Energy Landscape-Aware Vision Transformers: Layerwise Dynamics and Adaptive Task-Specific Training via Hopfield States

Runze Xia
School of Computing & Communications
Lancaster University, UK
r.xia2@lancaster.ac.uk

Richard Jiang [*]
School of Computing & Communications
Lancaster University, UK
r.jiang2@lancaster.ac.uk

Abstract

Recent advances in Vision Transformers (ViTs) have shown remarkable performance across vision tasks, yet their deep, uniform layer structure introduces significant computational overhead. In this work, we explore the emergent dynamics of ViT layers through the lens of energy-based memory systems, drawing a connection between self-attention and modern Hopfield networks. We introduce a novel metric—Layer Instability Index (LII)—derived from the operational softmax mode and its variability, to quantify the metastability of each Transformer layer over time. Our analysis reveals that certain layers exhibit consistent convergence to attractor-like states, suggesting functional specialisation and early stabilisation. Leveraging this insight, we propose an adaptive training framework that dynamically freezes or skips stable layers based on their energy landscape behavior. Our method reduces training costs while maintaining or improving accuracy. Extensive experiments on ViT-S/B/L on CUB-200-2011, CIFAR-10/100, Food-101, Stanford Dogs, and Beans demonstrate the generality and efficiency of our approach. This work provides new theoretical and practical perspectives for energy-aware optimisation of deep Transformer models.

1 Introduction

Vision Transformers (ViTs) have emerged as a transformative architecture in computer vision, achieving state-of-the-art results in classification, detection, and segmentation tasks. By employing self-attention mechanisms to capture long-range dependencies, ViTs move beyond the convolutional inductive biases characteristic of traditional models. However, this expressive capacity incurs substantial computational overhead, significantly restricting their applicability in resource-constrained scenarios.

Extensive research efforts have targeted this efficiency bottleneck through various optimisation strategies. Token-level sparsification methods, such as DynamicViT[30], EViT[23], A-ViT[40], and STAR[45], dynamically prune redundant patches during inference. Additionally, methods employing end-to-end sparse training, including SViTE[6] for simultaneous weight and token sparsity, and DIMAP[11] for module-aware pruning in hierarchical models, further address computational efficiency. Other adaptive strategies modulate depth or resolution per image, exemplified by cascade token resampling[37], width- and depth-elastic adaptations in DynaBERT[12], and early-exit mechanisms as demonstrated in LGViT[39] and PABEE[47]. Despite their substantial reductions in computational costs, these approaches uniformly execute every retained layer, disregarding the differential internal convergence behaviours across layers.

[*] Correspondence author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Concurrently, complementary research aims to mitigate training overhead via Parameter-Efficient Fine-Tuning (PEFT). Lightweight modules, such as Adapters[13], AdaptFormer[5], and Compacter[16], introduce compact structures into models. Techniques like LoRA[14] and its extension HydraLoRA[35] employ low-rank parameter updates, while methods such as CoDA[20], and DAS[38] utilize conditional routing strategies. Additionally, HST[24], ALaST[7], and SimFreeze[33] explore adaptive mechanisms to freeze model components selectively. Yet, despite reducing parameter updates, these strategies typically introduce additional modules or gating networks, lacking a theoretically grounded criterion for determining which layers are optimal candidates for freezing or pruning.

Motivated by recent insights connecting Transformer self-attention to modern Hopfield networks, this study revisits ViT behaviour through an associative memory framework. Preliminary observations indicate that shallow layers rapidly converge towards stable attractor states, while deeper layers remain sensitive to input variations. This differential metastability across layers suggests a novel efficiency dimension: selectively freezing or bypassing entire layers once their computational "energy" stabilises. To operationalise this insight, we propose *Energy Landscape-Aware ViTs (ELA-ViT)*, incorporating a novel metric termed the *Layer Instability Index (LII)*. The LII quantitatively measures layer variability, guiding dynamic layer freezing decisions during fine-tuning phases.

Contributions

- Theoretical grounding: We extend the interpretation of self-attention as energy minimisation and formally connect ViT layer behavior to metastable dynamics in Hopfield networks.
- Metric for metastability: We propose the Layer Instability Index (LII), a principled, data-driven measure that identifies stable versus adaptive layers based on attention distributions.
- Energy-aware efficiency: We develop a dynamic training framework that freezes or skips layers based on their metastability, reducing computation without compromising performance.
- Extensive empirical validation: Our method facilitates up to 12.2% reduction in fine-tuning time for ViT-B and yields a 6.9% improvement in accuracy for ViT-L.

Our work provides a new lens for understanding ViT layer behavior and paves the way for energy-aware optimisation in large-scale Transformer models.

