YOU ONLY REPROGRAM ONCE: RETHINKING PROLONGED TRAINING FOR VISUAL REPROGRAMMING

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual reprogramming is a parameter-efficient method for adapting pretrained models, yet its training can remain computationally expensive: even with a frozen backbone, visual prompts are often optimized through the full model for hundreds of epochs. Before changing what the pretrained model sees, we ask whether we are fully using what it already tells us. We find that modeling the full source response can already yield strong downstream predictions without prompt optimization. Motivated by this observation, we introduce You Only Reprogram Once (YORO), which constructs a downstream predictor from the frozen response space in a single forward-only traversal. Its Bayesian Discriminant Mapping (BDM) derives a covariance-aware affine mapping from streaming class statistics, requiring no backpropagation, optimizer updates, or repeated visits to the training set. When further input adaptation helps, YORO-FP optionally refines the visual prompt for 20 epochs. BDM also extends naturally to CLIP by treating attribute-prompt similarities as source responses. Across three full-data settings, YORO improves average accuracy over the strongest prior gradient-free mapping by 18.4–24.4%. On 16-shot CLIP, it raises the four-backbone average from 71.4% to 77.2%. YORO-FP provides further gains on selected tasks, while validation often retains the one-pass predictor. These results suggest a different default for visual reprogramming: read out the frozen response first, and optimize the input only when needed. [1]

1 INTRODUCTION

Visual reprogramming adapts a pretrained model to a downstream task by modifying lightweight input and output interfaces while keeping the backbone frozen (Elsayed et al., 2018; Tsai et al., 2020; Chen et al., 2023; Tsao et al., 2024). A visual prompt transforms downstream inputs into the source input space, while a label mapping converts source responses into target predictions. Despite optimizing few parameters, gradient-based visual reprogramming still requires repeated backpropagation through the frozen backbone, and recent methods commonly train for hundreds of epochs (Cai et al., 2024a;b). This raises an important question: *does effective visual reprogramming actually require prolonged prompt optimization?*

A key difficulty is that prompt optimization and label mapping are usually evaluated together. Poor initial performance may therefore reflect either insufficient downstream information or an insufficient readout of information already present in the source response. RLM, FLM, and ILM construct one-to-one source–target correspondences (Elsayed et al., 2018; Tsai et al., 2020; Chen et al., 2023), while BLM and BLM+ use predicted source labels or selected probabilities (Cai et al., 2024a). However, these mappings still use only part of the source response and may discard information that separates downstream classes.

Our central observation is that the output readout can change the apparent need for prompt training. As illustrated in Figure 1, prior label mapping methods model Top-1 or Top-K source-label information, whereas our method models the full source response. Figure 1(b) shows the case where, using the same ResNet–18 backbone and the same watermark-style visual prompt parameterization, a stronger readout of the initial frozen response already outperforms a weaker mapping paired with prolonged prompt optimization. Moreover, continued prompt training reduces the training loss without improving validation accuracy.

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

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

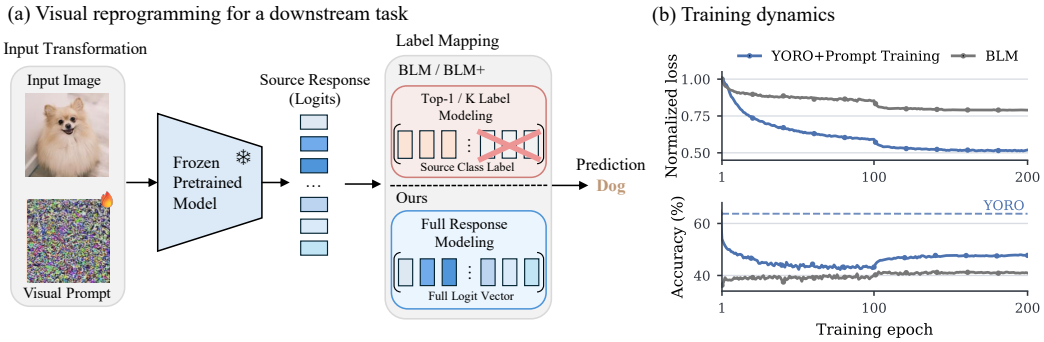


Figure 1: **Stronger label mapping, less prompt training.** Compared with Cai et al. (2024a), which relies on Top-1 or Top-K source-label information, our method (YORO) models the full source response. Using a single traversal of the training set, YORO already outperforms BLM trained for 200 epochs on CIFAR-100, while further visual-prompt optimization may reduce training loss without improving validation accuracy.

Motivated by this observation, we introduce *You Only Reprogram Once (YORO)*, a one-pass visual reprogramming framework. YORO fixes the input transformation at initialization, traverses the labeled training set once using forward passes only, and constructs the downstream predictor directly from the resulting responses. It requires no backward pass, prompt update, intermediate features, or access to model parameters, and can therefore operate on output-access black-box models when full response vectors are available.

To make this one-pass formulation effective, we formulate *Bayesian Discriminant Mapping (BDM)*, an output label mapping method that treats the complete response as a class-conditional observation. Following Fisher's criterion (Fisher, 1936), target-class means encode separation and a shared covariance measures within-class variation. The corresponding Gaussian decision rule is affine, so its weights and biases directly form the label mapping. To stabilize high-dimensional covariance estimation, BDM shrinks the empirical covariance toward a trace-matched identity (Friedman, 1989). BDM dynamically selects how much correlation structure to retain, emphasizing stable discriminative directions without amplifying noisy near-null directions. Streaming statistics recover the entire regularized mapping in one traversal, without backpropagation or storing individual responses.

This perspective also clarifies when additional prompt optimization can still be useful. If the initial frozen response does not contain sufficient downstream structure, modifying the input can reshape the response space and improve the readout. We therefore introduce *YORO-FP*, an optional few-pass extension that starts from the one-pass YORO predictor and performs limited visual-prompt refinement. Rather than treating prolonged training as the default, YORO-FP makes input adaptation an additional step whose value can be assessed against the one-pass predictor.

Across three full-data backbone–input settings, YORO improves average accuracy over the strongest prior gradient-free label mapping by 18.4–24.4% and over the reported learned linear mapping by 6.3–12.3%. The same formulation extends naturally to CLIP by treating attribute-prompt similarities as the source response. In the 16-shot setting, YORO raises the four-backbone average accuracy from 71.4% to 77.2% while requiring no prompt optimization. YORO-FP provides additional gains on selected tasks, although validation frequently retains the original one-pass predictor. These results establish YORO as a strong one-pass baseline for visual reprogramming, showing that substantial downstream structure can be recovered from the frozen response space before prompt optimization. Our contributions are:

- We show that prolonged prompt optimization can be unnecessary when the frozen response space is modeled effectively.
- We introduce BDM, a covariance-aware mapping that models the full source response and recovers an affine predictor in one pass.
- We introduce YORO, requiring one forward-only traversal and no prompt optimization, and YORO-FP for optional few-pass refinement.
- Across full-data visual reprogramming and few-shot CLIP adaptation, YORO substantially improves prior mappings and establishes a strong one-pass baseline.

2 Related Work

Visual reprogramming and label mapping. Visual reprogramming adapts a frozen pretrained model through lightweight input and output interfaces (Elsayed et al., 2018; Tsai et al., 2020; Chen, 2024). Pixel-space methods typically learn either padding patterns (Tsai et al., 2020; Chen et al., 2023; Tsao et al., 2024) or watermark-style perturbations (Bahng et al., 2022; Cai et al., 2024b; Oh et al., 2023). Because source and downstream tasks generally use different label spaces, an output label mapping is required to convert pretrained responses into downstream predictions. RLM uses random source-to-target assignments (Elsayed et al., 2018), FLM selects frequently predicted source labels (Tsai et al., 2020), and ILM iteratively updates this one-to-one mapping during prompt training (Chen et al., 2023). BLM extends this design to a dense probabilistic mapping from top-1 source predictions, while BLM+ incorporates top-K predicted probabilities (Cai et al., 2024a). Although these mappings can be constructed without gradient-based fitting, the overall reprogramming pipelines still rely on multi-epoch optimization of the input prompt.

Recent CLIP-based visual reprogramming instead improves image–text alignment. AttrVR (Cai et al., 2025a) introduces multiple attribute descriptions for each downstream class, DVP (Cai et al., 2025b) learns multiple visual prompts with complementary roles, and Dual-Granularity Alignment (Wu et al., 2026) exploits semantic label hierarchies together with multi-scale visual representations.

In contrast, BDM operates on the complete pretrained response rather than compressed predictions or selected similarities, and models its downstream class structure directly. Building on BDM, YORO turns this response geometry into a streaming, one-pass reprogramming interface with dataset-size-independent memory.

Training-efficient adaptation. Prior work mainly reduces or avoids prompt optimization. DAM-VP uses meta-initialized cluster-specific prompts with 10-epoch adaptation (Huang et al., 2023), AutoVP selects task-specific prompt, backbone, and mapping configurations (Tsao et al., 2024), and LoR-VP adopts a low-rank prompt parameterization (Jin et al., 2025). Related token-prompt methods improve initialization or optimization through gradient regulation (Li et al., 2023), downstream prototypes (Wang et al., 2024a), one-pass initialization (Park & Chung, 2025), or stabilized prompt dynamics (Wang et al., 2026). Other methods reduce training further: CARPRT reweights prompt scores using unlabeled target images (Dong et al., 2026). Tip-Adapter builds a few-shot feature cache for lightweight CLIP adaptation (Zhang et al., 2022). Wang et al. (2024b) apply Gaussian discriminant analysis directly in the CLIP visual-feature space and therefore assume access to intermediate features. In contrast, BDM operates directly on model responses, allowing the same formulation to be applied to both conventional classifiers and vision–language models without requiring feature-space access. AReS instead reduces repeated model access by using one service-API pass to prime a local encoder before reprogramming the proxy (Zhang et al., 2026). YORO goes further by constructing the downstream predictor from labeled data in a single traversal, while YORO-FP tests whether a small 20-epoch prompt budget provides additional value.

