CAN VISUAL PROMPTS INHERIT FULL FINE-TUNING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained vision transformers have demonstrated strong transfer ability and are increasingly shared across many downstream tasks. This trend has driven growing interest in parameter-efficient adaptation, which aims to adapt one frozen backbone to each task with only a few task-specific parameters. Visual prompt tuning (VPT) is a representative approach that learns a small set of prompt tokens and a classifier for each task. However, existing VPT methods learn prompts only on the frozen pretrained backbone and therefore cannot use the task-specific adaptation that full fine-tuning (FT) provides. These limitations leave compact prompts well below the accuracy their architecture can reach. Therefore, this paper proposes *gradual withdrawal*. Specifically, gradual withdrawal first fine-tunes a copy of the backbone on the target task, and then trains visual prompts while smoothly returning the backbone to its original pretrained weights. The deployed model is exactly that of direct VPT: the original backbone plus a prompt and a classifier per task. Crucially, this ordered return lets the prompt track a sequence of nearby solutions, as our local tracking analysis explains. On supervised ViT-B/16, gradual withdrawal raises mean accuracy from 69.4% to 76.2% across 19 VTAB-1k tasks and from 89.1% to 92.3% across five fine-grained classification tasks, although the FT model itself scores only 65.6% and 88.5%. Extensive experiments show that gradual withdrawal outperforms matched distillation, abrupt-return, and shuffled-path controls, with gains that hold across eight backbones spanning 86M to 1.1B parameters. More broadly, these results establish the training path as a new axis of parameter-efficient adaptation: a compact model can learn from far more than it deploys, so better training alone can make it stronger.

1 INTRODUCTION

A pretrained vision transformer (ViT) learns general visual representations from large-scale data and is then adapted to downstream tasks (Dosovitskiy et al., 2021). By transferring these representations to a wide range of recognition problems, pretrained ViTs have become a core component of many modern vision systems. Full fine-tuning (FT) adapts all backbone weights to each task, which typically requires storing a separate backbone for every task.

As these systems naturally shift toward serving many tasks from one shared model, parameter-efficient adaptation emerges as a critical research topic. Visual prompt tuning (VPT) keeps the backbone frozen and learns only a few task-specific tokens and a classifier (Jia et al., 2022), so the cost of adding a task stays small as the number of tasks grows. Improving the accuracy of such compact models is a key step toward fully unlocking the practical value of shared vision backbones.

However, accurate prompt tuning remains challenging. The accuracy of a prompt depends on two distinct factors: what the prompt can express and how training finds that solution. The usual comparison between FT and VPT mixes these factors, since the methods differ in both their deployed models and their optimization problems, and their relative strengths vary across tasks (Han et al., 2024; Mai et al., 2025). Existing methods improve initialization and optimization (Wang et al., 2024; Li et al., 2024; Wang et al., 2026a) or enrich prompt expressiveness (Le et al., 2026), but *they all train the prompt on the same frozen pretrained backbone*. Crucially, existing work transfers full fine-tuning into prompts only through its outputs, and does not ask whether a prompt can learn inside the fine-tuned weights themselves when none of them are kept at deployment. This motivates the central question of this paper: *Can visual prompts inherit full fine-tuning without enlarging the deployed model?*

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.

To answer this question, we propose *gradual withdrawal*, based on the insight that *training can use adaptation that deployment does not retain*. Specifically, we first fine-tune a copy of the pretrained backbone on the target task. We then optimize prompts and a classifier while gradually removing the task-specific weight update, and a final phase refines them on the exact original backbone (Figure 1). The FT checkpoint serves only as a temporary training environment; deployment retains one shared pretrained backbone and each task's prompt and classifier.

Extensive experiments show why this design matters. On supervised ViT-B/16, gradual withdrawal improves VTAB-1k accuracy from 69.4% to 76.2% and fine-grained classification accuracy from 89.1% to 92.3%, surpassing six recent prompt methods with the same deployed model. The FT checkpoint itself scores only 65.6% and 88.5%, below direct VPT on both benchmarks, so *its value as a training guide exceeds its value as a standalone predictor*. Matched controls identify the source of this value: ordered withdrawal exceeds abrupt return by 2.8 VTAB points, shuffled withdrawal by 4.0 points, and fixed-teacher logit distillation (Hinton et al., 2015) by 5.5 points.

Our contributions are threefold. First, we provide a fixed-deployment view of the VPT–FT gap, which separates what a prompt can express from what training attains (Eq. (1)) and turns the training path into a controlled variable for visual adaptation. Second, we propose gradual withdrawal, a simple procedure that transfers full-model adaptation into prompts while deploying exactly the VPT architecture on the original backbone, together with a local tracking analysis that explains why the ordered path works. Third, extensive experiments across eight backbones and three pretraining families show VTAB gains of 4.1 to 8.1 points that persist with 100 prompt tokens per layer, and diagnostics show that withdrawal reaches a flatter basin and yields prompts that reorganize patch interactions rather than absorb attention. More broadly, our work establishes the training path as a new axis of parameter-efficient adaptation: additional training compute can make compact, shared models stronger without enlarging the model that is deployed.

2 Related Work

Visual Prompt Tuning. VPT adapts frozen vision transformers with shallow or deep learnable tokens (Jia et al., 2022). Later methods improve token interactions, frequency structure, prompt placement, and cross-layer communication (Han et al., 2023; Das et al., 2023; Zeng et al., 2024; Zhou et al., 2024; Huang et al., 2025; Shang et al., 2025). Recent work develops prompt experts, interpretable concepts, cascaded semantics, layer-specific fusion, and information allocation (Le et al., 2026; Wang et al., 2026b; Xiao et al., 2026a;b; Li et al., 2026); PAE improves cross-layer optimization dynamics (Wang et al., 2026a). Other work improves prompt initialization (Wang et al., 2024; Park & Chung, 2025), adapts prompts to the data distribution or to individual instances (Huang et al., 2023; Xiao et al., 2025a), targets self-supervised backbones (Yoo et al., 2023), or learns prompts in pixel space (Jin et al., 2025). All of these methods train the prompt on the frozen pretrained backbone. We study a different and complementary degree of freedom: the sequence of backbones through which a fixed prompt architecture is learned.

Training Paths and Full-Model Guidance. VPT and FT offer complementary strengths across tasks and distributions (Han et al., 2024). Progressive language-prompt training (Huang et al., 2022) and low-rank compression (Hwang et al., 2024) use staged adaptation in different settings. Gradual withdrawal instead exposes visual prompts to a task-specific FT update and removes it before deployment. Fixed-teacher distillation (Hinton et al., 2015) provides a complementary control, transferring endpoint predictions while the student's backbone stays fixed. Our schedule is instead a continuation method (Allgower & Georg, 2003; Wortsman et al., 2022; Ilharco et al., 2023): it solves a sequence of nearby problems whose last member is the deployed problem, so the solution of each step initializes the next.

Parameter-Efficient Fine-Tuning. Beyond prompts, parameter-efficient methods adapt a frozen backbone through side networks (Zhang et al., 2020), bias terms (Zaken et al., 2022), bottleneck adapters (Houlsby et al., 2019; Chen et al., 2022), low-rank updates (Hu et al., 2022), or combinations of these designs (Dong et al., 2023; Luo et al., 2025). These methods change where task parameters enter the network. Gradual withdrawal is complementary: it changes how a fixed set of task parameters is trained. We focus on visual prompts because both their deployment savings and their accuracy gap to full fine-tuning are well documented (Jia et al., 2022; Han et al., 2024); the same procedure could train adapters or low-rank modules, which we leave to future work.

3 PRELIMINARIES

Models and endpoint. Let $D_{\mathrm{tr}} = \{(x_i, y_i)\}_{i=1}^n$ be a downstream training set. A pretrained vision transformer has backbone weights θ_0 (Dosovitskiy et al., 2021). Deep visual prompt tuning inserts trainable tokens $P = (P_\ell)_{\ell \in \mathcal{I}}$ at fixed transformer layers and trains a classification head g_ϕ while freezing the backbone (Jia et al., 2022). Write $z = (P, \phi)$ and $h_{\theta,z}(x) = g_\phi(f_\theta(x; P))$ for its logits. The empirical cross-entropy is $\mathcal{L}(z, \theta) = n^{-1} \sum_{i=1}^n \ell_{\mathrm{CE}}(h_{\theta,z}(x_i), y_i)$. Full fine-tuning starts from the *same* θ_0 and trains a prompt-free backbone and its own head, yielding $(\theta_{\mathrm{FT}}, \phi_{\mathrm{FT}})$. We use only the backbone update $\Delta\theta = \theta_{\mathrm{FT}} - \theta_0$; the FT head is never interpolated. Every variant in the matched path comparison uses the same insertion sites, token count, head architecture, and data split.

What the performance gap can mean. For population cross-entropy $\mathcal{R}(\theta, z)$, let $\mathcal{R}_P^* = \inf_z \mathcal{R}(\theta_0, z)$, let z_D be a directly trained VPT solution, and let $\mathcal{R}_{\mathrm{FT}}$ be the risk of the prompt-free FT model. The gap between direct VPT and FT then splits into two terms:

$$\mathcal{R}(\theta_0, z_D) - \mathcal{R}_{\mathrm{FT}} = \underbrace{\mathcal{R}(\theta_0, z_D) - \mathcal{R}_P^*}_{\text{attainment gap}} + \underbrace{\mathcal{R}_P^* - \mathcal{R}_{\mathrm{FT}}}_{\text{best-class difference}} . \tag{1}$$

The first term is the trained prompt's excess population risk within its deployment class; the second compares that class's best risk with the achieved FT reference and can have either sign (Jia et al., 2022; Han et al., 2024). Holding the prompt class fixed therefore lets our path comparisons isolate changes in the attainment gap, that is, in how close training gets to the best prompt.

4 METHOD

4.1 WITHDRAWING THE FULL-TUNING UPDATE

Figure 1 summarizes the training path. Rather than training on a single backbone, we expose the prompt to a continuous family of frozen backbones,

$$\theta(\alpha) = \theta_0 + \alpha\Delta\theta, \qquad \alpha \in [0, 1], \tag{2}$$

and optimize only the prompt parameters z on $F(z, \alpha) = \mathcal{L}(z, \theta(\alpha))$:

$$\min_z F(z, \alpha_t) = \frac{1}{n} \sum_{i=1}^n \ell_{\mathrm{CE}}\big(g_\phi(f_{\theta(\alpha_t)}(x_i; P)), y_i\big) . \tag{3}$$

The FT checkpoint and $\Delta\theta$ remain fixed. With T_w warm-up, T_d withdrawal, and T_f final updates, we use $\alpha = 1$ for the first T_w steps, $\alpha(s_j) = [1 + \cos(\pi s_j)]/2$ with $s_j = (j + 1)/(T_d + 1)$ for $j = 0, \ldots, T_d - 1$, then $\alpha = 0$ for T_f steps. Warm-up fits the prompt and classifier to the task-adapted backbone; withdrawal lets them adjust as the FT update is removed; the final phase optimizes the exact deployed model $h_{\theta_0, z_{\mathrm{final}}}$. Prompt insertion changes the attention slots, so warm-up optimizes the prompted model at θ_{FT}. This continuation strategy (Allgower & Georg, 2003) uses task adaptation to shape prompt learning and ends with the original backbone.

4.2 WHY GRADUAL WITHDRAWAL HELPS

Gradual withdrawal lets prompt learning follow a sequence of nearby solutions. Let $z^\star(\alpha)$ be a differentiable stationary branch of F. Its velocity is $dz^\star/d\alpha = -H^{-1}\nabla^2_{z\theta}\mathcal{L}\,\Delta\theta$, where $H = \nabla^2_{zz}F(z^\star(\alpha), \alpha)$. The prompt's response therefore depends on the coupling between the FT update and prompt gradients, filtered by the local curvature of the prompt loss.

