

VISUAL PROMPT-AGNOSTIC EVOLUTION

Junze Wang^{1*}, Lei Fan^{2*}, Dezheng Zhang¹, Weipeng Jing³, Donglin Di⁴, Yang Song²,
Sidong Liu⁵, Cong Cong^{5†}

¹University of Science and Technology Beijing, ²University of New South Wales

³Northeast Forestry University, ⁴Tsinghua University, ⁵Macquarie University

wangjunze@yeah.net, lei.fan1@unsw.edu.au, thomas.cong@mq.edu.au

ABSTRACT

Visual Prompt Tuning (VPT) enables effective adaptation of a frozen Vision Transformer (ViT) to downstream tasks by inserting a small number of learnable prompt tokens into the token sequence at each layer. However, we observe that existing VPT variants often suffer from unstable training dynamics, characterized by gradient oscillations. A closer layer-wise analysis reveals that shallow-layer prompts tend to stagnate early, while deeper-layer prompts exhibit high-variance oscillations, leading to a cross-layer mismatch. These issues contribute to slower convergence and degraded final performance. To address these challenges, we propose the Prompt-Agnostic Evolution (PAE) method, which can strengthen vision prompt tuning by explicitly modeling the dynamics of learnable prompts. From a frequency-domain perspective, we initialize prompts in a task-aware direction by uncovering and propagating frequency shortcut patterns that the backbone inherently exploits for recognition. To ensure coherent evolution across layers, we further employ a shared Koopman operator, which imposes a global linear transformation rather than uncoordinated, layer-specific updates. Finally, inspired by Lyapunov stability theory, we introduce a regularizer that constrains error amplification during evolution. Extensive experiments demonstrate that using PAE with VPT variants not only accelerates convergence with an average $1.41\times$ speedup but also yields 1–3% gains on 25 datasets with multi downstream tasks. Beyond performance, PAE remains prompt-agnostic and lightweight, and it integrates seamlessly with diverse VPT variants without backbone modification or inference-time changes, providing a practical and scalable solution for advancing prompt tuning¹.

1 INTRODUCTION

With the advent of large-scale vision foundation models, particularly those based on Vision Transformers (ViTs) (Dosovitskiy et al., 2020), significant progress has been made in learning general-purpose visual representations that can be transferred to a wide range of downstream tasks (Zhu et al., 2025; Fan et al., 2025a;b; Cong et al., 2025). However, effectively adapting these powerful pretrained models to specific tasks remains a key challenge, especially under constraints such as limited labeled data and computational resources. To address this, prompt tuning has emerged as a lightweight and parameter-efficient solution, enabling task-aware adaptation without modifying the backbone network (He et al., 2023; Mai et al., 2025). Among these approaches, Visual Prompt Tuning (VPT) (Jia et al., 2022) has shown particular promise for ViTs. By introducing a small set of learnable prompt tokens, VPT enables effective adaptation to new tasks while preserving generalizable knowledge encoded in the pretrained backbone (Xiao et al., 2025c).

Subsequent VPT variants improve upon this along four main directions: *Structured prompts* (Jia et al., 2022; Han et al., 2023; Tu et al., 2023; Xu et al., 2025; Ren et al., 2025; Wang et al., 2025b), which enrich input with additional channels, e.g., E2VPT (Han et al., 2023) adds key/value prompts and prunes tokens for efficiency, and ProVP (Xu et al., 2025) reorganizes prior-layer outputs and applies contrastive learning to reduce mismatch; *Adaptive prompting mechanisms* (Wang et al., 2022;

*Equal contribution.

†Corresponding author.

¹The code is available at <https://github.com/reeive/PAE>.

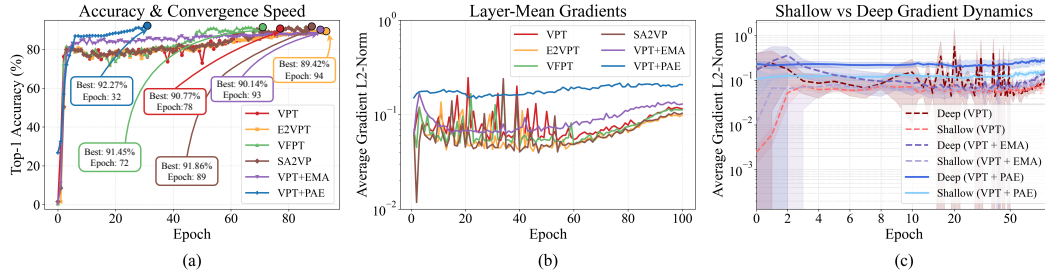


Figure 1: (a) Comparison of accuracy and convergence speed is shown in multiple VPT variants, including VPT, VPT+EMA, E2VPT, VFPT, and SA2VP. (b) Gradient oscillation (12 layers mean) is observed in multiple VPT variants, *i.e.*, VPT, E2VPT, VFPT, and SA2VP. (c) VPT hierarchically exhibits shallow-layer (Layers 1–4) stagnation and deep-layer (Layers 9–12) oscillations.

Dong et al., 2023; Sohn et al., 2023), where methods like LPT (Dong et al., 2023) dynamically combine shared and group-specific prompts for long-tailed tasks; *Projection-based prompts* (Xiao et al., 2025a;b; Liu et al., 2025) tightly couple input projection with prompt optimization, and *Perception-driven designs* (Xu et al., 2024; Zeng et al., 2024; Wang et al., 2024; Zhou et al., 2024), such as VFPT (Zeng et al., 2024), which reweights spectral components in the Fourier domain. Intuitively, since VPT and its variants update only a small number of prompt parameters, one might expect faster and more efficient training. In practice, however, they often suffer from slower convergence and sub-optimal accuracy (Figure 1a). To understand this discrepancy, we analyze their gradient behaviors during training. Figure 1(b) reveals that many VPT variants exhibit pronounced gradient oscillations, particularly in the early and middle training stages. Further, our layer-wise gradient analysis in Figure 1(c) shows a clear mismatch in optimization dynamics between shallow and deep layers. Specifically, prompts inserted into shallow layers experience an early surge in gradient magnitude followed by stagnation, remaining close to their initialization, while deeper-layer prompts begin to oscillate heavily after this stagnation sets in. These phenomena suggest poor coordination across layers and contribute directly to the observed degradation in training efficiency and performance.

We attribute these phenomena to two persistent challenges in VPT variants. (1) **Task-agnostic Prompt Initialization.** While prior works have explored various initialization strategies for prompt tokens (Jia et al., 2022; Wang et al., 2024; Xu et al., 2025; Kang et al., 2025), these initializations are generally agnostic to the downstream task. As a result, early gradient updates tend to align the prompts with the pretrained backbone rather than the target task (Wang et al., 2025b). To expedite this alignment phase, many methods employ aggressive optimization setting with high initial learning rates which can exacerbate instability and induce gradient oscillations, ultimately hindering convergence and degrading performance. (2) **Independent prompt design across layers.** In most VPT variants, prompts are pre-pended and optimized independently at each transformer layer. During training, only representations propagate forward through the model, while gradients must back-propagate through many frozen layers. This leads to significantly weakened gradient signals in the shallow layers, causing their prompts to stagnate near initialization values (Merlin et al., 2023). Meanwhile, deeper-layer prompts are repeatedly adjusted in compensation, resulting in amplified gradient oscillations. The lack of explicit cross-layer coordination introduces an optimization mismatch across layers, contributing to slower convergence and reduced final accuracy.

To address Challenge (1), we propose Modal Pre-Alignment (MPA), inspired by recent findings that pretrained vision backbones often rely on specific frequency shortcuts to make correct prediction (Wang et al., 2025a). MPA performs a lightweight search to identify these task-aware shortcuts and uses them to initialize the visual prompts. To address challenge (2), we reformulate prompt optimization as a Koopman (Mezić, 2013)-Lyapunov (Shevitz & Paden, 1994) discrete dynamical system (KLD). In this formulation, the prompt embedding at each layer is treated as the system state, and its evolution is governed by a single shared operator that maps the prompt state from one layer to the next. Rather than optimizing each layer’s prompts independently, this perspective explicitly couples them through a common dynamical rule, thereby establishing cross-layer dependencies.

Building on these two components, we introduce Prompt-Agnostic Evolution (PAE), a unified method designed to stabilize and accelerate prompt learning for VPT variants. PAE operates in two sequential stages: First, MPA provides task-aware prompt initialization by leveraging frequency-domain cues aligned with the backbone’s inductive biases. Then, KLD evolves these prompts across

layers using a shared Koopman operator, enforcing smooth inter-layer transitions. A Lyapunov-style regularizer further enhances stability by constraining error accumulation during evolution. Notably, PAE is lightweight, introduces no inference-time overhead, and can be seamlessly integrated into existing methods without modifying the backbone. Our contributions are summarized as follows:

- To the best of our knowledge, this is the first work to reframe VPT as the control of prompt trajectories within a dynamical system. This offers an explicit cross-layer association perspective for VPT variants.
- We propose MPA, a task-aware prompt initialization which utilizes frequency-domain cues to produce prompts that are aligned with the downstream objective from the start.
- We propose KLD which employs a Koopman evolution operator to establish cross-layer association of prompts, while ensuring the stability of the entire optimization process by a Lyapunov-style regularizer.
- Extensive experiments on 25 datasets establish new state-of-the-art performance by incorporating PAE into various VPT variants, yielding faster convergence without incurring additional inference-time costs.

2 METHOD

2.1 PRELIMINARIES AND PROBLEM FORMULATION

Formulation. Given a dataset $\mathcal{D}$ consisting of input-label pairs (x, y) , where $x \in \mathbb{R}^{H \times W \times 3}$ is an image and $y \in \{1, \dots, C\}$ is the corresponding class label, the image x is processed by a Vision Transformer (ViT) backbone F , composed of L encoder blocks $\{E_i\}_{i=1}^L$. Specifically, x is first passed through a patch embedding layer, which splits it into N patch tokens. As shown in Figure 2(a), a learnable visual prompt $\mathbf{P}_i \in \mathbb{R}^{T \times d}$ is pre-pended to the patch tokens at each transformer layer i , where T denotes the prompt length and d is the embedding dimension. We denote the set of prompts across all layers as $\mathcal{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_L\}$. These prompts are optimized while keeping the ViT backbone F frozen. The final classification head H is trained jointly with $\mathcal{P}$ and the overall training objective is defined as:

$$\min_{\mathcal{P}} \mathbb{E}_{(x,y) \sim \mathcal{D}} \mathcal{L}_{\text{task}}(H(F(x; \mathcal{P})), y), \quad (1)$$

where $\mathcal{L}_{\text{task}}$ is a task-aware loss function. The goal of PAE is to accelerate and stabilize the learning of prompts $\mathcal{P}$ by combining two complementary components: MPA, which provides task-aware initialization, and KLD, which enforces coordinated optimization across layers.