3 Method

3.1 Setting and the one-pass question

Following Cai et al. (2024a), let the source and target label spaces be $\mathcal{Y}^{\mathrm{S}} = \{1, \ldots, k_{\mathrm{S}}\}$ and $\mathcal{Y}^{\mathrm{T}} = \{1, \ldots, k_{\mathrm{T}}\}$. A frozen source classifier $f_{\mathrm{pre}}\colon \mathcal{X}^{\mathrm{S}} \to \mathbb{R}^{k_{\mathrm{S}}}$ and an input transformation $f_{\mathrm{in}}(\cdot \mid \theta)\colon \mathcal{X}^{\mathrm{T}} \to \mathcal{X}^{\mathrm{S}}$ produce

$$z_i(\theta) = (f_{\mathrm{pre}} \circ f_{\mathrm{in}})(x_i^{\mathrm{T}}; \theta) \in \mathbb{R}^{k_{\mathrm{S}}}. \tag{1}$$

Here θ denotes the trainable parameters of the input visual prompt, whereas f_{pre} remains frozen. This response-space formulation also covers CLIP. Let $\mathcal{A} = \{a_{c,j} : c \in [k_{\mathrm{T}}], j \in [m]\}$ be the attribute-prompt bank used by DVP, with m descriptions per downstream class. Given frozen CLIP image and text encoders ϕ_{I} and ϕ_{T}, we define

$$z_{i,c,j}(\theta) = \tau_{\mathrm{C}} \left\langle \bar{\phi}_{\mathrm{I}}(f_{\mathrm{in}}(x_i^{\mathrm{T}} \mid \theta)), \bar{\phi}_{\mathrm{T}}(a_{c,j}) \right\rangle, \tag{2}$$

where bars denote ℓ_2 normalization and τ_{C} is CLIP's frozen logit scale (Radford et al., 2021; Cai et al., 2025b). Flattening (c, j) gives $z_i \in \mathbb{R}^{mk_{\mathrm{T}}}$, so we set $k_{\mathrm{S}} = mk_{\mathrm{T}}$ and apply BDM without modification. The attribute prompts and both CLIP encoders remain fixed. YORO-FP may update only

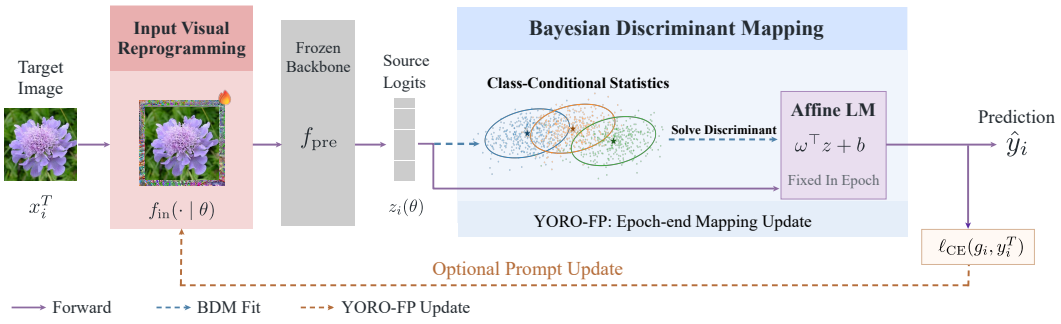


Figure 2: **One pass first, optional prompt refinement when needed.** YORO fits BDM to the full frozen response in one pass. BDM estimates target class means and a shared covariance in the source-response space, then solves these statistics into an affine label mapping. The response can be classifier logits or CLIP image–text similarities. The dashed orange path denotes optional YORO-FP refinement, where only the visual prompt is optimized and the mapping is updated between epochs.

the visual input prompt. Since these logits lie in the visual embedding subspace, our implementation uses an exact low-rank identity to avoid forming the $(mk_{\text{T}}) \times (mk_{\text{T}})$ covariance (Appendix B.1). It accumulates statistics and solves the system in the visual embedding dimension while producing exactly the same logit-space mapping. We use an affine label mapping

$$f_{\text{out}}^{\omega,b}(z) = \omega^\top z + b, \qquad \omega \in \mathbb{R}^{k_{\text{S}} \times k_{\text{T}}}, \quad b \in \mathbb{R}^{k_{\text{T}}}. \tag{3}$$

The conventional objective jointly adapts this interface through many visits to the labeled target set. We instead first hold $\theta = \theta_0$ fixed and ask whether (ω, b) can be constructed after one visit to each training example. The complete one-pass construction and its optional refinement path are summarized in Figure 2.

3.2 Bayesian Discriminant Mapping

Mathematically, BDM repurposes regularized linear discriminant analysis (LDA) (Fisher, 1936; Friedman, 1989) as an output label mapping over the full pretrained response space. Existing mappings reduce the source response to labels, selected probabilities, or predefined associations before predicting the target class. BDM instead retains the complete response vector and asks which response directions separate target classes relative to their within-class variation. Target-class means describe the between-class geometry, while a shared covariance provides the within-class metric and captures correlations among response dimensions. A shared-covariance Gaussian model turns this criterion into a Bayesian decision rule with the affine form in Equation (3). Its parameters depend only on first- and second-order statistics. We therefore model

$$Z \mid Y^{\text{T}} = c \sim \mathcal{N}(\mu_c, \Sigma), \qquad \pi_c = p(Y^{\text{T}} = c). \tag{4}$$

Here μ_c is the mean response for target class c, and Σ is the shared within-class response covariance. The shared covariance supplies a common metric between class means and avoids estimating a separate covariance for every target class. Up to terms independent of c, Bayes' rule gives

$$g_c(z) = z^\top \Sigma^{-1} \mu_c - \tfrac{1}{2} \mu_c^\top \Sigma^{-1} \mu_c + \log \pi_c. \tag{5}$$

Thus, the Gaussian decision rule is exactly an affine output mapping. Between classes c and d, its decision direction is $\Sigma^{-1}(\mu_c - \mu_d)$, which emphasizes mean differences that are large relative to within-class variation. Each covariance-adjusted mean $\Sigma^{-1}\mu_c$ becomes the mapping vector for target class c, while the bias encodes its class-conditional location and prior. This Fisher view is the core of BDM: it converts the complete pretrained response geometry directly into target logits, recovers the mapping from streaming statistics, and requires no gradient-based optimization.

The key challenge is reliable covariance estimation. When k_{S} is large relative to target supervision, especially in few-shot settings, the empirical covariance can be noisy or singular. BDM therefore combines streaming estimation with shrinkage. For each target class, it stores a count and response sum together with one global second moment:

$$n_c = \sum_i \mathbf{1}[y_i^{\text{T}} = c], \qquad s_c = \sum_{i: y_i^{\text{T}} = c} z_i, \qquad Q = \sum_i z_i z_i^\top. \tag{6}$$

Let $\mathbf{n} = (n_1, \ldots, n_{k_\mathrm{T}})$ and $S = [s_1, \ldots, s_{k_\mathrm{T}}]$. After one traversal,

$$\widehat{\mu}_c = \frac{s_c}{n_c}, \quad \widehat{\pi}_c = \frac{n_c}{n}, \quad \widehat{\Sigma} = \frac{Q - \sum_c s_c s_c^\top / n_c}{n - k_\mathrm{T}}. \tag{7}$$

To stabilize covariance estimation in the high-dimensional response space, BDM shrinks the covariance toward a trace-scaled identity:

$$\widehat{\Sigma}_\rho = (1-\rho)\widehat{\Sigma} + \rho \frac{\mathrm{tr}(\widehat{\Sigma})}{k_\mathrm{S}} I + \delta I, \qquad \rho \in [0,1], \tag{8}$$

where ρ trades off empirical covariance structure against isotropic stability, and $\delta \geq 0$ is added only when required for stable factorization. Shrinkage suppresses poorly estimated directions while retaining informative covariance structure when supported by the data.

We select ρ by validation accuracy. All candidates reuse the same streaming statistics and cached validation responses, so the search requires only one training traversal and one validation traversal. Candidate-specific work is limited to forming $\widehat{\Sigma}_\rho$, solving the analytic output mapping, and scoring the cached responses. Ties are assigned to the smaller ρ.

Let $M = [\widehat{\mu}_1, \ldots, \widehat{\mu}_{k_\mathrm{T}}]$. BDM computes all target class weights jointly:

$$\widehat{\Sigma}_\rho \omega_0 = M, \qquad b_{0,c} = -\tfrac{1}{2}\widehat{\mu}_c^\top (\omega_0)_{:,c} + \log \widehat{\pi}_c. \tag{9}$$

The resulting predictor is $f_{\mathrm{out}}^{\omega_0, b_0}(z) = \omega_0^\top z + b_0$.

Proposition 1 (Exact streaming recovery) *For fixed responses $\{z_i\}$ and shrinkage ρ, the affine mapping computed by BDM from the streaming statistics (n, S, Q) is exactly the same as that obtained by fitting BDM jointly to all training responses.*

Thus, the one-pass formulation introduces no approximation: the class means and shared covariance required by BDM are fully determined by (n, S, Q). YORO can therefore recover the exact BDM mapping in a single traversal, without storing individual responses or revisiting the training set. The proof is given in Appendix B.

3.3 YORO-FP: CONTROLLED FEW-PASS REFINEMENT

BDM can recover target structure only when the frozen responses preserve it. When they do not, the input interface can reshape the response space first. YORO-FP therefore starts from $(\theta_0, \omega_0, b_0)$ and permits 20 additional prompt epochs while f_{pre} remains frozen. During epoch t, the active label mapping is fixed and optimizer updates only θ:

$$\mathcal{L}_{\mathrm{FP}}(\theta; \omega_t, b_t) = \frac{1}{n} \sum_i \ell_{\mathrm{CE}}\left(\omega_t^\top z_i(\theta) + b_t, y_i^{\mathrm{T}}\right). \tag{10}$$

The same minibatch logits, detached before the prompt update, accumulate a new set of BDM statistics. At the epoch boundary these statistics yield a candidate $(\widehat{\omega}_{t+1}, \widehat{b}_{t+1})$. By default, we use the Momentum update to stabilize the LM as prompt optimization gradually changes the response geometry. At each epoch, the newly estimated BDM parameters are blended with the previous mapping:

$$\omega_{t+1} = \beta \omega_t + (1-\beta)\widehat{\omega}_{t+1}, \qquad b_{t+1} = \beta b_t + (1-\beta)\widehat{b}_{t+1}, \tag{11}$$

with the fixed smoothing coefficient $\beta = 0.9$. This smooths epoch-to-epoch changes in the discriminant mapping while allowing it to track the evolving response space, without any additional refitting traversal. Algorithm 1 in the appendix summarizes the shared one-pass initialization and optional few-pass refinement.