Suppose F is smooth in a tube around this branch, with $\mu I \preceq \nabla^2_{zz}F \preceq LI$ in gauge-fixed coordinates. Define $B = \sup_\alpha \|\partial_\alpha \nabla_z F(z^\star(\alpha), \alpha)\|$. For full-batch steps of size $0 < \eta \leq 1/L$, put $q = 1 - \eta\mu$, $e_t = \|z_t - z^\star(\alpha_t)\|$, and $\delta_t = \alpha_t - \alpha_{t+1} \geq 0$. A sufficiently small withdrawal step keeps the iterates in the tube and gives the tracking bound

$$e_T \leq q^T e_0 + \frac{B}{\mu} \sum_{t=0}^{T-1} q^{T-1-t}\delta_t. \tag{4}$$

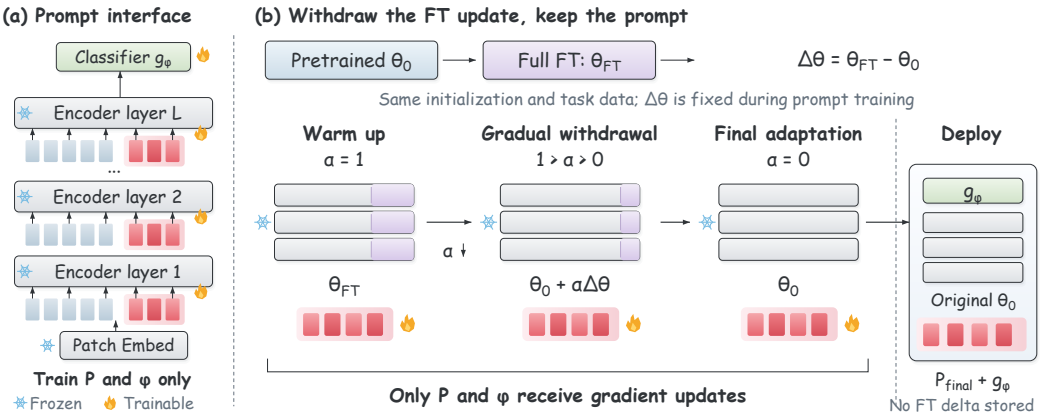

Figure 1: Training path and deployment. Prompts P and the classifier are learned while the backbone follows $\theta(\alpha) = \theta_0 + \alpha\Delta\theta$. Purple bands denote the temporary FT update, which vanishes at $\alpha = 0$. Snowflakes indicate no backbone gradients; flames indicate trainable parameters. The deployed model uses the original backbone and the final prompt and head.

The two terms separate warm-up error from the cost of moving the backbone. With exact warm-up and a uniform T-step schedule, endpoint error is $O(T^{-1})$; K final steps at $\alpha = 0$ reduce excess training loss to $O(q^{2K}T^{-2})$. This explains the three phases: warm-up enters a useful basin, gradual withdrawal tracks it, and endpoint refinement consolidates the prompt. Appendix A gives the complete assumptions, the tube condition, and the full proof.

4.3 Measuring prompt control

The shared endpoint makes the gain attributable to the training route. We compare gradual withdrawal with fixed-teacher logit distillation (Hinton et al., 2015), an abrupt return, a shuffled path through the same intermediate states, and a norm-matched FT update from another task. Prompt initialization, data order, and update budgets are matched; Table 1 summarizes them.

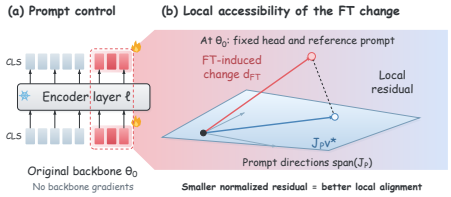

Figure 2: Local prompt control at θ_0. Projecting d_{FT} onto prompt directions gives the normalized residual in Eq. (5).

To probe local prompt control (Figure 2), fix a direct-VPT solution $(\bar{P}, \bar{\phi})$ and a validation batch, and concatenate its logits as $H(\theta, P, \phi)$. Evaluate both backbones with the *same* prompt slots and fixed head: $d_{\mathrm{FT}} = H(\theta_{\mathrm{FT}}, \bar{P}, \bar{\phi}) - H(\theta_0, \bar{P}, \bar{\phi})$ and $J_P = \partial_P H(\theta_0, \bar{P}, \bar{\phi})$, and define the normalized residual

$$\varepsilon_{\tan} = \frac{\min_v \|J_P v - d_{\mathrm{FT}}\|_2}{\|d_{\mathrm{FT}}\|_2}, \qquad d_{\mathrm{FT}} \neq 0. \tag{5}$$

The residual measures local alignment between the FT-induced logit change and prompt directions. Fixing the head isolates prompt control; matched paths evaluate the complete prompt/head model.

5 Experiments

We test three questions: how much the training path improves a fixed prompt model, whether the gain scales with model capacity, and which properties of the path produce it.

5.1 Setup

Our core evaluation covers all 19 VTAB-1k tasks and five fine-grained visual classification (FGVC) tasks with supervised ViT-B/16 and ViT-L/16 (Jia et al., 2022; Dosovitskiy et al., 2021). We use five paired seeds and the standard VTAB 800/200 training/validation split for model selection, followed

Table 1: **Matched training paths.** All prompt variants deploy the original backbone; $\Delta\theta_{\text{other}}$ is an FT update from a different task, norm matched to $\Delta\theta$.

Method	Backbone during prompt tuning	FT signal	Deploy
Full FT (2022)	$\theta_0 \to \theta_{\text{FT}}$	weights	θ_{FT}
Direct VPT (2022)	fixed θ_0	none	θ_0
VPT + KD (2015)	fixed θ_0	logits	θ_0
Abrupt return	$\theta_{\text{FT}} \to \theta_0$	weights	θ_0
Shuffled path	$\theta_0 + \alpha_{\pi(t)}\Delta\theta$	weights	θ_0
Other-task delta	$\theta_0 + \alpha_t\Delta\theta_{\text{other}}$	weights	θ_0
Gradual withdrawal	$\theta_0 + \alpha_t\Delta\theta$, $1 \to 0$	weights	θ_0

by refitting on all 1,000 examples (Zhai et al., 2019; Jia et al., 2022). Scaling adds MAE B/L/H and DINOv2 B/L/g with three paired seeds (He et al., 2022; Oquab et al., 2023). Results are grouped by pretraining family. The primary configuration uses 20 deep prompts per layer and cosine withdrawal. Warm-up and final-phase fractions are selected on eight development tasks that are fixed before any test score is inspected. The full dataset, label-budget, and shift protocols are in Appendix C.

The main comparison includes direct VPT and six recent prompt methods, alongside fixed-teacher distillation, abrupt return, shuffled withdrawal, and other-task guidance. Matched paths share the deployed architecture, prompt/head initialization, data order, and prompt-update budget. Target-task variants also share the FT checkpoint; ordered and shuffled paths use the same intermediate backbone states. These controls isolate path order and task alignment. Table 1 defines each path, and Appendix Table 9 gives the source-cited baseline roster.

5.2 A BETTER PROMPT THROUGH A BETTER TRAINING PATH

Gradual withdrawal achieves 76.2% mean accuracy on VTAB-1k and 92.3% on FGVC (Table 2), improving direct VPT by 6.8 and 3.2 points with the same deployed backbone and prompt architecture. *Gains span Natural, Specialized, and Structured tasks*: 5.8, 4.6, and 8.8 points, respectively. The largest gain occurs on Structured tasks, where direct VPT starts with the lowest accuracy. Gradual withdrawal also exceeds the strongest recent prompt method in every group: PRO-VPT on Natural, Specialized, and the VTAB mean, PIB on Structured tasks and FGVC. These methods redesign the prompt, whereas we change only how the VPT prompt is trained, so the two are complementary. The spread among the six recent methods is 4.4 VTAB and 2.9 FGVC points, while changing only how the original VPT prompt is trained adds 6.8 and 3.2 points: *the training path moves accuracy more than the choice among recent prompt designs*.

Table 2: **ViT-B/16 transfer accuracy (%).** Means cover 19 VTAB and five FGVC tasks.

Method	FGVC [5]	VTAB-1k [19]			
		Natural [7]	*Special.* [4]	*Struct.* [8]	Mean
Original pretrained backbone θ_0					
VPT-D (2022)	89.1	78.5	82.4	55.0	69.4
E^{2}VPT (2023)	89.2	80.0	84.4	57.4	71.4
CVPT (2025)	90.5	83.3	86.4	61.7	74.9
PRO-VPT (2025)	91.7	<u>84.1</u>	<u>86.7</u>	63.1	<u>75.8</u>
PAE (2026a)	91.0	81.7	84.5	58.3	72.4
VAPT (2026)	89.6	81.4	85.1	59.3	72.9
PIB (2026)	<u>92.1</u>	83.0	85.8	<u>63.2</u>	75.3
Matched path controls (all deploy θ_0)					
VPT-D + logit KD (2022; 2015)	89.8	79.6	83.8	56.4	70.7
Abrupt return	90.8	82.2	85.4	59.8	73.4
Shuffled path	90.2	81.3	84.7	58.1	72.2
Other-task delta	89.7	79.0	83.5	55.8	70.2
▶ **Ours (gradual withdrawal)**	**92.3**	**84.3**	**87.0**	**63.8**	**76.2**
Updated backbone θ_{FT} (contextual)					
Full FT (2022)	88.5	75.9	83.4	47.6	65.6

The order of guidance matters. Withdrawal exceeds fixed-teacher distillation by 5.5 VTAB points and abrupt return by 2.8 points. Shuffling the same intermediate backbone states costs 4.0 points.

FGVC shows the same ordering, with corresponding margins of 2.5, 1.5, and 2.1 points. Access to the FT checkpoint alone is insufficient to reproduce the gain: *the prompt benefits from continually adapting to the changing backbone throughout an ordered return*.

Weights guide better than logits. Fixed-teacher distillation and abrupt return use the same FT teacher, yet abrupt return gains 4.0 VTAB points over direct VPT while distillation gains only 1.3. Training the prompt inside the teacher's weights, even briefly, transfers far more than matching the teacher's outputs, and abrupt return alone already surpasses three of the six recent prompt methods. *The FT update is most useful as a training environment, not as a target to imitate.*

A weaker predictor can be a useful guide. Full FT scores 65.6% on VTAB and 88.5% on FGVC, below direct VPT on both aggregates, yet its update guides prompts to 76.2% and 92.3%. *The FT model's standalone accuracy therefore understates its value for training another model.* We distinguish the path gain $\Delta_{\text{path}} = A_{\text{ours}}(\theta_0) - A_{\text{VPT}}(\theta_0)$ from the FT reference gap $\Delta_{\text{FT}} = A_{\text{FT}}(\theta_{\text{FT}}) - A_{\text{VPT}}(\theta_0)$, where A denotes accuracy. Useful guidance can have positive Δ_{path} even when $\Delta_{\text{FT}} < 0$, which is exactly the case on both VTAB-1k and FGVC. The effect is strongest on Structured tasks, where FT reaches only 47.6%, 7.4 points below direct VPT, yet guides the largest group gain of 8.8 points among the three VTAB groups.

5.3 Scaling from 86M to 1.1B parameters

The gain persists across all eight backbones in Table 3. VTAB improvements range from 4.1 to 8.1 points, while FGVC improvements range from 1.3 to 5.6 points. Within each pretraining family, increasing model size preserves the benefit: supervised ViT-B/L gains are 6.8/7.2 points, MAE B/L/H gains are 7.8/8.1/7.9 points, and DINOv2 B/L/g gains are 4.2/4.4/4.1 points. Each paired comparison fixes the checkpoint, prompt interface, task split, and seeds; family-specific trends account for differences in pretraining and distillation (Oquab et al., 2023).