2.2 INITIALIZATION VIA MODAL PRE-ALIGNMENT (MPA)

To initialize $\{\mathbf{P}_i^{\text{init}}\}_{i=1}^L$ with task awareness, we propose the MPA strategy. As illustrated in Figure 2(b), MPA consists of two stages. First, we identify frequency shortcuts by probing F using the training dataset $\mathcal{D}$. Next, we use the identified shortcuts to construct the first-layer prompt $\mathbf{P}_1^{\text{init}}$, which is then propagated through F to generate the complete initialization set $\{\mathbf{P}_i^{\text{init}}\}_{i=2}^L$.

Phase I — Discovering Frequency Shortcuts. Neural networks often exploit the most distinctive frequency components in data as shortcuts during training (Wang et al., 2023). However, such frequency biases are not explicitly represented in the spatial domain, making them difficult to observe or control directly. To address this, we transform images into the spectral domain, where frequency characteristics are more accessible for analysis. Specifically, we begin by sampling a mini-batch $\{(x_j, y_j)\}_{j=1}^B$ from the training set. Each image x_j is transformed into the frequency domain using the 2D Fourier transform, yielding $\mathcal{F}(x_j) \in \mathbb{R}^{H \times W}$. To identify task-aware regions in the frequency spectrum, we define a set of binary masks $\{\mathbf{M}_s\}_{s=1}^S$, each corresponding to a local region of the frequency space. These masks are generated by sliding a window of size $w \times w$ with stride r along both height and width, resulting in $S = (\lfloor \frac{H-w}{r} \rfloor + 1) \times (\lfloor \frac{W-w}{r} \rfloor + 1)$ distinct window locations. Each mask $\mathbf{M}_s \in \{0, 1\}^{H \times W}$ has ones in its selected $w \times w$ region and zeros elsewhere. We then apply each mask $\mathbf{M}_s$ to the frequency representation $\mathcal{F}(x_j)$ by element-wise multiplication, and reconstruct the corresponding spatial-domain image using the inverse Fourier transform:

$$\hat{x}_{s,j} = \mathcal{F}^{-1}(\mathbf{M}_s \odot \mathcal{F}(x_j)), \quad (2)$$

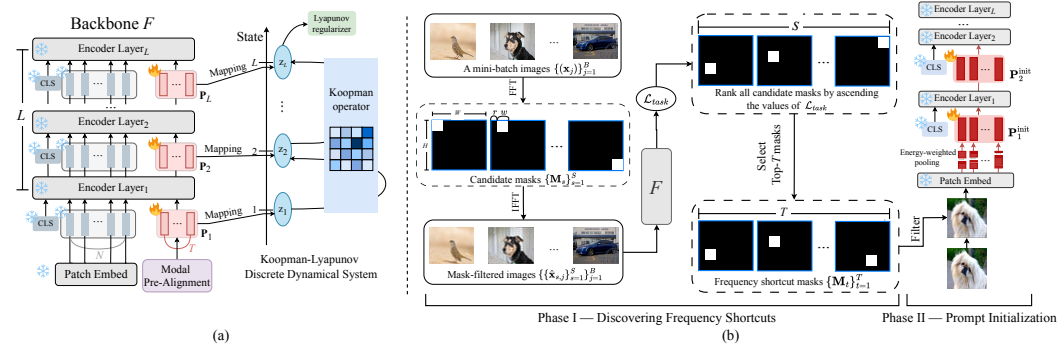


Figure 2: (a) PAE pipeline: MPA first initialize per-layer prompts. KLD then propagates prompts across layers via a shared Koopman operator, with a Lyapunov-style regularizer constraining error growth. (b) MPA pipeline: Frequency-domain transformations generate candidate masks, from which top ones are selected to build the initial prompt and propagate it across layers.

where $\odot$ denotes element-wise multiplication. To evaluate the predictive utility of each frequency region, we use the reconstructed images $\{\hat{x}_{s,j}\}_{j=1}^B$ to compute the task loss for each mask M_s . We then rank all masks $\{M_s\}_{s=1}^S$ in ascending order of their corresponding $\mathcal{L}_{task,s}$. A lower loss implies that the masked frequency region retains more class-discriminative information for the task.

Phase II — Prompt Initialization. We select the top- T masks, *i.e.*, the T frequency masks with the lowest task losses $\mathcal{L}_{task,s}$, to initialize the first-layer visual prompt $P_1^{init} \in \mathbb{R}^{T \times d}$. Each selected mask M_t serves as a frequency shortcut, capturing a localized spectral region that preserves strong task-discriminative information. To encode these frequency shortcuts into prompt tokens, we proceed as follows. For each selected mask $M_t \in \{M_1, \dots, M_T\}$ and each image $x_j \in \{x_1, \dots, x_B\}$ in a mini-batch, we apply M_t to the image’s Fourier spectrum and reconstruct the filtered image $\hat{x}_{t,j}$ using Equation 2. Each $\hat{x}_{t,j}$ is then passed through the frozen patch embedding module to yield patch tokens $\{p_{t,j,n} \in \mathbb{R}^{1 \times d}\}_{n=1}^N$, where N is the number of image patches. To aggregate the discriminative content emphasized by each frequency shortcut, we employ a token energy-weighted pooling strategy. The intuition is that tokens with higher activation norms tend to carry stronger semantic signals. Specifically, we compute each token’s energy and normalize it across the batch:

$$e_{t,j,n} = \|p_{t,j,n}\|^2, \quad w_{t,j,n} = \frac{e_{t,j,n}}{\sum_{j'=1}^B \sum_{n'=1}^N e_{t,j',n'}}. \quad (3)$$

We compute the representative token ρ_t and stack them to obtain the initial prompt matrix P_1^{init} :

$$\rho_t = \sum_{j=1}^B \sum_{n=1}^N w_{t,j,n} \cdot p_{t,j,n}, \quad P_1^{init} = \begin{bmatrix} \rho_1^T \\ \vdots \\ \rho_T^T \end{bmatrix} \in \mathbb{R}^{T \times d}. \quad (4)$$

To initialize deeper layers, we propagate P_1^{init} through the frozen transformer encoder blocks $\{E_i\}_{i=1}^{L-1}$. Starting from the first-layer, each block processes only the prompt and outputs the prompt for the next layer:

$$P_{i+1}^{init} = E_i(P_i^{init}), \quad \text{for } i = 1, \dots, L-1. \quad (5)$$

This propagation transfers frequency-aligned semantics through the transformer hierarchy, forming a coherent, task-aware prompt initialization trajectory across all layers.

2.3 OPTIMIZATION VIA KOOPMAN-LYAPUNOV DISCRETE DYNAMICAL SYSTEM (KLD)

Following the MPA initialization, we use KLD to establish cross-layer association of prompts $\{P_i\}_{i=1}^L$. KLD is divided into two parts. Firstly, the cross-layer association is established through the Koopman operator, which enables linear evolution in a lifted latent space, facilitating prediction and control from nonlinear data (Mezić, 2013). Then, a Lyapunov-style regularizer is employed to limit the error of the Koopman operator calculation across layers.

Prompt Linear Projection. To establish cross-layer associations, KLD first linearly projects the T prompt tokens at each layer into a shared latent space. We introduce a global learnable projection matrix $\mathbf{U} \in \mathbb{R}^{d \times K}$, where K is the dimensionality of the Koopman space. This matrix is initialized using Kaiming-uniform initialization (He et al., 2015), and projects the prompt matrix $\mathbf{P}_i \in \mathbb{R}^{T \times d}$ from layer i into a linear representation $\mathbf{z}_i \in \mathbb{R}^{T \times K}$ as follows:

$$\mathbf{z}_i = \mathbf{P}_i \mathbf{U}. \quad (6)$$

Within this shared latent space, we model prompt evolution using a global shared Koopman operator $\mathbf{K} \in \mathbb{R}^{K \times K}$, which is initialized as the identity matrix.

Koopman Operator Evolution.

Unlike conventional VPT variants that tune prompts independently per layer, we introduce cross-layer coupling via $\mathbf{K}$. Specifically, we evolve the current state $\mathbf{z}_i$ into the next state $\hat{\mathbf{z}}_{i+1}$ using:

$$\hat{\mathbf{z}}_{i+1} = \mathbf{z}_i \mathbf{K}. \quad (7)$$

This formulation replaces per-layer independence with a global consistent evolution process, as visualized in Figure 3.

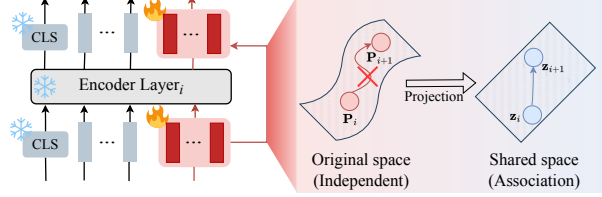


Figure 3: Illustration of Koopman evolution. In standard VPT, prompts at different layers are independent. In contrast, KLD maps each layer’s prompt into a shared latent space, where a global shared Koopman operator $\mathbf{K}$ governs their evolution, enabling smooth cross-layer transitions.

Learning the Koopman Operator. Effective evolution requires learning both $\mathbf{U}$ and $\mathbf{K}$. The matrix $\mathbf{U}$ learns to map prompts into a space where transitions between layers are approximately linear, while $\mathbf{K}$ captures the shared layer-to-layer dynamics. To jointly learn these parameters, we introduce the *Koopman consistency loss* $\mathcal{L}_{\text{kp}}$, which minimizes the difference between the predicted evolution $\hat{\mathbf{z}}_{i+1}$ and the actual projected state $\mathbf{z}_{i+1}$:

$$\mathcal{L}_{\text{kp}} = \sum_{i=1}^{L-1} \|\mathbf{z}_{i+1} - \hat{\mathbf{z}}_{i+1}\|_F^2 = \sum_{i=1}^{L-1} \|\mathbf{P}_{i+1} \mathbf{U} - \mathbf{P}_i \mathbf{U} \mathbf{K}\|_F^2, \quad (8)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This loss provides explicit gradient flow across layers by coupling all prompts $\mathcal{P} = \{\mathbf{P}_i\}_{i=1}^L$ via the shared parameters $\mathbf{U}$ and $\mathbf{K}$. The gradient with respect to each prompt $\mathbf{P}_i$ is:

$$\frac{\partial \mathcal{L}_{\text{kp}}}{\partial \mathbf{P}_i} = \underbrace{2(\mathbf{P}_i \mathbf{U} - \mathbf{P}_{i-1} \mathbf{U} \mathbf{K}) \mathbf{U}^\top}_{\text{preceding-layer consistency: } (i-1) \rightarrow i} + \underbrace{2(\mathbf{P}_i \mathbf{U} \mathbf{K} - \mathbf{P}_{i+1} \mathbf{U}) (\mathbf{U} \mathbf{K})^\top}_{\text{succeeding-layer consistency: } i \rightarrow (i+1)}. \quad (9)$$

This formulation reveals how each layer is pulled toward consistency with both its preceding and succeeding layers, enforcing global smooth and coherent prompt evolution².