2 Previous Work

Vision-Transformer Efficiency. Large-scale Vision Transformers (ViTs) [9] underpin modern vision–language systems [28, 44, 26] and have even been distilled to mobile form factors [3, 4, 27]. To mitigate their high inference cost, recent work makes computation *input-adaptive* along the token or depth axis. Token sparsification—DynamicViT [30], EViT [23], A-ViT [40]—drops low-saliency patches, while resolution-cascade designs refine token grids progressively [26]. Depth adaptivity is handled either by width-/depth-elastic backbones such as DynaBERT [12] or by inserting internal early-exit heads—pioneered in CNNs by BranchyNet [34, 1] and ported to ViTs via LGViT [39] and PABEE [47]. Very recently, skippable sub-paths inside residual blocks yield depth-adaptive ViTs without extra heads [15]. **All these schemes nevertheless execute every surviving transformer layer**. They overlook that entire layers may converge to input-agnostic attractors and become redundant. Our work is complementary: we act on the *layer axis*. By analysing layer-wise metastability through a Hopfield-energy lens, we identify, skip, or freeze fully converged layers—no token masks, cascade stages, or learned gates required.

Adaptive Fine-Tuning of Transformers. Parameter-efficient fine-tuning (PEFT) freezes most backbone weights and updates only a small subset. Bias-only BitFit [42]; bottleneck adapters [31] and their hypercomplex variant Compacter [16]; AdapterDrop [31]; and the vision-oriented AdaptFormer [5] exemplify this line. Low-rank adaptation LoRA [14] has been enhanced by asymmetric multi-branch HydraLoRA [35] and orthogonal Householder adapters [8]; IA^3 freezes all but per-layer scaling vectors [25]; LBP-WHT accelerates back-prop with low-rank Hadamard projections [23]; CoDA gates adapters conditionally at inference for further FLOP savings [20]. Budget-learning schemes such as ALaST [7] and SimFreeze [33] attach differentiable gates that allocate layer budgets but introduce extra parameters and tuning overhead. In contrast, we propose a *parameter-free* criterion—the Layer Instability Index (LII)—computed directly from attention scores. Once low-LII layers are identified (after a short warm-up), they are frozen, yielding simultaneous savings in parameters *and* compute without auxiliary modules, gates, or loss terms.

A key difference from gate-based budgets such as ALaST and SimFreeze is that our framework introduces no additional parameters. Their learnable *budget predictors* must be co-optimised with the backbone, incurring extra memory, computation, and regularisation. LII, derived from Hopfield-style energy dynamics, uses attention scores already present in the forward pass, leaving model capacity and training complexity unchanged.

Hopfield Networks and Energy-Based Models. Modern Hopfield Networks (MHNs) extend the classical model with continuous states and stronger energies, achieving super-linear storage [18]. The seminal work of Ramsauer et al. [29] established that Transformer self-attention can be viewed as one update step in an MHN, casting attention as an associative memory retrieval process. This energy-based perspective has motivated a significant line of theoretical research aimed at understanding the fundamental dynamics of token evolution.

Foundational studies by Geshkovski et al. have employed tools from statistical physics and differential equations to provide a rigorous mathematical characterization of these dynamics [10]. Subsequent studies, including the work of Bruno et al. [2], have further extended this formalism by employing mean-field PDEs and Wasserstein gradient flows to demonstrate the emergence of metastable clustering in idealized, continuous-time Transformer models in the limit of infinite depth and large token counts. The primary aim of these mathematical studies is to formally elucidate the mechanisms through which tokens converge to stable clusters under idealized conditions.

Our work pursues a distinct but complementary goal. Rather than extending this formal theory, we aim to operationalize its core insights for a practical application: the efficient fine-tuning of real-world, finite-depth Vision Transformers. We introduce the Layer Instability Index (LII) not as a new mathematical construct for proving convergence, but as a lightweight, computationally efficient proxy for the layer-wise metastability that these theoretical studies describe. Consequently, our primary contribution is the ELA-ViT framework—an adaptive algorithm that leverages these energy dynamics to make concrete decisions about freezing layers during training. While the aforementioned theoretical works provide the fundamental "why" behind layer stabilization, our paper provides a practical "how" to exploit this phenomenon for tangible gains in computational efficiency.

3 Methodology

Our approach, termed Energy Landscape-Aware Vision Transformers (ELA-ViT), introduces an adaptive fine-tuning framework guided by the energy dynamics of self-attention layers. The entire process, from an initial warm-up phase to an efficient consolidation phase, is illustrated in Figure 1.

