Layer-Specific Prompt Fusion Discovery via Differentiable Search in Vision Foundation Models

Xi Xiao[1], Xingjian Li[2], Yunbei Zhang[3], Cheng Han[4], Tianming Liu[5], Tianyang Wang[1*,†], Runmin Jiang[2], Jihun Hamm[3], Xiao Wang[6], and Min Xu[2*]

[1] University of Alabama at Birmingham
[2] Carnegie Mellon University
[3] Tulane University
[4] University of Missouri-Kansas City
[5] University of Georgia
[6] Oak Ridge National Laboratory

Abstract. Visual prompt tuning has emerged as a parameter-efficient fine-tuning approach for adapting large-scale Vision Transformers (ViTs) to downstream tasks. As its learnable prompts are applied in input and feature spaces, prior to jointly going through attention in transformer layers, the most commonly used scheme for fusing image and prompt tokens is concatenation or addition. In this paper, we aim to study a fundamental yet essential problem in visual prompt tuning: whether a single fusion scheme tends to yield better results, and whether that would be beneficial to develop a hybrid fusion scheme. To this end, we formulate the task as a bi-level optimization problem, and solve it leveraging differentiable architecture search. In this context, the learnable prompts and their fusion schemes are jointly optimized. To enrich the search space in the architecture search, we propose two additional fusion schemes, namely, affine transformation and cross-attention, in addition to concatenation and addition. Extensive experiments on 34 datasets spanning VTAB-1k, FGVC, and HTA show consistent gains over prompt-tuning baselines. With a frozen ViT backbone, our method delivers a favorable accuracy–latency–parameter trade-off compared with VPT-Deep and recent variants. Our findings reveal that how prompts fuse with image tokens plays a significant role in visual prompt tuning, and a hybrid fusion fashion can more effectively leverage layer semantics of ViTs, contributing a novel perspective for visual prompt-tuning research.

1 Introduction

Vision transformers (ViTs) have demonstrated impressive performance across a wide range of visual recognition tasks [11, 40, 64]. However, if full fine-tuning is employed, adapting such models to downstream scenarios inevitably demands high computing load, and a potential overfitting problem to small downstream datasets. *Visual Prompt Tuning* (VPT) [28], a *Parameter-efficient fine-tuning* (PEFT)

*Corresponding authors. †Project lead.

solution, has recently emerged to address this challenge. VPT incorporates learnable prompts (*i.e.*, a certain type of parameters) into input and feature spaces, and solely updates these prompts during fine-tuning. The core operation in VPT is how to fuse the learnable prompts with image tokens and/or feature tokens before the fused elements are fed into transformer layers. To date, the most commonly used scheme is either concatenation or addition [28, 39, 68, 69].

Nevertheless, it remains unknown whether these fusion schemes contribute differently to ViT's fine-tuning. If so, should we always use a single fusion scheme or is there an optimal hybrid fusion fashion that takes the best of both concatenation and addition to fully leverage **layer-wise** information [35, 47, 53]? Moreover, if a hybrid fusion scheme can be optimal, what else single fusion scheme can be developed to facilitate the design of the hybrid fashion? Such questions are fundamental yet essential to visual prompt tuning research, which are rarely explored in previous literature.

In this work, we aim to answer the aforementioned questions in a unified task, and formulate the task as a bi-level optimization problem, consisting of inner and outer optimizations. In the former, the learnable prompts will be optimized, while in the latter, the best combination of fusion schemes will be learned. We tackle this problem by developing a novel method with differentiable neural architecture search (NAS), fully leveraging layer-wise information, where another challenge naturally arises: how to design a search space that is indispensable in NAS? A natural solution is to incorporate the two commonly used single fusion schemes directly. However, combining these two simple fusions is unlikely to yield optimal results (Sec. 3.3). To enrich the search space to discover better hybrid prompting fashions, we thus propose two new single fusion schemes, namely, affine transformation and cross-attention. The rationale lies in that they are simple and light-weight, and can be seamlessly used to fuse prompt and image tokens efficiently in the VPT context (Sec. 3.5). Together, our proposed method is highly scalable and can accommodate any new single fusion schemes by including them in the search space.

We validate the proposed method from both interpretive and empirical perspectives. We interpret our method through the lens of *Information Bottleneck* (IB) [56], and observe that jointly optimizing prompts and fusion schemes leads to a lower empirical IB surrogate than optimizing prompts alone, supporting the role of hybrid fusion in prompt tuning. Empirically, we conduct extensive experiments on 34 datasets across three widely used benchmarks: VTAB-1k [81], FGVC [58], and HTA [25]. The results (*e.g.*, 77.01% on VTAB-1k mean, 92.5% on HTA mean, 91.6% on FGVC mean, and only 0.75% of the ViT's parameters tuned) reveal that our method consistently and significantly outperforms existing state-of-the-art baselines. More importantly, these results help answer the aforementioned crucial questions as follows: ① there is an optimal hybrid fusion fashion that is significantly better than single fusion scheme in visual prompt tuning; and ② the hybrid fashion is highly scalable and can seamlessly include new single fusion schemes.

Our main contributions are summarized as follows:

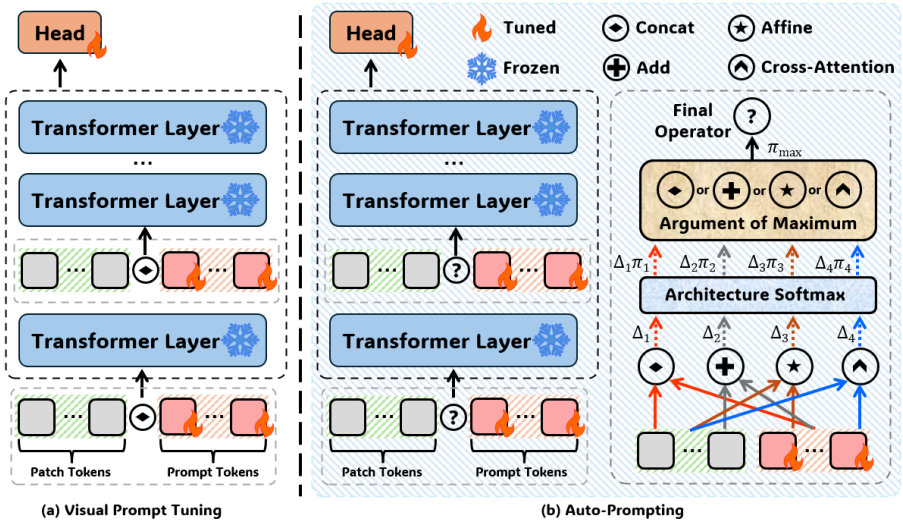

Fig. 1: Overview of standard Visual Prompt Tuning *vs*. Auto-Prompting framework. **(a) Visual Prompt Tuning:** Employs a *single, fixed fusion operator* to combine learnable prompt tokens with patch tokens at every layer. The Transformer Layers remain frozen. **(b) Auto-Prompting (Ours):** We replace the fixed operator with a *searchable fusion module* that operates in two phases. **Search Phase:** An 'Architecture Softmax' module generates layer-wise preference weights $(\pi_1, \pi_2, \pi_3, \pi_4)$ for a set of fundamental operators (Concat $\diamond$, Add $+$, Affine $\star$, Cross-Attn $\wedge$). Concurrently, the operators process the inputs in parallel to produce their respective outputs $(\Delta_1, \Delta_2, \Delta_3, \Delta_4)$. **Discretization Phase:** After training, we discretize each layer by selecting the final operator with the largest learned preference.

- We raise fundamental yet essential questions on fusing prompt and image tokens in visual prompt tuning research, and reveal that using a single fusion scheme fails to yield optimal performance.
- We propose a hybrid fusion method and formulate it as a bi-level optimization problem. To solve the new optimization problem, we develop a novel method with differentiable neural architecture search.
- We provide both a theoretical interpretation (via Information Bottleneck) and extensive empirical evaluations to demonstrate our method's state-of-the-art and parameter-efficient performance across diverse datasets. We use the observed results, along with the analysis, to answer the raised key questions in visual prompt tuning research.