Table 3: **Backbone scaling over 19 VTAB-1k and five FGVC tasks.** Accuracy is in %; Δ is gradual withdrawal minus direct VPT in percentage points. FT is a contextual teacher endpoint.

Backbone	Params	VTAB-1k, 19-task mean				FGVC
		FT	VPT	Ours	Δ	Δ
Supervised ImageNet-21K 5 paired seeds						
ViT-B/16	86M	65.6	69.4	76.2	+6.8	+3.2
ViT-L/16	307M	68.8	72.6	79.8	+7.2	+2.8
MAE ImageNet-1K 3 paired seeds						
ViT-B/16	86M	61.3	37.2	45.0	+7.8	+5.6
ViT-L/16	304M	65.0	42.1	50.2	+8.1	+4.9
ViT-H/14	632M	67.6	45.8	53.7	+7.9	+4.2
DINOv2 3 paired seeds						
ViT-B/14	86M	73.5	72.0	76.2	+4.2	+1.9
ViT-L/14	300M	75.1	73.8	78.2	+4.4	+1.6
ViT-g/14	1.1B	76.8	75.2	79.3	+4.1	+1.3

Across 24 tasks, the grid contains 672 task–backbone–seed settings and 2,016 FT/VPT/withdrawal runs, before path controls and recent-method reruns. This spans a thirteen-fold range in backbone size (Jia et al., 2022; He et al., 2022; Oquab et al., 2023). *Gains remain positive at every scale*, and within each family they stay nearly constant as the model grows, varying by at most 0.4 points, whereas they differ by up to 4.0 points across families. *The benefit is set mainly by how the backbone was pretrained, not by its size.* Across families, gains are consistently largest for MAE, whose direct-VPT baseline is the weakest by a wide margin.

Prompts can surpass their teacher. The FT column of Table 3 shows two regimes. On supervised and DINOv2 backbones, the withdrawal prompt exceeds its own FT teacher on all five models, by 2.5 to 11.0 VTAB points. On MAE, where direct VPT trails FT by more than 20 points, the teacher remains stronger, yet withdrawal recovers about one third of the VPT–FT gap at every scale. In both regimes, *the prompt gains from the FT update regardless of whether the FT model itself is the better predictor*, which matches the path-gain view of Section 5.2.

5.4 Ablation studies

Table 4 shows that the path advantage persists as prompt capacity grows. With cosine withdrawal, VTAB gains are 5.7–6.8 points on ViT-B and 6.0–7.2 points on ViT-L across 5, 20, and 100 tokens per layer. Even at 100 tokens, the gains remain 6.5 and 7.0 points. *A larger prompt interface therefore leaves substantial room for improvement through training.* ViT-L also gains 0.3 to 0.5 points more than ViT-B at every prompt size, so larger backbones benefit slightly more from the path. The schedules share the teacher, prompt updates, warm-up, endpoint refinement, and total intermediate exposure $\sum_t \alpha_t$; the shuffled control permutes the linear path's intermediate values.

Table 4: **Withdrawal schedule $\times$ prompt capacity.** Entries are VTAB-19 gains over direct VPT (pp); all paths have matched training steps and intermediate $\sum_t \alpha_t$.

Schedule	ViT-B/16: tokens/layer			ViT-L/16: tokens/layer		
	5	20	100	5	20	100
Linear withdrawal	+5.4	+6.6	+6.3	+5.8	+7.0	+6.8
Cosine withdrawal	**+5.7**	**+6.8**	**+6.5**	**+6.0**	**+7.2**	**+7.0**
Staged withdrawal	+5.0	+6.1	+5.9	+5.5	+6.6	+6.4
Shuffled order	+2.0	+2.8	+2.6	+2.2	+3.0	+2.8

Cosine withdrawal gives the largest gain in every capacity setting, exceeding linear withdrawal by 0.2–0.3 points. The effect of ordering is larger: linear withdrawal exceeds its shuffled counterpart by 3.4–4.0 points. Thus, *preserving an orderly return contributes more than the particular smooth schedule among those tested*. The phase ablations in Table 5 further support the three-stage procedure. Removing warm-up reduces the VTAB gain from 6.8 to 5.3 points, and halving endpoint refinement reduces it to 5.6 points. Doubling the endpoint phase yields 6.4 points under the same total update budget, showing the importance of allocating updates across both withdrawal and refinement. The duration sweep in Appendix Figure 7 agrees: every sampled task peaks at a nonzero fraction, and Specialized and Structured tasks peak at 0.75 or higher, whereas Natural tasks peak at 0.5, which suggests that tasks farther from the pretraining data benefit from a longer return.

A better teacher helps, but any task teacher helps. Table 5 also varies the FT checkpoint under a fixed schedule. An early checkpoint still improves direct VPT by 4.9 VTAB points and a late checkpoint by 5.9 points, compared with 6.8 points for the best-validation checkpoint. *The gain grows with teacher quality, but it does not require a carefully tuned teacher*, which keeps the method practical when validating many teacher checkpoints is costly.

The delta-source ablation in Table 6 identifies task-aligned weight structure as a major source of the gain. The target-task update delivers 6.8 VTAB points, compared with 0.8 for another task's update, 1.7 for a layer-shuffled update, and 0.1 for a random direction. The corresponding FGVC gains are 3.2, 0.6, 1.0, and 0.0 points. Strikingly, a layer-shuffled target-task update still outperforms an intact update from another task, so *which task the update comes from matters more than where its weights sit*. All nonzero updates are norm matched and follow the same schedule, prompt architecture, data order, head initialization, and endpoint phase. The other-task teacher shares θ_0; layer shuffling permutes only homologous transformer blocks; the random direction is sampled tensorwise. The strong advantage of the intact target-task update shows that *effective guidance depends on both task relevance and the organization of the learned weight changes*.

Table 6: **FT-delta source.** Gains over direct VPT (pp); nonzero deltas are norm matched.

Delta source	VTAB	FGVC
Target-task FT	**+6.8**	**+3.2**
Other-task FT	+0.8	+0.6
Layer-shuffled FT	+1.7	+1.0
Random direction	+0.1	+0.0

5.5 What the training path changes

Every comparison below deploys the same backbone θ_0 and prompt architecture, so any difference between the compared prompts is produced by the training path alone. We ask where the resulting prompts look, where they sit in the loss landscape, and how they change attention.

Object evidence follows path order. Figure 3 compares input attributions on eight vehicle and bird photographs. Direct VPT often spreads weak evidence over the background (V1, B1) or fixates

Table 5: **Teacher quality and training-phase ablations.** All rows keep the same total prompt updates; results are gains over direct VPT (pp) on ViT-B/16.

Intervention	FT checkpoint	Warm-up	$\alpha = 0$ tail	VTAB	FGVC
Selected protocol	best validation	selected	selected	**+6.8**	**+3.2**
Early teacher	early	same	same	+4.9	+2.1
Late teacher	late	same	same	+5.9	+2.7
No warm-up	best validation	none	same	+5.3	+2.4
Half-length tail	best validation	same	half	+5.6	+2.5
Double-length tail	best validation	same	double	<u>+6.4</u>	<u>+3.0</u>

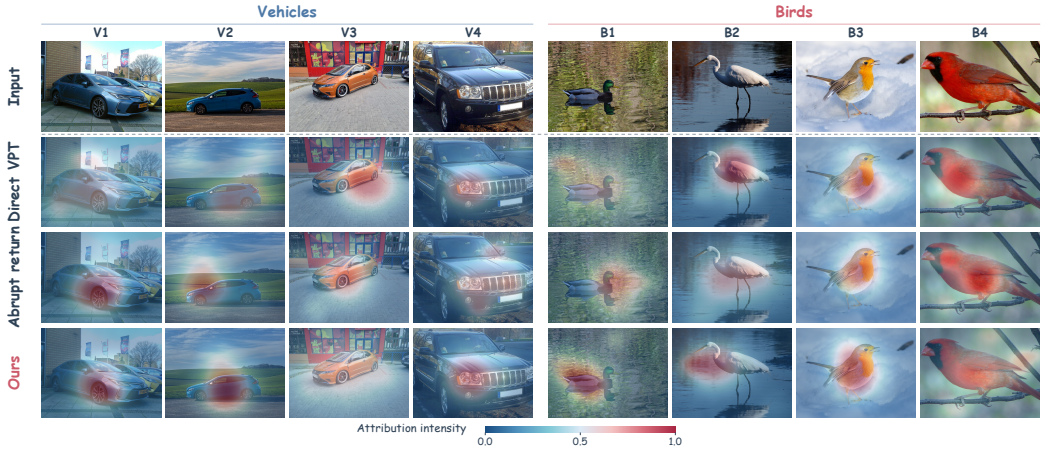

Figure 3: **Attribution across image categories.** Columns show four vehicle and four bird photographs; rows show the input, direct VPT, abrupt return, and gradual withdrawal, all evaluated at θ_0. Color scale and opacity are shared. Image credits are in Appendix L.

on a single part, such as one headlight of the SUV (V4). Abrupt return moves evidence onto the object but covers it unevenly. Gradual withdrawal most consistently covers the object body, for both rigid vehicles and articulated birds. The qualitative ordering, direct VPT $<$ abrupt return $<$ gradual withdrawal, reproduces the accuracy ordering in Table 2: *path order shapes where the prompt looks, not only how accurate it is*. Appendix Figure 8 repeats the comparison on the same eight images under controlled occlusion and reframing of each object.

The path reaches a different basin. Figure 4 slices the prompt loss around each fitted solution along two shared directions. Direct VPT sits in a sharply curved region with negative curvature, i.e., not a well-conditioned minimum. Gradual withdrawal sits in a wide basin with sparse contours, and negative curvature is confined to the slice boundary. The withdrawal prompt is therefore not a faster route to the direct-VPT solution; *it is a different, flatter solution that optimization from θ_0 misses*. This is the attainment gap of Eq. (1) made visible, and it matches the tracking analysis of Eq. (4): warm-up enters a well-conditioned basin at θ_{FT}, and gradual withdrawal carries it to θ_0. Appendix I gives the full slice protocol, the shared directions, and the definition of ρ. This flatness is not imposed: the objective in Eq. (3) has no sharpness penalty, so the wide basin is a property of the solution that the path selects. Figure 6a shows the same effect over training: our prompt first trails direct VPT at θ_0 and overtakes it as the basin is carried to the pretrained endpoint θ_0. This mirrors the analysis in Appendix A: an abrupt return can start outside the endpoint basin, whereas a sufficiently gradual return stays inside the moving tube around the solution branch $z^\star(\alpha)$.

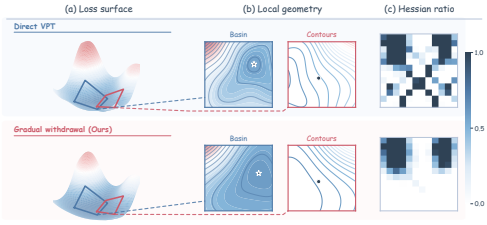

Figure 4: **Optimization geometry at θ_0.** (a) Loss surfaces. (b) Basin and contour views. (c) Negative-curvature ratio ρ of each surface's Hessian. Scales are shared.

Prompts steer patch interactions rather than absorb attention. Figure 5 fixes the images, backbone, prompt count, and layer. Under direct VPT, almost every query attends to the prompt columns while patch–patch attention is diffuse: the prompts act as an attention sink that overrides the backbone's computation rather than reorganizing it. After gradual withdrawal, the prompts keep a moderate share of attention, and structure reappears among patches, with a locality diagonal and a few patch columns attended by most queries. The attribution overlays in panel (b) show the same contrast: direct VPT spreads weak evidence across the image, whereas our prompt concentrates it on the subject. This points to how a prompt inherits fine-tuning. The FT update reorganizes patch interactions; *a prompt trained through its withdrawal learns to induce a comparable reorganization in the frozen backbone*.