4 EXPERIMENTS

Protocols. We evaluate YORO in two complementary settings. For full-data visual reprogramming, we use ImageNet-1K-pretrained ResNet–18 (He et al., 2016; Russakovsky et al., 2015) and

Table 1: Performance comparison of visual reprogramming methods with different output label mappings under the padding visual prompt setting (top-1 accuracy, %). YORO and YORO-FP report mean ± sample standard deviation over three seeds. Baseline results are directly taken from Cai et al. (2024a). YORO results are highlighted, the best prior gradient-free label mapping is shown in **bold**, and the gradient-based method (Deep) is shown in gray. Our methods achieve the best overall results.

	ResNet–18 (ImageNet–1K)								ResNeXt–101–32x8d (Instagram)						
Dataset	RLM	FLM	ILM	BLM	BLM+	Deep	YORO	YORO-FP	FLM	ILM	BLM	BLM+	Deep	YORO	YORO-FP
Flowers102	11.0	20.0	27.9	44.4	50.1	76.7	**88.0**±0.0	87.9±0.3	22.5	27.9	31.5	30.1	85.2	94.0±0.0	**94.3**±0.1
DTD	16.3	32.4	35.3	42.0	43.9	49.1	59.0±0.0	**59.6**±0.5	40.3	41.4	47.8	49.4	64.0	**70.3**±0.0	**70.3**±0.0
UCF101	6.6	18.9	23.9	30.9	32.0	46.0	**61.3**±0.0	**61.3**±0.0	41.9	43.1	48.3	50.1	68.3	74.9±0.0	**75.4**±0.3
Food101	3.8	12.8	14.8	23.2	25.1	34.1	**49.8**±0.0	**49.8**±0.0	20.5	23.0	29.6	31.4	58.7	**70.3**±0.0	**70.3**±0.0
GTSRB	46.1	45.5	52.0	54.8	54.3	63.1	79.0±0.1	**80.2**±0.4	56.2	59.9	62.9	63.0	74.4	79.3±0.1	**79.6**±0.5
EuroSAT	82.4	83.8	85.2	86.7	86.7	92.4	**92.8**±0.0	92.5±0.2	87.8	86.2	87.6	88.3	93.2	93.0±0.0	**93.4**±0.3
OxfordPets	9.3	62.9	65.4	69.8	70.6	73.0	**86.9**±0.0	**86.9**±0.0	76.8	78.9	82.4	83.0	91.8	94.8±0.0	**95.1**±0.1
StanfordCars	0.9	2.7	4.5	5.4	7.7	14.3	34.9±0.0	**37.1**±0.1	4.6	7.0	8.3	9.3	50.5	71.1±0.0	**72.2**±0.3
SUN397	1.0	10.4	13.0	16.2	18.7	26.3	**47.8**±0.0	**47.8**±0.0	21.6	23.7	30.1	32.0	51.5	**61.9**±0.0	**61.9**±0.0
CIFAR10	63.0	65.7	65.5	66.7	66.8	72.1	**86.2**±0.1	**86.2**±0.1	80.3	81.7	82.2	82.2	83.4	**95.3**±0.0	**95.3**±0.0
CIFAR100	12.9	18.1	24.8	29.6	30.6	46.7	**64.8**±0.1	**64.8**±0.1	39.7	45.9	47.8	47.8	56.2	**76.3**±0.1	**76.3**±0.1
SVHN	73.5	73.1	75.2	74.5	74.2	82.1	72.3±0.1	**77.0**±0.3	79.0	**81.4**	79.8	79.3	85.7	56.5±0.0	68.6±0.6
Average	27.2	37.2	40.6	45.3	46.7	56.3	68.6±0.0	**69.3**±0.1	47.6	50.0	53.2	53.8	71.9	78.2±0.0	**79.4**±0.1

Instagram-pretrained ResNeXt–101–32x8d (Mahajan et al., 2018). Both expose 1,000 source logits. We evaluate on Flowers102 (Nilsback & Zisserman, 2008), DTD (Cimpoi et al., 2014), UCF101 (Soomro et al., 2012), Food101 (Bossard et al., 2014), GTSRB (Stallkamp et al., 2012), EuroSAT (Helber et al., 2019), OxfordPets (Parkhi et al., 2012), StanfordCars (Krause et al., 2013), SUN397 (Xiao et al., 2010), CIFAR-10/100 (Krizhevsky & Hinton, 2009), and SVHN (Netzer et al., 2011). The watermark setting uses the same 12 tasks with ResNet–18. We compare against RLM, FLM, ILM, BLM, BLM+ (Elsayed et al., 2018; Tsai et al., 2020; Chen et al., 2023; Cai et al., 2024a), and Deep. Deep is a gradient-based label mapping that uses the same frozen backbone and input interface, but learns a dense single-layer linear output mapping jointly with the visual prompt. All baselines follow their original 200-epoch training protocols.

For few-shot adaptation, we follow the 16-shot DVP protocol (Cai et al., 2025b), covering 11 datasets and four frozen CLIP backbones (Radford et al., 2021). By default, DVP refers to the variant using unsupervised prompt clustering. YORO uses the same DVP attribute-prompt bank and treats the complete image–text similarity vector in Equation (2) as its frozen response.

Unless stated otherwise, we report top-1 classification accuracy (%) and mean ± sample standard deviation over three random seeds. We consider two input reprogramming forms: *padding* places a learnable prompt around the resized target image, whereas *watermarking* overlays a learnable perturbation on the image itself. Following Cai et al. (2025a;b), we use 16-pixel padding. YORO-FP performs 20 additional prompt epochs. Further implementation details are provided in Appendix A.

4.1 Results on Image Classifiers

Tables 1 and 2 show that YORO outperforms prior gradient-free label mappings on average across all three settings, without iterative prompt optimization. The gains span diverse tasks, including StanfordCars, CIFAR-100, UCF101, Food101, OxfordPets, and SUN397.

For padding, we follow Cai et al. (2025a;b) and resize images to 192×192 within a 224×224 canvas, leaving a 16-pixel border. This standardizes the image area across datasets, avoids excessively small image regions under legacy padding, and limits the prompt to 39,936 trainable parameters during optional refinement. We also evaluate YORO under the legacy configuration, which applies no resizing and instead uses dynamic padding to match the pretrained

Table 2: Watermarking-based visual reprogramming on ResNet–18 (top-1 accuracy, %). Baseline results are taken from Cai et al. (2024a). Our methods achieve the best overall results.

	ResNet–18 (ImageNet–1K)					
Dataset	ILM	BLM	BLM+	Deep	YORO	YORO-FP
Flowers102	23.2	39.2	44.1	82.4	**88.8**±0.0	**88.8**±0.0
DTD	29.0	40.1	43.0	48.9	**62.5**±0.0	**62.5**±0.0
UCF101	24.4	32.9	35.4	53.1	**65.1**±0.0	**65.1**±0.0
Food101	13.2	21.5	22.9	30.4	**52.8**±0.0	**52.8**±0.0
GTSRB	76.8	82.1	82.0	89.5	75.9±0.1	**84.5**±0.6
EuroSAT	84.3	84.4	84.8	89.2	**93.2**±0.0	**93.2**±0.0
OxfordPets	70.0	72.4	73.3	77.6	**88.5**±0.0	**88.5**±0.0
StanfordCars	3.4	5.5	7.4	30.7	37.8±0.0	**38.6**±0.5
SUN397	13.4	18.4	19.4	32.9	**52.0**±0.0	**52.0**±0.0
CIFAR10	68.9	74.9	75.7	71.7	**86.9**±0.1	**86.9**±0.1
CIFAR100	33.8	41.2	41.6	39.9	**64.8**±0.1	**64.8**±0.1
SVHN	78.3	79.2	78.8	83.7	60.6±0.1	**84.7**±0.2
Average	43.2	49.3	50.7	60.8	69.1±0.0	**71.9**±0.0

Table 3: Performance comparison on 16-shot downstream classification using ViT-B/32 and ResNet–50 CLIP backbones (accuracy %). Prior results are taken from Cai et al. (2025b). Highlighted YORO columns report mean ± sample standard deviation over three seeds using DVP attribute-prompt similarities as source responses. Best results are in **bold**. Our methods achieve the best overall results.

	ViT-B/32 (CLIP)						ResNet–50 (CLIP)					
Dataset	VP	AR	AttrVR	DVP	YORO	YORO-FP	VP	AR	AttrVR	DVP	YORO	YORO-FP
FGVC Aircraft	24.3	21.8	24.5	26.1	36.6±0.8	**36.9**±1.2	16.2	18.6	20.7	22.1	36.2±1.5	**36.3**±1.4
Caltech101	92.3	92.7	92.0	92.9	**94.0**±0.0	93.7±0.6	80.1	86.5	89.1	89.8	**91.3**±0.1	91.0±0.4
StanfordCars	58.6	56.9	56.6	56.5	**71.8**±0.6	71.0±1.2	44.0	53.9	53.9	54.5	**68.6**±0.2	68.5±0.3
DTD	54.9	49.9	56.8	57.2	**65.8**±1.6	65.5±1.8	43.4	46.4	54.4	55.9	**64.4**±0.2	63.8±0.8
EuroSAT	85.9	85.6	**88.6**	88.5	82.3±1.5	82.1±2.4	59.7	66.6	72.0	72.2	**82.9**±0.7	79.7±5.7
Flowers102	71.2	66.7	77.8	82.5	**94.0**±0.1	**94.0**±0.1	53.6	60.9	74.8	80.0	**92.6**±0.5	92.5±0.7
Food101	75.0	75.7	**77.2**	77.0	75.1±1.3	75.1±1.3	65.3	74.2	**75.3**	75.0	72.8±0.2	72.6±0.4
OxfordPets	86.8	84.7	**89.8**	89.2	87.9±1.0	87.5±1.1	77.2	82.5	**88.9**	**88.9**	87.5±0.9	87.5±0.9
SUN397	61.0	59.9	62.8	64.2	**70.6**±0.2	**70.6**±0.2	48.8	56.8	59.9	61.1	**67.6**±0.3	**67.6**±0.3
UCF101	67.3	63.5	67.9	70.5	**77.8**±1.5	**77.8**±1.5	52.0	59.7	63.6	65.9	**75.1**±0.2	**75.1**±0.2
RESISC45	73.9	71.6	73.9	76.0	**82.6**±0.3	82.0±0.5	47.7	58.4	58.2	60.8	**79.4**±1.1	78.6±1.6
Average	68.3	66.3	69.8	71.0	**76.2**±0.3	76.0±0.2	53.5	60.4	64.6	66.0	**74.4**±0.2	73.9±0.6

input size, as detailed in Section A.2. YORO still substantially outperforms BLM+ on both backbones, demonstrating gains from the output mapping without the change in input resizing.