2 Method

We propose a generalized prompt tuning framework that *treats the fusion between prompts and visual tokens as a searchable, differentiable component* (Figure 1).

Concretely, each Transformer layer chooses one of several candidate fusion operators that determine *how* prompts interact with tokens before entering the frozen ViT block. This section clarifies the injection point and notation, specifies the operator set (Sec. 2.1), introduces Differentiable Architecture Search (DARTS) [36] and our bilevel optimization with stability regularizers (Sec. 2.2), and gives an *Information Bottleneck* (IB) perspective (Sec. 2.3) on why learning the fusion rule improves adaptation.

Let f_ψ be a frozen ViT encoder. At block l, let $x^{(l-1)} \in \mathbb{R}^{k\times d}$ be the token sequence (incl. CLS) and $p^{(l)} \in \mathbb{R}^{m\times d}$ be m learnable prompt tokens. We use the Pre-LayerNorm (Pre-LN) form, where a standard block is:

$$\begin{aligned} y &= x^{(l-1)} + \mathrm{MSA}\big(\mathrm{LN}(x^{(l-1)})\big), \\ x^{(l)} &= y + \mathrm{MLP}\big(\mathrm{LN}(y)\big). \end{aligned} \tag{1}$$

Denote $\tilde{x}^{(l-1)} = \mathrm{LN}(x^{(l-1)})$. We inject fusion *after* this internal LayerNorm and *before* the frozen MSA sublayer:

$$\begin{aligned} \widehat{x}^{(l-1)}_{\mathrm{msa}} &= \Delta^{(l)}\big(p^{(l)},\, \tilde{x}^{(l-1)}\big), \\ y &= x^{(l-1)} + \mathrm{MSA}\big(\widehat{x}^{(l-1)}_{\mathrm{msa}}\big). \end{aligned} \tag{2}$$

No extra normalization is applied after $\Delta^{(l)}$. By default, we inject fusion once per block (before MSA). We split trainable variables into two groups. (1) *Model parameters* ϕ: prompt tokens $\{p^{(l)}\}$ and operator-internal weights (*e.g.*, the reduction matrix $R^{(l)}$ in `concat`, FiLM MLPs in `affine`, and W_Q, W_K, W_V in `cross` when not shared). (2) *Architecture parameters* $\alpha = \{\alpha^{(l)}\}_{l=1}^{L}$: for each layer l, $\alpha^{(l)} \in \mathbb{R}^{|\mathcal{S}|}$ contains one logit (unnormalized preference) per candidate operator in $\mathcal{S} = \{\texttt{concat}, \texttt{add}, \texttt{affine}, \texttt{cross}\}$. These logits are converted into mixing weights by:

$$\pi^{(l)} = \mathrm{softmax}\big(\alpha^{(l)}/\tau\big), \quad \tau > 0, \quad \pi_i^{(l)} \geq 0,\ \sum_{i\in\mathcal{S}} \pi_i^{(l)} = 1,$$

and define the soft fusion at layer l:

$$\Delta^{(l)}_{\mathrm{soft}}(p^{(l)},\, \tilde{x}^{(l-1)}) = \sum_{i\in\mathcal{S}} \pi_i^{(l)}\, \Delta_i\big(p^{(l)},\, \tilde{x}^{(l-1)}\big).$$

At discretization we pick $\widehat{i}^{(l)} = \arg\max_{i\in\mathcal{S}} \pi_i^{(l)}$ and keep only $\Delta_{\widehat{i}^{(l)}}$.

2.1 Fusion Operator Design

We place a fusion module $\Delta^{(l)}(\cdot)$ before each frozen block, with the constraint that its output is always $k \times d$ (token count k incl. CLS) so the backbone interface never changes. We consider

$$\mathcal{S} = \{\Delta_{\texttt{concat}},\, \Delta_{\texttt{add}},\, \Delta_{\texttt{affine}},\, \Delta_{\texttt{cross}}\}.$$

This set covers four complementary fusion operators while keeping the interface fixed: `concat` augments the sequence and mixes prompt cues with a light token-preserving reduction; `add` injects a prompt-dependent residual bias at negligible cost; `affine` applies prompt-conditioned channel-wise scale–shift (with `add` as a special case); and `cross-attention` enables content-adaptive routing from a prompt memory when stronger fusion is needed. The motivation for involving these four operators is that many heavier fusion designs (*e.g.*, gated residuals, dynamic projections) can be **precisely composed** from or **closely approximated** by these four primitives. Searching within this basis lets each layer select the right fusion without redesigning the frozen ViT and exposes an accuracy–cost frontier that we regularize during search (Sec. 2.2). We elaborate on them in detail below.

(1) ***`concat`.*** Unlike the original VPT that feeds $(m{+}k)$ tokens into the frozen block, we keep the backbone interface fixed at $k \times d$ to avoid changing positional embeddings, attention map size, and FLOPs inside the block. We first prefix prompts and then reduce back to k tokens with a light, token preserving mixer:

$$\widetilde{x} = [\, p^{(l)};\, \tilde{x}^{(l-1)} \,], \tag{3a}$$

$$R^{(l)} = \mathrm{softmax}_{\mathrm{col}}\big(U^{(l)}\big), \tag{3b}$$

$$\Delta_{\texttt{concat}}\big(p^{(l)}, \tilde{x}^{(l-1)}\big) = (R^{(l)})^{\top}\, \widetilde{x}. \tag{3c}$$

Here $U^{(l)} \in \mathbb{R}^{(m+k)\times k}$ collects the per-column logits and $R^{(l)} = \mathrm{softmax}_{\mathrm{col}}(U^{(l)})$ is column-stochastic (each column is nonnegative and sums to 1), so every output token is a convex combination of the $(m{+}k)$ inputs. This operator uses no $W_Q/W_K/W_V$ projections and is therefore cheaper and more stable than full cross-attention. By default $R^{(l)}$ is layer-shared to keep parameters minimal (More analysis in Appendix).

(2) ***`add`.*** We summarize prompts by $s = \mathrm{LN}(\mathrm{mean}(p^{(l)})) \in \mathbb{R}^d$ and broadcast it across tokens:

$$\Delta_{\texttt{add}}(p^{(l)}, \tilde{x}^{(l-1)}) \;=\; \tilde{x}^{(l-1)} \;+\; \mathbf{1}_k\, s^{\top}, \tag{4}$$

where $\mathbf{1}_k$ repeats s along the token axis.

(3) ***`affine`.*** Feature-wise Linear Modulation (FiLM) applies channel-wise scale and shift conditioned on an external signal. We reuse the same prompt summary s as in Eq. (4) and generate per-channel scale/bias via two small MLPs:

$$\phi_t(s) = W_t^2\, \sigma_h\big(W_t^1 s + b_t^1\big) + b_t^2, \quad t \in \{\gamma, \beta\}. \tag{5}$$

Then

$$\begin{aligned} \gamma = \sigma\big(\phi_\gamma(s)\big), \quad \beta = \phi_\beta(s), \\ \Delta_{\texttt{affine}}\big(p^{(l)}, \tilde{x}^{(l-1)}\big) = \gamma \odot \tilde{x}^{(l-1)} + \mathbf{1}_k\, \beta^{\top}, \end{aligned} \tag{6}$$

with σ a squashing nonlinearity and $\odot$ the Hadamard product. Eq. (4) is the special case $\gamma = \mathbf{1}$ and $\beta = s$. The summary vector is used only for `add`

and `affine`, whose outputs are channel-wise modulations over the fixed $k \times d$ image-token interface. In contrast, `concat` and `cross-attention` consume all m prompt tokens directly. The LayerNorm-mean summary gives a permutation-invariant task descriptor and prevents scale drift in these modulation branches; Appendix D reports ablations against first-token, unnormalized mean, and raw-token conditioning.