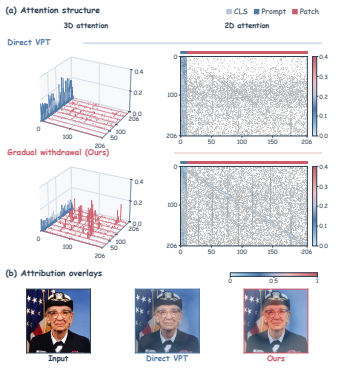

Figure 5: **Attention and attribution.** (a) 3D and 2D attention; blue-gray, blue, and rose mark CLS, prompt, and patch tokens. (b) Attribution overlays.

5.6 Training resources

Figure 6 separates prompt-stage memory from the cost of obtaining the full-fine-tuned teacher. The latter is charged to every task, while deployment retains only the shared original backbone, prompts, and heads. Appendix K gives the accounting protocol; Appendix J varies the time spent withdrawing the teacher update.

Gradual withdrawal trades training compute for accuracy while leaving deployment unchanged. Panel (a) shows how this compute is spent. Early in training, our prompt is fit to the fine-tuned backbone and trails direct VPT when evaluated at θ_0. As the update is withdrawn, it overtakes direct VPT and finishes about seven points higher, while abrupt return plateaus in between. Peak prompt-stage memory is nearly identical to direct VPT at every batch size (panel b), because the teacher update is stored once and interpolated in place. Total training cost roughly doubles: the teacher adds a fixed per-task cost, and the withdrawal stage costs about as much as direct VPT (panel c). Deployment, in contrast, is untouched. For 50 tasks, independent FT models require about 8.6 GB of FP16 weights, whereas one shared backbone with per-task prompts and heads requires about 0.2 GB (panel d), a 43-fold reduction that direct VPT and gradual withdrawal share exactly.

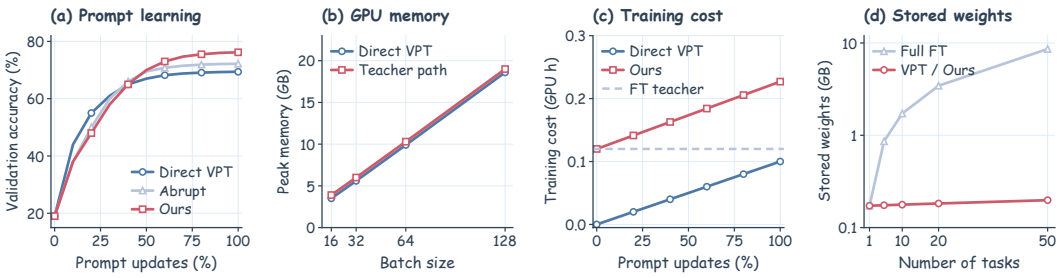

Figure 6: **Training resources and deployment cost.** (a) Validation accuracy at the original backbone. (b) Peak GPU memory during prompt optimization, excluding teacher fitting. (c) Cumulative GPU-hours including the task-specific teacher; the dashed line marks teacher cost alone. (d) FP16 weight storage across tasks; VPT and our method coincide.

6 Conclusion

In this work, we asked whether visual prompts can inherit full fine-tuning without enlarging the deployed model, and showed that they can. Gradual withdrawal trains prompts while a task-fine-tuned backbone returns to its pretrained weights, and it deploys exactly the direct-VPT model. On ViT-B/16, it raises VTAB-1k accuracy from 69.4% to 76.2% even though its FT teacher is weaker than direct VPT, and the gain holds from 86M to 1.1B parameters. More broadly, these results identify the training path as a new axis of parameter-efficient adaptation: training compute, rather than deployed parameters, can make compact models stronger.

AI USE STATEMENT

Generative AI assisted with literature search, manuscript drafting, mathematical exposition, and language revision. The authors are responsible for the methods, results, and final manuscript.

REFERENCES

Eugene L Allgower and Kurt Georg. *Introduction to numerical continuation methods*. SIAM, 2003.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in neural information processing systems*, 35:16664–16678, 2022.

Rajshekhar Das, Yonatan Dukler, Avinash Ravichandran, and Ashwin Swaminathan. Learning expressive prompting with residuals for vision transformers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3366–3377. IEEE, 2023.

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*, pp. 248–255. Ieee, 2009.

Wei Dong, Dawei Yan, Zhijun Lin, and Peng Wang. Efficient adaptation of large vision transformer via adapter re-composing. *Advances in Neural Information Processing Systems*, 36:52548–52567, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Cheng Han, Qifan Wang, Yiming Cui, Zhiwen Cao, Wenguan Wang, Siyuan Qi, and Dongfang Liu. E^2VPT: An effective and efficient approach for visual prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17491–17502, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Han_E2VPT_An_Effective_and_Efficient_Approach_for_Visual_Prompt_Tuning_ICCV_2023_paper.html.

Cheng Han, Qifan Wang, Yiming Cui, Wenguan Wang, Lifu Huang, Siyuan Qi, and Dongfang Liu. Facing the elephant in the room: Visual prompt tuning or full finetuning? *arXiv preprint arXiv:2401.12902*, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, June 2022.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Lingyun Huang, Jianxu Mao, Junfei Yi, Ziming Tao, and Yaonan Wang. Cvpt: Cross visual prompt tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 848–858. IEEE, 2025.

Qidong Huang, Xiaoyi Dong, Dongdong Chen, Weiming Zhang, Feifei Wang, Gang Hua, and Nenghai Yu. Diversity-aware meta visual prompting. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10878–10887. IEEE, 2023.

Yufei Huang, Yujia Qin, Huadong Wang, Yichun Yin, Maosong Sun, Zhiyuan Liu, and Qun Liu. Fpt: Improving prompt tuning efficiency via progressive training. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 6877–6887, 2022.

Injoon Hwang, Haewon Park, Youngwan Lee, Jooyoung Yang, and SunJae Maeng. Pc-lora: Low-rank adaptation for progressive model compression with knowledge distillation. *arXiv preprint arXiv:2406.09117*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

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*, pp. 709–727. Springer, 2022.

Can Jin, Ying Li, Mingyu Zhao, Shiyu Zhao, Zhenting Wang, Xiaoxiao He, Ligong Han, Tong Che, and Dimitris Metaxas. Lor-vp: Low-rank visual prompting for efficient vision model adaptation. In *International Conference on Learning Representations*, volume 2025, pp. 73004–73021, 2025.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pp. 5637–5664. PMLR, 2021.

Minh Le, Anh Nguyen, Huy Nguyen, Chau Nguyen, Anh T Tran, and Nhat Ho. Revisit visual prompt tuning: The expressiveness of prompt experts. In *International Conference on Learning Representations*, volume 2026, pp. 107257–107304, 2026.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/a41b3bb3e6b050b6c9067c67f663b915-Abstract.html.

Mengke Li, Ye Liu, Yang Lu, Yiqun Zhang, Yiu-ming Cheung, and Hui Huang. Improving visual prompt tuning by gaussian neighborhood minimization for long-tailed visual recognition. *Advances in Neural Information Processing Systems*, 37:103985–104009, 2024.

Yuqi Li, Xi Xiao, Yunbei Zhang, Lin Zhao, Yu Li, Aiden Zhao, Tianyang Wang, Hao Xu, and Yingli Tian. Rethinking layer-wise information allocation for vision foundation model adaptation. *arXiv preprint arXiv:2607.21973*, 2026.

Siqi Luo, Haoran Yang, Yi Xin, Mingyang Yi, Guangyang Wu, Guangtao Zhai, and Xiaohong Liu. Tr-pts: Task-relevant parameter and token selection for efficient tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4360–4369. IEEE, 2025.

Zheda Mai, Ping Zhang, Cheng-Hao Tu, Hong-You Chen, Quang-Huy Nguyen, Li Zhang, and Wei-Lun Chao. Lessons and insights from a unifying study of parameter-efficient fine-tuning (peft) in visual recognition. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14845–14857. IEEE, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Jaekyun Park and Hye Won Chung. VIPAMIN: Visual prompt initialization via embedding selection and subspace expansion. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4075. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b0bc711f48724237b38823c4d9cee10b-Abstract-Conference.html.

Wenjie Pei, Tongqi Xia, Fanglin Chen, Jinsong Li, Jiandong Tian, and Guangming Lu. Sa^2vp: spatially aligned-and-adapted visual prompt. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4450–4458, 2024.

Li Ren, Chen Chen, Liqiang Wang, and Kien Hua. Da-vpt: Semantic-guided visual prompt tuning for vision transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4353–4363. IEEE, 2025.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Selvaraju_Grad-CAM_Visual_Explanations_ICCV_2017_paper.html.

Chikai Shang, Mengke Li, Yiqun Zhang, Zhen Chen, Jinlin Wu, Fangqing Gu, Yang Lu, and Yiuming Cheung. Pro-vpt: Distribution-adaptive visual prompt tuning via prompt relocation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1558–1568. IEEE, 2025.

Visipedia. iNaturalist 2018 Competition, 2018. URL https://github.com/visipedia/inat_comp/blob/master/2018/README.md.

Junze Wang, Lei Fan, Dezheng Zhang, Weipeng Jing, Donglin Di, Yang Song, Sidong Liu, and Cong Cong. Visual prompt-agnostic evolution. *arXiv preprint arXiv:2601.20232*, 2026a.

Yubin Wang, Xinyang Jiang, De Cheng, Xiangqian Zhao, Zilong Wang, Dongsheng Li, and Cai Zhao. Exploring interpretability for visual prompt tuning with cross-layer concepts. In *International Conference on Learning Representations*, volume 2026, pp. 32636–32656, 2026b.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50233–50247. PMLR, 2024. URL https://proceedings.mlr.press/v235/wang24i.html.

Yuzhu Wang, Manni Duan, and Shu Kong. Attention to the burstiness in visual prompt tuning! In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4253–4263. IEEE, 2025.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7959–7971, June 2022.

Xi Xiao, Yunbei Zhang, Xingjian Li, Tianyang Wang, Xiao Wang, Yuxiang Wei, Jihun Hamm, and Min Xu. Visual instance-aware prompt tuning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 2880–2889, 2025a.

Xi Xiao, Yunbei Zhang, Yanshuh Li, Xingjian Li, Tianyang Wang, Jihun Hamm, Xiao Wang, and Min Xu. Visual variational autoencoder prompt tuning. *arXiv preprint arXiv:2503.17650*, 2025b.

Xi Xiao, Xingjian Li, Cheng Han, Tianyang Wang, Lin Zhao, Yunbei Zhang, Guosheng Hu, Runmin Jiang, Xi Li, Xiao Wang, et al. Adapting vision foundation models with cascaded semantics. *arXiv preprint arXiv:2608.05393*, 2026a.

Xi Xiao, Xingjian Li, Yunbei Zhang, Cheng Han, Tianming Liu, Tianyang Wang, Runmin Jiang, Jihun Hamm, Xiao Wang, and Min Xu. Layer-specific prompt fusion discovery via differentiable search in vision foundation models. In *European Conference on Computer Vision*, pp. 391–409. Springer, 2026b.

Seungryong Yoo, Eunji Kim, Dahuin Jung, Jungbeom Lee, and Sungroh Yoon. Improving visual prompt tuning for self-supervised vision transformers. In *International conference on machine learning*, pp. 40075–40092. PMLR, 2023.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Runjia Zeng, Cheng Han, Qifan Wang, Chunshu Wu, Tong Geng, Lifu Huang, Ying N Wu, and Dongfang Liu. Visual fourier prompt tuning. *Advances in Neural Information Processing Systems*, 37:5552–5585, 2024.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

Jeffrey O Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-tuning: a baseline for network adaptation via additive side networks. In *European conference on computer vision*, pp. 698–714. Springer, 2020.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Nan Zhou, Jiaxin Chen, and Di Huang. ivpt: Improving task-relevant information sharing in visual prompt tuning by cross-layer dynamic connection. *arXiv preprint arXiv:2404.05207*, 2024.