For watermarking, all methods resize images to the pretrained model's input resolution before applying the visual prompt. Under this matched protocol, YORO achieves 69.1% average accuracy on ResNet–18, compared with 50.7% for BLM+ and 60.8% for Deep, without iterative prompt optimization. YORO-FP further reaches 71.9%, with the largest gains on GTSRB and SVHN, where input adaptation is more useful.

These gains also come at substantially lower adaptation cost. While the baselines optimize visual prompts for 200 epochs, YORO requires only one forward-only traversal and an analytic mapping solve. YORO-FP uses only 20 additional prompt-training epochs, around 10% of the baseline training budget.

4.2 Few-Shot Results on CLIP

YORO can also be applied to CLIP-based few-shot learning. Following DVP (Cai et al., 2025b), we construct the source response from cosine similarities between image embeddings and attribute prompts, and apply BDM to learn the downstream class geometry in a single pass. Full per-dataset results for ViT-B/32 and ResNet–50 are reported in Table 3, with the four-backbone summary in Table 4. Across all four backbones, the highest average accuracy is achieved by either YORO or YORO-FP.

Table 4: Aggregate 16-shot accuracy (%) over 11 datasets. YORO results report mean ± sample standard deviation over three seeds. Overall averages the four frozen CLIP backbones equally. Our methods achieve the best overall results.

Method	RN50	RN101	ViT-B/32	ViT-B/16	Overall
VP	53.5	57.5	68.3	74.4	63.4
AR	60.4	62.7	66.3	76.5	66.5
AttrVR	64.6	67.2	69.8	78.5	70.0
DVP	66.0	68.8	71.0	79.7	71.4
YORO	**74.4**±0.2	**76.8**±0.2	**76.2**±0.3	81.2±0.1	**77.2**±0.1
YORO-FP	73.9±0.6	76.7±0.3	76.0±0.2	**81.4**±0.3	77.0±0.3

The gains are particularly strong on the weaker backbones, reaching 8.4% and 8.0% on ResNet–50 and ResNet–101, compared with 5.2% on ViT-B/32 and 1.5% on ViT-B/16. At the dataset level, YORO provides large improvements on fine-grained tasks such as FGVC Aircraft and StanfordCars, as well as Flowers102, SUN397, UCF101, and RESISC45. For example, on ViT-B/32 it improves DVP from 26.1% to 36.6% on FGVC Aircraft and from 56.5% to 71.8% on StanfordCars. This pattern suggests that modeling the full similarity response can recover class structure that is not captured by the original DVP readout, especially when the frozen backbone provides less separable features.

YORO-FP provides little additional benefit on average. It improves the backbone-level average only on ViT-B/16, while matching or slightly trailing YORO on the other three backbones. This indicates that, for most CLIP settings, the stronger full-response readout already captures most of the available adaptation signal before prompt refinement, with additional input optimization becoming useful only in selected cases.

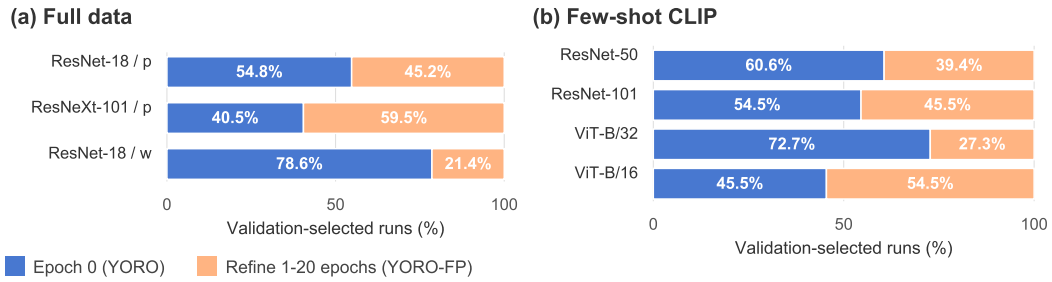


Figure 3: **Prompt refinement is beneficial only for a subset of runs.** Fraction of three-seed runs whose best validation checkpoint is Epoch 0 (YORO) or Epochs 1–20 (YORO-FP) for (a) full-data visual reprogramming and (b) 16-shot CLIP. Here, p and w denote padding and watermarking, respectively.

4.3 Does visual prompt training still help?

Figure 3 examines whether continued prompt refinement is selected by validation. Across five of the seven backbone–setting combinations, Epoch 0 is selected more often than refined checkpoints, including 78.6% of runs for ResNet–18 watermarking and 72.7% for ViT-B/32 CLIP. Refinement is selected more often for ResNeXt–101 padding and ViT-B/16 CLIP, showing that additional input adaptation remains useful in some settings.

SVHN provides the clearest dataset-level example: YORO-FP improves over YORO by 4.7%, 12.1%, and 24.1% across the three full-data settings, with gains also observed on GTSRB. These tasks differ substantially from the natural-image categories used for source pretraining, making prompt refinement more useful for reshaping the response space.

4.4 Accuracy and Training Cost.

The accuracy–efficiency advantage is clear on Oxford Flowers with CLIP ViT-B/32 under the 16-shot setting (Table 5). YORO achieves the highest accuracy, 94.0%, in only 17.1 seconds. DGA (Wu et al., 2026), the closest baseline in accuracy, reaches 91.8% but requires 6562.1 seconds. Other methods show the same gap: DVP and AttrVR require thousands of seconds while reaching 80.9% and 76.2%, respectively, while AutoVP, BLM, and BLM+ achieve only 72.7–81.5% with 2754.4–5019.9 seconds of training.

Table 5: Accuracy and training time on Oxford Flowers with CLIP ViT-B/32 under the 16-shot setting. Results are mean ± sample standard deviation over three seeds. YORO achieves the best accuracy with the lowest training cost. All methods in this table are rerun under the same hardware protocol.

Method	Test accuracy (%)	Training time (s)
YORO	$\mathbf{94.0 \pm 0.1}$	$\mathbf{17.1 \pm 6.2}$
DGA	91.8 ± 4.8	6562.1 ± 34.5
AutoVP	81.5 ± 0.4	2754.4 ± 21.3
DVP	80.9 ± 1.1	3714.9 ± 7.3
AttrVR	76.2 ± 0.6	2683.2 ± 7.4
BLM	73.1 ± 1.2	4996.0 ± 122.3
BLM+	72.7 ± 1.0	5019.9 ± 20.9

YORO therefore improves efficiency without sacrificing accuracy. Instead of repeatedly optimizing the visual prompt through the frozen backbone, it constructs the downstream mapping from a single forward traversal of the labeled data. This supports our central hypothesis that prolonged prompt optimization is often unnecessary when the frozen response space can already be read out effectively.

4.5 Ablation Study

We study how the affine label mapping should be handled during the optional YORO-FP prompt refinement. As the visual prompt is updated, the frozen backbone produces a gradually changing response space, raising the question of whether the initial YORO mapping should remain fixed or adapt to this change. We compare three strategies. **Frozen** keeps the initial YORO mapping (ω_0, b_0) unchanged throughout refinement. **Refresh** directly replaces the mapping with the BDM estimate accumulated during the current prompt epoch. **Momentum** instead smooths each new mapping with the previous one using Equation (11) with $\beta = 0.9$. Equivalently, Frozen and Refresh correspond to $\beta = 1$ and $\beta = 0$, respectively. By default, we use the **Momentum** update strategy.

Table 6: **Full-data mapping-update ablation** (top-1 accuracy, %). Entries are 14-dataset macro-averages over three seeds. YORO is the one-pass predictor, while Frozen, Refresh, and Momentum denote YORO-FP variants with $\beta = 1$, 0, and 0.9, respectively. P/W denote padding/watermarking.

Setting	YORO	Frozen	Refresh	Momentum
RN18–P	71.0±0.0	71.1±0.0	71.1±0.0	**71.6**±0.1
RN18–W	71.5±0.0	73.3±0.1	73.6±0.1	**73.9**±0.0
RNXT101–P	79.8±0.0	80.2±0.0	80.6±0.1	**80.9**±0.1
Overall	74.1±0.0	74.9±0.0	75.1±0.1	**75.4**±0.0

Table 7: **Few-shot CLIP mapping-update ablation** (top-1 accuracy, %). Entries are 11-dataset macro-averages over three seeds. YORO is the one-pass predictor, while Frozen, Refresh, and Momentum denote YORO-FP variants with $\beta = 1$, 0, and 0.9, respectively.

Backbone	YORO	Frozen	Refresh	Momentum
ResNet–50	**74.4**±0.2	74.0±0.2	73.8±0.4	73.9±0.6
ResNet–101	**76.8**±0.2	76.5±0.2	76.4±0.0	76.7±0.3
ViT-B/32	**76.2**±0.3	76.0±0.4	76.1±0.2	76.0±0.2
ViT-B/16	81.2±0.1	80.9±0.2	81.3±0.3	**81.4**±0.3
Overall	**77.2**±0.1	76.8±0.1	76.9±0.1	77.0±0.3

The two ablations use task suites tailored to their protocols. The full-data study covers 14 datasets: the 12 datasets in the published-baseline comparison plus Caltech101 (Fei-Fei et al., 2004) and RESISC45 (Cheng et al., 2017). The few-shot study follows DVP and averages over its 11 datasets and four CLIP backbones.

Table 6 shows that Momentum performs best in all three full-data settings. Compared with one-pass YORO, it improves average accuracy by 0.6% with ResNet–18 padding, 2.4% with ResNet–18 watermarking, and 1.1% with ResNeXt–101 padding. Refresh generally improves over Frozen, indicating that adapting the mapping to the evolving response space can be useful, while Momentum provides a further benefit by avoiding abrupt epoch-to-epoch changes. The larger gain under watermarking is concentrated in tasks such as SVHN and GTSRB, where prompt adaptation changes the response space more substantially. Per-dataset results are reported in Table 15.

The few-shot CLIP results in Table 7 show a similar but weaker effect. Momentum is the strongest YORO-FP update rule overall and gives the best result on ViT-B/16, but one-pass YORO remains best when averaged across all four backbones, with 77.2% accuracy. Thus, smoothing the mapping is useful when prompt refinement meaningfully changes the response space, while the broader result remains unchanged: in most settings, the initial one-pass mapping already captures most of the available downstream structure.

5 Limitations

YORO assumes that the frozen response space retains useful structure for the target labels. SVHN shows that this assumption may weaken when the target label space differs substantially from the source label space, in which case visual prompt adaptation remains useful for reshaping the responses. Our experiments also focus on image classification. Extending the one-pass principle to other tasks and modalities remains an interesting direction.

