under this notation, the JS divergence admits the simplified form:

$$\mathrm{JS}\big(\mathrm{P}^{\mathrm{des}}_y\|\mathrm{P}^{\mathrm{dis}}_y\big) \le C_y\|\delta_y\|_2^2 = 4C_y\sin^2(\theta_y/2). \tag{68}$$

Proof. Write $G^{\mathrm{des}}_{y,k} := \mathcal{N}(m^{\mathrm{des}}_{y,k}, \nu^{\mathrm{des}}_{y,k})$, $G^{\mathrm{dis}}_{y,k} := \mathcal{N}(m^{\mathrm{dis}}_{y,k}, \nu^{\mathrm{dis}}_{y,k})$, so that, by Lemma 1, we can obtain $\mathrm{P}^{\mathrm{des}}_y = \sum_k \pi_{y,k}G^{\mathrm{des}}_{y,k}$, $\mathrm{P}^{\mathrm{dis}}_y = \sum_k \pi_{y,k}G^{\mathrm{dis}}_{y,k}$. Next, introduce the mixture components:

$$M_{y,k} := \frac{1}{2}\Big(G^{\mathrm{des}}_{y,k} + G^{\mathrm{dis}}_{y,k}\Big), \qquad M_y := \frac{1}{2}\Big(\mathrm{P}^{\mathrm{des}}_y + \mathrm{P}^{\mathrm{dis}}_y\Big) = \sum_k \pi_{y,k}M_{y,k}. \tag{69}$$

By the joint convexity of the KL divergence and the definitions above, we have:

$$\mathrm{KL}(\mathrm{P}^{\mathrm{des}}_y\|M_y) \le \sum_{k=1}^{K_y}\pi_{y,k}\,\mathrm{KL}(G^{\mathrm{des}}_{y,k}\|M_{y,k}), \quad \mathrm{KL}(\mathrm{P}^{\mathrm{dis}}_y\|M_y) \le \sum_{k=1}^{K_y}\pi_{y,k}\,\mathrm{KL}(G^{\mathrm{dis}}_{y,k}\|M_{y,k}). \tag{70}$$

Consequently, the JS divergence is bounded as follows:

$$\mathrm{JS}\big(\mathrm{P}^{\mathrm{des}}_y\|\mathrm{P}^{\mathrm{dis}}_y\big) \le \sum_{k=1}^{K_y}\pi_{y,k}\,\mathrm{JS}\big(G^{\mathrm{des}}_{y,k}\|G^{\mathrm{dis}}_{y,k}\big). \tag{71}$$

Applying Lemma 3 to each component pair with $\underline{\nu} = \underline{\nu}_{y,k}$ gives:

$$\mathrm{JS}\big(G^{\mathrm{des}}_{y,k}\|G^{\mathrm{dis}}_{y,k}\big) \le \frac{\big(m^{\mathrm{des}}_{y,k} - m^{\mathrm{dis}}_{y,k}\big)^2}{4\underline{\nu}_{y,k}} + \frac{\big(\nu^{\mathrm{des}}_{y,k} - \nu^{\mathrm{dis}}_{y,k}\big)^2}{8\underline{\nu}_{y,k}^2}. \tag{72}$$

Then Lemma 2 further implies:

$$\big(m^{\mathrm{des}}_{y,k} - m^{\mathrm{dis}}_{y,k}\big)^2 \le \|\mu_{y,k}\|_2^2\,\|\delta_y\|_2^2, \quad \big(\nu^{\mathrm{des}}_{y,k} - \nu^{\mathrm{dis}}_{y,k}\big)^2 \le 4\|\Sigma_{y,k}\|_{\mathrm{op}}^2\,\|\delta_y\|_2^2. \tag{73}$$

Substituting these two bounds into Equation (72) and factoring out the common term $\|\delta_y\|_2^2$ gives Equation (66). The factor $\|\delta_y\|_2^2$ appears because the component-wise mean and variance gaps are linear in the direction mismatch, whereas the Gaussian divergence bound depends quadratically on these gaps. Finally, since t^{des}_y and t^{dis}_y are unit vectors, Equation (47) gives $\|\delta_y\|_2^2 = 4\sin^2(\theta_y/2)$, which directly yields Equation (68). This shows that the JS discrepancy is controlled by the angular mismatch between the two attribute directions and decreases as they become more aligned. □

Interpretation. Proposition 1 shows that the JS divergence between the descriptive and distinctive score laws is controlled by the mismatch between the two attribute directions. From the prediction-score perspective, an easy sample is expected to receive a high score for the correct class and low scores for competing classes under both attribute views, yielding a large positive-negative margin and consistent posteriors. In contrast, a hard sample often has a smaller score margin, because descriptive cues may capture common visual patterns while distinctive cues emphasize subtler discriminative evidence. Such samples may therefore induce more view-dependent posterior responses; the resulting dual-attribute discrepancy provides a complementary cue for distinguishing informative hard samples from semantically unstable pseudo-labels.

Empirical support. This interpretation is supported by the positive-negative attribute margin analysis in Appendix C.1. As shown in Table 6, supervised VR variants such as VP and AR exhibit decreasing margins as the number of labeled samples becomes smaller, indicating weakened separation between correct and incorrect attributes. In contrast, CURE maintains relatively large margins even in low-shot settings, suggesting stronger score-level separation and more robust image-attribute alignment. These results provide empirical evidence that the proposed dual-attribute consistency design helps preserve discriminative attribute alignment under limited supervision.