(4) `cross-attention` *(prompt-as-memory).* With H heads and $d_h{=}d/H$, image tokens attend to a prompt memory per head:

$$\begin{aligned} Q_x^{(h)} &= \tilde{x}^{(l-1)} W_Q^{(h)} \in \mathbb{R}^{k \times d_h}, \\ K_p^{(h)} &= p^{(l)} W_K^{(h)} \in \mathbb{R}^{m \times d_h}, \\ V_p^{(h)} &= p^{(l)} W_V^{(h)} \in \mathbb{R}^{m \times d_h}, \\ A^{(h)} &= \mathrm{softmax}\Big(\tfrac{Q_x^{(h)} (K_p^{(h)})^\top}{\sqrt{d_h}}\Big) \in \mathbb{R}^{k \times m}, \\ O^{(h)} &= A^{(h)} V_p^{(h)} \in \mathbb{R}^{k \times d_h}. \end{aligned} \tag{7}$$

We then concatenate head outputs along the channel dimension and add a residual:

$$\Delta_{\texttt{cross}}\big(p^{(l)}, \tilde{x}^{(l-1)}\big) = \big[\mathrm{Concat}_{h=1}^{H} O^{(h)}\big] \; + \; \tilde{x}^{(l-1)} \; \in \; \mathbb{R}^{k \times d}.$$

The projection matrices $\{W_Q^{(h)}, W_K^{(h)}, W_V^{(h)}\}$ are shared across layers unless stated (More details see Appendix. B.4).

2.2 Differentiable Layer-wise Operator Search

Differentiable Architecture Search (DARTS) [36] relaxes discrete operator choices into a softmax over learnable logits, producing mixing weights per layer; model weights are optimized on the training split and architecture parameters on the validation split, yielding a bilevel program that is later discretized by an arg max choice. For layer l, architecture logits $\alpha^{(l)} \in \mathbb{R}^{|\mathcal{S}|}$ parameterize a soft mixture:

$$\begin{aligned} \pi_i^{(l)}(\tau) &= \mathrm{softmax}\big(\alpha^{(l)}/\tau\big)_i, \\ \Delta_{\mathrm{soft}}^{(l)}(p^{(l)}, \tilde{x}^{(l-1)}; \alpha^{(l)}, \tau) &= \sum_{i \in \mathcal{S}} \pi_i^{(l)}(\tau)\, \Delta_i(p^{(l)}, \tilde{x}^{(l-1)}), \end{aligned} \tag{8}$$

where the temperature τ is annealed from high (exploration) to low (selection). We adopt a DARTS–style bilevel objective [36]:

$$\min_{\alpha} \; \mathcal{L}_{\mathrm{val}}\big(\phi^*(\alpha), \alpha\big), \quad \text{s.t.} \;\; \phi^*(\alpha) = \arg\min_{\phi} \mathcal{L}_{\mathrm{train}}(\phi, \alpha). \tag{9}$$

Gradients $\nabla_\alpha \mathcal{L}_{\mathrm{val}}$ use the one-step unrolled approximation with Hessian–vector [1] products (See Appendix B.2):

$$\phi' = \phi - \eta_\phi \nabla_\phi \mathcal{L}_{\mathrm{train}}, \tag{10}$$

$$\nabla_\alpha \mathcal{L}_{\mathrm{val}} \approx \nabla_\alpha \mathcal{L}_{\mathrm{val}}(\phi', \alpha) - \eta_\phi \big[\nabla^2_{\alpha,\phi} \mathcal{L}_{\mathrm{train}}\big] \nabla_\phi \mathcal{L}_{\mathrm{val}}(\phi', \alpha). \tag{11}$$

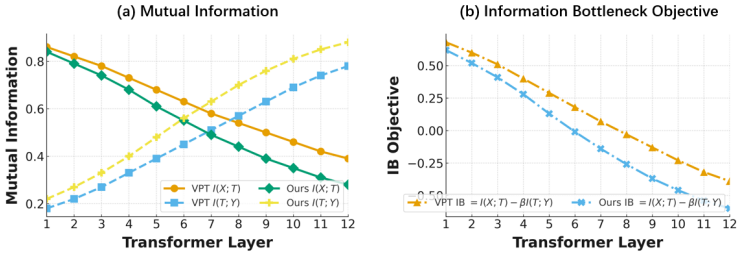

Fig. 2: Mutual information and per-layer IB objective (CUB-200, ViT-B/16). Comparison between VPT [28] and our search-based fusion. (a) $\widehat{I}(T^{(l)};X)$ (noise compression) decreases and $\widehat{I}(T^{(l)};Y)$ (semantic retention) increases more markedly for our method in deeper blocks. (b) The resulting empirical IB surrogate remains consistently lower for ours, suggesting a favorable compression–relevance trade-off under a frozen backbone.

To avoid premature collapse and to softly prefer cheaper operators when accuracies are similar, we use two unweighted terms:

$$\widetilde{\mathcal{R}}_{\text{ent}}(\alpha) = -\sum_{l} H\big(\pi^{(l)}\big), \tag{12a}$$

$$\widetilde{\mathcal{R}}_{\text{cost}}(\alpha) = \sum_{l}\sum_{i\in\mathcal{S}} c_i\,\pi_i^{(l)}, \tag{12b}$$

where $H(\cdot)$ is Shannon entropy [2]. We set the normalized cost prior to $c = [0, 0.06, 0.30, 1.00]$ for the ordered operators (`add`, `affine`, `concat`, `cross`), based on per-operator microbenchmarks. The outer objective places the weights here:

$$\mathcal{L}_{\text{outer}} = \mathcal{L}_{\text{val}} + \lambda_{\text{ent}}\,\widetilde{\mathcal{R}}_{\text{ent}} + \lambda_{\text{cost}}\,\widetilde{\mathcal{R}}_{\text{cost}}. \tag{13}$$

Positive λ_{ent} on the negative-entropy term encourages exploration; as the temperature τ is annealed the mixture sharpens toward a discrete choice. The cost term biases against unnecessary heavy operators when accuracies are similar. At epoch e_{disc}, we discretize each layer by $\widehat{i}^{(l)} = \arg\max_i \pi_i^{(l)}(\tau)$, freeze α, and continue optimizing ϕ for a few epochs with the selected operators only. This yields a compact, deployable blueprint with the same inference path length as a fixed-fusion method (More details in Appendix B.2).

2.3 Observation: From the Information Bottleneck (IB) Perspective

Let $T^{(l)} = \Delta^{(l)}\big(p^{(l)}, \tilde{x}^{(l-1)}\big)$ be the representation entering the l-th frozen block. We adopt a per-layer Information Bottleneck (IB) objective [56]:

$$\mathcal{L}_{\text{IB}}^{(l)}\big(T^{(l)}\big) = I\big(T^{(l)};X\big) \;-\; \beta\, I\big(T^{(l)};Y\big), \qquad \beta > 0, \tag{14}$$

where X is the input image, Y the label, and $I(\cdot;\cdot)$ denotes mutual information. Since exact MI is intractable, we report the variational surrogate $\widehat{\mathcal{L}}_{\text{IB}}^{(l)} = \widehat{I}\big(T^{(l)};X\big) - \beta\,\widehat{I}\big(T^{(l)};Y\big)$ estimated with standard MI estimators (*i.e.*, MINE/InfoNCE/CLUB; details in Appendix B.5).

Why learning the fusion operator under IB? Fix prompts p and consider the family $\{T_\Delta\}$ induced by $\Delta \in \mathcal{S}$. Allowing each layer to select its own fusion operator provides a broader range of possible representations than a single fixed operator. Figure. 2 reports the MI metrics on CUB-200 (ViT-B/16, β=1). As shown in Fig. 2(a), our model exhibits a larger drop in $\widehat{I}\big(T^{(l)}; X\big)$ in deep layers while maintaining higher $\widehat{I}\big(T^{(l)}; Y\big)$. This pattern suggests stronger compression of input-dependent variation together with better label alignment. The consistently lower $\widehat{\mathcal{L}}_{\mathrm{IB}}^{(l)}$ is therefore empirical evidence that search-based fusion reaches a more favorable compression–relevance trade-off, rather than a causal proof of background removal or accuracy gains.

Operator schedule as information flow. The learned operator schedule induces a coherent information flow: shallow blocks tend to use lightweight primitives (`concat`/`add`) that preserve structural cues, whereas deeper blocks favor semantic primitives (`affine`/`cross-attention`) that emphasize label-relevant patterns. This layer-wise shaping is consistent with the IB view and with the attention trends in Sec. 3.4.