A Lyapunov-style Regularizer for Evolution Stability. Since the linear approximation is imperfect and errors can accumulate across layers (Philipp et al., 2024), we introduce a Lyapunov-style regularizer $\mathcal{L}_{\text{stab}}$ to mitigate error growth and improve stability. Specifically, $\mathcal{L}_{\text{stab}}$ employs a Lyapunov function $V(\mathbf{z}) = \text{tr}(\mathbf{z} \mathbf{Q} \mathbf{z}^\top)$, where $\mathbf{Q} \in \mathbb{R}^{K \times K}$ is a learnable symmetric positive definite matrix. We regard the evolution as stable when the cross-layer error is non-increasing, *i.e.*, when associations between successive layers do not deteriorate (Haber & Ruthotto, 2017). In other words, stability requires $V(\mathbf{z}_{i+1}) \leq V(\mathbf{z}_i)$. This condition can be implemented as follows:

$$\mathcal{L}_{\text{stab}} = \sum_{i=1}^{L-1} \max(0, V(\mathbf{z}_{i+1}) - V(\mathbf{z}_i)). \quad (10)$$

As a stability regularizer, $\mathcal{L}_{\text{stab}}$ penalizes increases in the evolution error between successive layers. The penalty is adaptive, and is applied only when cross-layer association degrades, so the evolution remains stable without excessive constraints³.

²See Appendix A.2 for full derivation.

³The gradient derivation is in Appendix A.3.

Table 1: The speedup($\times$) and image classification accuracy (%) for ViT-Base/16 pretrained on supervised ImageNet-21k. Values are raw accuracy with PAE gains in parentheses.

Method with PAE	Speedup($\times$)	FGVC	VTAB-1k			
			<i>Natural</i>	<i>Specialized</i>	<i>Structured</i>	Mean Total
Full Fine-tune	-	88.54	75.88	83.36	47.64	68.96
VPT (Jia et al., 2022)	1.78	89.11 +1.91	78.48 +3.25	82.43 +2.09	54.98 +3.30	71.96 +2.88
E2VPT (Han et al., 2023)	1.65	89.22 +1.74	80.01 +1.38	84.43 +1.33	57.39 +2.34	73.94 +1.68
LPT (Dong et al., 2023)	1.44	89.94 +1.38	79.24 +1.77	83.40 +1.62	57.39 +2.02	73.34 +1.81
VQT (Tu et al., 2023)	1.52	89.41 +1.21	79.46 +2.93	82.23 +3.11	57.47 +2.36	73.05 +2.80
VFPT (Zeng et al., 2024)	1.27	89.24 +2.24	81.35 +0.72	84.93 +1.03	60.19 +0.77	75.39 +0.94
SA2VP (Pei et al., 2024)	1.60	90.08 +1.12	80.97 +1.89	85.73 +0.85	60.80 +2.25	75.83 +1.66
ProVP (Xu et al., 2025)	1.19	89.56 +1.62	80.35 +1.98	84.07 +1.34	60.31 +1.41	74.91 +1.57
BPT (Wang et al., 2025b)	1.37	90.86 +1.35	80.24 +2.22	84.45 +1.88	60.39 +1.66	75.02 +1.92

Total Training Objective. Finally, our complete model is trained end-to-end by minimizing a total objective function that combines the primary downstream task loss, $\mathcal{L}_{\text{task}}$, with our two dynamics-based regularizers:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \alpha \mathcal{L}_{\text{kp}} + \beta \mathcal{L}_{\text{stab}}, \quad (11)$$

where α and β are hyperparameters that balance the enforcement of coherent dynamics and stability against the task-aware objective.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Datasets We conduct experiments on 2 classification benchmarks: Fine-Grained Visual Classification (FGVC) and VTAB-1k (Zhai et al., 2019). VTAB-1k comprises 19 tasks across *Natural*, *Specialized*, and *Structured* domains, with a constrained budget of 1,000 labeled training samples per task. The FGVC benchmark refers to a collection of five fine-grained classification datasets. Additionally, we evaluate semantic segmentation performance on the ADE20K (Zhou et al., 2017). All 25 datasets follow the training and testing split protocols defined in VPT (Jia et al., 2022)⁴.

Baseline Models To evaluate the effectiveness of PAE, we incorporate PAE into several state-of-the-art VPT variants, including structured prompt designs (VPT (Jia et al., 2022), E2VPT (Han et al., 2023), VQT (Tu et al., 2023), ProVP (Xu et al., 2025), BPT (Wang et al., 2025b)), adaptive prompting (LPT (Dong et al., 2023)), and perception-driven priors (VFPT (Zeng et al., 2024), SA2VP (Pei et al., 2024)). For each method, we report both its original performance and the enhanced version with PAE, denoted as “Baseline $+\Delta$ ”, to isolate the contribution of our framework. Then, we quantify convergence rate with speedup: the baseline epochs to reach its best val acc divided by the epochs required after using PAE. Values >1 mean faster convergence. In addition, we include *Full Fine-tuning*, where all parameters are updated. The experiments are conducted by 2 Transformer architectures: 3 variants (ViT-B/16, -L/16, and -H/14,) of ViT (Dosovitskiy et al., 2020) and Swin-B (Liu et al., 2021), where are pre-trained on ImageNet-21k (Deng et al., 2009). Moreover, we evaluate a self-supervised ViT B/16 backbone: MAE (He et al., 2022), and a segmentation ViT L/16 backbone: SETR (Zheng et al., 2021).

Table 2: Results of ADE20K datasets with ViT-L. SS/MS denote single/multi-scale inference. Values are mIoU with PAE gains.

Methods	Speedup($\times$)	mIoU-SS	mIoU-MS
Full-tuning	-	47.60	49.18
SPT-LoRA	-	45.40	47.50
VPT	1.29 $\times$	44.08 + 2.73	46.01 + 1.96
E2VPT	1.18 $\times$	44.61 + 2.32	46.56 + 2.84
VFPT	1.15 $\times$	45.32 + 2.75	47.17 + 2.09

Implementation Details In MPA, we set the window size $w=16$ with stride $r=8$. We randomly selected a input mini-batch to discover frequency shortcuts. Then, the dimension of Koopman operator was set to 256. The weights were set to $\alpha = 0.5$ for $\mathcal{L}_{\text{kp}}$ and $\beta = 0.2$ for $\mathcal{L}_{\text{stab}}$ ⁵. Cross-entropy (CE) loss was used for $\mathcal{L}_{\text{task}}$. After adding PAE, the initial learning rate was set to 0.25, which was different from the larger learning rate of VPT variants. The batch size was set to 128. All experiments were conducted on an NVIDIA A800 GPU.

⁴Detailed descriptions and statistics of these datasets are provided in Appendix A.4.

⁵More ablation studies on window size, stride, loss weights are provided in Appendix A.5.

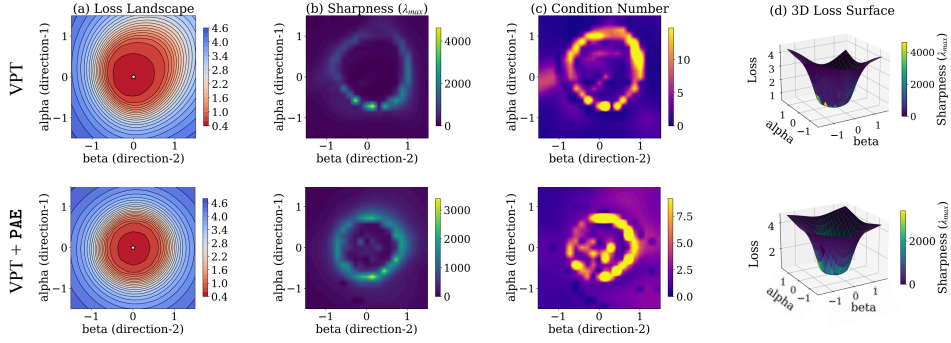


Figure 4: Loss landscape comparisons (Li et al., 2018) show (left to right): 2-D loss contours, sharpness (max Hessian eigenvalue), anisotropy (Hessian condition number), and a 3-D surface colored by curvature.

3.2 RESULTS

The results⁶ on the FGVC and VTAB-1k benchmarks are summarized in Table 1. With only a slight increase in preprocessing time, MPA completes initialization in just 74.17 seconds, roughly equivalent to 5.3 training epochs. Incorporating PAE consistently boosts performance across all settings, underscoring its versatility as a prompt-agnostic enhancement with no inference-time overhead. For instance, integrating PAE into VPT (Jia et al., 2022) raises FGVC accuracy on 91.02% (89.11%+**1.91%**) and VTAB-1k mean accuracy on 74.84% (71.96%+**2.88%**) with a convergence speedup of $1.78\times$. Similarly, when applied to SA2VP (Pei et al., 2024), it improves VTAB-1k performance on 77.49% (75.83%+**1.66%**), and achieves a $1.60\times$ speedup in terms of convergence. Furthermore, adding PAE brings improvements on several VPT variants on ADE20K (Table 2). Specifically, adding PAE to VPT, E2VPT, and VFPT increases mIoU by about 2-3% under both single- and multi-scale evaluation, while also speeding up by roughly 1.15 to $1.29\times$. These results show that PAE is still beneficial for dense prediction tasks such as semantic segmentation.

Furthermore, we visualize the loss landscape on a 2D subspace and report curvature statistics in Figure 4. Applying PAE results in a substantially larger low-loss region with near-circular contours, indicating reduced anisotropy. The Hessian-based sharpness map shows diminished high-curvature rings when PAE is applied, and the condition number heatmap displays uniformly lower values, both suggesting a more isotropic loss surface. The 3D surface plot further illustrates that VPT with PAE converges to a wider and flatter minimum, whereas the baseline VPT remains sharp and narrow. Then, we employ Grad-CAM (Selvaraju et al., 2017) to visualize the impact of integrating PAE into VPT on the CUB dataset, which consists of 200 fine-grained bird species. As shown in Figure 5, at epoch 5, VPT displays diffuse and non-discriminative attention, while VPT+PAE already focuses on task-relevant regions such as the *wing* and *beak* when classifying the Tennessee warbler. By epoch 30, VPT begins to localize the bird, but VPT+PAE delineates its full body with greater precision. At epoch 50, VPT+PAE produces sharper and more consistent saliency maps, indicating improved discriminative localization. These observations collectively indicate that using PAE leads to flatter, more stable minima, enhances discriminative localization, and accelerates convergence.