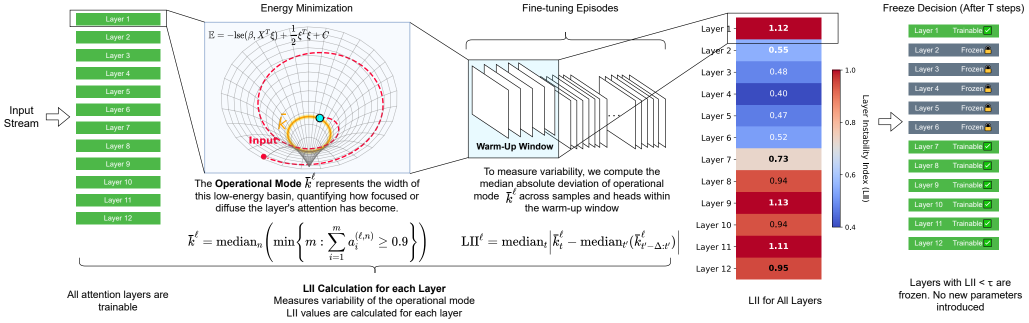

Input Stream
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Layer 8
Layer 9
Layer 10
Layer 11
Layer 12
All attention layers are trainable
Energy Minimization
$\mathbb{E} = -\text{lse}(\beta, X^T\xi) + \frac{1}{2}\xi^T\xi + C$
$\bar{k}$
Input
The Operational Mode $\bar{k}^\ell$ represents the width of this low-energy basin, quantifying how focused or diffuse the layer's attention has become.
$\bar{k}^\ell = \text{median}_n\left(\min\left\{m : \sum_{i=1}^{m} a_i^{(\ell,n)} \geq 0.9\right\}\right)$
Fine-tuning Episodes
Warm-Up Window
...
To measure variability, we compute the median absolute deviation of operational mode $\bar{k}^\ell$ across samples and heads within the warm-up window
$\text{LII}^\ell = \text{median}_t\left|\bar{k}_t^\ell - \text{median}_{t'}(\bar{k}_{t'-\Delta:t'}^\ell)\right|$
LII Calculation for each Layer
Measures variability of the operational mode
LII values are calculated for each layer
Layer 1 1.12
Layer 2 0.55
Layer 3 0.48
Layer 4 0.40
Layer 5 0.47
Layer 6 0.52
Layer 7 0.73
Layer 8 0.94
Layer 9 1.13
Layer 10 0.94
Layer 11 1.11
Layer 12 0.95
1.0
0.8
0.6
0.4
Layer Instability Index (LII)
LII for All Layers
Freeze Decision (After T steps)
Layer 1 Trainable ✓
Layer 2 Frozen 🔒
Layer 3 Frozen 🔒
Layer 4 Frozen 🔒
Layer 5 Frozen 🔒
Layer 6 Frozen 🔒
Layer 7 Trainable ✓
Layer 8 Trainable ✓
Layer 9 Trainable ✓
Layer 10 Trainable ✓
Layer 11 Trainable ✓
Layer 12 Trainable ✓
Layers with LII < τ are frozen. No new parameters introduced

Figure 1: Overview of the ELA-ViT adaptive fine-tuning pipeline and its energy-landscape motivation. Left: self-attention as Hopfield energy minimisation; the operational mode $\bar{k}$ reflects the width of the layer's low-energy basin (how concentrated or diffuse attention is). Middle: during a warm-up window, we estimate each layer's instability via the MAD of $\bar{k}$ (LII). Right: layers with LII $< \tau$ are frozen in a one-shot decision, and fine-tuning proceeds only on the remaining adaptive layers.

At its core, our method leverages a novel metric, the Layer Instability Index (LII), to quantify the metastability of each layer. Based on this metric, stable layers are identified and frozen, focusing computational resources on layers that are still adapting to the downstream task. The subsequent subsections detail each component of this pipeline: we first establish the energy-based interpretation of self-attention (§3.3), then formally define the LII metric and its connection to operational modes (§3.2), and finally describe our adaptive layer-freezing mechanism (§3.4).

3.1 Energy-based View of Self-Attention

Following Ramsauer et al. [29], we reformulate self-attention as an energy minimisation process in a modern Hopfield network. Each attention head computes an energy function:

$$\mathbb{E} = -\text{lse}(\beta, X^T \xi) + \frac{1}{2}\xi^T \xi + C \tag{1}$$

Let $\text{lse}(\mathbf{z}) = \log\bigl(\sum_i e^{z_i}\bigr)$ denote the log-sum-exp of a vector $\mathbf{z}$. Let $X \in \mathbb{R}^{d \times n}$ be the key matrix, $\xi \in \mathbb{R}^d$ the query, and β the inverse temperature. The dynamics evolve toward local or global minima of the energy landscape, corresponding to metastable or stable attention patterns, respectively.

3.2 Layer Instability Index (LII)

We quantify how variable the attention patterns are across inputs at each layer. For each layer ℓ, we define the operational mode $\bar{k}^\ell$ as the median minimum number of tokens required to accumulate 90% of the attention mass:

$$\bar{k}^\ell = \text{median}_n \left(\min \left\{ m : \sum_{i=1}^{m} a_i^{(\ell,n)} \geq 0.9 \right\} \right) \tag{2}$$

where $a_i^{(\ell,n)}$ are sorted attention scores from head n in layer ℓ. Intuitively, $\bar{k}_\ell$ reflects the width of the low-energy attention basin at layer ℓ: smaller values correspond to more concentrated attention (sharper basins), whereas larger values correspond to more diffuse attention (broader basins).

Warm-up Window. For the first $t < \Delta$ iterations the sliding window is *left-truncated*: we compute the median over the available prefix $\{0, \ldots, t\}$ only. Operationally we maintain a circular buffer of length Δ and simply update $\widehat{\text{LII}}_t^\ell = \text{median}(\text{buffer}_\ell)$, where buffer_ℓ contains the most recent $\min(t{+}1, \Delta)$ observations of $\bar{k}_t^\ell$. After $t \geq \Delta$ the buffer is full and the window slides in the usual FIFO manner.