2.4 Summary of Training Procedure.

As detailed in Algorithm 1, our framework operates in two distinct stages. First, the differentiable search phase jointly optimizes the prompts and the architecture logits via a bilevel objective. Following this, the discretization step selects the dominant fusion operator for each layer, succeeded by a final short fine-tuning phase to recover performance with the pruned architecture.

2.5 Discussion on Operator Orthogonality and Redundancy

While the inclusion of both 'Add' and 'Affine' operators might suggest a functional overlap, maintaining this diversity in the search space $\mathcal{S}$ is essential for both optimization stability and computational efficiency. We justify this design from three primary perspectives. ❶ Regarding the optimization landscape and gradient flow, although the 'Add' operator can be viewed as a special case of 'Affine' transformation, optimizing 'Affine' parameters $(\phi_\gamma, \phi_\beta)$ via MLPs from scratch presents a significantly more complex optimization landscape. Following the principles of Residual Learning [20], the parameter-free 'Add' acts as a stable identity-like connection that facilitates smooth gradient flow during the early stages of architecture search. Furthermore, our ❷ outer objective leverages a cost-driven regularizer ($\mathcal{R}_{cost}$) to exploit this apparent redundancy for implicit pruning. In layers where the extra expressivity of an 'Affine' calibration does not yield a significant reduction in validation loss, the optimization naturally gravitates toward the zero-cost 'Add' operator. Removing the simpler 'Add' would force the model to employ the heavier 'Affine' operator (which incurs approximately 2× the operator latency) even for basic bias injections, leading to unnecessary computational overhead [45]. Additionally, ❸ empirical evidence

Algorithm 1 Layer-wise Search and Discretization for Prompt–Token Fusion

Require: Frozen ViT f_ψ; operator set $\mathcal{S}$; cost prior $c = \{c_i\}$; train/val splits
Require: Init prompts $\{p^{(l)}\}$ and operator-internal weights in ϕ; architecture logits $\alpha = \{\alpha^{(l)}\}$
Require: Hyperparams: learning rates η_ϕ, η_α; temperatures $\tau_{\max}, \tau_{\min}$; weights $\lambda_{\text{ent}}, \lambda_{\text{cost}}$; search epochs E_{search}; fine-tune epochs E_{ft}
$\tau \leftarrow \tau_{\max}$ ▷ **Phase I: differentiable search**
for $e = 1$ **to** E_{search} **do**
  (Inner/train) Compute $\pi^{(l)} = \text{softmax}(\alpha^{(l)}/\tau)$ and $\Delta^{(l)}_{\text{soft}}$ (Eq. 8); update $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathcal{L}_{\text{train}}(\phi, \alpha)$
  (Outer/val) Build $\widetilde{\mathcal{R}}_{\text{ent}}, \widetilde{\mathcal{R}}_{\text{cost}}$ (Eq. 12); compute one-step unrolled gradient (Eq. 11); update $\alpha \leftarrow \alpha - \eta_\alpha \nabla_\alpha \big[\mathcal{L}_{\text{val}} + \lambda_{\text{ent}} \widetilde{\mathcal{R}}_{\text{ent}} + \lambda_{\text{cost}} \widetilde{\mathcal{R}}_{\text{cost}}\big]$
  Anneal $\tau \leftarrow \text{CosAnneal}(\tau_{\max} \to \tau_{\min},\, e,\, E_{\text{search}})$
end for
Discretize: for all l, set $\widehat{i}^{(l)} \leftarrow \arg\max_{i \in \mathcal{S}} \pi_i^{(l)}$; freeze α and *discard* inactive operators
▷ **Phase II: short fine-tuning with discrete operators**
for $e = 1$ **to** E_{ft} **do**
  Forward with $\Delta^{(l)} = \Delta_{\widehat{i}^{(l)}}$ only; update $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathcal{L}_{\text{train}}(\phi, \widehat{i}^{(1\ldots L)})$
end for
Return: prompts $\phi^\star$ and discrete operators $\{\widehat{i}^{(l)}\}_{l=1}^{L}$

confirms that a diverse basis set is vital for stable search dynamics, as shown in Appendix Table S7. Without the stabilization and regularization strategies that utilize this operator redundancy, the search process suffers a 65% collapse rate to suboptimal solutions. In summary, these four operators form a complete basis set for feature modulation: 'Add' for bias injection [20], 'Affine' for feature calibration [45], and 'Cross-Attention' for dynamic content-adaptive routing [57].

3 Experiments

3.1 Experimental Setup

Benchmarks. We follow VPT [28] on VTAB-1k [81] (19 datasets across Natural/Specialized/Structured), and include fine grained datasets such as CUB-200-2011 [58] and Oxford Flowers-102 [43]. To test robustness with different pretraining biases, we additionally evaluate under MAE [19] and MoCo v3 [7]. For architectural generality, we further report results on a hierarchical backbone (Swin-Base [40]).

Fusion candidates and search. Each Transformer layer selects its fusion operator from `concat`, `add`, `affine`, and `cross-attention` via a differentiable relaxation (Sec. 2). The backbone stays frozen throughout. We adopt a bilevel schedule: inner loop updates optimize prompts and fusion weights on training data, while outer loop updates refine architecture logits on held out validation data. A temperature annealing is applied to stabilize discretization toward a single operator per layer at convergence.

Training details. We use AdamW with learning rate 1×10^{-3}, weight decay 0.01, cosine schedule, 100 epochs, batch size 64. We report top-1 accuracy averaged over three seeds per dataset. All runs are on NVIDIA A100 GPUs. To ensure comparability, Tuned/% includes the task-specific head and follows the macro averaging convention of the compared works; per task counts are provided in Appendix C.

Table 1: Comparison of fine-tuning methods with ViT-Base/16 backbone. Bold and underlined indicate best and second-best results. We report mean accuracy across datasets for each benchmark (FGVC: 5 datasets, HTA: 10 datasets, VTAB-1k: 19 datasets) and for each VTAB-1k category (Natural, Specialized, and Structured). See Appendix. C for per-task results. Same for Table 2.

ViT-Base/16 [11] (85.8M)	Tuned/Total (%)	Extra Params	FGVC Mean (%)	HTA Mean (%)	VTAB-1k [81] *Natural*	*Specialized*	*Structured*	VTAB-1k Mean (%)
Full [CVPR22] [27]	100.00	—	88.54	85.8	75.88	83.36	47.64	65.57
Linear [CVPR22] [27]	0.08	—	79.32	75.7	68.93	77.16	26.84	52.94
Partial-1 [NeurIPS14] [78]	8.34	—	82.63	80.8	69.44	78.53	34.17	56.52
MLP-3 [CVPR20] [6]	1.44	✓	79.80	78.5	67.80	72.83	30.62	53.21
Sidetune [ECCV20] [84]	10.08	—	78.35	72.3	58.21	68.12	23.41	45.65
Bias [NeurIPS17] [49]	0.80	—	88.41	82.1	73.30	78.25	44.09	62.05
Adapter [NeurIPS20] [3]	1.02	✓	85.46	80.6	70.67	77.80	33.09	62.41
LoRA [ICLR22] [23]	—	✓	89.46	85.5	78.26	83.78	56.20	72.25
AdaptFormer [NeurIPS22] [5]	—	✓	—	—	80.56	84.88	58.83	72.32
ARC_{att} [NeurIPS23] [10]	—	✓	89.12	89.0	80.41	85.55	58.38	72.32
VPT-S [ECCV22] [28]	0.16	✓	84.62	85.5	76.81	79.66	46.98	64.85
VPT-D [ECCV22] [28]	0.73	✓	89.11	85.5	78.48	82.43	54.98	69.43
E2VPT [ICCV23] [17]	0.39	✓	89.22	88.5	80.01	84.43	57.39	71.42
EXPRES [CVPR23] [8]	—	✓	—	—	79.69	84.03	54.99	70.02
DAM-VP [CVPR23] [25]	—	✓	—	88.5	—	—	—	—
SA^2VP [AAAI24] [44]	0.81	✓	90.08	91.5	80.97	**85.73**	60.80	75.83
SPT [ICML24] [65]	—	✓	90.10	—	81.44	83.65	52.86	72.65
VFPT [NeurIPS24] [80]	0.66	✓	89.24	—	81.35	84.93	60.19	75.49
LoR-VP [ICLR25] [29]	—	✓	91.22	—	80.25	85.12	58.71	74.69
DA-VPT [CVPR25] [50]	—	✓	89.32	—	79.91	83.16	60.01	74.36
Ours	0.75	✓	**91.60**	**92.5**	**82.88**	85.61	**62.55**	**77.01**

Table 2: Comparison of the fine-tuning methods under different pretraining paradigms. We report VTAB-1k [81] accuracy using ViT-Base [11] as the frozen backbone, pretrained with MAE [19] and MoCo v3 [7], respectively.