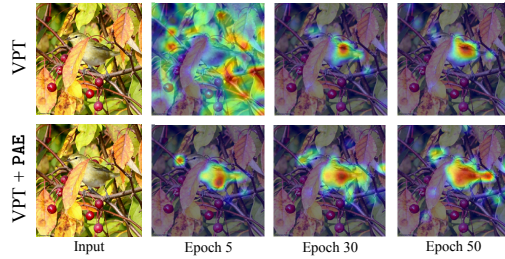


Figure 5: Grad-CAM visualizations at early training stages (epochs 5, 30, and 50) show that VPT+PAE rapidly concentrates attention on class-discriminative regions and stabilizes the saliency patterns, showing faster convergence than VPT.

To assess the efficiency and scalability of PAE across diverse architectures and model sizes, we evaluate VPT, E2VPT, VFPT, and VPT+PAE across three VTAB-1k groups on 4 backbones that vary in different *scale* and *architecture* (ViT-B/16, Swin-B, ViT-L/16, ViT-H/14) in Figure 6. Across all three groups, VPT+PAE consistently improves over VPT on every backbone and remains competitive

⁶Complete per-dataset scores are provided in Appendix A.7.

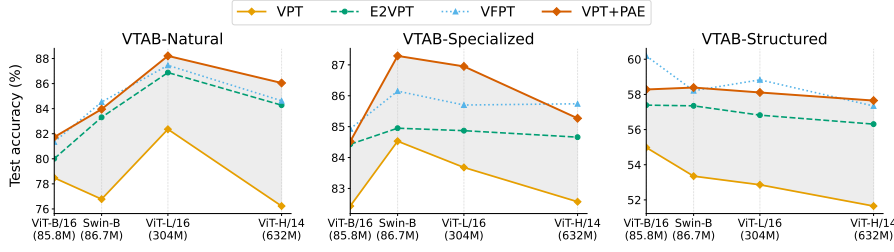


Figure 6: VPT, E2VPT, VFPT, and VPT+PAE across three VTAB groups and four backbones (ViT-B/16 $\rightarrow$ ViT-H/14). The shaded area marks the gap between VPT and VPT+PAE.

with or better than E2VPT and VFPT, indicating that PAE scales smoothly from base to huge ViTs and transfers to other architectures, such as Swin-B.

Beyond performance, we further analyze cross-layer prompt interactions by visualizing prompt CKA (Kornblith et al., 2019) across layers on an MAE backbone, following Yoo et al. (2023). As shown in Figure 7, VPT and VFPT exhibit large blocks of high CKA, reflecting highly redundant prompts and weak depth dependent structure. GatePT (Yoo et al., 2023) slightly smooths these correlations but still maintains globally high similarity across most layers. In contrast, VPT+PAE produces a sharp diagonal band in the CKA matrix, where similarity is strongest locally and gradually decreases with layer distance. This pattern indicates a progressive, depth-aware evolution of the prompt state rather than a globally entangled representation. The diagonal structure aligns with our formulation: a single shared Koopman operator governs prompt dynamics and induces a stable, forward-evolving trajectory across layers.

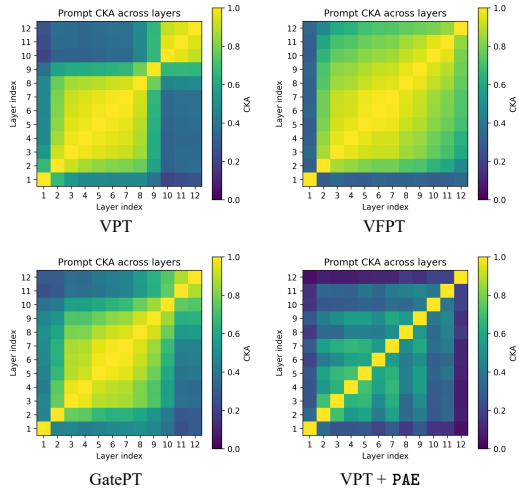


Figure 7: Prompt CKA (Kornblith et al., 2019) across layers for different VPT variants on MAE.

We additionally analyze how PAE behaves on classes with different levels of intra-class variance (Cong et al., 2024) on CUB-200-2011. For each class, we compute its intra-class variance and sort classes along the x-axis from low to high variance. In Fig. 8(a), we plot the per-class accuracy of VPT+PAE against intra-class variance. We observe a mild negative Pearson correlation ($\text{corr} = -0.290$), this shows high-variance classes tend to have lower accuracy, confirming that intra-class variance is a good proxy for class difficulty. In Fig. 8(b), we plot the relative accuracy gain of VPT+PAE over vanilla VPT for each class, using the same ordering by intra-class variance. Here we observe a mild positive correlation ($\text{corr} = 0.207$), this shows although high-variance classes are absolutely harder, they receive larger relative improvements from PAE. This indicates that PAE provides the greatest benefit on the difficult, high-variance categories where VPT struggles most.

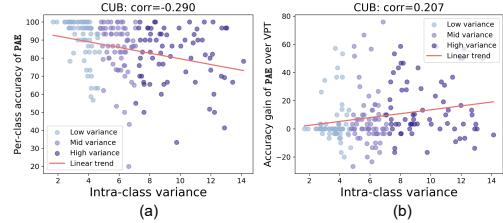


Figure 8: (a) VPT+PAE per-class accuracy vs. intra-class variance. (b) PAE improvement over VPT vs. intra-class variance.

3.3 ABLATION STUDIES

We conduct an ablation study to dissect the PAE framework and quantify the contributions of core components. Results are presented in Table 3. The study reveals two key findings. First, the MPA strategy is the most critical contributor to performance. When used in isolation, MPA yields a +2.06% improvement on the VTAB-1k mean score over the baseline. This gain significantly exceeds that of using only the Koopman consistency loss ($\mathcal{L}_{kp}$). Second, $\mathcal{L}_{kp}$ and the Lyapunov stability loss $\mathcal{L}_{stab}$

Table 3: Ablation of PAE core elements on ViT-Base/16 using VPT (Jia et al., 2022) as baseline.

MPA(init)	KLD ($\mathcal{L}_{kp}$)	KLD ($\mathcal{L}_{stab}$)	FGVC	VTAB-1k			
				<i>Natural</i>	<i>Specialized</i>	<i>Structured</i>	Mean
$\times$	$\times$	$\times$	89.11	78.48	82.43	54.98	71.96
$\checkmark$	$\times$	$\times$	89.63	81.35	84.07	56.64	74.02
$\times$	$\checkmark$	$\times$	90.56	80.27	83.28	55.82	73.13
$\times$	$\checkmark$	$\checkmark$	90.78	81.46	83.85	57.96	74.42
$\checkmark$	$\checkmark$	$\checkmark$	91.02	81.73	84.52	58.28	74.84

Table 4: Ablation studies on ViT-B/16, using VPT with PAE: (Left) Comparison across different prompt initialization strategies. (Right) Variants of frequency shortcut injection.

(a) Prompt Initialization Strategies						(b) Frequency-Shortcut Injection Variants					
Method	FGVC	VTAB-1k				Variant	FGVC	VTAB-1k			
		<i>Natural</i>	<i>Specialized</i>	<i>Structured</i>	Mean			<i>Natural</i>	<i>Specialized</i>	<i>Structured</i>	Mean
Random	89.11	78.48	82.43	54.98	71.96	Shallow	89.37	79.14	82.36	55.47	72.32
Xavier	89.55	80.67	82.32	53.91	72.30	Copy	89.83	80.05	83.11	56.34	73.17
UnPro	90.23	80.46	83.18	57.89	73.85	Layer-wise	89.86	81.27	83.68	57.93	74.29
MPA (Ours)	91.02	81.73	84.52	58.28	74.84	Shared (Ours)	91.02	81.73	84.52	58.28	74.84

are also essential and demonstrate strong synergy. While $\mathcal{L}_{kp}$ alone yields notable improvement, adding $\mathcal{L}_{stab}$ results in a further +1.29% gain. The full PAE framework, integrating all three components, achieves the highest performance, confirming that each part plays a complementary role in addressing the underlying optimization challenges.

To further validate the effectiveness of MPA, we compare it with several commonly used initialization strategies, as shown in Table 4(a). For fair comparison, all methods use the $\mathcal{L}_{kp}$ and $\mathcal{L}_{stab}$. Random initialization yields a VTAB-1k mean score of 71.96%, while Xavier initialization (Xu et al., 2025) slightly improves this to 72.30% but lacks task-aware alignment. A stronger baseline uses features derived from Unsupervised Prototypes (Wang et al., 2024), achieving 73.85% by leveraging general-purpose visual representations. In contrast, MPA consistently outperforms all these alternatives, reaching 74.84%. This highlights its unique advantage: rather than relying on generic representations, MPA conducts a task-aware search to identify frequency shortcuts that are most relevant to the target task.

We further investigate how prompts are propagated across layers in MPA in Table 4(b). After generating frequency shortcut masks, MPA constructs the first-layer prompt and propagates it through the frozen backbone to initialize prompts in subsequent layers—achieving the best performance at 74.84% VTAB-1k mean. This outperforms two alternative designs. First, copying the first-layer prompt to all layers yields 73.17%, indicating that building a layer-specific prompt hierarchy is crucial. Second, performing a layer-wise independent search achieves 74.29%, which still underperforms MPA. These findings show the effectiveness of our single-search and propagate strategy.

With batch size B and hyperparameters fixed ($w=16$, $r=8$, $B=128$), we examine the effect of randomly selected input batches on MPA, as shown in Table 5. Batch $B^{(1)}$ is used as the baseline, and paired t -tests are performed against it. Deviations across task categories remain small: $FGVC \leq 0.40$, $Natural \leq 0.39$, $Specialized \leq 0.44$, and $Structured \leq 0.38$, yielding $p \in [0.008, 0.092]$. These results demonstrate that the randomness introduced by selecting different input batches for MPA has a negligible impact on performance.

Finally, we study the effect of the Koopman operator’s latent dimension K . As shown in Figure 9, K controls the expressivity of the linear dynamics model. The results show a clear trade-off: a small dimension (e.g., $K = 64$) leads to underfitting and poor performance (67.11%). As K increases, performance improves, peaking at $K = 256$ with a mean VTAB-1k score of 74.84%. However, further increasing K to 384 degrades performance (68.24%), suggesting that an overly large latent

Table 5: Ablation on different input mini-batches using VPT+PAE.