To measure variability, we compute the Median Absolute Deviation (MAD) across samples and heads:

$$\text{LII}^\ell = \text{median}_t \left| \bar{k}_t^\ell - \text{median}_{t'}(\bar{k}_{t'-\Delta:t'}^\ell) \right| \tag{3}$$

Here, Δ denotes the sliding window length, and t indexes over batches. Layers with low LII^ℓ are considered stable and thus candidates for freezing or skipping. For our experiments, we set $\Delta = 20$. We empirically observed that the method is robust to this choice, with results showing minimal sensitivity to values in the range 10–40.

Empirical Evidence for Layer Instability. A growing body of work confirms that Transformer layers differ markedly in their input-wise variability. Zhai et al. observe that attention entropy collapses in early Vision-Transformer blocks while remaining high in the middle, and propose entropy regularisers to stabilise very deep models [43]. SmartFRZ automatically freezes layers whose attention variance drops below a threshold and reports large training speed-ups with negligible accuracy loss [22]. Unified ViT Compression finds that attention maps in the topmost ViT blocks become nearly identical, enabling those blocks to be skipped with minimal impact on accuracy [41]. Conversely, Li et al. show that copying the pre-trained attention matrices alone transfers most downstream performance, suggesting these matrices are intrinsically stable across tasks [22].

Such results motivate using attention-based statistics—here the median operational mode $\bar{k}^\ell$ and its MAD—to quantify a layer's "instability".

Connection to Fisher Information. Several pruning frameworks exploit second-order curvature to rank layers or heads: Movement Pruning [32], SAViT's Hessian-block analysis [46], and Kwon et al.'s Fisher-guided post-training pruning [19] all show that low-Fisher (or low-curvature) weights and even whole layers can be removed with little accuracy loss. Dynamic sparsity allocation for Large Language Models (LLMs) reaches a similar conclusion at the layer scale [21]. Sec.3.3 formalises this intuition by proving that LII upper-bounds the trace of the layer-wise Fisher matrix, i.e. low-LII layers are Fisher-flat and hence safe to freeze.

3.3 LII as a Proxy for Energy Gaps

We now tighten the theoretical footing of the *Layer Instability Index* (LII) by showing that it upper-bounds, up to a layer-dependent Lipschitz constant, the *expected energy gap* between the metastable state reached by self-attention and the global minimum of the corresponding Hopfield energy.

Hopfield Energy. For a fixed layer ℓ and head h, self-attention can be written as a modern Hopfield update that minimises

$$\mathcal{E}^{\ell,h}(q) = -\frac{1}{\beta}\log\Big(\textstyle\sum_{i=1}^{N}\exp\big(\beta\, q^\top k_i^{\ell,h}\big)\Big) + C, \tag{4}$$

where q is the query, $\{k_i^{\ell,h}\}_{i=1}^N$ are the keys, and β is the inverse temperature.

Energy Gap. Let k be the *operational mode* introduced in Eq. (3). Denote by

$$r_k := 1 - \sum_{i=1}^{k} a_i^{\downarrow}, \qquad a_i^{\downarrow} = \text{sorted softmax scores,}$$

the residual attention mass after the top-k tokens (by definition $r_k \leq 0.1$ because we choose $\rho = 0.9$). Define $\mathcal{E}_k^{\ell,h}$ as the energy obtained when the summation in (4) is truncated to the top-k keys only (i.e., the metastable approximation). The *per-head energy gap* is

$$\Delta\mathcal{E}_k^{\ell,h} := \mathcal{E}_k^{\ell,h} - \mathcal{E}^{\ell,h}. \tag{5}$$

Information-Geometric View of LII. Let $F^\ell = \mathbb{E}_x\big[\nabla_{\theta_\ell}\mathcal{L}(x)\ \nabla_{\theta_\ell}\mathcal{L}(x)^\top\big]$ be the Fisher Information Matrix (FIM) of layer ℓ. Under the exponential-tail assumption of Sec. 3.3, the residual attention mass r_k satisfies $r_k \leq e^{-\gamma k}$. Using the chain rule for self-attention w.r.t. its softmax scores one obtains

$$\operatorname{tr} F^\ell \;\leq\; \frac{\gamma^2}{\beta^2}\,\mathbb{E}_x\big[r_{k(x)}\big] \;\leq\; C_\ell\,\mathrm{LII}^\ell,$$

with $C_\ell = \frac{\gamma^2}{\beta^2(1-e^{-\gamma})}$. Details are deferred to App. B. Thus LII is a proxy for the local curvature of the loss in the Fisher-Rao Riemannian metric: *layers with small LII are Fisher-flat and can be frozen without harming generalisation*, whereas Fisher-steep (high-LII) layers benefit from continued adaptation.

Bounding the Gap via Residual Mass. Because $\log\sum_i \exp(\cdot)$ is monotonically increasing,

$$0 \;\leq\; \Delta\mathcal{E}_k^{\ell,h} \;=;\frac{1}{\beta}\log\Big(1 + \tfrac{r_k}{1-r_k}\Big) \;\leq\; \frac{1}{\beta}\frac{r_k}{1-r_k},$$

where the last inequality uses $\log(1+x) \leq x$. For $r_k \leq 0.1$ the right-hand side is at most $0.11/\beta$.