Pretrained Method	MAE [19] Tuned(%)	*Natural*	*Specialized*	*Structured*	MoCo v3 [7] Tuned(%)	*Natural*	*Specialized*	*Structured*
Full [CVPR22] [27]	100.00	59.31	**79.68**	53.82	100.00	71.95	84.72	51.98
Linear [CVPR22] [27]	0.04	18.87	53.72	23.70	0.04	67.46	81.08	30.33
Partial-1 [NeurIPS14] [78]	8.30	58.44	78.28	47.64	8.30	72.31	84.58	47.89
Bias [NeurIPS17] [49]	0.16	54.55	75.68	47.70	0.16	72.89	81.14	53.43
Adapter [NeurIPS20] [3]	0.87	54.90	75.19	38.98	1.12	74.19	82.66	47.69
VPT-S [ECCV22] [28]	0.05	39.96	69.65	27.50	0.06	67.34	82.26	37.55
VPT-D [ECCV22] [28]	0.31	36.02	60.61	26.57	0.22	70.27	83.04	42.38
GPT [ICML23] [77]	0.05	47.61	76.86	36.80	0.06	74.84	83.38	49.10
VFPT [NeurIPS24] [80]	0.38	53.59	77.75	36.15	0.22	77.47	85.76	58.74
LoR-VP [ICLR25] [29]	0.40	51.23	74.21	33.16	0.29	72.28	82.89	46.92
DA-VPT [CVPR25] [50]	—	**62.14**	79.14	**54.31**	—	74.24	83.21	55.23
Ours	0.34	55.12	78.19	39.01	0.25	**79.60**	**86.86**	**61.01**

3.2 Main Results

ViT-Base. In Table 1, we show that our method attains the best mean accuracy on ViT-Base across VTAB-1k, FGVC, and HTA. Specifically, relative to the fixed

fusion VPT-Deep, gains are significant and consistent: VTAB-1k mean +7.58, FGVC +2.49, HTA +7.0, with category-wise improvements on *Natural* +4.40, *Specialized* +3.18, and *Structured* +7.57, respectively. Compared with VFPT, we lift VTAB-1k mean by +1.52 and improve each VTAB category (*Natural* +1.53, *Specialized* +0.68, *Structured* +2.36) at a comparable tunable budget (0.75% *vs.* 0.66%), yielding a favorable accuracy–parameter trade-off (roughly +0.17 points VTAB-mean per additional 0.01% tuned). Against SA^2VP, which is particularly strong on *Specialized* (85.73%), we obtain the top results on *Natural* (82.88%) and *Structured* (62.55%) and the best overall VTAB-1k mean (77.01%, +1.18) with a similar parameter ratio. Taken together, these patterns indicate that *how* prompts interact with tokens (*i.e.*, the *fusion rule* is a first-order factor for generalization): allowing each layer to choose between lightweight (`concat`/`add`) and semantic (`affine`/`cross-attn`) operators systematically reduces domain sensitivity (notably on *Structured*) without resorting to heavier heads or hand-crafted inductive biases.

Different Pretrained Objectives. Table 2 shows that our fusion search is competitive under reconstruction-centric MAE and clearly strongest under contrastive MoCo v3. Under MAE, compared with VFPT (0.38% tuned), we achieve higher accuracy in all VTAB-1k categories with a lower budget (0.34%): *Natural* 55.12 *vs.* 53.59 (+1.53), *Specialized* 78.19 *vs.* 77.75 (+0.44), *Structured* 39.01 *vs.* 36.15 (+2.86). Although the absolute category bests on MAE come from less parameter-efficient baselines (Partial-1 at 8.30% on Natural/Specialized, Bias at 0.16% on Structured), our results offer a balanced profile at PEFT scale, aligned with observation in [18]. Under MoCo v3, our method attains the top results across all categories while tuning only 0.25%: *Natural* 79.60 (+2.13 over VFPT at 0.22%), *Specialized* 86.86 (+1.10), and *Structured* 61.01 (+2.27). The consistent gains over a strong frequency domain prompt at near identical budgets suggest that learning the fusion rule synergizes with contrastive features, likely by exploiting discriminative token relations via deeper affine/cross-attention choices. In short, across different pretrained objectives, we trade a modest tuned ratio for reliable cross-category improvements, narrowing MAE's structured gap and delivering SOTA performance on MoCo v3.

Swin-Base. Table 3 shows that our fusion search remains effective on Swin-Base [40]. Specifically, we achieve the best results in all VTAB-1k categories, surpassing VFPT at 0.27% and SA^2VP at 0.29%. Gains over VPT-Deep are substantial despite a nearly identical budget (0.26% *vs.* 0.25%): *Natural* +8.74%, *Specialized* +3.16%, *Structured* +10.51%. The largest margins are seen on the *Structured*, indicating that the layer-wise operator selection effectively adapts to Swin's stage-wise shifts and transitions from local to global reasoning. In the early blocks, lightweight fusion is favored to maintain locality within windows, while the deeper blocks choose more complex operators to integrate evidence across different windows. Additionally, across stage transitions, affine calibration helps stabilize the scales of the tokens. This schedule yields stronger geometry and counting-heavy performance without resorting to heavier per-layer heads,

Table 3: VTAB-1k results using Swin-Base. Our method outperforms all baselines with fewer tunable parameters.

Swin-Base [40]	Tuned	*Nat.*	*Spec.*	*Struc.*
Full [27]	100.00	79.10	86.21	59.65
Linear [27]	0.06	73.52	80.77	33.52
Bias [49]	0.30	76.78	83.33	51.85
VPT-D [28]	0.25	76.78	83.33	51.85
E^2VPT [17]	0.21	83.31	84.95	57.35
SA^2VP [44]	0.29	80.81	<u>86.30</u>	<u>60.03</u>
VFPT [80]	0.27	<u>84.53</u>	86.15	58.21
LoR-VP [29]	0.29	83.51	85.22	57.61
Ours	**0.26**	**85.52**	**86.49**	**62.36**

Table 4: Prompt fusion strategy ablation. Search-based fusion consistently outperforms any single fixed operator.

Fusion	Search	*Nat.*	*Spec.*	*Struc.*	Gain
`Concat`	✗	78.42	82.10	56.33	-1.73
`Add`	✗	77.63	81.47	55.98	-2.32
`Affine`	✗	78.86	82.62	57.40	-1.05
`Cross-Attn`	✗	<u>79.91</u>	<u>83.44</u>	<u>58.67</u>	0.00
Ours	✓	**82.88**	**85.61**	**62.55**	**+3.00**

explaining why we simultaneously reduce domain sensitivity and maintain a competitive tunable ratio.

3.3 Ablation Studies

Table 4 varies only the fusion rule while holding the prompt length and backbone fixed. Among single operators, cross-attention is the strongest. Our searchable fusion improves to 82.88 / 85.61 / 62.55 on Natural / Specialized / Structured, *i.e.*, +2.97, +2.17, and +3.88 over the best fixed baseline, for a +3.00 mean gain across VTAB-1k groups. Concat, Add, and Affine trail cross-attention by 1.73, 2.32, and 1.05 mean points, respectively, indicating that no single operator is uniformly optimal. The largest margin on Structured suggests that layer-wise operator choice is particularly helpful for token calibration and geometric reasoning. Overall, the results validate our hypothesis: learning the fusion rule via per layer selection delivers consistent, task robust gains beyond improving prompt content alone.