Mini-batch	FGVC	VTAB-1k				p-value
		<i>Natural</i>	<i>Specialized</i>	<i>Structured</i>	Mean	
$B^{(1)}$	91.02	81.73	84.52	58.28	74.84	-
$B^{(2)}$	90.62	81.47	84.54	57.95	74.65	0.081
$B^{(3)}$	91.07	81.39	84.23	57.92	74.51	0.092
$B^{(4)}$	90.98	82.01	84.96	58.66	75.21	0.089
$B^{(5)}$	90.64	81.34	84.14	57.91	74.46	0.008

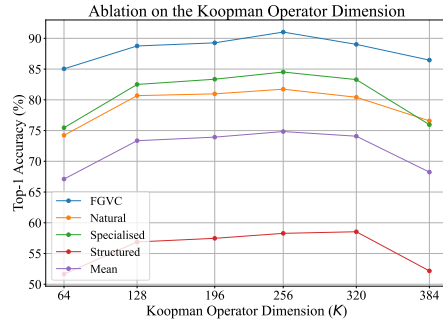


Figure 9: Ablation on the Koopman operator dimension in VPT+PAE on ViT-B/16.

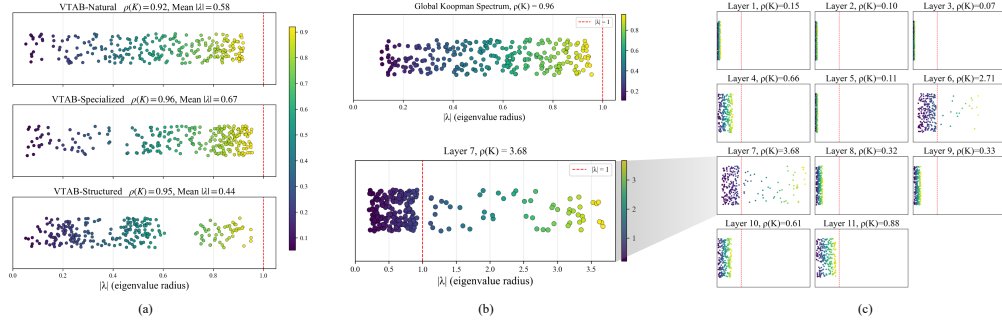


Figure 10: (a) Global Koopman spectra computed for the Natural, Specialized, and Structured sub-sets of VTAB-1k. (b) Comparison between the spectrum of the global Koopman operator (top) and that of the layer-wise operator at layer 7 (bottom). (c) The layer-wise design exhibits the individual spectra for each layer-specific operator.

space introduces optimization challenges and increases the risk of overfitting. Based on this analysis, we set $K = 256$ as the default configuration.

4 ANALYSIS

According to classical linear systems theory, Eq.7 is globally asymptotically stable if the spectral radius $\rho(\mathbf{K}) < 1$ (Guglielmi & Overton, 2011). In Koopman theory, eigenvalue magnitude dictates persistence: values near one decay slowly, while small ones vanish quickly (Budišić et al., 2012; Williams et al., 2015). In practice, we analyze the spectrum via $\mathbf{K}v = \lambda v$. Specifically, Eq.8 ($\mathcal{L}_{kp}$) is used in the shared latent space, and Eq.10 ($\mathcal{L}_{stab}$) penalizes growth of the energy. These losses favor contraction without rotation and concentrate the spectrum on the positive real axis.

We first compare the global Koopman spectra learned by PAE on VTAB-1k benchmark from the Natural, Specialized, and Structured groups. As shown in Fig. 10(a), all three learned global operators are stable with $\rho(\mathbf{K}) < 1$, but their spectra exhibit clear group-specific patterns. Specifically, Natural tasks show a moderate mean radius (mean $|\lambda| = 0.58$), indicating that PAE only needs to gently reshape pretrained features when transferring to other natural-image tasks. Specialized tasks attain the largest mean radius (mean $|\lambda| = 0.67$) with slow modes near unity, suggesting long effective memory for strong domain shifts. This helps cope with stronger domain shifts such as histopathology and remote sensing. In contrast, structured tasks yield the smallest radius (mean $|\lambda| = 0.44$), consistent with aggressive damping of texture variations. These group-dependent spectra reveal dynamical patterns in how PAE adapts its cue dynamics to different downstream tasks.

Next, we compare the spectrum of the global operator learned by PAE against a layer-wise design, in which an independent learnable operator $\mathbf{K}_i$ is assigned to each layer transition $i \rightarrow i+1$. Fig. 10(b) shows that the global operator concentrates eigenvalues along the positive real axis with $\rho(\mathbf{K}) < 1$, ensuring coherent evolution. In contrast, the eigenvalue radius at layer 7 exceeds 3, reaching 3.68. As shown in Fig. 10(c), the layer-wise variant yields highly heterogeneous dynamics. Several layers exhibit large spectral radii (eigenvalue radius ≥ 1), introducing mismatched time scales and unstable modes that are significantly harder to regularize.

5 CONCLUSION

In this paper, we reframe visual prompt tuning as a discrete dynamical system and introduce PAE, coupling MPA and KLD. MPA employs a supervision mechanism to find task-aware frequency shortcuts, which convert into initial prompts. KLD applies a shared Koopman evolution with a Lyapunov-style regularizer. This establishes cross-layer dependencies explicit, stabilizes gradients, accelerates convergence, and improves performance. Across multiple variants and image-classification benchmarks, PAE improves accuracy and convergence. In particular, PAE remains prompt-agnostic: it drops into various VPT variants with no backbone changes and no inference-time overhead, offering a practical and scalable path to improve visual prompt tuning. **Limitation:** MPA may be less suitable for low- or zero-label regimes unless appropriate surrogate signals are available.

REFERENCES

- Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4), 2012.
- Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. Tinytl: Reduce memory, not parameters for efficient on-device learning. *NeurIPS*, 33:11285–11297, 2020.
- Cong Cong, Shiyu Xuan, Sidong Liu, Shiliang Zhang, Maurice Pagnucco, and Yang Song. Decoupled optimisation for long-tailed visual recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 1380–1388, 2024.
- Cong Cong, Yang Song, Antonio Di Ieva, Angela Chou, Anthony J Gill, Enrico Coiera, and Sidong Liu. Flexible and scalable federated learning with deep feature prompts for digital pathology. 2025.
- 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.
- Bowen Dong, Pan Zhou, Wangmeng Zuo, et al. Lpt: Long-tailed prompt tuning for image classification. In *The Eleventh International Conference on Learning Representations*, 2023.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.
- Lei Fan, Dongdong Fan, Zhiguang Hu, Yiwen Ding, Donglin Di, Kai Yi, Maurice Pagnucco, and Yang Song. Manta: A large-scale multi-view and visual-text anomaly detection dataset for tiny objects. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 25518–25527, 2025a.
- Lei Fan, Junjie Huang, Donglin Di, Anyang Su, Tianyou Song, Maurice Pagnucco, and Yang Song. Salvaging the overlooked: Leveraging class-aware contrastive learning for multi-class anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21419–21428, 2025b.
- Zhengyi Fang, Yue Wang, Ran Yi, and Lizhuang Ma. Dropout mixture low-rank adaptation for visual parameters-efficient fine-tuning. In *European Conference on Computer Vision*, pp. 369–386. Springer, 2024.
- Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, and Li Fei-Fei. Fine-grained car detection for visual census estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.
- Nicola Guglielmi and Michael L Overton. Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1166–1192, 2011.
- Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse problems*, 34(1):014004, 2017.
- Cheng Han, Qifan Wang, Yiming Cui, Zhiwen Cao, Wenguan Wang, Siyuan Qi, and Dongfang Liu. E²vpt: An effective and efficient approach for visual prompt tuning. *arXiv preprint arXiv:2307.13770*, 2023.
- Haoyu He, Jianfei Cai, Jing Zhang, Dacheng Tao, and Bohan Zhuang. Sensitivity-aware visual parameter-efficient fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11825–11835, 2023.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

- 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*, pp. 16000–16009, 2022.
- 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 *ICML*, pp. 2790–2799. PMLR, 2019.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022.
- 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.
- Zhaolu Kang, Junhao Gong, Jiaxu Yan, Wanke Xia, Yian Wang, Ziwen Wang, Huaxuan Ding, Zhuo Cheng, Wenhao Cao, Zhiyuan Feng, et al. Hssbench: Benchmarking humanities and social sciences ability for multimodal large language models. *arXiv preprint arXiv:2506.03922*, 2025.
- Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, 2011.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMIR, 2019.
- Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.
- Yiyang Liu, James C Liang, Heng Fan, Wenhao Yang, Yiming Cui, Xiaotian Han, Lifu Huang, Dongfang Liu, Qifan Wang, and Cheng Han. All you need is one: Capsule prompt tuning with a single vector. *arXiv preprint arXiv:2510.16670*, 2025.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.
- 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14845–14857, 2025.
- Gabriele Merlin, Vedant Nanda, Ruchit Rawal, and Mariya Toneva. What happens during finetuning of vision transformers: An invariance based investigation. In *Conference on Lifelong Learning Agents*, pp. 601–619. PMLR, 2023.
- Igor Mezić. Analysis of fluid flows via spectral properties of the koopman operator. *Annual review of fluid mechanics*, 45(1):357–378, 2013.
- Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, 2008.
- Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pp. 69–84. Springer, 2016.
- Wenjie Pei, Tongqi Xia, Fanglin Chen, Jinsong Li, Jiandong Tian, and Guangming Lu. Sa²vp: Spatially aligned-and-adapted visual prompt. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 4450–4458, 2024.
- Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapter-fusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020a.

- Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pp. 46–54, Online, 2020b. Association for Computational Linguistics.
- Friedrich Philipp, Manuel Schaller, Karl Worthmann, Sebastian Peitz, and Feliks Nüske. Error bounds for kernel-based approximations of the koopman operator. *Applied and Computational Harmonic Analysis*, 71, 2024.
- Li Ren, Chen Chen, Liqiang Wang, and Kien Hua. Da-vpt: Semantic-guided visual prompt tuning for vision transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4353–4363, 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.
- Daniel Shevitz and Brad Paden. Lyapunov stability theory of nonsmooth systems. *IEEE Transactions on automatic control*, 39(9):1910–1914, 1994.
- Kihyuk Sohn, Huiwen Chang, José Lezama, Luisa Polania, Han Zhang, Yuan Hao, Irfan Essa, and Lu Jiang. Visual prompt tuning for generative transfer learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19840–19851, 2023.
- Cheng-Hao Tu, Zheda Mai, and Wei-Lun Chao. Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7725–7735, 2023.
- Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2015.
- Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- Shunxin Wang, Raymond Veldhuis, Christoph Brune, and Nicola Strisciuglio. What do neural networks learn in image classification? a frequency shortcut perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1433–1442, 2023.
- Shunxin Wang, Raymond Veldhuis, and Nicola Strisciuglio. Do imagenet-trained models learn shortcuts? the impact of frequency shortcuts on generalization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 25198–25207, 2025a.
- 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*, pp. 50233–50247, 2024.
- Yuzhu Wang, Manni Duan, and Shu Kong. Attention to the burstiness in visual prompt tuning! In *International Conference on Computer Vision (ICCV)*, 2025b.
- Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 139–149, 2022.
- Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

- 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, Yunbei Zhang, Lin Zhao, Yiyang Liu, Xiaoying Liao, Zheda Mai, Xingjian Li, Xiao Wang, Hao Xu, Jihun Hamm, et al. Prompt-based adaptation in large-scale vision models: A survey. *arXiv preprint arXiv:2510.13219*, 2025c.
- Chen Xu, Yuhan Zhu, Haocheng Shen, Boheng Chen, Yixuan Liao, Xiaoxin Chen, and Limin Wang. Progressive visual prompt learning with contrastive feature re-formation. *International Journal of Computer Vision*, 133(2):511–526, 2025.
- Wenhao Xu, Rongtao Xu, Changwei Wang, Shibiao Xu, Li Guo, Man Zhang, and Xiaopeng Zhang. Spectral prompt tuning: Unveiling unseen classes for zero-shot semantic segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 6369–6377, 2024.
- 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.
- Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *NeurIPS*, 27, 2014.
- 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 Nian 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 Ruysen, 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 *ECCV*, pp. 698–714. Springer, 2020.
- Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pp. 649–666. Springer, 2016.
- Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6881–6890, 2021.
- Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.
- Shihao Zhou, Jinshan Pan, Jinglei Shi, Duosheng Chen, Lishen Qu, and Jufeng Yang. Seeing the unseen: A frequency prompt guided transformer for image restoration. In *European Conference on Computer Vision*, pp. 246–264. Springer, 2024.
- Zhijie Zhu, Lei Fan, Maurice Pagnucco, and Yang Song. Interpretable image classification via non-parametric part prototype learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9762–9771, 2025.

A APPENDIX

A.1 THE USES OF LLM

We used a large language model (LLM) only for language polishing (grammar, clarity, and style). All ideas, analyses, and conclusions are the authors' own, and the authors conducted the final review and approval.

A.2 GRADIENT FOR THE KOOPMAN CONSISTENCY LOSS

Here we provide the detailed derivation for the gradient of the Koopman consistency loss $\mathcal{L}_{\text{kp}}$ with respect to an intermediate prompt $\mathbf{P}_i$. The loss is given by:

$$\mathcal{L}_{\text{kp}} = \sum_{i=1}^{L-1} \left\| \underbrace{\mathbf{z}_{i+1}}_{\mathbf{P}_{i+1}\mathbf{U}} - \underbrace{\mathbf{z}_i\mathbf{K}}_{\hat{\mathbf{z}}_{i+1}} \right\|_F^2. \quad (12)$$

The prompt $\mathbf{P}_i$ (and its Koopman representation $\mathbf{z}_i = \mathbf{P}_i\mathbf{U}$) influences two adjacent terms in this sum: the term for the transition from $i-1$ to i , which we denote $l_{i-1} = \|\mathbf{z}_i - \mathbf{z}_{i-1}\mathbf{K}\|_F^2$, and the term for the transition from i to $i+1$, denoted $l_i = \|\mathbf{z}_{i+1} - \mathbf{z}_i\mathbf{K}\|_F^2$.

The gradient is the sum of the gradients of these two terms. First, we compute the gradient with respect to $\mathbf{z}_i$:

$$\frac{\partial l_{i-1}}{\partial \mathbf{z}_i} = 2(\mathbf{z}_i - \mathbf{z}_{i-1}\mathbf{K}) \quad (13)$$

$$\frac{\partial l_i}{\partial \mathbf{z}_i} = -2(\mathbf{z}_{i+1} - \mathbf{z}_i\mathbf{K})\mathbf{K}^\top \quad (14)$$

Combining these gives the full gradient with respect to $\mathbf{z}_i$:

$$\frac{\partial \mathcal{L}_{\text{kp}}}{\partial \mathbf{z}_i} = 2(\mathbf{z}_i - \mathbf{z}_{i-1}\mathbf{K}) - 2(\mathbf{z}_{i+1} - \mathbf{z}_i\mathbf{K})\mathbf{K}^\top = 2(\mathbf{z}_i - \hat{\mathbf{z}}_i) - 2(\mathbf{z}_{i+1} - \hat{\mathbf{z}}_{i+1})\mathbf{K}^\top. \quad (15)$$

Finally, using the chain rule with $\mathbf{z}_i = \mathbf{P}_i\mathbf{U}$, we obtain the gradient with respect to the prompt $\mathbf{P}_i$:

$$\frac{\partial \mathcal{L}_{\text{kp}}}{\partial \mathbf{P}_i} = \frac{\partial \mathcal{L}_{\text{kp}}}{\partial \mathbf{z}_i} \frac{\partial \mathbf{z}_i}{\partial \mathbf{P}_i} = [2(\mathbf{z}_i - \hat{\mathbf{z}}_i) - 2(\mathbf{z}_{i+1} - \hat{\mathbf{z}}_{i+1})\mathbf{K}^\top] \mathbf{U}^\top. \quad (16)$$

This final gradient provides the direct supervisory signal used to update the prompts.

A.3 GRADIENT FOR THE LYAPUNOV STABILITY REGULARIZER

The Lyapunov stability regularizer is defined as:

$$\mathcal{L}_{\text{stab}} = \sum_{i=1}^L \max(0, \Delta V_i), \quad \text{where} \quad \Delta V_i \triangleq V(\mathbf{z}_{i+1}) - V(\mathbf{z}_i). \quad (17)$$

The Lyapunov function is $V(\mathbf{z}) = \text{tr}(\mathbf{z}\mathbf{Q}\mathbf{z}^\top)$. Its gradient with respect to $\mathbf{z}$ is $\partial V(\mathbf{z})/\partial \mathbf{z} = 2\mathbf{z}\mathbf{Q}$, given that $\mathbf{Q}$ is symmetric.

To handle the non-differentiability of the $\max(0, x)$ function at $x = 0$, we can use an indicator function in the derivation. Let $\eta_i = \mathbf{1}_{\{\Delta V_i > 0\}}$ be an indicator that is 1 if the error increases from step i to $i+1$, and 0 otherwise. The loss can be expressed as $\mathcal{L}_{\text{stab}} = \sum_{i=1}^L \eta_i \Delta V_i$.

Let's compute the gradient with respect to an intermediate state $\mathbf{z}_i$ (where $0 < i < L$). The state $\mathbf{z}_i$ influences two terms in the sum: $\eta_{i-1}\Delta V_{i-1} = \eta_{i-1}(V(\mathbf{z}_i) - V(\mathbf{z}_{i-1}))$ and $\eta_i\Delta V_i = \eta_i(V(\mathbf{z}_{i+1}) - V(\mathbf{z}_i))$. The gradient is therefore:

$$\begin{aligned} \frac{\partial \mathcal{L}_{\text{stab}}}{\partial \mathbf{z}_i} &= \frac{\partial}{\partial \mathbf{z}_i} \left[\eta_{i-1}(V(\mathbf{z}_i) - V(\mathbf{z}_{i-1})) \right] + \frac{\partial}{\partial \mathbf{z}_i} \left[\eta_i(V(\mathbf{z}_{i+1}) - V(\mathbf{z}_i)) \right] \\ &= \eta_{i-1} \frac{\partial V(\mathbf{z}_i)}{\partial \mathbf{z}_i} - \eta_i \frac{\partial V(\mathbf{z}_i)}{\partial \mathbf{z}_i} \\ &= 2\eta_{i-1}\mathbf{z}_i\mathbf{Q} - 2\eta_i\mathbf{z}_i\mathbf{Q} \\ &= 2(\eta_{i-1} - \eta_i)\mathbf{z}_i\mathbf{Q}. \end{aligned} \quad (18)$$

This resulting gradient for $\mathbf{z}_i$ is non-zero only if the stability condition changes at the boundary of step i (i.e., violated for step $i - 1$ but not for step i , or vice-versa).

Finally, using the chain rule with $\mathbf{z}_i = \mathbf{P}_i \mathbf{U}$, we obtain the gradient with respect to the prompt $\mathbf{P}_i$:

$$\frac{\partial \mathcal{L}_{\text{stab}}}{\partial \mathbf{P}_i} = \frac{\partial \mathcal{L}_{\text{stab}}}{\partial \mathbf{z}_i} \frac{\partial \mathbf{z}_i}{\partial \mathbf{P}_i} = 2(\eta_{i-1} - \eta_i) \mathbf{z}_i \mathbf{Q} \mathbf{U}^\top. \quad (19)$$

This provides the corrective signal to update the prompts, active only when necessary to enforce the error constraint.

A.4 DETAILED DATASET DESCRIPTIONS

We provide comprehensive statistics for the datasets used in our evaluation, corresponding to the experimental setup described in the main text.

Fine-Grained Visual Classification (FGVC). This benchmark includes five datasets that require distinguishing between highly similar subordinate categories. The specific data splits are as follows:

- **CUB-200-2011** (Wah et al., 2011): 200 classes (5,394 train, 600 val, 5,794 test).
- **NABirds** (Van Horn et al., 2015): 55 classes (21,536 train, 2,393 val, 24,633 test).
- **Oxford Flowers-102** (Nilsback & Zisserman, 2008): 102 classes (1,020 train, 1,020 val, 6,149 test).
- **Stanford Dogs** (Khosla et al., 2011): 120 classes (10,800 train, 1,200 val, 8,580 test).
- **Stanford Cars** (Gebru et al., 2017): 196 classes (7,329 train, 815 val, 8,041 test).