Technical Appendices

Table of Contents

A Local Tracking of the Withdrawal Path **15**

B Implementation and Reproduction Details **17**
B.1 Training procedure 17
B.2 Configuration and checkpoint selection 18

C Experimental Protocol and Baseline Coverage **19**

D Run Matrix and Reproducibility **20**

E Task-Level Results and Paired Variability **21**
E.1 Scores for individual tasks 21
E.2 Paired gains and aggregation 21

F Equal Total-Compute Comparison **22**

G Prompt Representation after Head Refitting **22**

H Teacher and Prompt Seed Stability **23**

I Additional Representation Diagnostics **23**

J Withdrawal Duration Analysis **24**

K Resource Accounting Protocol **24**

L Additional Qualitative Visualizations **26**

A LOCAL TRACKING OF THE WITHDRAWAL PATH

The analysis makes the benefit of gradual withdrawal precise: a sequence of small backbone changes can keep prompt optimization inside a moving basin, and refinement at the pretrained endpoint reduces the remaining error. We analyze full-batch gradient descent in coordinates that remove the symmetries of the loss, since no Hessian can be positive definite along a direction in which the loss is constant. For an affine softmax head, adding the same vector to every class weight, or the same scalar to every class bias, leaves the loss unchanged; class-centered weights and biases remove these directions. Deep prompts carry two further symmetries through LayerNorm, which the lemma at the end of this section removes. Positive curvature on the resulting coordinates is an explicit local assumption.

Proposition (local tracking and endpoint refinement). Let $z^\star : [0,1] \to \mathbb{R}^d$ be a continuously differentiable stationary branch, $\nabla_z F(z^\star(\alpha), \alpha) = 0$. Suppose F is twice continuously differentiable on an open neighborhood of the closed tube $\mathcal{T}_r = \{(z,\alpha) : \alpha \in [0,1],\ \|z - z^\star(\alpha)\| \le r\}$, and $\mu I \preceq \nabla^2_{zz} F \preceq LI$ throughout $\mathcal{T}_r$, where $r > 0$ and $0 < \mu \le L$. Define $B = \sup_{\alpha\in[0,1]} \|\partial_\alpha \nabla_z F(z^\star(\alpha), \alpha)\|$ and $c = B/\mu$. For $T \ge 1$ and $1 = \alpha_0 \ge \cdots \ge \alpha_T = 0$, take $z_{t+1} = z_t - \eta \nabla_z F(z_t, \alpha_t)$ with $0 < \eta \le 1/L$. Put $q = 1 - \eta\mu$, $e_t = \|z_t - z^\star(\alpha_t)\|$, and $\delta_t = \alpha_t - \alpha_{t+1}$. If $e_0 \le r$ and $qr + c \max_t \delta_t \le r$, then every iterate stays in the tube and

$$e_T \le q^T e_0 + c \sum_{t=0}^{T-1} q^{T-1-t} \delta_t. \tag{6}$$

After K further gradient steps at $\alpha = 0$, the endpoint training loss satisfies

$$0 \le F(z_{T+K}, 0) - F(z^\star(0), 0) \le \frac{L}{2} q^{2K} e_T^2. \tag{7}$$

Proof. Differentiate stationarity along the branch. With $H(\alpha) = \nabla^2_{zz} F(z^\star(\alpha), \alpha)$,

$$\frac{dz^\star}{d\alpha} = -H(\alpha)^{-1} \partial_\alpha \nabla_z F(z^\star(\alpha), \alpha) = -H(\alpha)^{-1} \nabla^2_{z\theta} \mathcal{L}\, \Delta\theta. \tag{8}$$

The mixed Hessian is evaluated at $(z^\star(\alpha), \theta(\alpha))$. The second identity follows from $F(z,\alpha) = \mathcal{L}(z, \theta_0 + \alpha\Delta\theta)$. Since $\|H^{-1}\| \le 1/\mu$, we have $\|dz^\star/d\alpha\| \le c$, hence $\|z^\star(\alpha_{t+1}) - z^\star(\alpha_t)\| \le c\delta_t$. Suppose $e_t \le r$. The segment between $z^\star(\alpha_t)$ and z_t lies inside the corresponding tube cross-section. Its average Hessian

$$A_t = \int_0^1 \nabla^2_{zz} F\big(z^\star(\alpha_t) + s(z_t - z^\star(\alpha_t)), \alpha_t\big)\, ds$$

satisfies $\mu I \preceq A_t \preceq LI$. By stationarity and the fundamental theorem of calculus, $z_{t+1} - z^\star(\alpha_t) = (I - \eta A_t)(z_t - z^\star(\alpha_t))$. The step-size condition gives $\|I - \eta A_t\| \le 1 - \eta\mu = q$. Therefore

$$e_{t+1} \le q e_t + c\delta_t \le qr + c \max_t \delta_t \le r.$$

Starting from $e_0 \le r$, induction proves tube invariance, and unrolling the recurrence proves Eq. (6). At the fixed endpoint, the same argument gives $\|z_{T+K} - z^\star(0)\| \le q^K e_T$. Taylor's theorem along the segment to $z^\star(0)$, together with stationarity and the Hessian bounds, yields Eq. (7). □

What gradual withdrawal gains. For a uniform schedule, $\delta_t = 1/T$, Eq. (6) becomes

$$e_T \le q^T e_0 + \frac{B}{\eta\mu^2 T}(1 - q^T). \tag{9}$$

With exact warm-up ($e_0 = 0$), this gives an $O(T^{-1})$ endpoint parameter error and an $O(q^{2K} T^{-2})$ excess training loss, for fixed B, μ, L, η. The controlling quantity is the response $H^{-1}\nabla^2_{z\theta}\mathcal{L}\,\Delta\theta$: task-dependent curvature and coupling determine how rapidly the prompt must adapt. The bound explains the complementary roles of small withdrawal steps and final endpoint refinement.

The same warmed-up solution can be outside the endpoint basin even when gradual tracking is certified. If $D = \|z^\star(1) - z^\star(0)\| > r + e_0$, then $\|z_0 - z^\star(0)\| \ge D - e_0 > r$. An abrupt return therefore starts outside the radius-r neighborhood certified at $\alpha = 0$, whereas a sufficiently gradual path satisfying the proposition stays inside the moving tube. This is a local optimization advantage under the stated assumptions; the matched experiments in Section 5 establish its empirical relevance for predictive accuracy.

What the bound predicts about the schedule. Equation (6) contains two schedule-dependent quantities: the forcing term $S(\delta) = \sum_t q^{T-1-t}\delta_t$ and the tube condition $c\max_t \delta_t \le (1-q)r = \eta\mu r$. Together they make three predictions, each of which the experiments confirm.

Corollary (minimum withdrawal length and optimal split). A uniform schedule with T_d withdrawal steps satisfies the tube condition if and only if $T_d \ge T_{\min} = B/(\eta\mu^2 r)$. Fix the warm-up and a budget of $N = T_d + T_f$ withdrawal and final steps with $T_{\min} \le N$. Then

$$\|z_N - z^\star(0)\| \le q^N e_0 + \frac{c\,q^N}{1-q}\cdot\frac{q^{-T_d}-1}{T_d}, \tag{10}$$

and the right-hand side is minimized by the shortest certified withdrawal, $T_d^\star = \lceil T_{\min}\rceil$.

Proof. The tube condition reads $qr + c/T_d \le r$, which is $T_d \ge c/(\eta\mu r)$. Equation (6) with $\delta_t = 1/T_d$ gives $e_{T_d} \le q^{T_d}e_0 + c(1-q^{T_d})/((1-q)T_d)$, and T_f contracting steps at $\alpha = 0$ multiply this by q^{T_f}; using $q^{T_f}(1-q^{T_d}) = q^N(q^{-T_d}-1)$ yields Eq. (10). With $\lambda = \log(1/q) > 0$, the map $x \mapsto (e^{\lambda x}-1)/x$ is increasing on $x > 0$, because its derivative has the sign of $e^y(y-1)+1 > 0$ for $y = \lambda x$. □

The certified optimum is interior, and it moves later as $B/(\mu^2 r)$ grows: a task whose solution moves farther relative to the size of its basin needs a longer return, every remaining step is best spent refining at $\alpha = 0$, and dropping the final phase ($T_f = 0$) gives the largest certified bound. Abrupt return, $T_d = 1$, is certified only when $B \le \eta\mu^2 r$. The duration sweep in Figure 7 has exactly this shape: accuracy peaks at an interior fraction on every sampled task, and the peak moves from 0.5 on Natural tasks to 0.75 or higher on Specialized and Structured tasks.

Proposition (schedule shape). Fix T decrements $\delta_0, \dots, \delta_{T-1} \ge 0$ with $\sum_t \delta_t = 1$. (i) For a fixed multiset of decrements, $S(\delta)$ is smallest when the decrements are applied in nonincreasing order. (ii) If δ is symmetric, $\delta_t = \delta_{T-1-t}$, and nondecreasing up to the middle index, then $S(\delta) \le S(\mathbf{1}/T)$, with equality only for the uniform schedule.

Proof. The weights $w_t = q^{T-1-t}$ increase with t, so (i) is the rearrangement inequality. For (ii), let $m = \lfloor (T-1)/2 \rfloor$ and fold the sum at the middle: $S(\delta) - S(\mathbf{1}/T) = \sum_{t=0}^{m} \nu_t a_t \tilde{g}_t$, where $a_t = \delta_t - 1/T$, $\tilde{g}_t = (w_t + w_{T-1-t})/2$, and $\nu_t = 2$ except $\nu_m = 1$ when T is odd. The sequence a_t is nondecreasing, $\tilde{g}_t = (q^t + q^{T-1-t})/2$ is strictly decreasing on $t \le m$ by convexity of $t \mapsto q^t$, and $\sum_t \nu_t a_t = 0$. The weighted Chebyshev sum inequality gives $\sum_t \nu_t a_t \tilde{g}_t \le (\sum_t \nu_t a_t)(\sum_t \nu_t \tilde{g}_t)/\sum_t \nu_t = 0$, with equality only when a is constant, that is, $a = 0$. □

Cosine withdrawal has decrements $\frac{1}{2}[\cos(\pi s_j) - \cos(\pi s_{j-1})]$ on a uniform grid of s, which are symmetric and unimodal. *Its forcing term is therefore strictly smaller than that of linear withdrawal with the same number of steps*, because it slows down as the backbone approaches θ_0, where the bound weights errors most. Its largest decrement is about $\pi/2$ times the linear one, so it needs proportionally more steps to satisfy the tube condition. Table 4 agrees: cosine withdrawal exceeds linear withdrawal at every prompt capacity and on both backbones. Staged withdrawal applies the return in a few discrete drops; each drop is a large decrement, so the tube condition holds only for small stages, and staged withdrawal ranks below both smooth schedules.

Non-monotone paths. The proof never uses $\delta_t \ge 0$. For any sequence (α_t), it gives $e_{t+1} \le qe_t + c|\alpha_t - \alpha_{t+1}|$, so the forcing term becomes $\sum_t q^{T-1-t}|\alpha_t - \alpha_{t+1}|$ and the tube condition constrains the largest jump. A monotone path has total variation one. The shuffled control visits the same values in random order: consecutive values differ by about $1/3$ on average, so its total variation grows linearly with T, and single jumps approach the full range. *The shuffled path loses the certificate for the same reason as abrupt return, and does so repeatedly*, which is why it ranks last in Table 4 despite visiting exactly the same backbones as linear withdrawal.