3.4 Layer-wise Fusion Patterns

Depth-wise Patterns. Figure 3 shows a clear depth profile. Early blocks concentrate probability mass on `concat`/`add`. These operators minimally perturb token statistics and act as content-preserving routes, allowing prompts to tag the stream without overwriting low-level evidence. As depth increases, the search shifts toward `affine`/`cross-attention`. The former calibrates feature scales and

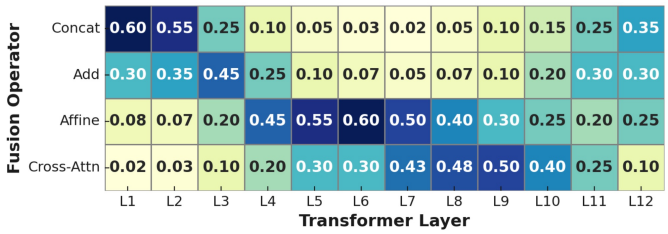

Fig. 3: Fusion operator selection by layer. Early layers favor `concat` and `add`, while deeper layers shift toward `affine` and `cross-attention`.

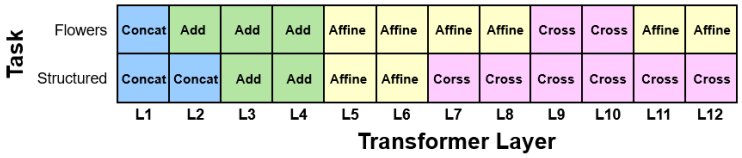

Fig. 4: Task-specific fusion preferences. The discovered operator distributions differ between Flowers and Structured tasks: early layers consistently favor lightweight fusion (`Concat`/`Add`), while deeper layers shift toward semantic operators (`Affine`/`Cross`), revealing task-specific adaptation patterns.

offsets (helpful when domain statistics drift), while the latter enables content-adaptive routing between prompts and tokens. Aggregated over VTAB-1k, layers 1–4 assign 0.69 probability mass to lightweight operators, layers 5–8 have the highest entropy, and layers 9–12 assign 0.73 mass to `affine`/`cross-attention`. The final operators agree on 9–10 of 12 layers across three seeds, with most variation concentrated in the middle blocks. Consistent with the ablation, the layers that finally prefer `cross-attention` are also those whose replacement with a single fixed rule produces the most significant accuracy drop, suggesting that *where* we switch operators is as important as *which* operator we pick.

Task-dependent Preferences. Figure 4 compares architectures discovered on FGVC versus Structured regimes. FGVC consistently pushes `cross-attention` deeper, which is coherent with the need to bind subtle, instance-specific attributes to category cues. Structured tasks select `cross-attention`/`affine` more often than Natural tasks, reflecting a stronger emphasis on geometric calibration and token rescaling. Any static recipe does not capture these task-aware shifts and helps explain the empirical pattern in Table 4: the most significant margin over the best fixed rule appears on the Structured group, where per-layer calibration is particularly valuable.

Layer-wise Attention Analysis. We select four representative layers and visualize the CLS→patch attention for VPT [28] and our model (Figure 5). At layer 1, both models distribute attention broadly. By layer 3, our model begins to concentrate attention on the head region, while VPT still shows several scattered peaks. At

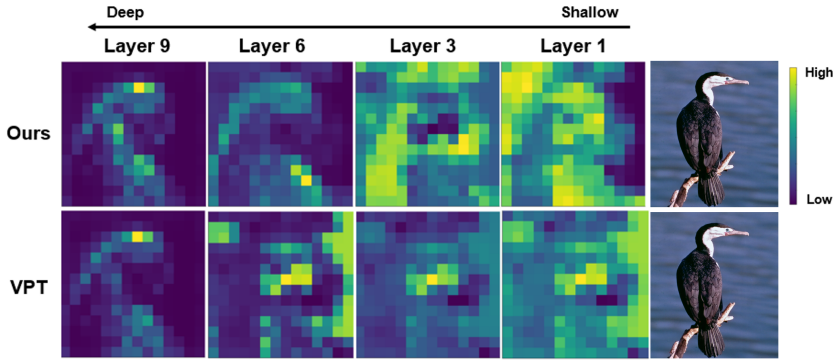

Fig. 5: Layer-wise attention visualization. We compare CLS→patch attention (mean over heads) at four representative layers. Each column uses the same color scale for a fair comparison. Our model shows more concentrated attention over object regions in deeper layers compared to VPT [28].

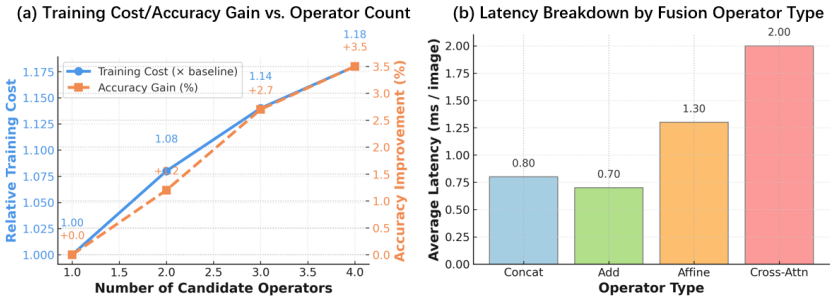

Fig. 6: Efficiency analysis. (a) Training cost scales sublinearly with more candidate operators, while accuracy steadily improves. (b) Latency breakdown shows lightweight `add`/`concat` and moderate overhead from `affine`/`cross-attention`. We use ViT-B/16 as backbone and a batch size of 64 on an NVIDIA A100 GPU.

layer 6, our model assigns less attention to peripheral patches near the right boundary and more to the eye/beak area, whereas VPT continues to spread attention across multiple regions. By layer 9, our model creates a compact hotspot on the eye, while VPT remains more diffuse. These observations should be read as attention-concentration evidence rather than proof of literal background removal: background can be task-relevant, and cross-attention may also act by aggregating prompt-token evidence. Across images, attention entropy decreases with depth and object-region energy increases for our model [21].

3.5 Analysis of Efficiency and Inference Latency

To evaluate the deployment cost of our method, we measure both the training overhead during the search stage and the inference latency after discretization

on VTAB-1k. During search, each transformer block evaluates a soft mixture of candidate fusion operators, which adds a few extra forward passes proportional to the number of candidates. As shown in Figure 6(a), the training cost increases slowly with more operators. With four candidates, the search cost is 1.38× that of VPT-Deep [28] (10.8 vs. 7.8 GPU hours), while the VTAB-1k mean improves by 3.5 points. This extra cost mainly comes from deeper layers that evaluate `cross-attention` during search. After discretization, inference uses only one operator per layer. Our model runs at 15.9 ms per image compared with 14.8 ms for VPT-Deep, a 7.4% increase. Under matched controls, latency-matched and parameter-matched variants still reach 75.83% and 76.42% VTAB-1k mean, respectively, while VPT-Deep reaches 69.43%; full numbers are in Appendix F. As shown in Figure 6(b), `add` and `concat` are very fast, while `affine` and `cross-attention` take longer, matching their computational complexity. The efficiency comes from late-stage discretization, which prunes inactive fusion paths, and from operator heterogeneity, which lets shallow layers use cheap operations while deeper layers use more expressive ones only when needed.

4 Conclusion

In this paper, we challenge the common practice of using a fixed operator to combine prompts and image tokens in visual prompt tuning. We propose a novel approach that treats fusion as a learnable component, jointly optimizing the prompt parameters and their mode of fusion through differentiable architecture search. This exploration encompasses four fusion methods: concatenation, addition, affine transformation, and cross-attention. This allows us to discover diverse and layer-specific fusion techniques that adapt to the representational needs of the frozen backbone. By introducing fusion operator search into the prompt tuning paradigm, our work provides a fresh perspective on understanding and enhancing prompt utilization. This work bridges the gap between neural architecture search and lightweight fine-tuning, presenting an efficient approach to designing prompting solutions for large-scale vision models.