Implication for dual-attribute consistency. The above analysis clarifies why the JS divergence between descriptive and distinctive posteriors can serve as a semantic consistency signal. If the two attribute views produce similar posteriors, the pseudo-label is more likely to be stable across complementary textual descriptions. If their posteriors diverge, the prediction may rely on view-specific cues and become less reliable as pseudo supervision. Therefore, the dual-attribute consistency weight acts as a semantic reliability gate, up-weighting samples with consistent attribute-view predictions and down-weighting samples with unstable pseudo-labels.

H Task Scope and Robustness to Unknown-Class Contamination

CURE is primarily designed for **image classification**. Its pseudo-labeling mechanism and dual-attribute consistency objective are both defined over class-level prediction distributions. Therefore, the current formulation is not directly applicable to structured prediction tasks such as object detection or semantic segmentation without task-specific modifications. For example, extending CURE to these tasks would require adapting pseudo-label generation and consistency regularization to region-level or pixel-level predictions. Although our main experiments follow the standard closed-set learning SSL setting, unlabeled data in practical scenarios may contain samples from categories outside the predefined label space. We further evaluate the robustness of CURE when the closed-set SSL assumption is violated by introducing **unknown-class samples** into the unlabeled pool.

Specifically, for each dataset, we randomly designate 80% of the categories as **known classes** and the remaining 20% as **unknown classes**. The unknown classes are completely excluded from the classifier output space, text prompts, labeled training set, and hyperparameter selection. Starting from the original unlabeled pool, we replace 0%, 25%, or 50% of the known-class unlabeled samples with samples from unknown classes while keeping the total size of the unlabeled pool unchanged. All other training settings remain identical. We report both the **average** accuracy and the **worst-case** accuracy across the three contamination levels.

Table 18: Robustness to unknown-class contamination in the unlabeled pool. Each entry reports average / worst-case accuracy (%) across contamination ratios of 0%, 25%, and 50%.

Method	DTD	Aircraft	Cars	UCF101
AR+FlexMatch	59.21 / 58.77	30.76 / 29.79	68.33 / 68.19	74.53 / 74.40
AttrVR+FlexMatch	63.01 / 62.28	35.49 / 35.21	67.03 / 66.54	73.57 / 72.77
CURE	**66.81 / 66.23**	**37.08 / 36.56**	**71.50 / 71.04**	**77.12 / 76.52**

As shown in Table 18, CURE achieves the highest average and worst-case accuracy on all four datasets. In particular, the small gap between its average and worst-case performance indicates that CURE remains relatively stable even when up to 50% of the unlabeled pool is replaced by samples

from unknown categories. These results suggest that the proposed framework is not overly dependent on a perfectly clean closed-set unlabeled pool and can tolerate unknown-class contamination.

We emphasize that this experiment evaluates **robustness to violations of the closed-set SSL assumption**, rather than extending CURE to a new prediction task or to explicit open-set recognition. The prediction space remains restricted to the known categories throughout training and evaluation, and the model is not required to identify or classify unknown categories. Extending CURE to other prediction settings, such as detection, segmentation, or explicit open-set recognition, would require corresponding task-specific formulations of pseudo-labeling and consistency learning, which we leave for future work.

I Broader impacts

This work investigates semi-supervised visual reprogramming for adapting frozen vision-language models under limited labeled supervision. The proposed framework may have positive societal impacts by reducing annotation costs and computational barriers for downstream visual classification, especially in scenarios where labeled data are scarce but unlabeled data are available. However, improved data-efficient adaptation could also be misused in sensitive visual recognition applications, such as surveillance or identity-related classification, and may inherit biases from labeled data, unlabeled data, or automatically generated attributes. Therefore, we recommend careful dataset auditing, fairness evaluation, and task-specific validation before deploying such methods in high-stakes or human-centered applications.

J Limitations and future work

The main limitation of CURE lies in the scope of unlabeled data. Following the common setting of semi-supervised learning, CURE assumes that the unlabeled images are drawn from the same target-domain distribution as the labeled examples. Under this assumption, unlabeled samples can provide useful target-domain structure for improving visual reprogramming under limited labeled supervision. However, when the unlabeled data contain substantial out-of-domain or distribution-shifted samples, such as cartoon-like, sketch-like, or otherwise visually shifted variants, the reliability of pseudo supervision may degrade. In such cases, confidence-based weighting and dual-attribute consistency can reduce the influence of unreliable predictions to some extent, but they do not fully eliminate the fundamental dependence on in-domain unlabeled data. Therefore, the contribution of CURE should be understood as improving the utilization of in-domain unlabeled data for semi-supervised visual reprogramming, rather than solving domain adaptation or out-of-domain unlabeled data learning.

This limitation is shared by many semi-supervised learning methods, where the quality and distribution of unlabeled data play a central role in the effectiveness of pseudo-label-based training [29]. Recent studies have investigated more challenging settings such as class-distribution mismatch [19] and out-of-domain unlabeled data learning [40]. Extending CURE to explicitly detect, reweight, or selectively exploit out-of-domain unlabeled samples is an important direction for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We include detailed information in Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide the limitations in Appendix J.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the assumptions and proofs of dual-attributed consistency regularization in Appendix G.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the details of experiments settings in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the details of reproducibility code in anonymous link `https://github.com/iLearn-Lab/NeurIPS26-CURE/`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the experimental details in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The error bars are included in Table 1 and Table 2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the experiment compute resources in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: In every respect in the paper, we follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide the broader impacts in Appendix I.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All data, models, and code in the paper respect the license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not include crowdsourcing and human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We use LLMs to automatically generate class-level descriptive and distinctive attributes, as described in Appendix D. The LLMs are not used for sample-level annotation or pseudo-label generation.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.