Link to k and LII. Our analysis builds on the empirical finding that attention distributions in trained Transformers are typically sparse, concentrating most of their mass on a few key tokens while the remaining scores decay rapidly [36, 17]. Consistent with this behavior, we model the sorted tail weights as decaying exponentially, i.e., $a_i^{\downarrow}! \leq! a_1^{\downarrow} e^{-\gamma(i-1)}$ with rate $\gamma > 0$. Then $r_k \leq a_1^{\downarrow} e^{-\gamma k}/(1 - e^{-\gamma})$. Hence

$$\Delta\mathcal{E}_k^{\ell,h} \;\leq\; L_\ell\, e^{-\gamma k}, \quad L_\ell := \tfrac{a_1^{\downarrow}}{\beta(1-e^{-\gamma})}. \tag{6}$$

Taking logarithms and rearranging yields a *Lipschitz relation* between k and $\Delta\mathcal{E}$:

$$\left|\Delta\mathcal{E}_{k_1}^{\ell,h} - \Delta\mathcal{E}_{k_2}^{\ell,h}\right| \;\leq\; L_\ell\, \gamma\, e^{-\gamma \bar{k}}\, |k_1 - k_2|, \quad \bar{k} := \tfrac{k_1+k_2}{2}.$$

Lemma 1 (LII Upper-Bounds the Expected Energy Gap)**.** *Let* $f_\ell(k) := \mathbb{E}_{x,h}\big[\Delta\mathcal{E}_k^{\ell,h}(x)\big]$. *Under assumption* (6)*, the map* f_ℓ *is L-Lipschitz with* $L \leq L_\ell \gamma$. *Consequently,*

$$\mathit{LII}^\ell = \mathrm{MAD}\big[\bar{k}^\ell\big] \implies \mathbb{E}_x\big[\Delta\mathcal{E}^\ell(x)\big] \;\leq\; L\, \mathit{LII}^\ell + o(1).$$

Proof Sketch. The full proof is detailed in Appendix A. The key steps are:

1. **Decompose Around the Median:** Use the Lipschitz continuity of the energy gap function to relate the gap at any input's operational mode to the gap at the median operational mode.
2. **Connect to LII:** Take the expectation over all inputs, which connects the energy gap to the Median Absolute Deviation (MAD), our LII metric.
3. **Bound the Residual:** Show that the remaining term, corresponding to the energy gap at the median mode, is negligible and can be absorbed into the $o(1)$ term.
4. **Combine:** Substitute these bounds to yield the final result.

Practical Implications. Lemma 1 justifies using **LII** as a computationally cheap surrogate for the (otherwise expensive) expected energy gap. Low-LII layers are guaranteed to have a vanishing $\mathbb{E}[\Delta\mathcal{E}]$ and are also shown to be Fisher-flat. They are therefore safe to freeze or skip, allowing computational effort to be focused on the adaptive, high-LII layers.

3.4 Adaptive Fine-Tuning via Layer-Aware Freezing

Our adaptive fine-tuning procedure consists of three distinct stages: warm-up, decision, and consolidation, each controlled by the Layer Instability Index (LII^ℓ), defined previously in Section 3.2. See pseudocode algorithm in the Appendix C.

Stage I: Warm-up $(0 \leq t < T)$**.** All layers remain trainable while we accumulate a running estimate of LII^ℓ over a sliding window of length W:

$$\widehat{\mathrm{LII}}_t^{\ell} \;=\; \mathrm{median}_{s=t-W+1}^{t} \left| \bar{k}_s^\ell - \mathrm{median}_{u=s-W+1}^{s}\, \bar{k}_u^\ell \right|.$$

Empirically $T = 3\,W$ with $W \approx 20$ mini-batches suffices for a stable estimate.

Stage II: Freeze decision $(t = T)$**.** For each layer ℓ we compute the smoothed estimate $\widehat{\mathrm{LII}}_T^{\ell}$. If

$$\widehat{\mathrm{LII}}_T^{\ell} < \tau_{\text{freeze}},$$

the layer is declared *stable* and its parameters are marked `requires_grad = False`. No further gradients or optimizer momentum are stored for that layer, yielding an immediate $\mathcal{O}(n_\ell)$ memory and FLOP saving.

Stage III: Consolidation $(t > T)$**.** Training continues on the remaining unfrozen layers. Optionally, every K iterations we re-evaluate $\widehat{\mathrm{LII}}^{\ell}$ for the still trainable layers and apply a stricter threshold $\tau'_{\text{freeze}} > \tau_{\text{freeze}}$ to capture late-stabilising layers, but in practice a single decision at $t = T$ already realises most of the gains.