Acknowledgements

This manuscript was co-authored by Oak Ridge National Laboratory (ORNL), operated by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. Any subjective views or opinions expressed in this paper do not necessarily represent those of the U.S. Department of Energy or the United States Government. This research was supported by the National Science Foundation under Grant No. 2450068. This work was supported in part by U.S. NIH grants R35GM158094 and R01GM134020, NSF grants DBI-2238093, DBI-2422619, IIS-2211597, and MCB-2205148. This work was supported in part by an Amazon Research Award (Fall 2025 CFP).

References

1. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)
2. Bromiley, P., Thacker, N., Bouhova-Thacker, E.: Shannon entropy, renyi entropy, and information. Statistics and Inf. Series (2004-004) **9**(2004), 2–8 (2004)
3. Cai, H., Gan, C., Zhu, L., Han, S.: Tinytl: Reduce memory, not parameters for efficient on-device learning. Advances in Neural Information Processing Systems **33**, 11285–11297 (2020)
4. Chen, M., Peng, H., Fu, J., Ling, H.: Autoformer: Searching transformers for visual recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12270–12280 (2021)
5. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. Advances in Neural Information Processing Systems **35**, 16664–16678 (2022)
6. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
7. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9640–9649 (2021)
8. Das, R., Dukler, Y., Ravichandran, A., Swaminathan, A.: Learning expressive prompting with residuals for vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3366–3377 (2023)
9. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms. Advances in neural information processing systems **36**, 10088–10115 (2023)
10. Dong, W., Yan, D., Lin, Z., Wang, P.: Efficient adaptation of large vision transformer via adapter re-composing. Advances in Neural Information Processing Systems **36**, 52548–52567 (2023)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations
12. Du, S., Bai, Y., Ma, J., Liu, M., Li, Y.: Frequency-decoupled learning for joint thin-cloud removal and pansharpening. In: ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 9282–9286. IEEE (2026)
13. Du, S., Zou, Y., Li, J., Liu, M., Li, Y., Shang, C., Shen, Q.: Pansharpening for thin-cloud contaminated remote sensing images: a unified framework and benchmark dataset. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 3696–3704 (2026)
14. Du, S., Zou, Y., Wang, Z., Li, X., Li, Y., Shang, C., Shen, Q.: Unsupervised hyperspectral image super-resolution via self-supervised modality decoupling. International Journal of Computer Vision **134**(4), 152 (2026)
15. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey. Journal of Machine Learning Research **20**(55), 1–21 (2019)
16. Gong, C., Wang, D., Li, M., Chen, X., Yan, Z., Tian, Y., Chandra, V., et al.: Nasvit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In: International Conference on Learning Representations (2021)

17. Han, C., Wang, Q., Cui, Y., Cao, Z., Wang, W., Qi, S., Liu, D.: E 2 vpt: An effective and efficient approach for visual prompt tuning. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17445–17456. IEEE (2023)
18. Han, C., Wang, Q., Cui, Y., Wang, W., Huang, L., Qi, S., Liu, D.: Facing the elephant in the room: Visual prompt tuning or full finetuning? In: The Twelfth International Conference on Learning Representations
19. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
21. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11936–11945 (2021)
22. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: International conference on machine learning. pp. 2790–2799. PMLR (2019)
23. Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations
24. Huang, L., Mao, J., Yi, J., Tao, Z., Wang, Y.: Cvpt: Cross visual prompt tuning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 848–858 (2025)
25. Huang, Q., Dong, X., Chen, D., Zhang, W., Wang, F., Hua, G., Yu, N.: Diversity-aware meta visual prompting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10878–10887 (2023)
26. Huang, Z., Pedapati, T., Chen, P.Y., Gao, J.: Differentiable prompt learning for vision language models. In: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence. pp. 5444–5452 (2025)
27. Iofinova, E., Peste, A., Kurtz, M., Alistarh, D.: How well do sparse imagenet models transfer? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12266–12276 (2022)
28. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European conference on computer vision. pp. 709–727. Springer (2022)
29. Jin, C., Li, Y., Zhao, M., Zhao, S., Wang, Z., He, X., Han, L., Che, T., Metaxas, D.: Lor-vp: Low-rank visual prompting for efficient vision model adaptation. In: International Conference on Learning Representations. vol. 2025, pp. 73004–73021 (2025)
30. Karimi Mahabadi, R., Henderson, J., Ruder, S.: Compacter: Efficient low-rank hypercomplex adapter layers. Advances in neural information processing systems **34**, 1022–1035 (2021)
31. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 conference on empirical methods in natural language processing. pp. 3045–3059 (2021)
32. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4582–4597 (2021)

33. Li, Z., Song, Y., Cheng, M.M., Li, X., Yang, J.: Advancing textual prompt learning with anchored attributes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3618–3627 (2025)
34. Liang, Y., Chongjian, G., Tong, Z., Song, Y., Wang, J., Xie, P.: Evit: Expediting vision transformers via token reorganizations. In: International Conference on Learning Representations
35. Liu, C., Sun, X., Xiao, X., Van Tassel, A., Xu, K., Reimann, K., Liao, D., Gerstein, M., Wang, T., Wang, X., Krishnaswamy, S.: Dispersion loss counteracts embedding condensation and improves generalization in small language models. In: International Conference on Machine Learning. PMLR (2026)
36. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. In: International Conference on Learning Representations
37. Liu, M., Liu, J., Yang, M.Y., Jiang, C., Li, J., Zhang, Y., Wang, H., Nex, F., Cheng, H.: Streamvlo: Streaming visual-lidar odometry with cumulative drift compensation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 39086–39097 (2026)
38. Liu, M., Zhang, D., Liu, J., Cui, J., Xie, H., Chen, G., Ye, H., Yang, M.Y., Nex, F., Cheng, H.: Driveva: Video action models are zero-shot drivers. arXiv preprint arXiv:2604.04198 (2026)
39. Liu, Y., Liang, J., Fan, H., Yang, W., Cui, Y., Han, X., Huangg, L., Liu, D., Wang, Q., Han, C.: All you need is one: Capsule prompt tuning with a single vector. Advances in Neural Information Processing Systems **38**, 88139–88166 (2026)
40. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
41. Mai, Z., Chowdhury, A., Zhang, P., Tu, C.H., Chen, H.Y., Pahuja, V., Berger-Wolf, T., Gao, S., Stewart, C., Su, Y., et al.: Fine-tuning is fine, if calibrated. Advances in neural information processing systems **37**, 136084–136119 (2024)
42. Mai, Z., Zhang, P., Tu, C.H., Chen, H.Y., Nguyen, Q.H., Zhang, L., Chao, W.L.: 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)
43. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian conference on computer vision, graphics & image processing. pp. 722–729. IEEE (2008)
44. Pei, W., Xia, T., Chen, F., Li, J., Tian, J., Lu, G.: Sa^2vp: Spatially aligned-and-adapted visual prompt. In: Proceedings of the AAAI conference on artificial intelligence. vol. 38, pp. 4450–4458 (2024)
45. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
46. Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., Cho, K., Gurevych, I.: Adapterhub: A framework for adapting transformers. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. pp. 46–54 (2020)
47. Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A.: Do vision transformers see like convolutional neural networks? Advances in neural information processing systems **34**, 12116–12128 (2021)

48. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. Advances in neural information processing systems **34**, 13937–13949 (2021)
49. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. Advances in neural information processing systems **30** (2017)
50. Ren, L., Chen, C., Wang, L., Hua, K.: Da-vpt: Semantic-guided visual prompt tuning for vision transformers. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 4353–4363 (2025)
51. Shang, C., Li, M., Zhang, Y., Chen, Z., Wu, J., Gu, F., Lu, Y., Cheung, Y.m.: Pro-vpt: Distribution-adaptive visual prompt tuning via prompt relocation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1558–1568 (2025)
52. Shin, T., Razeghi, Y., Iv, R.L.L., Wallace, E., Singh, S.: Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). pp. 4222–4235 (2020)
53. Skean, O., Arefin, M.R., Zhao, D., Patel, N.N., Naghiyev, J., Lecun, Y., Shwartz-Ziv, R.: Layer by layer: Uncovering hidden representations in language models. In: International Conference on Machine Learning. pp. 55854–55875. PMLR (2025)
54. Spirtes, P., Glymour, C., Scheines, R.: Causation, prediction, and search, vol. 81. Springer Science & Business Media (2012)
55. Su, X., You, S., Xie, J., Zheng, M., Wang, F., Qian, C., Zhang, C., Wang, X., Xu, C.: Vitas: Vision transformer architecture search. In: European Conference on Computer Vision. pp. 139–157. Springer (2022)
56. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. arXiv preprint physics/0004057 (2000)
57. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
58. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
59. Wallace, E., Feng, S., Kandpal, N., Gardner, M., Singh, S.: Universal adversarial triggers for attacking and analyzing nlp. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). pp. 2153–2162 (2019)
60. Wang, J., Zhang, Y., Ding, Z., Hamm, J.: Doctor approved: Generating medically accurate skin disease images through ai-expert feedback. In: Advances in Neural Information Processing Systems (NeurIPS) (2025)
61. Wang, J., Chen, Z., Yuan, C., Li, B., Ma, W., Hu, W.: Hierarchical curriculum learning for no-reference image quality assessment. International Journal of Computer Vision **131**, 3074–3093 (2023)
62. Wang, J., Duan, Y., Tao, X., Xu, M., Lu, J.: Semantic perceptual image compression with a laplacian pyramid of convolutional networks. In: IEEE Transactions on Image Processing. vol. 30, pp. 4225–4237 (2021)
63. Wang, J., Yuan, C., Li, B., Deng, Y., Hu, W., Maybank, S.: Self-prior guided pixel adversarial networks for blind image inpainting. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 45, pp. 12377–12393 (2023)
64. Wang, W., Han, C., Zhou, T., Liu, D.: Visual recognition with deep nearest centroids. In: The Eleventh International Conference on Learning Representations

65. Wang, Y., Cheng, L., Fang, C., Zhang, D., Duan, M., Wang, M.: Revisiting the power of prompt for visual tuning. In: Forty-first International Conference on Machine Learning
66. Wang, Z., Mo, M., Xiao, X., Liu, C., Ma, C., Zhang, Y., Wang, X., Krishnaswamy, S., Wang, T.: Ctr-lora: Curvature-aware and trust-region guided low-rank adaptation for large language models. In: ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1396–1400. IEEE (2026)
67. Xiao, X., Ma, C., Zhang, Y., Liu, C., Wang, Z., Li, Y., Zhao, L., Hu, G., Wang, T., Xu, H.: Not all directions matter: Towards structured and task-aware low-rank model adaptation. arXiv preprint arXiv:2603.14228 (2026)
68. Xiao, X., Tsaris, A., Tabassum, A., Lagergren, J., York, L.M., Wang, T., Wang, X.: Focus: Fused observation of channels for unveiling spectra (2026), https://arxiv.org/abs/2507.14787
69. Xiao, X., Wang, Z., Mo, M., Liu, C., Ma, C., Li, Y., Krishnaswamy, S., Wang, X., Wang, T.: Self-supervised visual prompting for cross-domain road damage detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3514–3524 (2026)
70. Xiao, X., Zhang, Y., Li, X., Wang, T., Wang, X., Wei, Y., Hamm, J., Xu, M.: Visual instance-aware prompt tuning. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 2880–2889 (2025)
71. Xiao, X., Zhang, Y., Li, Y., Li, X., Wang, T., Hamm, J., Wang, X., Xu, M.: Visual variational autoencoder prompt tuning. arXiv preprint arXiv:2503.17650 (2025)
72. Xiao, X., Zhang, Y., Zhao, L., Liu, Y., Liao, X., Mai, Z., Li, X., Wang, X., Xu, H., Hamm, J., Lin, X., Xu, M., Wang, Q., Wang, T., Han, C.: Prompt-based adaptation in large-scale vision models: A survey. Transactions on Machine Learning Research (2026), https://openreview.net/forum?id=UwtXDttgsE
73. Yan, L., Han, C., Xu, Z., Liu, D., Wang, Q.: Prompt learns prompt: Exploring knowledge-aware generative prompt collaboration for video captioning. In: IJCAI. pp. 1622–1630 (2023)
74. Yin, J., Chen, T., Chen, Y., Pei, G., Shu, X., Yao, Y., Shen, F.: Pca-seg: Revisiting cost aggregation for open-vocabulary semantic and part segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 27633–27643 (June 2026)
75. Yin, J., Chen, T., Pei, G., Liu, H., Yao, Y., Nie, L., Hua, X.: Semi-supervised semantic segmentation with multi-constraint consistency learning. IEEE Transactions on Multimedia **27**, 6449–6461 (2025)
76. Yin, J., Jiang, X., Chen, T., Pei, G., Yao, Y., Shen, F., Shen, H.T.: Depmatch: Boosting semi-supervised semantic segmentation by exploring depth difference knowledge. IEEE Transactions on Image Processing **35**, 3256–3270 (2026)
77. Yoo, S., Kim, E., Jung, D., Lee, J., Yoon, S.: Improving visual prompt tuning for self-supervised vision transformers. In: International Conference on Machine Learning. pp. 40075–40092. PMLR (2023)
78. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? Advances in neural information processing systems **27** (2014)
79. Zaken, E.B., Goldberg, Y., Ravfogel, S.: 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)
80. Zeng, R., Han, C., Wang, Q., Wu, C., Geng, T., Huang, L., Wu, Y.N., Liu, D.: Visual fourier prompt tuning. Advances in Neural Information Processing Systems **37**, 5552–5585 (2024)

81. Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A.S., Neumann, M., Dosovitskiy, A., et al.: A large-scale study of representation learning with the visual task adaptation benchmark. arXiv preprint arXiv:1910.04867 (2019)
82. Zhang, H., Huang, B., Li, Z., Xiao, X., Leong, H.Y., Zhang, Z., Long, X., Wang, T., Xu, H.: Sensitivity-lora: Low-load sensitivity-based fine-tuning for large language models. arXiv preprint arXiv:2509.09119 (2025)
83. Zhang, H., Li, Z., Bao, R., Gao, Y., Xiao, X., Zhang, H., Zhang, S., Huang, B., Wu, Y., Wang, T., et al.: Hyperadalora: Accelerating lora rank allocation during training via hypernetworks without sacrificing performance. arXiv preprint arXiv:2510.02630 (2025)
84. Zhang, J.O., Sax, A., Zamir, A., Guibas, L., Malik, J.: Side-tuning: a baseline for network adaptation via additive side networks. In: European conference on computer vision. pp. 698–714. Springer (2020)
85. Zhang, P., Jia, Z., Liu, K., Weng, S., Li, S., Shi, B.: Stage: Storyboard-anchored generation for cinematic multi-shot narrative. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 659–669 (2026)
86. Zhang, P., Weng, S., Zhu, C., Tang, B., Jia, Z., Li, S., Shi, B.: Affective image editing: Shaping emotional factors via text descriptions. International Journal of Computer Vision **134**(1), 16 (2026)
87. Zhang, Y., Cai, C., Liu, F., Hamm, J.: Prime once, then reprogram locally: An efficient alternative to black-box service model adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2026)
88. Zhang, Y., Mehra, A., Hamm, J.: Ot-vp: Optimal transport-guided visual prompting for test-time adaptation. In: IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1122–1132. IEEE (2025), oral Presentation
89. Zhang, Y., Mehra, A., Niu, S., Hamm, J.: Dpcore: Dynamic prompt coreset for continual test-time adaptation. In: International Conference on Machine Learning. pp. 75757–75778. PMLR (2025)
90. Zhang, Y., Niu, S., Cai, C., Liu, F., Hamm, J.: Adapting in the dark: Efficient and stable test-time adaptation for black-box models. arXiv preprint arXiv:2604.15609 (2026)
91. Zhao, L., Li, H., Ning, X., Jiang, X.: Thinimg: Cross-modal steganography for presenting talking heads in images. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 5553–5562 (2024)
92. Zhao, L., Zhao, T., Lin, Z., Ning, X., Dai, G., Yang, H., Wang, Y.: Flasheval: Towards fast and accurate evaluation of text-to-image diffusion generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16122–16131 (2024)