6 Conclusion

Visual reprogramming often adopts long optimization schedules without first establishing whether they are necessary. YORO shows that a single labeled-data traversal can already produce a strong downstream predictor when the frozen response space is read out effectively. Using Bayesian Discriminant Mapping, YORO requires no backward pass or prompt update while consistently improving prior label mappings across full-data visual reprogramming and few-shot CLIP.

These results establish YORO as a strong one-pass baseline. They suggest that future methods should first determine how much downstream structure is already recoverable from frozen responses before introducing further prompt optimization. YORO-FP covers the complementary case where additional input adaptation is useful. This one-pass-to-few-pass perspective provides a simple reference for more efficient visual reprogramming.

AI USE STATEMENT

We used generative AI tools for language editing, LaTeX assistance, table formatting, and literature discovery for related work. We did not use generative AI tools to generate or alter data, run experiments, replace author decisions about the method, or make unsupported scientific claims. All AI-assisted text and code were reviewed by the authors. The authors designed the method and experiments, executed and tested the code, checked all numerical results against the recorded outputs, and manually verified the relevant literature. We take responsibility for the final content of this work, including any text, claims, code, or other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

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

REPRODUCIBILITY STATEMENT

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

REFERENCES

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

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

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

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

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

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

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

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

Yilun Chen, Ami Wiesel, Yonina C. Eldar, and Alfred O. Hero. Shrinkage algorithms for mmse covariance estimation. *Trans. Sig. Proc.*, 58(10):5016–5029, October 2010. ISSN 1053-587X. doi: 10.1109/TSP.2010.2053029. URL https://doi.org/10.1109/TSP.2010.2053029.

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

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

Ruijiang Dong, Zesheng Ye, Jianzhong Qi, Lei Feng, Feng Liu, Gang Niu, and Masashi Sugiyama. Carprt: Class-aware zero-shot prompt reweighting for black-box vision-language models. In *International Conference on Learning Representations (ICLR)*, 2026.

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

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

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. doi: https://doi.org/10.1111/j.1469-1809.1936.tb02137.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x.

Jerome H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989. doi: 10.1080/01621459.1989.10478752.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

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

Can Jin, Ying Li, Mingyu Zhao, Shiyu Zhao, Zhenting Wang, Xiaoxiao He, Ligong Han, Tong Che, and Dimitris N. Metaxas. Lor-VP: Low-rank visual prompting for efficient vision model adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5btFIv2PNb.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

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

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004. ISSN 0047-259X. doi: https://doi.org/10.1016/S0047-259X(03)00096-4. URL https://www.sciencedirect.com/science/article/pii/S0047259X03000964.

Juncheng Li, Minghe Gao, Longhui Wei, Siliang Tang, Wenqiao Zhang, Mengze Li, Wei Ji, Qi Tian, Tat-Seng Chua, and Yueting Zhuang. Gradient-regulated meta-prompt learning for generalizable vision-language models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2551–2562. IEEE, 2023.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

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

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NeurIPS workshop*, 2011.

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*. IEEE, 2008.

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

Jaekyun Park and Hye Won Chung. Vipamin: Visual prompt initialization via embedding selection and subspace expansion. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 122227–122262. Curran Associates, Inc., 2025. doi: 10.52202/085713-4075. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b0bc711f48724237b38823c4d9cee10b-Paper-Conference.pdf.

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

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

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

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

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2012.02.016. URL https://www.sciencedirect.com/science/article/pii/S0893608012000457. Selected Papers from IJCNN 2011.

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

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

Junze Wang, Lei Fan, Dezheng Zhang, Weipeng Jing, Donglin Di, Yang Song, Sidong Liu, and Cong Cong. Visual prompt-agnostic evolution. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3vmDyYC9Tn.

Yuzhu Wang, Lechao Cheng, Chaowei Fang, Dingwen Zhang, Manni Duan, and Meng Wang. Revisiting the power of prompt for visual tuning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Zhengbo Wang, Jian Liang, Lijun Sheng, Ran He, Zilei Wang, and Tieniu Tan. A hard-to-beat baseline for training-free CLIP-based adaptation. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=Js5PJPHDyY.

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

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

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

Yunbei Zhang, Chengyi Cai, Feng Liu, and Jihun Hamm. Prime once, then reprogram locally: An efficient alternative to black-box service model adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

A IMPLEMENTATION AND EVALUATION DETAILS

Full-data protocol. Following AttrVR and DVP (Cai et al., 2025a;b), we use the same padding for the principal experiment: a downstream image is resized to 192×192 and embedded in the 224×224 source canvas. This parameterization contains 39,936 trainable RGB values, substantially fewer than larger padding designs. The additive watermark is initially zero. The initial BDM pass is deterministic. YORO-FP uses random resized crops with scale $[0.7, 1.0]$ and ratio $[0.9, 1.1]$, plus horizontal flips except for GTSRB and SVHN. We use batch size 256 for ResNet–18 and 64 for ResNeXt. Adam optimizer (Kingma & Ba, 2015) updates only the prompt with learning rate 0.01, default $(\beta_1, \beta_2) = (0.9, 0.999)$, no weight decay, and no scheduler. The additional watermark experiment overlays a trainable input-space pattern. Full-data training-dynamics and update ablations include Caltech101 (Fei-Fei et al., 2004) and RESISC45 (Cheng et al., 2017) in addition to the 12 tasks in the published-baseline comparison. All experiments use three random seeds and the configurations specified above.

Few-shot CLIP protocol. The 16-shot experiments use FGVC Aircraft, Caltech101, StanfordCars, DTD, EuroSAT, Flowers102, Food101, OxfordPets, SUN397, UCF101, and RESISC45 (Maji et al., 2013; Fei-Fei et al., 2004; Krause et al., 2013; Cimpoi et al., 2014; Helber et al., 2019; Nilsback & Zisserman, 2008; Bossard et al., 2014; Parkhi et al., 2012; Xiao et al., 2010; Soomro et al., 2012; Cheng et al., 2017). We evaluate CLIP ResNet–50, ResNet–101, ViT-B/32, and ViT-B/16. We sample 16 training examples and at most four validation examples per class, and use batch size 64. Following DVP, each downstream class has 20 fixed attribute descriptions, making the source response dimension $20k_{\mathrm{T}}$. BDM is applied to the complete temperature-scaled image-to-attribute-prompt similarity vector in Equation (2). For CLIP, we accumulate equivalent statistics in the normalized visual-feature space and exploit an exact low-rank identity, avoiding construction of the full logit covariance. YORO-FP uses a prompt learning rate of 0.1. Its deterministic transform preserves aspect ratio before a 192×192 center crop and the same padding. YORO-FP uses the same weak crop-and-flip family as the full-data protocol before CLIP normalization.

A.1 DATASET COVERAGE

Table 8: Dataset coverage across the evaluated tasks and domains. Full and FS indicate evaluation under the full-data and few-shot settings used in this work, respectively.

Dataset	Task/domain	Full	FS
Flowers102 (Nilsback & Zisserman, 2008)	Fine-grained flowers	✓	✓
DTD (Cimpoi et al., 2014)	Texture attributes	✓	✓
UCF101 (Soomro et al., 2012)	Human actions (video clips)	✓	✓
Food101 (Bossard et al., 2014)	Food recognition	✓	✓
GTSRB (Stallkamp et al., 2012)	Traffic signs	✓	✓
EuroSAT (Helber et al., 2019)	Satellite land use	✓	✓
Oxford-IIIT Pets (Parkhi et al., 2012)	Fine-grained pets	✓	✓
Stanford Cars (Krause et al., 2013)	Fine-grained cars	✓	✓
SUN397 (Xiao et al., 2010)	Scene recognition	✓	✓
CIFAR-10 (Krizhevsky & Hinton, 2009)	Natural images	✓	✓
CIFAR-100 (Krizhevsky & Hinton, 2009)	Natural images	✓	✓
SVHN (Netzer et al., 2011)	Street-view digits	✓	
Caltech101 (Fei-Fei et al., 2004)	General objects	✓	✓
RESISC45 (Cheng et al., 2017)	Remote-sensing scenes	✓	✓
FGVC Aircraft (Maji et al., 2013)	Fine-grained aircraft		✓

Full-data suite. Following BLM and BLM+ (Cai et al., 2024a), the principal comparison uses the first 12 datasets in Table 8. They span fine-grained objects, textures, actions, scenes, remote sensing, traffic signs, and digits. The training-dynamics and mapping-update analyses add Caltech101 and RESISC45, giving the 14-task ablation suite. These two datasets are not part of the published-baseline comparison.

Few-shot CLIP suite. Following DVP (Cai et al., 2025b), the few-shot study uses the 11 datasets whose split column contains "FS". This suite deliberately combines fine-grained recognition with textures, scenes, actions, and remote sensing. We keep DVP's 20 attribute descriptions per class

fixed and treat the complete CLIP image–text similarity vector as BDM's response. Each class contributes 16 training examples and at most four validation examples, allowing us to test whether the same one-pass statistics remain useful when labels are scarce.

Model selection and evaluation. For each seed, we select $\rho \in \{0.01, 0.03, 0.1, 0.3, 0.5, 0.8\}$ on the validation split. The training statistics and validation responses are computed once and shared by all six candidates. Selection therefore adds only six analytic mapping solves and score evaluations, with no additional training traversal or frozen-model evaluation. Ties choose the smaller ρ. Official validation splits are used when available. CIFAR-10, CIFAR-100, GTSRB, and SVHN use a stratified 10% training holdout. YORO-FP evaluates the unrefined YORO predictor together with refinement epochs 1–20, then keeps the first checkpoint with the best validation accuracy. Test data are evaluated once after selection.

BDM uses empirical class priors and solves the regularized linear system directly. A small adaptive diagonal shift is added only when required for numerical positive definiteness. Training time excludes validation and test evaluation. YORO requires one labeled training-set traversal and no backward pass, whereas YORO-FP uses 21 traversals in total. All main-text results are macro-averaged across datasets and backbones.

Algorithm 1 Unified YORO and YORO-FP pseudocode. Within each minibatch, the pre-update response is reused for both BDM statistics and prompt optimization, so epoch-boundary mapping updates require no extra traversal.