VTAB-1k Benchmark. The Visual Task Adaptation Benchmark (VTAB-1k) (Zhai et al., 2019) limits the label budget to 1,000 training samples per task (800 for training, 200 for validation). It consists of 19 tasks grouped into three categories:

- **Natural:** CIFAR-100 (10,000 test), Caltech101 (6,084 test), DTD (1,880 test), Flowers102 (6,149 test), Pets (3,669 test), SVHN (26,032 test), and SUN397 (21,750 test).
- **Specialized:** Patch Camelyon (32,768 test), EuroSAT (5,400 test), RESISC45 (6,300 test), and Retinopathy (42,670 test).
- **Structured:** CLEVR/count (15,000 test), CLEVR/distance (15,000 test), DMLab (22,735 test), KITTI/distance (711 test), dSprites/location (73,728 test), dSprites/orientation (73,728 test), SmallNORB/azimuth (12,150 test), and SmallNORB/elevation (12,150 test).

Semantic Segmentation. For dense prediction tasks, we utilize the **ADE20K** (Zhou et al., 2017) dataset. Train/val splits were followed the same split protocol as in our main experimental setup.

A.5 ABLATION STUDIES ON HYPERPARAMETERS

As shown in Table 6, the frequency-grid hyperparameters show that ($w=16$, $r=8$) attains the highest cross-benchmark Mean = 74.84 and is best (or tied) on FGVC 91.02 and *Natural* 81.73. Although per-track optima differ—*Specialized* peaks at (16, 12) with 85.07 and *Structured* peaks at (24, 12) with 59.11, we choose ($w=16$, $r=8$) as the final configuration *because it maximizes the Mean*, which is our selection criterion across benchmarks. Unless otherwise stated, subsequent experiments use $\alpha=0.5$, $\beta=0.2$ and ($w=16$, $r=8$).

As shown in Table 7, introducing moderate loss weights consistently improves over the no-auxiliary baseline ($\alpha=\beta=0$, Mean = 71.96). The best average appears at $\alpha=0.5$, $\beta=0.2$ with Mean = 74.84 (+2.88), alongside gains on all tracks: FGVC 91.02 (+1.91), *Natural* 81.73 (+3.25), *Specialized* 84.52 (+2.09), and *Structured* 58.28 (+3.30). Further increasing α (e.g., $\alpha=0.9$, $\beta=0.3$) degrades the average (73.52), indicating over-regularization.

Table 6: Ablation on frequency window size w and stride r for ViT-B/16 with VPT+PAE. *Bold* denotes the best in column.

w	r	FGVC	VTAB-1k			
			<i>Natural</i>	<i>Specialized</i>	<i>Structured</i>	Mean
8	4	89.28	79.76	82.71	55.29	72.59
16	8	91.02	81.73	84.52	58.28	74.84
16	12	90.37	79.84	85.07	57.73	74.21
24	12	90.41	80.43	82.96	59.11	74.16
24	16	90.15	79.97	83.56	58.62	74.05

Table 7: Ablation on loss weights (α, β) for ViT-B/16 with VPT+PAE. *Bold* denotes the best in column.

α	β	FGVC	VTAB-1k			
			<i>Natural</i>	<i>Specialized</i>	<i>Structured</i>	Mean
0.0	0.0	89.11	78.48	82.43	54.98	71.96
0.1	0.1	89.58	79.02	83.01	55.67	72.57
0.3	0.1	90.72	80.31	84.24	57.83	74.13
0.5	0.2	91.02	81.73	84.52	58.28	74.84
0.7	0.2	90.63	82.18	84.17	58.09	74.81
0.9	0.3	89.37	79.95	82.90	57.71	73.52

A.6 SENSITIVITY ANALYSIS OF CONVERGENCE SPEED.

To rigorously evaluate the robustness of our hyperparameter selection, we analyze the impact of the regularization weights (α, β) on the convergence speed across four distinct benchmarks: FGVC, VTAB-Natural, VTAB-Specialized, and VTAB-Structured. As illustrated in Figure 11, a consistent **U-shaped correlation** is observed between the regularization strength and the number of epochs required for convergence across all task groups.

Specifically, the experimental results reveal three key regimes centering around an optimal configuration. The fastest convergence is uniformly achieved at the “**Sweet Spot**” setting of $(\alpha = 0.5, \beta = 0.2)$. In this region, the stability induced by PAE allows for aggressive optimization, reducing the training duration by approximately 40% $\sim$ 50% compared to the baseline $(\alpha = 0, \beta = 0)$. Deviating from this optimum compromises efficiency. In the **Under-regularization** regime (e.g., 0.1, 0.1), weak constraints fail to effectively suppress high-frequency gradient oscillations, leaving the optimization trajectory jagged and resulting in negligible epoch reduction. Conversely, **Over-regularization** (e.g., 0.9, 0.3) leads to a sharp rebound in training costs due to excessive constraints. This is particularly evident in the *FGVC* group, where “over-smoothing” prompts detrimentally affects the learning of fine-grained discriminative features, prolonging convergence to 96 epochs.

Notably, the *Natural* and *Specialized* groups exhibit the steepest improvements, benefiting most from the stabilized dynamics. While the *Structured* tasks show a flatter improvement curve due to their inherent optimization difficulty, they still achieve optimal efficiency at the same parameter configuration, confirming the universality of our default setup.

A.7 FULL EXPERIMENTAL RESULTS

For FGVC in Figure 8, we report per-dataset top-1 accuracy and convergence epoch for all evaluated methods, including baseline VPT variants and their PAE-augmented counterparts. For VTAB-1k, we provide per-task scores for every task: Natural in Figure 9, Specialized in Figure 10, and Structured in Figure 11, along with the group means and the overall mean. Training and evaluation settings match those in the main text unless otherwise noted.

We compare both variants of VPT with other commonly used fine-tuning protocols:

- Fully fine-tune all backbone and classification head parameters.
- Methods that focus on the classification head. They treat the pre-trained backbone as a feature extractor, whose weights are fixed during tuning:
 - Linear: only use a linear layer as the classification head.

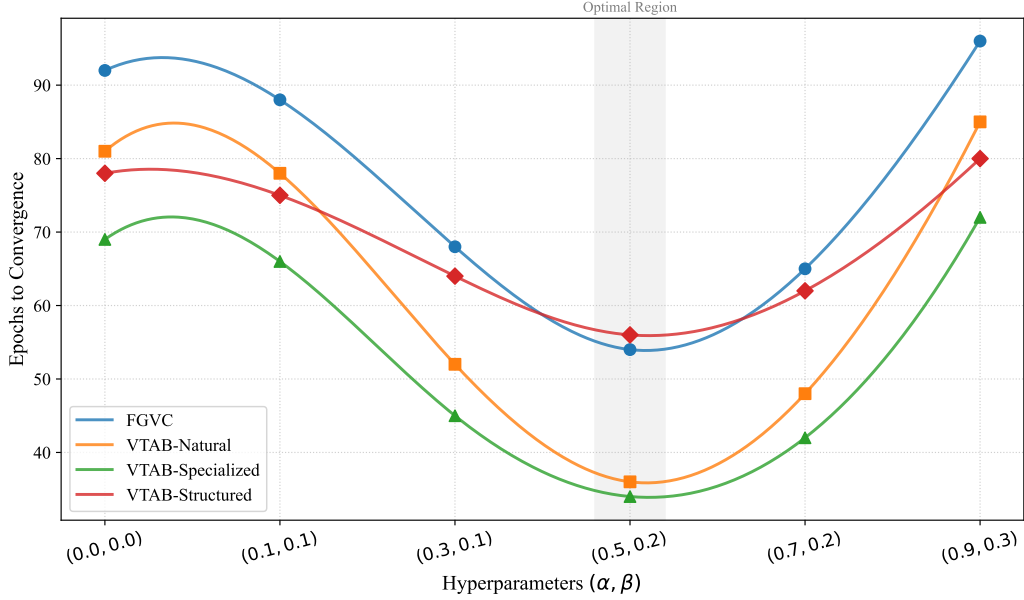


Figure 11: Sensitivity analysis of convergence speed with respect to hyperparameters (α, β) .

- **Partialft- k** : fine-tune the last k layers of the backbone while freezing the others, as adopted in (Yosinski et al., 2014; Zhang et al., 2016; Noroozi & Favaro, 2016; He et al., 2022). This redefines the boundary between the backbone and the classification head.
 - **Mlp- k** : utilize a multilayer perceptron (MLP) with k layers, instead of a linear layer, as the classification head.
- (c) Methods that update a subset of backbone parameters or add new trainable parameters to the backbone during fine-tuning:
- **Sidetune** (Zhang et al., 2020): train a “side” network and linearly interpolate between pre-trained features and side-tuned features before feeding them into the head.
 - **Bias** (Cai et al., 2020; Zaken et al., 2022): fine-tune only the bias terms of a pre-trained backbone.
 - **Adapter** (Houlsby et al., 2019; Pfeiffer et al., 2020a;b): insert new MLP modules with residual connections inside Transformer layers.
 - **LoRA** (Hu et al., 2022; He et al., 2023; Fang et al., 2024): insert low-rank trainable matrices into the projection layers while freezing the original backbone weights, enabling parameter-efficient adaptation.

Table 8: FGVC per-task results for ViT-Base/16 pretrained on supervised ImageNet-21k.

ViT-Base/16	FGVC					Mean	Convergence epoch
	CUB-200-2011	NABirds	Oxford Flowers	Stanford Dogs	Stanford Cars		
FULL	87.3	82.7	98.8	89.4	84.5	88.54	-
LINEAR	85.3	75.9	97.9	86.2	51.3	79.32	-
PARTIAL-1	85.6	77.8	98.2	85.5	66.2	82.63	-
MLP-2	85.7	77.2	98.2	85.4	54.9	80.28	-
MLP-3	85.1	77.3	97.9	84.9	53.8	79.80	-
MLP-5	84.2	76.7	97.6	84.8	50.2	78.71	-
MLP-9	83.2	76.0	96.2	83.7	47.6	77.31	-
SIDETUNE	84.7	75.8	96.9	85.8	48.6	78.35	-
BIAS	88.4	84.2	98.8	91.2	79.4	88.41	-
ADAPTER-256	87.2	84.3	98.5	89.9	68.6	85.70	-
ADAPTER-64	87.1	84.3	98.5	89.8	68.6	85.67	-
ADAPTER-8	87.3	84.3	98.4	88.8	68.4	85.46	-
LoRA	88.3	85.6	99.2	91.0	83.2	89.46	-
SPT-LoRA	88.6	83.4	99.5	91.4	87.3	90.04	-
DM-LoRA	89.8	86.6	99.5	91.8	85.7	90.68	-
VPT	88.5	84.2	99.0	90.2	83.6	89.11	92
VPT + PAE	89.7	87.4	99.2	92.0	86.8	91.02	54
E2VPT	89.1	84.6	99.1	90.5	82.8	89.22	96
E2VPT + PAE	90.3	87.4	99.3	91.4	86.4	90.96	58
LPT	89.6	86.1	99.0	90.9	84.2	89.96	87
LPT + PAE	90.8	87.6	99.4	91.7	87.2	91.32	58
VQT	88.9	85.5	99.2	90.2	83.1	89.41	85
VQT + PAE	90.0	86.6	99.1	91.5	85.9	90.62	56
VFPT	88.7	84.5	99.1	90.4	83.6	89.24	76
VFPT + PAE	90.5	88.4	99.3	92.7	86.5	91.48	61
SA2VP	89.1	85.8	99.3	92.1	84.1	90.08	81
SA2VP + PAE	89.3	88.1	99.5	92.0	87.1	91.20	50
ProVP	89.6	84.9	99.0	90.5	83.8	89.56	84
ProVP + PAE	90.4	87.3	99.2	92.3	86.7	91.18	70
BPT	90.2	87.5	99.7	90.1	86.8	90.86	93
BPT + PAE	91.2	89.0	99.6	91.9	89.4	92.21	63

Table 9: VTAB-1k *Natural* per-task results for ViT-Base/16 pretrained on supervised ImageNet-21k.