Proposition (inexact gradients). Replace the update by $z_{t+1} = z_t - \eta(\nabla_z F(z_t, \alpha_t) + \xi_t)$ with $\|\xi_t\| \le \sigma$, which covers mini-batch gradients with bounded deviation. If $e_0 \le r$ and $qr + \eta\sigma + c\max_t|\delta_t| \le r$, every iterate stays in the tube and

$$e_T \le q^T e_0 + c\sum_{t=0}^{T-1} q^{T-1-t}|\delta_t| + \frac{\sigma}{\mu}. \tag{11}$$

Proof. The contraction step now gives $\|z_{t+1} - z^\star(\alpha_t)\| \le qe_t + \eta\sigma$, so $e_{t+1} \le qe_t + \eta\sigma + c|\delta_t|$; induction and $\sum_{k\ge 0} q^k \eta\sigma = \sigma/\mu$ complete the argument. □

Gradient noise adds a floor σ/μ that does not depend on the schedule, so the schedule predictions above carry over unchanged to stochastic training.

Existence of the branch. The branch assumption is mild. If $z^\star(1)$ is stationary with $\nabla^2_{zz} F \succeq \mu I$, the implicit function theorem continues it to a C^1 branch around $\alpha = 1$, and Eq. (8) bounds its velocity by c. The continuation can stop only where the Hessian becomes singular, so it covers all of $[0, 1]$ whenever $\nabla^2_{zz} F \succeq \mu I$ and $\|\partial_\alpha \nabla_z F\| \le B$ hold along it, which are exactly the quantities the proposition uses.

Prompt response and local control. For cross-entropy, the prompt response has a Gauss–Newton form. Write h for the stacked logits, $W = \nabla^2_h \ell \succeq 0$, $J_z = \partial_z h$, and $d = J_\theta \Delta\theta$ for the first-order logit change caused by the FT update. Dropping terms that involve second derivatives of the logits, $\nabla^2_{zz} F \approx J_z^\top W J_z$ and $\partial_\alpha \nabla_z F \approx J_z^\top W d$, so

$$\frac{dz^\star}{d\alpha} \approx -\big(J_z^\top W J_z\big)^{-1} J_z^\top W\, d, \tag{12}$$

which is the coefficient of the W-weighted least-squares fit of the FT logit change by prompt directions. *Tracking asks the prompt to reproduce, step by step, the logit change that the FT update removes.* The residual $\varepsilon_{\tan}$ in Eq. (5) is the unweighted, finite-difference version of the same fit at θ_0: it measures how much of the FT change the prompt can follow at all, whereas B/μ measures how far the prompt must move to follow it.

Lemma (LayerNorm symmetries of deep prompts). In deep VPT, a prompt token $p \in \mathbb{R}^d$ inserted at layer ℓ affects the network only through $\mathrm{LN}_\ell(p)$, because its own output is replaced at layer $\ell + 1$. With the stabilizing constant of LayerNorm set to zero, the loss is therefore invariant under $p \mapsto ap + b\mathbf{1}$ for all $a > 0$ and $b \in \mathbb{R}$. The shift is linear and is removed by restricting each token to $\mathbf{1}^\perp$. For the scale, let f be differentiable on $\mathbb{R}^d \setminus \{0\}$ with $f(ap) = f(p)$ for all $a > 0$. Then $\langle \nabla f(p), p\rangle = 0$ and $\nabla f(ap) = a^{-1}\nabla f(p)$, and a gradient step $p^+ = p - \eta\nabla f(p)$ satisfies

$$\|p^+\|^2 = \|p\|^2 + \eta^2 \|\nabla f(p)\|^2, \qquad \frac{p^+}{\|p^+\|} = \frac{u - \eta_p \nabla f(u)}{\|u - \eta_p \nabla f(u)\|}, \qquad u = \frac{p}{\|p\|}, \quad \eta_p = \frac{\eta}{\|p\|^2}.$$

Proof. Differentiating $f(ap) = f(p)$ in a at $a = 1$ gives $\langle \nabla f(p), p\rangle = 0$; differentiating in p gives $a\nabla f(ap) = \nabla f(p)$. The norm identity follows from orthogonality. Writing $p = \|p\|u$ and $\nabla f(p) = \nabla f(u)/\|p\|$ gives $p^+ = \|p\|\big(u - \eta_p \nabla f(u)\big)$. □

Raw gradient descent on a deep prompt is thus normalized gradient descent on the unit sphere with the nonincreasing effective step size $\eta/\|p_t\|^2$; the norm of a prompt token carries no information about the loss. We therefore state the curvature assumption on the normalized, mean-zero prompt coordinates, where symmetry does not rule it out, and read the step size of the proposition as this effective step. A nonincreasing step changes the contraction factor of step t to $1 - \mu\eta/\|p_t\|^2$ and leaves the rest of the argument intact. With a positive stabilizing constant the scale symmetry holds only approximately, which makes the corresponding curvature nearly zero rather than positive; removing that direction is the accurate model.

B Implementation and Reproduction Details

B.1 Training procedure

The input is a pretrained backbone θ_0 and a labeled target-task training set. The output is a task-specific prompt P and classifier g_ϕ attached to the unchanged θ_0. Teacher fitting and prompt learning are separate optimization runs. The procedure below makes the state changes in Eqs. (2)–(3) explicit.

Table 7: **Configuration record.** Scope distinguishes the primary experiments from Figure 6.

Setting	Scope	Value or selection rule
Prompt capacity; path	Primary	20 deep prompts per layer; cosine withdrawal
Paired random seeds	Core; scaling	5; 3
Teacher checkpoint	Primary	Best validation performance
Phase-fraction selection	Primary	Eight fixed development tasks
Optimizer; momentum or betas	Teacher; prompt	AdamW; $\beta = (0.9, 0.999)$
Learning rate; LR schedule	Teacher; prompt	$5{\times}10^{-5}$; 10^{-3} for prompt/head; 5% LR warm-up, then cosine decay
Weight decay; exclusions	Teacher; prompt	Teacher 0.05, prompt 0, head 0.01; exclude bias/norm
Update budget; batch size	Benchmark runs	100 epochs each; batch 64
T_w, T_d, T_f or fractions	Primary	10%, 70%, 20% of prompt updates
Prompt/head initialization	Prompt runs	$P \sim \mathcal{N}(0, 0.02^2)$; Xavier-uniform head, zero bias
Optimizer reset at transitions	Prompt runs	Retain optimizer and LR state
Prompt insertion; feature readout	Prompt runs	Layerwise replacement before patches; native readout
Augmentation; dropout	Benchmark runs	Resize to 224; HFlip 0.5 except geometry tasks; prompt dropout 0.1
Prompt-checkpoint rule	Prompt runs	Best final-phase validation; refit to the selected epoch
Backbone; input	Resource figure	ViT-B/16; 224×224
Prompt interface	Resource figure	20 prompts in each of 12 layers
Hardware; precision	Resource figure	One node H100 80 GB; BF16 training
Updates; batch size	Resource figure	2,000 each for teacher and prompt; 64 except in the memory sweep

Step	Gradual withdrawal
1	**Fit the teacher.** Fine-tune a prompt-free copy of θ_0 and its classifier on the target training split. Select θ_{FT} using validation performance.
2	**Fix the guidance.** Store $\Delta\theta = \theta_{\mathrm{FT}} - \theta_0$. Keep both θ_0 and $\Delta\theta$ fixed throughout prompt learning; the teacher classifier is not interpolated.
3	**Initialize the student parameters.** Insert the chosen deep prompts and initialize $z = (P, \phi)$ using the same initialization as the matched prompt controls. The initialization rule is recorded in Table 7.
4	**Choose the current backbone.** Before each prompt update, set $\theta = \theta_0 + \alpha\Delta\theta$. Use $\alpha = 1$ for T_w warm-up updates, $\alpha = [1 + \cos(\pi(j+1)/(T_d+1))]/2$ for withdrawal updates $j = 0, \ldots, T_d - 1$, and $\alpha = 0$ for T_f final updates.
5	**Compute the supervised loss.** Pass the minibatch through the current prompted backbone and classifier, then evaluate the cross-entropy objective in Eq. (3).
6	**Update only the prompt and head.** Backpropagate to P and ϕ and apply their optimizer update. No gradient update changes θ, θ_0, or $\Delta\theta$. Repeat Steps 4–6 for all $T_w + T_d + T_f$ updates.
7	**Evaluate the deployment model.** Evaluate the retained prompt and classifier with the original backbone θ_0. Record the prompt-checkpoint selection rule and selected update index.
8	**Export.** Save P and ϕ with the identifier of θ_0. Deployment uses $h_{\theta_0, z}$ and requires neither θ_{FT} nor $\Delta\theta$.

Freezing the backbone removes its parameter updates, not the derivatives needed to learn the prompts. Gradients must still pass through the frozen transformer operations to P; detaching those activations or disabling gradient recording for the entire forward pass would block prompt learning. The preset α schedule is not optimized. The optimizer-state policy at phase boundaries remains a configuration field, rather than an additional step of the method.

B.2 Configuration and checkpoint selection

Table 7 separates the primary configuration from the settings used only for the resource analysis. The latter do not specify the optimizer or training budget of every benchmark. Schedule fractions are chosen on the eight development tasks fixed before test evaluation. VTAB model selection uses the 800/200 training/validation split, followed by refitting on all 1,000 examples; official held-out data determine the reported scores.

Checkpoint record. Each reproducible run record should identify the task and split, seed, pretrained checkpoint, selected teacher checkpoint, prompt interface and initialization, complete

teacher/prompt optimizer settings, phase lengths, and optimizer-state policy. It should also specify the validation metric, evaluation frequency, tie-breaking and checkpoint-selection rules, selected prompt update, and any refit procedure. Record prompt scores with the original backbone θ_0 and link them to the saved prompt/head checkpoint. These fields distinguish a change in training path from a change in data, initialization, selection, or optimization budget.

C Experimental Protocol and Baseline Coverage

Evaluation grid. Table 8 separates core transfer from pretraining, label-scale, and distribution-shift evaluations. Every FT teacher and prompt student start from the same pretrained checkpoint and use the target training split. Model and schedule selection use validation data; paired comparisons share their splits and seeds.

Table 8: **Evaluation grid.** Core transfer uses five paired seeds; scale, label, and shift tests use three.

Stratum	Backbone	Tasks / protocol	Seeds
Core transfer	Supervised ViT-B/16, L/16 (Dosovitskiy et al., 2021)	All 19 VTAB-1k tasks (Zhai et al., 2019) and five FGVC tasks (Jia et al., 2022)	5
Scale and pretraining	MAE B/L/H (He et al., 2022); DINOv2 B/L/g (Oquab et al., 2023)	Same transfer tasks where compatible; report each pretraining family separately	3
Label scale	Supervised B/L; MAE B/L	ImageNet-1K (supervised backbone only) (Deng et al., 2009) and Places365 (Zhou et al., 2017): 1%, 10%, full; iNat2018 (Visipedia, 2018): 10%, full	3
Shift	Supervised B/L; MAE B/L	WILDS iWildCam and FMoW, official splits and metrics (Koh et al., 2021)	3

We exclude ImageNet-1K from MAE transfer evaluation because it supplies MAE's pretraining images (He et al., 2022). DINOv2 results are reported separately to distinguish its pretraining corpus and distillation history. On iNat2018, 10% and full-data conditions preserve a class-complete protocol. WILDS evaluation uses macro-F1 for iWildCam and worst-region accuracy for FMoW (Koh et al., 2021).

Matched paths. All prompt variants in Table 1 deploy θ_0. Teacher-path variants share warm-up and final endpoint refinement. The shuffled path reorders the same intermediate α values; the other-task delta uses a teacher trained from the same θ_0 and is norm matched to the target-task update. These controls separate task alignment and path order from access to FT weights.

Baseline coverage. Table 9 records the broader comparison set. Our focused reruns include E^2VPT, CVPT, PRO-VPT, PAE, VAPT, and PIB (Han et al., 2023; Huang et al., 2025; Shang et al., 2025; Wang et al., 2026a; Le et al., 2026; Li et al., 2026). PAE provides a direct optimization baseline.