Require: $\mathcal{D}_{\mathrm{tr}}$, $\mathcal{D}_{\mathrm{val}}$, frozen f_{pre}, $f_{\mathrm{in}}(\cdot \mid \theta_0)$, shrinkage ρ, epochs E, momentum β
Ensure: Validation-best $(\theta^\star, \omega^\star, b^\star)$
$\theta \leftarrow \theta_0$; reset accumulator $\mathcal{R}$
for all $(X, Y) \in \mathcal{D}_{\mathrm{tr}}$ **do** ▷ one traversal
  $Z \leftarrow f_{\mathrm{pre}}(f_{\mathrm{in}}(X \mid \theta))$
  $\mathcal{R} \leftarrow \textsc{UpdateStats}(\mathcal{R}, Z, Y)$
end for
$(\omega, b) \leftarrow \mathrm{BDM}(\mathcal{R}, \rho)$; save the unrefined YORO candidate
if $E = 0$ **then**
  return the saved candidate ▷ YORO
end if
for $t = 0, \ldots, E-1$ **do**
  reset $\mathcal{R}_t$
  for all $(X, Y) \in \mathcal{D}_{\mathrm{tr}}$ **do**
    $Z \leftarrow f_{\mathrm{pre}}(f_{\mathrm{in}}(X \mid \theta))$
    $\mathcal{R}_t \leftarrow \textsc{UpdateStats}(\mathcal{R}_t, \mathrm{stopgrad}(Z), Y)$
    $\theta \leftarrow \textsc{AdamStep}(\theta, \nabla_\theta \ell_{\mathrm{CE}}(\omega^\top Z + b, Y))$
  end for
  $(\widehat{\omega}, \widehat{b}) \leftarrow \mathrm{BDM}(\mathcal{R}_t, \rho)$
  $(\omega, b) \leftarrow \beta(\omega, b) + (1-\beta)(\widehat{\omega}, \widehat{b})$
  save (θ, ω, b) if validation accuracy strictly improves
end for
return the saved checkpoint ▷ YORO-FP

A.2 Visual Prompt Configurations

Padding and watermarking. Visual reprogramming modifies the input space using a trainable visual pattern while keeping the pretrained backbone frozen. Following the experimental settings of Cai et al. (2025a;b), we distinguish two common prompt geometries: padding-based and watermarking-based visual prompts. A padding-based prompt first resizes the downstream image to a smaller spatial resolution and places it at the center of the pretrained model's input canvas. The remaining border is occupied by trainable parameters. In contrast, a watermarking-based prompt resizes the downstream image to the pretrained model's full input resolution and directly overlays the trainable pattern within the image support. Thus, padding introduces trainable parameters outside the resized image, whereas watermarking directly perturbs the resized image itself.

Following the notation used in the visual reprogramming literature, we use *VP* to denote the visual-prompting baseline based on Bahng et al. (2022), which uses a watermarking-style prompt, and

AR to denote adversarial reprogramming, which uses a padding-style prompt. Although both approaches optimize input-space visual patterns, their prompt widths have different meanings. For watermarking, the width specifies the region of the original image on which the learned pattern is superimposed. For padding, the width specifies the trainable border created after resizing the downstream image.

Prompt size. Unless otherwise stated, our padding-based experiments use a 16-pixel-wide visual prompt on each side, following DVP (Cai et al., 2025b). For a 224×224 input, the downstream image is resized to 192×192 and centered in the input, leaving a 16-pixel trainable frame on each side. This resize-and-padding design also reduces the number of trainable input parameters, since only the surrounding frame is optimized while the resized image region remains fixed. We use this configuration as the default padding setting throughout our experiments.

The reported VP and AR results follow the corresponding experimental settings in Cai et al. (2025a;b), rather than our default 16-pixel configuration. In particular, VP uses a watermarking prompt, whose width specifies the region overlaid on the original image, whereas AR uses a padding prompt, whose width specifies the border surrounding the resized image. These values are therefore not directly comparable as prompt widths. More generally, the resize-and-padding configuration can use fewer trainable input parameters than a wider watermarking region because the central resized image does not contain trainable prompt parameters. Differences in prompt geometry and size across VP, AR, and our default 16-pixel padding setting follow the experimental protocols of the corresponding methods.

Results under legacy padding. Table 9 further evaluates YORO under the legacy padding protocol used by Cai et al. (2024a). Unlike the visual prompt configurations adopted in later work (Cai et al., 2025a;b), this protocol does not first resize downstream images to a common spatial resolution. Instead, the original downstream image is directly placed at the center of the pretrained input canvas and the remaining area is filled by the visual program. Consequently, the relative area occupied by the image depends on its native resolution. For low-resolution datasets, only a small central region may contain image content, producing a substantial mismatch with the spatial scale seen by the pretrained model.

Later visual reprogramming methods have largely moved away from this legacy configuration and instead operate on rescaled downstream images (Cai et al., 2025a;b). Rescaling provides a more controlled input interface because downstream images occupy a consistent fraction of the pretrained input across datasets. This distinction also helps explain why YORO performs slightly better under our watermark setting: the resized input provides a more informative and spatially consistent source response, whereas legacy padding can substantially reduce the effective image content for low-resolution datasets. Since YORO itself requires no prompt training, its prediction is determined directly by the source response induced by the given visual program and the proposed output mapping.

Importantly, YORO remains strong even under this legacy padding protocol. With ResNet–18, YORO achieves 54.7% average accuracy, compared with 45.3% for BLM and 46.7% for BLM+, corresponding to gains of 9.4 and 8.0 percentage points, respectively. With ResNeXt–101–32x8d, YORO reaches 67.4%, improving over BLM and BLM+ by 14.2 and 13.6 points. These gains are obtained without iterative prompt optimization, even though the legacy padding scheme does not resize downstream images to a common spatial scale and can therefore provide a less consistent source response across datasets. The improvement under the same input parameterization used by BLM indicates that YORO's advantage primarily comes from its stronger use of the pretrained model's output response rather than from a more favorable visual prompt geometry. Together with the results under our default resized visual prompt settings, this also shows that YORO is effective across different input reprogramming configurations.

B Derivation and streaming equivalence

Affine Bayes rule. Under Equation (4),

$$\log p(z \mid c) + \log \pi_c = -\tfrac{1}{2}(z - \mu_c)^\top \Sigma^{-1}(z - \mu_c) + \log \pi_c + \text{const.}$$

Table 9: **Comparison under the legacy padding setting** (top-1 accuracy, %). Following Cai et al. (2024a), downstream images are directly padded to the pretrained input size without resizing. Despite this legacy input configuration, YORO substantially improves over BLM and BLM+ across both pretrained backbones.

	ResNet–18 (ImageNet–1K)								ResNeXt–101–32x8d (Instagram)						
Dataset	RLM	FLM	ILM	BLM	BLM+	Deep	YORO	YORO-FP	FLM	ILM	BLM	BLM+	Deep	YORO	YORO-FP
Flowers102	11.0	20.0	27.9	44.4	50.1	76.7	80.1	**80.7**	22.5	27.9	31.5	30.1	85.2	89.3	**89.4**
DTD	16.3	32.4	35.3	42.0	43.9	49.1	**55.5**	53.2	40.3	41.4	47.8	49.4	64.0	**67.4**	66.9
UCF101	6.6	18.9	23.9	30.9	32.0	46.0	**54.1**	52.9	41.9	43.1	48.3	50.1	68.3	68.4	**69.0**
Food101	3.8	12.8	14.8	23.2	25.1	34.1	**41.7**	35.5	20.5	23.0	29.6	31.4	58.7	**60.9**	58.2
GTSRB	46.1	45.5	52.0	54.8	54.3	63.1	47.2	**57.7**	56.2	59.9	62.9	63.0	74.4	58.2	**67.9**
EuroSAT	82.4	83.8	85.2	86.7	86.7	92.4	90.7	**91.1**	87.8	86.2	87.6	88.3	93.2	91.0	**93.1**
OxfordPets	9.3	62.9	65.4	69.8	70.6	73.0	**77.0**	76.6	76.8	78.9	82.4	83.0	91.8	**92.7**	**92.7**
StanfordCars	0.9	2.7	4.5	5.4	7.7	14.3	22.5	**22.7**	4.6	7.0	8.3	9.3	50.5	**53.2**	52.5
SUN397	1.0	10.4	13.0	16.2	18.7	26.3	**40.4**	35.5	21.6	23.7	30.1	32.0	51.5	**55.0**	54.1
CIFAR10	63.0	65.7	65.5	66.7	66.8	72.1	60.7	**66.9**	80.3	81.7	**82.2**	**82.2**	83.4	73.6	79.9
CIFAR100	12.9	18.1	24.8	29.6	30.6	46.7	36.2	**38.1**	39.7	45.9	47.8	47.8	56.2	46.7	**52.0**
SVHN	73.5	73.1	**75.2**	74.5	74.2	82.1	50.8	72.9	79.0	**81.4**	79.8	79.3	85.7	52.4	81.2
Average	27.2	37.2	40.6	45.3	46.7	56.3	54.7	**57.0**	47.6	50.0	53.2	53.8	71.9	67.4	**71.4**

Expanding the quadratic and dropping $-\frac{1}{2}z^\top\Sigma^{-1}z$, which is common to all classes, yields the score in Equation (5) and the affine parameters in Equation (9).

Proof of Proposition 1. For class c, let $\widehat{\mu}_c = s_c/n_c$. Its within-class scatter is

$$\sum_{i:y_i=c}(z_i-\widehat{\mu}_c)(z_i-\widehat{\mu}_c)^\top = \sum_{i:y_i=c} z_i z_i^\top - \frac{s_c s_c^\top}{n_c}.$$

Summing over classes gives $Q - \sum_c s_c s_c^\top/n_c$, exactly the numerator in Equation (7). Counts also recover the batch class means and priors. Applying the same deterministic shrinkage and linear solve therefore produces identical (ω_0, b_0). □

B.1 Optional exact low-rank CLIP implementation

BDM is defined entirely in the response space and does not require access to intermediate model features. When CLIP visual features are available, however, their low-dimensional structure can be exploited as an exact computational shortcut. If only the response vectors are observable, BDM can instead be fit directly from the response-space statistics in Equation (6).

Let $K = mk_\mathrm{T}$ and stack the normalized CLIP attribute embeddings as $T \in \mathbb{R}^{K\times d}$. For a normalized visual feature $v \in \mathbb{R}^d$, the response in Equation (2) is $z = \alpha T v$, where $\alpha = \tau_\mathrm{C}$. Let $M_v \in \mathbb{R}^{d\times k_\mathrm{T}}$ contain the target-class visual-feature means and let $C_v \in \mathbb{R}^{d\times d}$ be their pooled within-class covariance. The corresponding response-space statistics are

$$M_z = \alpha T M_v, \qquad \widehat{\Sigma}_z = \alpha^2 T C_v T^\top, \qquad \tau = \frac{\alpha^2}{K}\operatorname{tr}(C_v T^\top T). \tag{12}$$

Writing $a = \rho\tau + \delta$ and $\beta = 1-\rho$, the dense response-space BDM system admits the exact solution

$$\omega = \frac{\alpha}{a} T\left(I_d + \frac{\beta\alpha^2}{a} C_v T^\top T\right)^{-1} M_v. \tag{13}$$

This is a direct matrix-identity rewrite of $(aI_K + \beta\alpha^2 T C_v T^\top)\omega = \alpha T M_v$ and therefore produces exactly the same affine mapping as dense response-space BDM, rather than a low-rank approximation.