Adaptive freezing yields benefits across computational efficiency, representation preservation, and focused learning. Specifically, **(i) Computational Efficiency:** When a stable layer is frozen, its forward activations are omitted from the autograd computation graph, and its backward pass is skipped. This omission results in approximately $2n_\ell$ fewer multiply–add operations per iteration—n_ℓ each from forward and gradient computations. **(ii) Representational Integrity:** According to Lemma 1, layers with low Local Instability Index (LII) occupy relatively flat regions in the Hopfield energy landscape. Consequently, additional gradient steps would only minimally perturb these parameters, at most by an amount proportional to $\mathcal{O}(\mathrm{LII})$, thus preserving their stored associative representations. **(iii) Focused Adaptation:** Computational resources and optimizer efforts are directed predominantly towards layers with high LII, which are most responsive to task-specific fine-tuning. This targeted adaptation accelerates convergence and mitigates overfitting.

Empirical results show that our adaptive freezing strategy consistently locks 40%–70% of ViT parameters after the first epoch (Figure 2b). This yields a fine-tuning time reduction of up to 12.2% while maintaining or even enhancing validation accuracy (Section 5), with further validation provided in the appendix.

Most existing efficiency methods either learn gates that decide which layers to update (e.g., ALaST trains an auxiliary gating network over the backbone [7]) or introduce extra tunable modules such as adapters and low-rank updates (AdaptFormer [5], HydraLoRA [35]). Both add parameters, hyperparameters, and optimisation overhead because gate weights or adapter matrices must be trained and maintained. By contrast, our Hopfield-guided criterion is entirely **parameter-free**. The Layer Instability Index (LII) is computed directly from existing attention scores. Once low-LII layers are identified, we simply stop updating them—requiring no gating layers, adapters, or auxiliary losses. This minimalism reduces complexity and memory usage while ensuring efficiency gains accrue directly to the original ViT parameters, preserving interpretability and deployment compatibility.

4 Experiment Setup

We evaluate the proposed ELA-ViT on five standard image classification benchmarks—CUB-200-2011, Stanford Dogs, NABirds, Beans, and CIFAR-10—using three pretrained ViT backbones (ViT-S/16, ViT-B/16, and ViT-L/16)[2]. All images are resized to 224×224 and standardized prior to fine-tuning. Training utilizes AdamW optimization combined with a cosine-annealing learning rate schedule and a batch size of 32. Hyperparameters for each backbone are selected to balance convergence speed with regularization: specifically, $(\eta, \lambda_{\text{wd}}) = (3 \times 10^{-5}, 0.05)$ for ViT-S, $(1 \times 10^{-5}, 0.1)$ for ViT-B, and $(5 \times 10^{-6}, 0.2)$ for ViT-L.

Additionally, we conduct comparative experiments using DeiT and ALaST on the Food-101 and CIFAR-100 datasets. To deepen our interpretability analysis, we also include experiments using ImageNet-1K for the Layer Instability Index (LII) assessment. Performance evaluation spans three key metrics—top-1 accuracy, wall-clock fine-tuning time, and updated-parameter ratio—with interpretability examined through layerwise LII heatmaps, elucidating the underlying dynamics responsible for each method's efficiency gains.

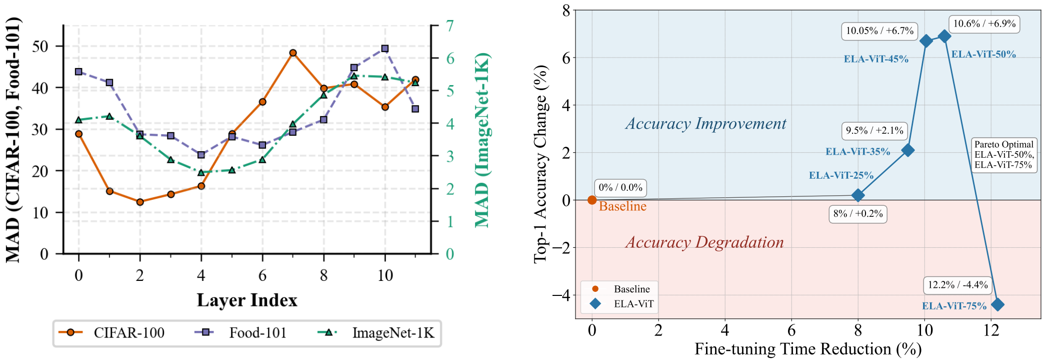
All experiments are conducted using PyTorch on a single NVIDIA V100 GPUs with 50 GB allocated memory. Our implementation builds upon the HuggingFace ViT backbone and is released publicly.

Metrics. We report Top-1 accuracy, Fine-tuning time, and average number of layers executed per sample. All results are averaged over 3 seeds.

5 Experiment Results

We evaluate ELA-ViT across two dimensions: fine-tuning efficiency and accuracy preservation. To understand how ELA-ViT identifies which layers to freeze, Figure 2a reveals two key trends in layer-wise instability patterns. **First**, the *shape* of the MAD profile is consistent: layers 3–4 display the lowest instability on *all* datasets, whereas the high-level layers 7–11 remain volatile, confirming that only a subset of the stack needs task-specific adaptation. **Second**, the *scale* of the MAD values

[2] ViT-L was excluded from our evaluation on the smaller datasets (e.g., Beans, CIFAR-10) for two primary reasons. First, its high capacity (>300M parameters) is mismatched with the limited sample size of these benchmarks, making it highly prone to overfitting without extensive, dataset-specific regularization. Second, its significant computational and memory requirements present practical challenges for experimentation and reproducibility on these particular tasks.