Table 9: **Source-cited baseline roster.** Full/Linear/Partial-1/MLP-3 protocols follow Jia et al. (2022).

Group	Methods and sources
Standard tuning	Full FT (Jia et al., 2022); linear probe (Jia et al., 2022); Partial-1 (Jia et al., 2022); MLP-3 (Jia et al., 2022); SideTune (Zhang et al., 2020); bias-only (Zaken et al., 2022); Adapter (Houlsby et al., 2019); LoRA (Hu et al., 2022); AdaptFormer (Chen et al., 2022); ARCatt (Dong et al., 2023); TR-PTS (Luo et al., 2025).
2022–23 prompts	VPT-S/D (Jia et al., 2022); E^2VPT (Han et al., 2023); EXPRES (Das et al., 2023); DAM-VP (Huang et al., 2023); GPT (Yoo et al., 2023).
2024 prompts	SA^2VP (Pei et al., 2024); SPT (Wang et al., 2024); VFPT (Zeng et al., 2024); iVPT (Zhou et al., 2024).
2025 prompts	LoR-VP (Jin et al., 2025); DA-VPT (Ren et al., 2025); ViaPT (Xiao et al., 2025a); CVPT (Huang et al., 2025); PRO-VPT (Shang et al., 2025); BPT (Wang et al., 2025); V^2APT (Xiao et al., 2025b).
2025 SSL	VIPAMIN (Park & Chung, 2025).
2026 prompts	PAE (Wang et al., 2026a); VAPT (Le et al., 2026); IVPT (Wang et al., 2026b); Prompt Fusion Discovery (Xiao et al., 2026b); Cascaded Semantics (Xiao et al., 2026a); PIB (Li et al., 2026).

DAM-VP uses meta training (Huang et al., 2023); GPT and VIPAMIN target self-supervised backbones (Yoo et al., 2023; Park & Chung, 2025). LoR-VP uses pixel-space prompts (Jin et al., 2025), while BPT reports FGVC but not VTAB in its original paper (Wang et al., 2025). These methods enter comparisons on supported, matched protocols. FT-then-PT and Mixed (Han et al., 2024) use changed deployment backbones and are distinct from our endpoint-matched controls.

D Run Matrix and Reproducibility

Paired core matrix. The 19 VTAB and five FGVC tasks, two supervised backbones, and five paired seeds give 240 task–backbone–seed teacher checkpoints. Each checkpoint supports six endpoint-matched prompt runs: direct VPT, fixed-logit KD, abrupt return, shuffled withdrawal, other-task delta, and gradual withdrawal. This gives 1,440 core prompt runs, in addition to reruns of compatible recent methods. The six-backbone MAE/DINOv2 extension contributes 432 teacher checkpoints at three seeds. Paired comparisons use the same checkpoint, split, prompt/head initialization, and seed.

Controlled sweeps. Eight development tasks are selected before inspecting test scores: two from each VTAB group, one FGVC task, and one shifted-domain task. They define the validation search for warm-up and final-$\alpha = 0$ fractions. The schedule-by-capacity matrix then covers all 19 VTAB tasks for supervised ViT-B/16 and ViT-L/16; the delta-source and teacher-phase controls cover all 19 VTAB and five FGVC tasks on ViT-B/16. The primary score sheet uses 20 prompts per layer and cosine withdrawal. Total prompt updates and validation-search trials are matched across causal controls. Cross-task deltas use another task's teacher from the same θ_0 and are norm-matched to the target delta; no source-task labels enter target prompt training.

Reporting. We retain paired-seed task scores, aggregates, selected configurations, and checkpoint IDs. Resource logs separate teacher/student GPU-hours, memory, wall time, trainable parameters, and deployment storage. Validation selects models; official held-out splits determine final scores.

E TASK-LEVEL RESULTS AND PAIRED VARIABILITY

E.1 SCORES FOR INDIVIDUAL TASKS

Tables 10 and 11 expand the benchmark means into their constituent tasks. Each task is an equal-weight unit in the benchmark average. VTAB Flowers102 and FGVC Flowers102 remain separate because their training protocols differ. The table schema uses the core supervised ViT-B/16 setting and five paired seeds; the same reporting format applies separately to each scaling backbone.

Table 10: **VTAB-1k task-level reporting.** Accuracy (%); task entries are mean ± standard deviation over five paired seeds. The last column reports the paired withdrawal gain (pp). The benchmark means repeat Table 2.

Task	FT	VPT	KD	Abrupt	Ours	Δ
Natural (7 tasks)						
CIFAR-100	75.2±0.8	79.0±0.6	80.0±0.6	82.5±0.7	84.7±0.5	+5.7±0.5
Caltech101	91.6±0.5	91.0±0.4	91.6±0.4	93.0±0.4	94.4±0.4	+3.4±0.4
DTD	60.7±1.0	65.9±0.8	67.8±0.8	72.1±0.9	75.3±0.7	+9.4±0.7
Flowers102	95.7±0.2	96.9±0.2	97.4±0.2	98.3±0.2	98.7±0.2	+1.8±0.2
Oxford-IIIT Pets	87.2±0.6	88.9±0.5	89.5±0.5	89.8±0.6	89.9±0.5	+1.0±0.4
SVHN	77.1±0.9	78.2±0.7	79.6±0.7	83.3±0.8	87.2±0.6	+9.0±0.6
SUN397	43.8±0.8	49.6±0.6	51.3±0.6	56.4±0.7	59.9±0.5	+10.3±0.5
Specialized (4 tasks)						
Patch Camelyon	82.7±0.6	81.9±0.5	83.4±0.5	85.2±0.6	87.2±0.5	+5.3±0.4
EuroSAT	97.5±0.2	96.2±0.2	97.2±0.2	98.3±0.2	98.5±0.2	+2.3±0.2
RESISC45	82.0±0.8	83.0±0.6	85.4±0.6	87.7±0.7	90.8±0.5	+7.8±0.5
Diabetic Retinopathy	71.4±0.9	68.5±0.7	69.2±0.7	70.4±0.8	71.5±0.6	+3.0±0.6
Structured (8 tasks)						
CLEVR: count	58.9±1.0	67.0±0.8	68.8±0.8	72.8±0.9	77.8±0.7	+10.8±0.7
CLEVR: distance	53.7±1.1	58.5±0.9	60.6±0.9	62.6±1.0	66.2±0.8	+7.7±0.8
DMLab	39.8±1.0	45.0±0.8	45.7±0.8	48.4±0.9	51.9±0.7	+6.9±0.7
KITTI: distance	68.0±0.9	72.7±0.7	73.1±0.7	74.6±0.8	76.0±0.6	+3.3±0.6
dSprites: location	56.4±1.4	67.2±1.1	70.0±1.2	76.3±1.2	83.9±1.0	+16.7±0.9
dSprites: orientation	41.3±1.2	48.3±1.0	49.7±1.1	54.2±1.1	58.9±0.9	+10.6±0.9
SmallNORB: azimuth	33.3±1.1	43.7±0.9	44.7±0.9	48.3±1.0	52.4±0.8	+8.7±0.8
SmallNORB: elevation	29.4±1.0	37.6±0.8	38.6±0.8	41.2±0.9	43.3±0.7	+5.7±0.7
19-task mean	65.6	69.4	70.7	73.4	**76.2**	**+6.8**

Table 11: **FGVC task-level reporting.** The same five-seed format and paired gain as Table 10; the last row repeats the main-table mean.

Task	FT	VPT	KD	Abrupt	Ours	Δ
CUB-200-2011	87.5±0.4	88.3±0.3	89.1±0.3	90.3±0.3	92.2±0.3	+3.9±0.3
NABirds	84.0±0.5	85.0±0.4	85.8±0.4	87.1±0.4	89.6±0.4	+4.6±0.4
Flowers102	98.3±0.1	98.6±0.1	98.8±0.1	99.0±0.1	98.9±0.1	+0.3±0.1
Stanford Dogs	85.0±0.5	85.9±0.4	86.6±0.4	87.5±0.4	89.7±0.4	+3.8±0.4
Stanford Cars	87.7±0.4	87.7±0.3	88.7±0.3	90.1±0.3	91.1±0.3	+3.4±0.3
5-task mean	88.5	89.1	89.8	90.8	**92.3**	**+3.2**

E.2 PAIRED GAINS AND AGGREGATION

For task d and paired seed s, let $A^m_{d,s}$ be the held-out accuracy of method m, and define $\Delta_{d,s} = A^{\text{ours}}_{d,s} - A^{\text{VPT}}_{d,s}$. Report the mean and sample standard deviation of $\Delta_{d,s}$ directly from these pairs; subtracting two independently computed error bars does not recover paired variability. For a benchmark of D tasks, first average within each seed, $\bar{A}^m_s = D^{-1}\sum_{d=1}^{D} A^m_{d,s}$, then summarize these seed-level means. The VTAB mean weights all 19 tasks equally, rather than weighting its three groups equally. Five-seed core results and three-seed scaling results are summarized separately.

Per-run records link each score to its task, split, backbone, method, teacher/prompt seeds, checkpoint, and configuration, enabling exact reconstruction of the aggregates.

F EQUAL TOTAL-COMPUTE COMPARISON

The matched paths in the main paper fix the prompt-update budget to isolate how training proceeds. A complementary question is how best to spend a fixed total training budget. The comparison uses supervised ViT-B/16 with 20 deep prompts per layer, the same task splits, and the hardware, precision, and batch-size settings in Appendix K.

Two budget views. The first view fixes U prompt updates, using the existing resource configuration of $U = 2{,}000$, and reports teacher and prompt costs separately. The second gives each method the same total budget B in GPU-hours. The table uses $B = 0.50$ GPU-hours per task and seed. For a teacher-guided run, $H_{\text{train}} = H_{\text{teacher}} + H_{\text{prompt}}$; direct VPT can spend the whole budget on prompt training. Teacher fitting is charged once for each independently deployable result, even when a checkpoint is physically shared across controls. KD includes its teacher forward passes or logit-generation cost. All methods deploy θ_0 and retain the same prompt architecture.

Using the budget. Direct VPT either extends one run or divides B among independent restarts, selecting one by validation accuracy. KD, abrupt return, and withdrawal use the remaining budget after teacher fitting; phase fractions are held fixed when their prompt schedules are rescaled. The common budget grid fix on development tasks and must permit every teacher-guided method to complete its schedule. Each point reports achieved GPU-hours, rather than equating update counts with compute. Validation and configuration search receive a separate, common GPU-hour allowance; their measured costs are recorded separately. All restart candidates and any final training refit are charged to B. Test scores do not select checkpoints, restarts, or budgets.

Table 12: **Two compute views.** Mean accuracy (%) over 19 VTAB-1k tasks at fixed prompt updates U and equal total budget B on supervised ViT-B/16. H_B is the per-task, per-seed cost at B.

Method	Accuracy at U	Accuracy at B	H_B (GPU-h)
Direct VPT, one run	69.4	70.2	0.50
Direct VPT, restarts	n/a	70.8	0.50
Fixed-teacher KD	70.7	71.5	0.50
Abrupt return	73.4	74.0	0.50
Gradual withdrawal	76.2	76.7	0.50

G PROMPT REPRESENTATION AFTER HEAD REFITTING

The learned prompt and classifier jointly determine endpoint accuracy. To test what remains in the prompt after removing its fitted classifier, the diagnostic freezes θ_0 and each method's learned prompt, discards the native head, and trains a new affine classifier on the resulting features. This directly tests whether the prompt supports a useful new readout.