In our CLIP implementation, we exploit this identity by accumulating class sums and the second moment in d dimensions, solving the $d \times d$ system in Equation (13), and forming the final $K \times k_\mathrm{T}$ mapping once. This use of visual features is purely an implementation shortcut and does not change the BDM formulation. Computation uses double-precision statistics, a symmetrized C_v, and the adaptive diagonal floor δ from Equation (8).

Complexity. For generic classifier logits, accumulation costs $\mathcal{O}(nk_\mathrm{S}^2)$ arithmetic and stores $\mathcal{O}(k_\mathrm{S}^2 + k_\mathrm{T}k_\mathrm{S})$ values; the final dense solve costs $\mathcal{O}(k_\mathrm{S}^3 + k_\mathrm{S}^2 k_\mathrm{T})$. Both image-classifier backbones in

Table 10: **Efficiency–accuracy comparison of YORO and YORO-FP.** Accuracy is the three-seed macro-average over 14 full-data tasks for ResNet–18 padding, or over 11 few-shot datasets and four CLIP backbones. Relative time is normalized by the corresponding YORO run. All timings use one NVIDIA A100 and exclude validation and test evaluation. "No refinement" reports the fraction of YORO-FP runs in which validation selects the original YORO checkpoint.

Protocol	Method	Accuracy (%)	Δ (%)	No refinement	Passes / time
ResNet18 (P)	YORO	71.0±0.0	–	–	1 / 1.0×
ResNet18 (P)	YORO-FP	**71.6**±0.1	+0.6	54.8%	21 / 20.8×
FS-CLIP	YORO	**77.2**±0.1	–	–	1 / 1.0×
FS-CLIP	YORO-FP	77.0±0.3	−0.2	58.3%	21 / 17.5×

our experiments have $k_{\mathrm{S}} = 1{,}000$, so the double-precision second moment occupies approximately 7.6 MiB. For CLIP, the exact formulation above stores $\mathcal{O}(d^2 + k_{\mathrm{T}}d + Kd)$ values and solves a $d \times d$ system. The largest response in our suite is SUN397 with $K = 20 \times 397 = 7{,}940$; a dense $K \times K$ double-precision moment would contain 63.0 million entries (approximately 481 MiB), whereas our implementation never materializes it. The complete one-pass timings, including the analytic fit, are reported in Table 10. All costs are independent of the number of prompt optimization epochs because YORO has none.

Reproducibility. Our implementation keeps the backbone frozen and enforces one training-set traversal for YORO. We will release the code, configurations, per-seed results, and evaluation scripts.

C Detailed results and ablations

Compute and accuracy. Table 10 shows diminishing returns from prompt optimization. YORO-FP uses 21 labeled passes and about 18–21× the training time of YORO. This additional computation raises the full-data mean by 0.6%, but lowers the few-shot CLIP mean by 0.2%. Validation selects the original YORO checkpoint in 54.8% and 58.3% of runs, respectively. YORO is therefore a strong accuracy–efficiency default, while YORO-FP provides an optional refinement when validation favors further prompt training.

C.1 Additional covariance and response-selection ablations

The additional covariance and response-selection comparisons are summarized in Table 11. These supplementary ablations follow the same frozen-response protocol as the main FS-CLIP experiments.

Table 11: **Covariance and response-selection ablations.** Results report mean ± sample standard deviation over three seeds (%), averaged across 11 datasets under the 16-shot FS-CLIP protocol. Overall further averages the four CLIP backbones.

Backbone	BDM	Conditional 95%	Adaptive	OAS	RBLW	Empirical LDA
ResNet–50	$\mathbf{74.4 \pm 0.2}$	74.1 ± 0.3	74.0 ± 0.3	73.3 ± 0.2	73.3 ± 0.2	69.2 ± 0.2
ResNet–101	$\mathbf{76.8 \pm 0.2}$	76.6 ± 0.3	76.6 ± 0.3	76.1 ± 0.2	76.1 ± 0.2	72.5 ± 0.4
ViT-B/32	$\mathbf{76.2 \pm 0.3}$	76.1 ± 0.4	76.1 ± 0.4	75.5 ± 0.0	75.5 ± 0.0	71.6 ± 0.1
ViT-B/16	$\mathbf{81.2 \pm 0.1}$	81.1 ± 0.1	81.0 ± 0.2	80.7 ± 0.1	80.7 ± 0.1	77.5 ± 0.5
Overall	$\mathbf{77.2 \pm 0.1}$	77.0 ± 0.0	76.9 ± 0.1	76.4 ± 0.1	76.4 ± 0.1	72.7 ± 0.2

Covariance regularization is more important than response pruning. On FS-CLIP, empirical LDA, which uses the unregularized shared sample covariance, reaches 72.7%, while default BDM reaches 77.2%. OAS and RBLW (Chen et al., 2010; Ledoit & Wolf, 2004) also improve substantially over unregularized LDA, reaching 76.4% without validation-based selection of the shrinkage coefficient.

Conditional 95% uses training labels to greedily select response coordinates according to their additional class-discriminative information conditioned on the coordinates already selected, and retains a fixed 95% of the response dimensions. Adaptive instead jointly selects the retention ratio, including the option to keep all coordinates, and the covariance-shrinkage coefficient using validation data for each dataset, backbone, and seed. It retains 97.3% of the coordinates on average, indicating that validation generally favors keeping most of the response.

Neither strategy improves on full-response BDM. Conditional 95% reaches 77.0%, and Adaptive reaches 76.9%, compared with 77.2% for BDM. We therefore retain the complete response by default and use shrinkage to stabilize covariance estimation rather than explicitly pruning response coordinates.

C.2 Contextual comparison with efficient CLIP adaptation

Table 12: **Computationally efficient CLIP adaptation on ten overlapping datasets with a ResNet–50 backbone** (accuracy %). We recompute each average over OxfordPets, Flowers102, FGVC Aircraft, DTD, EuroSAT, StanfordCars, Food101, SUN397, Caltech101, and UCF101. Tip-Adapter is reported by Wang et al. (2024b), while CARPRT, MPE, and WPE are reported by Dong et al. (2026). YORO is our three-seed result. All methods avoid iterative backbone updates, so the table provides a contextual efficiency-oriented comparison despite differences in data access.

Method	Pets	Flowers	Aircraft	DTD	EuroSAT	Cars	Food	SUN	Caltech	UCF	Avg.
CARPRT	85.7	65.6	16.9	41.3	36.8	56.4	76.9	61.3	88.5	63.7	59.3
MPE	76.0	57.0	16.1	41.7	30.3	55.7	75.5	59.3	86.4	60.1	55.8
WPE	78.4	58.8	16.1	40.9	30.7	56.0	76.2	59.7	86.7	61.5	56.5
Tip-Adapter	**88.1**	89.9	29.8	60.9	70.5	66.8	**77.8**	66.9	90.2	70.6	71.2
YORO	87.5	**92.6**	**36.2**	**64.4**	**82.9**	**68.6**	72.8	**67.6**	**91.3**	**75.1**	**73.9**

The broader comparison in Table 12 positions YORO among computationally efficient CLIP adaptation methods. On the ten-dataset overlap, YORO exceeds Tip-Adapter by 2.8% while using the complete attribute-prompt response as its only source representation. This comparison is intended to characterize efficiency-oriented alternatives rather than merge incompatible training protocols. YORO's advantage comes from its discriminant output mapping, not from an expensive feature adapter or prolonged backbone training.

Table 13: **SVHN isolates the benefit of input adaptation.** With BDM frozen, optimizing only the visual prompt improves every tested backbone–input setting. Accuracy is mean ± sample standard deviation over three seeds (%).

Backbone	Input	YORO	Frozen BDM	Δ (%)	YORO-FP
ResNet–18	Padding	72.3±0.1	76.0±0.2	+3.7	**77.0**±0.3
ResNeXt–101–32x8d	Padding	56.5±0.0	63.3±0.3	+6.8	**68.6**±0.6
ResNet–18	Watermark	60.6±0.1	83.4±0.2	+22.8	**84.7**±0.2

Why SVHN is different. With BDM frozen, prompt training raises SVHN accuracy by 3.7%, 6.8%, and 22.8% across the three settings in Table 13. The visual prompt therefore supplies most of the improvement even when the initial label mapping is held fixed. The particularly large watermarking gain shows that the benefit cannot be explained by repeatedly refitting the output mapping: input adaptation is reshaping the frozen response geometry itself. YORO-FP adds another 1.0%, 5.3%, and 1.3% by tracking that moving geometry without replacing the mapping abruptly. Together with the aggregate result in Table 10, SVHN identifies the exception that motivates YORO-FP: prompt training is valuable when the source responses and target classes are initially poorly aligned, rather than as a universal default.

Table 14: **Variants of YORO-FP mapping strategies.** Δ is measured relative to YORO, with 95% bootstrap confidence intervals over evaluation settings. W/T/L counts datasets for full data and dataset–backbone pairs in the few-shot CLIP setting. Refinement depth is the number of additional prompt-training epochs selected on the validation set. A depth of 0 corresponds to the original YORO checkpoint. A median depth of 0 means that validation selects YORO without further prompt training in at least half of the runs. The last column reports the fraction of runs for which validation retains the original YORO predictor. Frozen, Refresh, and Momentum use $\beta = 1$, 0, and 0.9, respectively.