ViT-Base/16	VTAB-1k <i>Natural</i>							Mean	Convergence epoch
	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	Sun397		
FULL	68.9	87.7	64.3	97.2	86.9	87.4	38.8	75.88	-
LINEAR	63.4	85.0	63.2	97.0	86.3	36.6	51.0	68.93	-
PARTIAL-1	66.8	85.9	62.5	97.3	85.5	37.6	50.6	69.44	-
MLP-2	63.2	84.8	60.5	97.6	85.9	34.1	47.8	67.70	-
MLP-3	63.8	84.7	62.3	97.4	84.7	32.5	49.2	67.80	-
MLP-5	59.3	84.4	59.9	96.1	84.4	30.9	46.8	65.98	-
MLP-9	53.1	80.5	53.9	95.1	82.6	24.4	43.7	61.90	-
SIDETUNE	60.7	60.8	53.6	95.5	66.7	34.9	35.3	58.21	-
BIAS	72.8	87.0	59.2	97.5	85.3	59.9	51.4	73.30	-
ADAPTER-256	74.1	86.1	63.2	97.7	87.0	34.6	50.8	70.50	-
ADAPTER-64	74.2	85.8	62.7	97.6	87.2	36.3	50.9	70.65	-
ADAPTER-8	74.2	85.7	62.7	97.8	87.2	36.4	50.7	70.67	-
LoRA	67.1	91.4	69.4	98.8	90.4	85.3	54.0	79.46	-
SPT-LoRA	73.5	93.3	72.5	99.3	91.5	87.9	55.5	81.93	-
DM-LoRA	74.0	90.7	73.9	99.3	92.2	91.1	56.4	82.51	-
VPT	78.8	90.8	65.8	98.0	88.3	78.1	49.6	78.48	81
VPT + PAE	79.5	92.3	70.4	98.6	91.7	88.2	51.4	81.73	36
E2VPT	78.6	89.4	67.8	98.2	88.5	85.3	52.3	80.01	91
E2VPT + PAE	79.1	91.3	69.4	98.0	89.8	87.6	54.5	81.39	59
LPT	77.2	89.9	66.4	97.8	89.1	83.4	50.9	79.24	88
LPT + PAE	78.8	91.5	69.7	98.0	91.3	86.1	51.7	81.01	61
VQT	77.9	88.4	68.2	98.1	89.6	82.8	51.2	79.46	80
VQT + PAE	80.3	91.0	73.6	99.2	92.5	85.7	54.4	82.39	52
VFPT	80.7	91.4	69.4	99.3	90.3	85.6	52.7	81.35	77
VFPT + PAE	82.1	91.8	69.5	99.6	91.7	87.2	52.6	82.07	62
SA2VP	73.0	91.9	70.5	99.1	90.8	84.7	56.8	80.97	86
SA2VP + PAE	77.5	93.2	70.1	99.6	91.3	87.8	56.5	82.86	54
ProVP	75.2	91.7	68.7	99.0	89.8	84.5	53.6	80.35	89
ProVP + PAE	80.6	92.8	71.3	99.4	89.3	86.7	56.2	82.33	74
BPT	74.6	90.8	69.4	99.5	90.2	84.5	52.7	80.24	79
BPT + PAE	79.3	93.5	70.8	99.5	92.7	86.1	55.3	82.46	61

Table 10: VTAB-1k *Specialized* per-task results for ViT-Base/16 pretrained on supervised ImageNet-21k.

ViT-Base/16	VTAB-1k <i>Specialized</i> (4)				Mean	Convergence epoch
	Patch Camelyon	EuroSAT	Resisc45	Retinopathy		
FULL	79.7	95.7	84.2	73.9	83.36	-
LINEAR	78.5	87.5	68.6	74.0	77.16	-
PARTIAL-1	78.6	89.8	72.5	73.3	78.53	-
MLP-2	74.3	88.8	67.1	73.2	75.86	-
MLP-3	77.0	88.0	70.2	56.1	72.83	-
MLP-5	73.7	87.2	64.8	71.5	74.31	-
MLP-9	78.5	83.0	60.2	72.3	73.49	-
SIDETUNE	58.5	87.7	65.2	61.0	68.12	-
BIAS	78.7	91.6	72.9	69.8	78.25	-
ADAPTER-256	76.3	88.0	73.1	70.5	76.98	-
ADAPTER-64	76.3	87.5	73.7	70.9	77.10	-
ADAPTER-8	76.9	89.2	73.5	71.6	77.80	-
LoRA	84.9	95.3	84.4	73.6	84.55	-
SPT-LoRA	85.7	96.2	85.9	75.9	85.93	-
DM-LoRA	85.6	96.5	87.0	76.1	86.30	-
VPT	81.8	96.1	83.4	68.4	82.43	69
VPT + PAE	83.1	96.8	85.7	72.5	84.52	34
E2VPT	82.5	96.8	84.8	73.6	84.43	90
E2VPT + PAE	83.6	97.4	86.1	75.9	85.76	56
LPT	81.6	95.5	84.7	71.8	83.40	83
LPT + PAE	82.8	96.4	85.3	75.6	85.02	55
VQT	81.4	94.8	84.2	72.5	82.23	78
VQT + PAE	83.5	96.1	86.8	74.9	85.34	51
VFPT	83.5	96.5	84.4	75.4	84.93	72
VFPT + PAE	83.4	97.8	85.9	76.7	85.96	58
SA2VP	86.0	95.9	85.8	75.2	85.73	81
SA2VP + PAE	85.7	97.5	86.9	76.2	86.58	53
ProVP	84.6	95.7	82.4	73.6	84.07	87
ProVP + PAE	84.8	96.1	85.0	75.7	85.41	73
BPT	85.8	96.2	81.5	74.9	84.45	88
BPT + PAE	87.4	97.6	84.1	76.2	86.33	60

Table 11: VTAB-1k *Structured* per-task results for ViT-Base/16 pretrained on supervised ImageNet-21k.

ViT-Base/16 (85.8M)	VTAB-1k <i>Structured</i>								Mean	Convergence epoch
	Clevr/ count	Clevr/ distance	DMLab	KITTI/ distance	dSprites/ location	dSprites/ orientation	SmallNORB/ azimuth	SmallNORB/ elevation		
FULL	56.3	58.6	41.7	65.5	57.5	46.7	25.7	29.1	47.64	-
LINEAR	34.3	30.6	33.2	55.4	12.5	20.0	9.6	19.2	26.84	-
PARTIAL-1	41.5	34.3	33.9	61.0	31.3	32.8	16.3	22.4	34.17	-
MLP-2	45.2	31.6	31.8	55.7	30.9	24.6	16.6	23.3	32.47	-
MLP-3	47.8	32.8	32.3	58.1	12.9	21.2	15.2	24.8	30.62	-
MLP-5	50.8	32.3	31.5	56.4	7.5	20.8	14.4	20.4	29.23	-
MLP-9	47.5	27.9	28.9	54.0	6.2	17.7	10.8	16.2	26.15	-
SIDETUNE	27.6	22.6	31.3	51.7	8.2	14.4	9.8	21.8	23.41	-
BIAS	61.5	55.6	32.4	55.9	66.6	40.0	15.7	25.1	44.09	-
ADAPTER-256	45.7	37.4	31.2	53.2	30.3	25.4	13.8	22.1	32.39	-
ADAPTER-64	42.9	39.9	30.4	54.5	31.9	25.6	13.5	21.4	32.51	-
ADAPTER-8	45.2	41.8	31.1	56.4	30.4	24.6	13.2	22.0	33.09	-
LoRA	82.9	69.2	49.8	78.5	75.7	47.1	31.0	44.4	59.83	-
T-LoRA	84.4	67.6	52.5	82.0	81.0	51.1	30.2	41.3	61.26	-
DM-LoRA	83.5	69.9	52.0	81.6	80.2	50.2	36.1	43.1	62.08	-
VPT	68.5	60.0	46.5	72.8	73.6	47.9	32.9	37.8	54.98	78
VPT + PAE	73.4	61.7	46.9	75.8	77.9	53.6	34.3	42.6	58.28	56
E2VPT	71.7	61.2	47.9	75.8	80.8	48.1	31.7	41.9	57.39	95
E2VPT + PAE	72.9	63.7	49.4	78.6	82.5	52.9	33.8	44.3	59.73	60
LPT	70.6	61.8	47.1	73.6	78.8	51.6	33.2	42.4	57.39	82
LPT + PAE	74.8	63.5	49.6	75.8	80.6	54.7	32.9	43.4	59.41	60
VQT	71.8	62.6	46.4	73.7	79.3	50.8	32.5	42.6	57.47	83
VQT + PAE	75.1	64.4	48.8	77.2	81.6	52.6	33.8	45.1	59.83	55
VFPT	75.8	63.2	48.3	79.3	81.5	56.0	34.1	43.4	60.19	74
VFPT + PAE	76.9	65.2	48.1	79.7	82.4	55.2	36.5	43.7	60.96	56
SA2VP	76.6	71.8	50.8	79.9	84.5	52.8	34.7	45.3	60.80	84
SA2VP + PAE	78.2	72.5	53.3	79.6	86.4	54.5	34.1	45.8	63.05	51
ProVP	75.5	67.2	49.3	77.6	83.7	51.9	33.8	43.5	60.31	84
ProVP + PAE	77.3	71.6	51.5	76.8	83.5	53.6	34.7	44.7	61.72	71
BPT	76.1	65.3	48.9	77.8	82.5	53.2	34.7	44.6	60.39	80
BPT + PAE	77.6	70.1	52.5	77.2	84.7	53.9	35.1	45.3	62.05	64