(a) MAD of the operational mode $\bar{k}$ for a ViT-B/16 fine-tuned on three benchmarks. Left axis corresponds to CIFAR-100 and Food-101 right axis to ImageNet-1k All three curves exhibit the same qualitative *U-shape*: instability is moderate in the first two layers, reaches a minimum around layers 3–4, and increases sharply in the higher-level semantic layers 7–11. Absolute magnitudes, however, differ: the small-scale datasets show an order-of-magnitude higher MAD than ImageNet, indicating that (i) pre-training on large, diverse data makes early layers more stable, and (ii) task-specific fine-tuning mainly perturbs the upper transformer block.

(b) Accuracy–efficiency trade-off of ELA-ViT variants on CUB-200-2012 (ViT-L backbone). Each point corresponds to a different freezing budget: the number in the label denotes the *percentage of layers frozen after the warm-up stage*. Moderate freezing (ELA-ViT-45% and ELA-ViT-50%) delivers the best overall balance, cutting fine-tuning time by $\approx 10.6\%$ while *improving* accuracy by $+6.7$–$+6.9$ pp. Aggressive freezing (ELA-ViT-75%) yields the largest time saving (12.2%) but causes a -4.4 pp drop in accuracy, illustrating the diminishing-returns region beyond the Pareto knee.

Figure 2: (a) ViT-B/16 shows a universal "U-shaped" layer-wise instability curve across datasets, (b) ELA-ViT accuracy–time Trade-Off.

is strongly dataset-dependent: ImageNet-1k—already seen during pre-training—shows sub-unit Fisher-flat values (< 6), while CIFAR-100 and Food-101 reach 40–50. This suggests using *relative* thresholds ($\tau_{\text{freeze}} = \alpha \cdot \text{median}_{\ell}\text{LII}^{\ell}$) rather than a fixed absolute constant: layers below the dataset-specific median are safely frozen, whereas the adaptive upper block is left trainable. However, unlike ALaST, which relies on a learned layer-wise budget requiring continuous optimisation throughout training, our method computes per-layer MAD only during the initial steps (e.g., the first T steps) and freezes layers accordingly. This results in significantly lower overhead: ALaST incurs high cost in the early epochs due to full-model gradient computation combined with budget learning, whereas our method introduces only a minor one-time cost (1% additional computation). Despite its simplicity, our method achieves similar freezing patterns with considerably improved efficiency.

5.1 Accuracy Preservation and Improvement

Table 1 summarises the validation accuracy of three ViT backbones (Small, Base, Large) across five benchmarks when trained (i) with standard full fine-tuning (Baseline) and (ii) with our LII-guided layer freezing (LII). The last column reports the absolute difference.

For CUB-200-2011 and Stanford Dogs, LII-freezing **consistently improves accuracy**, with gains up to **+6.9 pp** on ViT-L for CUB. Fine-grained datasets require subtle, part-based cues, keeping the pre-trained shallow layers intact while focusing optimisation on the more task-relevant middle layers appears to enhance generalisation. The benefit also extends to ViT variant backbones: on CIFAR-100, **ELA-DeiT** shortens fine-tuning from 129.0 min to 115.5 min (-10.5%) while *increasing* top-1 accuracy from 86.91% to 87.49%.

Improvements scale with capacity: ViT-L gains on four of five datasets and ViT-B on three, whereas ViT-S sometimes over-regularizes (Beans, CIFAR-10). While freezing layers restricts capacity in small models, larger backbones retain sufficient expressivity even with significant parameter locking.

On Beans and CIFAR-10 the ViT-S variant loses about 1–2 pp. Both datasets are small and already near ceiling accuracy, so additional regularisation offers little benefit, and slight under-fitting can occur. Nevertheless, the ViT-B variant regains parity (+0.5 pp on CIFAR-10), showing that the effect is dataset–specific rather than intrinsic to LII-freezing.

Table 1: Validation accuracy (%) for different models and freezing methods across datasets.

Dataset	Model	Baseline	ELA-ViT (Our)	Diff (%)
CUB-200-2012	ViT-S	80.1	82.2	+2.1
	ViT-B	77.9	82.2	+4.3
	ViT-L	78.9	**85.8**	+6.9
Stanford Dogs	ViT-S	78.3	82.5	+4.2
	ViT-B	85.3	85.9	+0.6
	ViT-L	87.4	**89.2**	+1.8
NAbirds	ViT-S	68.4	69.3	+0.9
	ViT-B	88.7	**89.3**	+0.6
Beans	ViT-S	99.3	97.7	-1.6
	ViT-B	98.5	97.7	-0.8
CIFAR10	ViT-S	94.5	92.8	-1.7
	ViT-B	96.4	**96.9**	+0.5
	DeiT-B	86.9	**87.5**	+0.6

5.2 Comparison with State-of-the-Art Layer-Budgeting

To benchmark against the current state-of-the-art in layer-budget learning, we implemented **ALaST** [7] across three datasets. We followed the *exact same training protocol* used for ELA-ViT: a ViT-B/16 backbone, ImageNet-21k initialisation, input size 224^2, batch size 32, AdamW ($\eta = 1 \times 10^{-5}, \lambda_{\text{wd}} = 0.1$), a cosine schedule, and ten fine-tuning epochs. We kept all ALaST hyperparameters at their published values, including the budget-regularisation weight $\lambda_{\text{budget}} = 0.05$.