Protocol	Strategy	Accuracy (%)	Δ (%)	95% CI	W/T/L	Refinement depth	No refinement
Full data	Frozen	71.1 ± 0.0	$+0.2$	$[-0.2, +0.8]$	2/7/5	4.8 ± 7.7	69.0%
	Refresh	71.1 ± 0.0	$+0.1$	$[-0.2, +0.7]$	2/10/2	2.2 ± 5.5	81.0%
	Momentum	$\mathbf{71.6 \pm 0.1}$	$\mathbf{+0.6}$	$[\mathbf{0.0}, \mathbf{+1.4}]$	5/6/3	6.5 ± 7.8	54.8%
FS-CLIP	Frozen	76.8 ± 0.1	-0.3	$[-0.5, -0.2]$	2/13/29	3.2 ± 5.6	66.7%
	Refresh	76.9 ± 0.1	-0.3	$[-0.5, 0.0]$	6/14/24	3.4 ± 5.7	60.6%
	Momentum	$\mathbf{77.0 \pm 0.3}$	$\mathbf{-0.2}$	$[\mathbf{-0.4}, \mathbf{0.0}]$	8/8/28	4.2 ± 6.3	58.3%

Selection behavior. Table 14 separates the value of prompt refinement from the way BDM follows the changing responses. In full data, Frozen and Refresh improve the mean by only 0.2% and 0.1%, whereas Momentum reaches $+0.6\%$ and produces the largest number of wins. This supports gradual tracking: an exponential moving average adapts the mapping when the prompt changes the response geometry, while avoiding the instability of replacing it after every epoch. The gain remains selective, however, because Momentum still retains the original YORO predictor in 54.8% of runs.

The few-shot regime is different. All three update rules reduce the aggregate accuracy and losses outnumber wins, with no refinement selected in 58.3–66.7% of runs. Limited supervision leaves little evidence for reliably updating both the prompt and its label mapping. Thus, the two protocols yield a consistent decision rule: begin with the strong one-pass YORO predictor, and spend the few-pass budget only when held-out validation demonstrates that response-space adaptation transfers.

Table 15: **Per-dataset full-data mapping-update ablation** (accuracy %). Each entry is the three-seed mean ± sample standard deviation. P/W denote padding/watermarking. Bold marks the highest mean within each dataset and setting. These results underlie the averages in Table 6.

Dataset	YORO	Frozen	Refresh	Momentum
(a) ResNet–18 / P				
Caltech101	92.0±0.0	92.0±0.0	92.0±0.0	**92.1**±0.1
CIFAR-10	**86.2**±0.1	86.0±0.4	**86.2**±0.1	**86.2**±0.1
CIFAR-100	**64.8**±0.1	**64.8**±0.1	**64.8**±0.1	**64.8**±0.1
DTD	59.0±0.0	59.0±0.0	59.1±0.2	**59.6**±0.5
EuroSAT	**92.8**±0.0	92.5±0.5	92.5±0.5	92.5±0.2
Flowers102	**88.0**±0.0	87.8±0.3	**88.0**±0.0	87.9±0.3
Food101	**49.8**±0.0	**49.8**±0.0	**49.8**±0.0	**49.8**±0.0
GTSRB	79.0±0.1	79.4±0.3	79.0±0.1	**80.2**±0.4
OxfordPets	**86.9**±0.0	**86.9**±0.0	**86.9**±0.0	**86.9**±0.0
RESISC45	**78.8**±0.0	**78.8**±0.0	**78.8**±0.0	78.7±0.2
StanfordCars	34.9±0.0	33.9±0.2	33.9±0.2	**37.1**±0.1
SUN397	**47.8**±0.0	**47.8**±0.0	**47.8**±0.0	**47.8**±0.0
SVHN	72.3±0.1	76.0±0.2	75.3±0.4	**77.0**±0.3
UCF101	**61.3**±0.0	**61.3**±0.0	**61.3**±0.0	**61.3**±0.0
Average	71.0±0.0	71.1±0.0	71.1±0.0	**71.6**±0.1
(b) ResNet–18 / W				
Caltech101	**92.4**±0.0	**92.4**±0.0	**92.4**±0.0	**92.4**±0.0
CIFAR-10	**86.9**±0.1	**86.9**±0.1	**86.9**±0.1	**86.9**±0.1
CIFAR-100	**64.8**±0.1	**64.8**±0.1	**64.8**±0.1	**64.8**±0.1
DTD	**62.5**±0.0	**62.5**±0.0	61.3±1.2	**62.5**±0.0
EuroSAT	**93.2**±0.0	**93.2**±0.0	**93.2**±0.0	**93.2**±0.0
Flowers102	**88.8**±0.0	**88.8**±0.0	**88.8**±0.0	**88.8**±0.0
Food101	**52.8**±0.0	**52.8**±0.0	**52.8**±0.0	**52.8**±0.0
GTSRB	75.9±0.1	79.5±0.6	**85.2**±0.9	84.5±0.6
OxfordPets	**88.5**±0.0	88.1±0.6	**88.5**±0.0	**88.5**±0.0
RESISC45	**79.5**±0.0	**79.5**±0.0	**79.5**±0.0	**79.5**±0.0
StanfordCars	37.8±0.0	37.8±0.0	37.8±0.0	**38.6**±0.5
SUN397	**52.0**±0.0	**52.0**±0.0	**52.0**±0.0	**52.0**±0.0
SVHN	60.6±0.1	83.4±0.2	82.3±0.4	**84.7**±0.2
UCF101	**65.1**±0.0	**65.1**±0.0	**65.1**±0.0	**65.1**±0.0
Average	71.5±0.0	73.3±0.1	73.6±0.1	**73.9**±0.0
(c) ResNeXt–101–32x8d / P				
Caltech101	**96.5**±0.0	96.2±0.1	96.4±0.2	96.4±0.0
CIFAR-10	**95.3**±0.0	**95.3**±0.0	**95.3**±0.0	**95.3**±0.0
CIFAR-100	**76.3**±0.1	**76.3**±0.1	**76.3**±0.1	**76.3**±0.1
DTD	**70.3**±0.0	**70.3**±0.0	**70.3**±0.0	**70.3**±0.0
EuroSAT	93.0±0.0	92.8±0.4	93.0±0.2	**93.4**±0.3
Flowers102	94.0±0.0	94.0±0.0	94.0±0.0	**94.3**±0.1
Food101	**70.3**±0.0	**70.3**±0.0	**70.3**±0.0	**70.3**±0.0
GTSRB	79.3±0.1	79.3±0.1	79.3±0.1	**79.6**±0.5
OxfordPets	94.8±0.0	94.9±0.1	**95.2**±0.1	95.1±0.1
RESISC45	82.5±0.0	82.3±0.1	82.5±0.0	**83.2**±0.3
StanfordCars	71.1±0.0	70.9±0.2	70.0±0.2	**72.2**±0.3
SUN397	**61.9**±0.0	61.8±0.2	**61.9**±0.0	**61.9**±0.0
SVHN	56.5±0.0	63.3±0.3	68.1±0.7	**68.6**±0.6
UCF101	74.9±0.0	74.8±0.1	75.1±0.2	**75.4**±0.3
Average	79.8±0.0	80.2±0.0	80.6±0.1	**80.9**±0.1

D NOTATION

Tables 16 and 17 list the symbols used in the method and result tables. Superscripts S and T always mean source and target.

Table 16: Symbols used in YORO and YORO-FP.

	Spaces and Indexing
$\mathcal{X}^{\mathrm{S}}, \mathcal{X}^{\mathrm{T}}$	Source and target input spaces.
$\mathcal{Y}^{\mathrm{S}}, \mathcal{Y}^{\mathrm{T}}$	Source and target label spaces.
k_{S}	Number of source-response coordinates.
k_{T}	Number of target classes.
$[q]$	Index set $\{1, \ldots, q\}$.
i	Example index.
c	Target-class index.
j	Attribute-prompt index.
$\mathbb{R}^d$	Real vector space of dimension d.
$\langle \cdot, \cdot \rangle$	Inner product.
$(x_i^{\mathrm{T}}, y_i^{\mathrm{T}})$	Target example and label.
n	Sample count.
$\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{val}}$	Training and validation sets.
	BDM and Output Mapping
μ_c	Class mean for target class c.
π_c	Class prior for target class c.
Σ	Shared response covariance.
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and covariance Σ.
$p(\cdot)$	Probability model.
$\widehat{\mu}_c$	One-pass estimate of the class mean.
$\widehat{\pi}_c$	One-pass estimate of the class prior.
$\widehat{\Sigma}$	Empirical covariance estimate.
ρ	Covariance shrinkage coefficient.
δ	Numerical diagonal shift.
$\widehat{\Sigma}_\rho$	Regularized covariance used by BDM.
I	Identity matrix.
$\mathrm{tr}(\cdot)$	Matrix trace.
$\mathbf{1}[\cdot]$	Indicator function.
M	Class-mean matrix $[\widehat{\mu}_1, \ldots, \widehat{\mu}_{k_{\mathrm{T}}}]$.
ω	Affine label-mapping weights.
b	Affine label-mapping bias.
$f_{\mathrm{out}}^{\omega,b}$	Affine output mapping.
g_c	BDM score for class c.
$\widehat{\omega}, \widehat{b}$	BDM candidate from the current pass. Subscript 0 marks the first pass.
n_c	Number of responses assigned to class c.
s_c	Sum of responses assigned to class c.
S	Class-sum matrix.
Q	Global uncentered second moment.
$\mathcal{R}, \mathcal{R}_t$	Running BDM statistics $(\mathbf{n}, S, Q)$ for the first pass and epoch t.

Table 17: Symbols used in YORO and YORO-FP (continued).

Reprogramming and Optimization	
f_{pre}	Frozen classifier or CLIP encoders.
$f_{\mathrm{in}}(\cdot \mid \theta)$	Input reprogramming function with parameters θ.
θ_0	Prompt initialization.
$\mathcal{L}_{\mathrm{FP}}$	YORO-FP objective.
ℓ_{CE}	Cross-entropy loss.
∇_θ	Gradient with respect to prompt parameters.
X	Minibatch images.
Y	Minibatch labels.
Z	Minibatch response vectors.
$z_i(\theta)$	Response vector for example i.
$\mathrm{stopgrad}(\cdot)$	Blocks gradients through its argument.
UPDATESTATS	Updates the streaming class statistics.
BDM	Analytic discriminant mapping fit.
ADAMSTEP	One Adam optimizer update of the prompt.
E	YORO-FP epoch budget.
t	Epoch index.
β	Momentum weight.
$(\cdot)^\star$	Validation-selected checkpoint.
CLIP and Evaluation	
$\mathcal{A}$	CLIP attribute-prompt bank.
$a_{c,j}$	Prompt j for class c.
m	Number of prompts per class.
ϕ_{I}	Frozen CLIP image encoder.
ϕ_{T}	Frozen CLIP text encoder.
$\bar{\phi}$	Normalized CLIP output.
τ_{C}	Frozen CLIP logit scale.
Δ	Accuracy change in percent.
CI	Confidence interval.
W/T/L	Win, tie, and loss counts.