Shared readout protocol. Within each paired seed, every method receives the same classifier initialization, training examples, augmentation sequence, optimizer, learning-rate schedule, regularization, and update budget. The backbone and prompt remain in evaluation mode throughout head fitting, with gradients enabled only for the new head. A common readout configuration is selected on development tasks and then held fixed. For VTAB, selection retains the 800/200 split; the selected protocol is refit on all 1,000 training examples before held-out evaluation. Features, normalization, and classifier capacity are treated identically across methods. Shared convergence checks track the training objective and validation accuracy so that insufficient readout optimization is visible.

Before and after. For each task and seed, report accuracy A_m^q and paired gain $\Delta_m^q = A_m^q - A_{\text{VPT}}^q$ for native and refitted heads, $q \in \{\text{native}, \text{refit}\}$. The refit gain measures how much of the advantage survives replacement of the learned classifier. Separate refitting accommodates method-specific feature coordinates. Summarize per-task means and seed variation before benchmark aggregation.

Table 13: **Shared head refitting on supervised ViT-B/16.** Accuracy is the mean over 19 VTAB-1k tasks; gains are paired differences from direct VPT (pp).

Method	Native (%)	Δ^{native}	Refit (%)	Δ^{refit}
Direct VPT	69.4	reference	69.2	reference
Fixed-teacher KD	70.7	+1.3	70.2	+1.0
Abrupt return	73.4	+4.0	72.5	+3.3
Shuffled path	72.2	+2.8	71.4	+2.2
Gradual withdrawal	76.2	+6.8	75.0	+5.8

H Teacher and Prompt Seed Stability

A crossed seed design separates variation in the FT guide from variation in prompt optimization. The study uses three FT seeds and three prompt seeds on the six VTAB tasks in the already fixed development pool, two from each group, with supervised ViT-B/16. Task identities and the protocol are fixed before inspecting these diagnostics. All runs share the data split, pretrained checkpoint, prompt architecture, and training budget. Prompt seeds control prompt/head initialization and data order; teacher seeds control the FT head initialization, data order, and stochastic training.

For each teacher $i \in \{1, 2, 3\}$, KD, abrupt return, and gradual withdrawal use the same checkpoint and all three prompt seeds $j \in \{1, 2, 3\}$. Direct VPT is run once per prompt seed, since it has no teacher. Pairwise differences use the same (i, j) for teacher-guided methods and the same j for direct VPT. The three teacher-conditioned means $A_i = \frac{1}{3}\sum_j A_{ij}$ summarize sensitivity to teacher choice; the within-teacher standard deviations across j summarize prompt variation. The same summaries are computed for paired gains. The full 3×3 score matrix is retained for each task.

Table 14 report the mean, the standard deviation of the three teacher-conditioned means, and the mean within-teacher standard deviation. These are descriptive summaries: the nine cells share teacher or prompt seeds and are not nine independent teachers. Reusing each direct-VPT result across teachers does not increase its sample count. Any aggregate across tasks first averages paired gains within each task, then weights tasks equally.

Table 14: **Crossed teacher and prompt seeds.** All columns average per-task summaries over the six VTAB development tasks. Teacher SD is across three teacher-conditioned means; prompt SD averages within-teacher SDs. Accuracy is in %; gains and SDs are in pp.

Method	Mean accuracy	Teacher SD	Prompt SD	Gain vs. VPT
Direct VPT	70.4	n/a	0.46	reference
Fixed-teacher KD	71.8	0.25	0.42	+1.4
Abrupt return	74.1	0.42	0.38	+3.7
Gradual withdrawal	76.9	0.35	0.31	+6.5

I Additional Representation Diagnostics

Optimization geometry. Following loss-landscape and visual-prompt diagnostics (Li et al., 2018; Zeng et al., 2024), Figure 4 compares prompt-loss slices at the deployed backbone θ_0. For model measurements, the protocol varies each fitted prompt along two shared, normalized directions, with its classifier and validation batch fixed. Shared directions, coefficient ranges, and loss levels make the local views directly comparable.

The Hessian refers to the two-dimensional slice. With extremal eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$ and $s = \max(|\lambda_{\min}|, |\lambda_{\max}|)$, the negative-curvature ratio is $\rho = \max(0, -\lambda_{\min})/s$, with $\rho = 0$ if $s = 0$. This summarizes the relative negative curvature within the displayed plane.

Attention and attribution. Figure 5 pairs token interactions with input-space overlays. A comparable attention analysis fixes images, backbone, prompt count, and layer across direct VPT and gradual withdrawal. It averages heads after softmax with dropout disabled and preserves row normalization. The 3D and 2D views depict the same matrix under a common token order and color range. Token-group totals distinguish a prominent column from a larger overall share of attention.

For input-space attribution, a comparable analysis fixes the target class and feature layer and applies the same scaling and opacity to both methods. Grad-CAM (Selvaraju et al., 2017) is one gradient-based option; attention weights answer a different question. We interpret these complementary views alongside prediction accuracy and matched path controls. Appendix L extends the comparison across categories and input transformations.

J Withdrawal Duration Analysis

To isolate the withdrawal schedule from prompt capacity and total optimization time, we fix warm-up at T_w updates and vary $T_d/(T_d+T_f)$ while holding T_d+T_f constant. Here T_d is the withdrawal phase and T_f is the final phase at θ_0. A ratio of zero is abrupt return; a ratio of one leaves no final adaptation phase. Every point uses the same prompt architecture and is evaluated at the original backbone. Duration is selected on development tasks before the full evaluation.

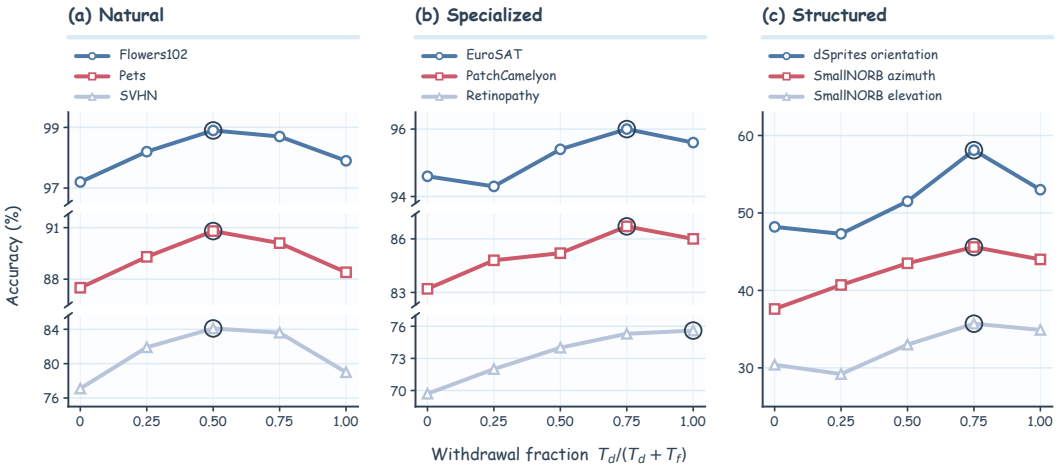

Figure 7: **Effect of withdrawal duration.** Accuracy on three tasks from each VTAB group as $T_d/(T_d+T_f)$ varies. Rings mark per-task maxima; axis breaks separate accuracy bands.

K Resource Accounting Protocol

Figure 6 uses ViT-B/16, 20 deep prompts in each of 12 layers, 224×224 inputs, BF16 training, and one H100 80 GB GPU. Teacher fitting and prompt optimization each have 2,000 updates, with batch size 64 except in the memory sweep. Panel (a) averages validation accuracy across the 19 VTAB tasks at matched update fractions; all prompt checkpoints are evaluated after restoring θ_0. Direct VPT, abrupt return, and gradual withdrawal share prompt and head initialization, data order, and update budget. The two teacher-path methods also share a teacher and warm-up.

Training cost. Prompt-stage peak memory includes weights, activations, gradients, optimizer states, the resident teacher update, and interpolation buffers under a common activation-checkpointing setting. Teacher fitting is excluded from panel (b); the full-pipeline peak is $\max(M_{\text{teacher}}, M_{\text{prompt}})$. Abrupt return and gradual withdrawal have the same peak-memory curve because both retain the teacher update during warm-up. GPU-hours equal device count times elapsed training hours. With $u\in[0,1]$ denoting the completed prompt-update fraction and constant mean time per update, the cumulative costs are $H_{\text{VPT}}(u)=uH_{\text{VPT}}$ and $H_{\text{ours}}(u)=H_{\text{teacher}}+uH_{\text{withdrawal}}$. Teacher fitting is charged once per task. Validation and hyperparameter search are excluded from the per-run view and belong in the total experiment ledger.

Deployment storage. We count FP16 weights in decimal GB and discard the teacher, weight update, optimizer states, and checkpoint metadata. For N tasks sharing a C-class head design, a backbone with P_B parameters, L prompted layers, m tokens per layer, and width d, independent full-tuned models use $S_{\text{FT}}(N)=2N[P_B+C(d+1)]$ bytes. A shared backbone and task-specific prompts and heads use $S_{\text{prompt}}(N)=2[P_B+N(Lmd+C(d+1))]$ bytes. Panel (d) sets $P_B=86\times10^6$,

$L = 12$, $m = 20$, $d = 768$, and $C = 100$. The common head size isolates scaling with task count rather than reproducing VTAB's different class counts; VPT and our method have identical storage under this accounting.

L Additional Qualitative Visualizations

Figure 8 applies cropping or partial occlusion to the examples of Figure 3 and places each input above direct VPT, abrupt return, and gradual withdrawal. The attribution protocol fixes the backbone, target class, feature layer, and attribution rule and evaluates every checkpoint at θ_0. The visual layout shares crops, color limits, and opacity across methods, with the same IDs and transformations in each row.

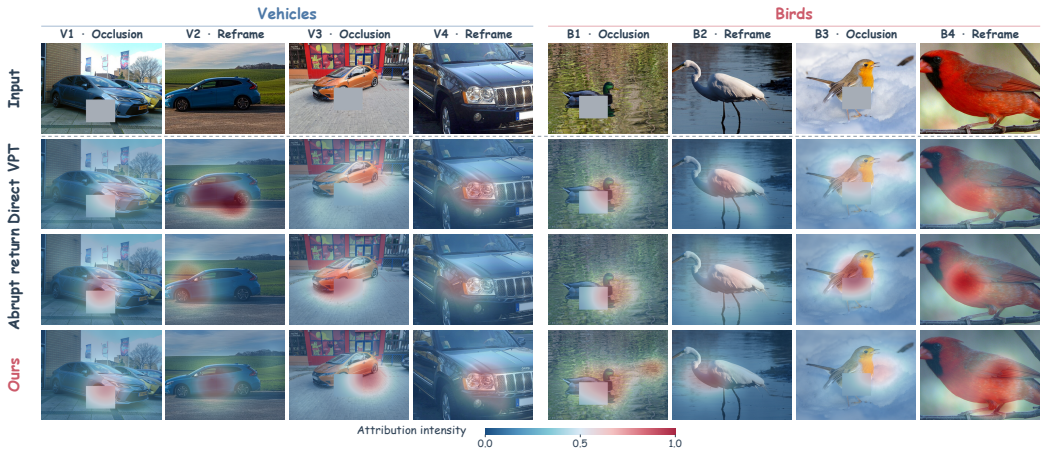

Figure 8: **Qualitative overlays under controlled input changes.** The same examples undergo cropping or partial occlusion, with row order and display settings preserved. Column IDs connect each view to its source image above.

Image credits. Source pages are linked by ID; photographs are resized/cropped and overlaid, with additional cropping or occlusion in Figure 8. V1: Dennis Elzinga (CC BY 2.0); V2: Rowan Heuvel (CC0); V3: (G)jabz (CC BY 4.0); V4: M 93, revised by Thomas doerfer (CC BY-SA 3.0); B1: AmosDoyle (CC BY-SA 3.0); B2 and B4: Ken Thomas (public domain); B3: Alexis Lours (CC BY 4.0). Adapted V4 and B1 tiles remain under CC BY-SA 3.0.