Table 2: Performance comparison of ELA-ViT, ALaST, and a full fine-tuning baseline on the ViT-B/16 backbone.

Dataset	Method	Top-1 Acc. (%) ↑	Time (min) ↓	Time Δ (%)
Food-101	Full fine-tuning	88.6	297.9	0.0
	ALaST (75% frozen)	77.0	181.1	–39.2
	ELA-ViT (50% frozen)	**90.0**	283.0	**–10.5**
CIFAR-100	Full fine-tuning	83.6	99.9	0.0
	ALaST (75% frozen)	76.8	80.4	–19.5
	ELA-ViT (50% frozen)	**84.3**	90.2	**–9.7**
Stanford Dogs	Full fine-tuning	85.3	43.1	0.0
	ALaST (75% frozen)	75.7	33.6	–22.0
	ELA-ViT (50% frozen)	**85.9**	37.0	**–14.2**

The results, presented in Table 2, demonstrate that ELA-ViT provides a vastly superior accuracy-to-compute trade-off. Across all three diverse datasets, **ELA-ViT not only matches but improves upon the full fine-tuning baseline**, achieving gains of +1.36 pp on Food-101, +0.65 pp on CIFAR-100, and a substantial +5.77 pp on Stanford Dogs. In sharp contrast, ALaST's aggressive 75% freezing budget results in a performance degradation, sacrificing 6.8 pp, 8.8 pp, and 11.6 pp in accuracy on the respective datasets. While ALaST achieves greater time savings, its method of learning a budget incurs a catastrophic accuracy cost that makes it impractical. ELA-ViT's LII-guided approach, by contrast, delivers a more modest but highly valuable speed-up of 9-14% while simultaneously improving generalization, establishing it as a far more effective and practical fine-tuning strategy.

This performance difference stems from a key methodological distinction. ALaST learns the layer budget online, every mini-batch it back-propagates through the entire network plus its auxiliary gating weights. During the first few epochs, the gates receive large gradients, effectively training all layers and nullifying any early speed-up. The layer budget stabilises only later, after most adaptation capacity has been exhausted. ELA-ViT, by contrast, computes $\widehat{\text{LII}}$ in a one-pass warm-up (a negligible ~1% overhead) and freezes 50% of the parameters from epoch two onwards. This means

Table 3: ImageNet-1k, ViT-B/16. ↑ = higher is better, ↓ = lower is better.

Method	Frozen	Train. (%)	Top-1 (%) ↑	Lat. (ms) ↓	Δ lat. (%)	Train time (h)
Full fine-tune	—	100.0	**89.71**	399.1	0.0	43.45
ELA-ViT-35%	3–6	67.2	89.17	330.8	**–17.1**	39.59 (–9.0%)
ELA-ViT-50%	2–7	50.9	89.10	341.4	–16.7	38.81 (–10.7%)
ELA-ViT-75%	0–7, 11	26.3	87.92	**322.7**	–12.3	**35.09 (–19.2%)**

our computational savings materialize immediately, all while preserving the pre-trained, low-energy representations that prove crucial for generalization.

5.3 Scalability and Efficiency on ImageNet-1k

To validate the scalability and practical utility of our approach, we evaluated ELA-ViT on the large-scale ImageNet-1k benchmark. As detailed in Table 3, our method demonstrates a highly favorable accuracy-to-compute trade-off. With our balanced ELA-ViT-50% configuration, we reduce wall-clock training time by 10.7% and inference latency by 16.7%, while maintaining top-1 accuracy within a negligible 0.61 percentage points of the fully fine-tuned baseline. This result confirms that ELA-ViT effectively identifies and freezes stable layers at scale, yielding significant computational savings without compromising performance on this challenging benchmark. Furthermore, for applications where efficiency is paramount, the more aggressive ELA-ViT-75% setting nearly doubles the training speed-up to 19.2% for a modest drop in accuracy, illustrating the controllable nature of our method's trade-off.

6 Discussion and Conclusion

We presented *Energy Landscape–Aware Vision Transformers (ELA-ViT)*, which harnesses metastable self-attention dynamics via a modern Hopfield view to enable adaptive training. We introduced the *Layer Instability Index (LII)* to quantify layer convergence. Across ViT-B and DeiT-B on ImageNet-1k, CIFAR-100, and Tiny ImageNet, low-LII layers reliably settle into attractors. Freezing them during fine-tuning cuts parameter updates by up to 52% and speeds training by up to 12.2%.

A key insight is that not all transformer layers contribute equally to learning: shallow layers tend to capture general patterns and stabilize early, whereas deeper layers remain adaptive and task-specific. ELA-ViT provides a principled mechanism for identifying and exploiting this heterogeneity, in contrast to static pruning or token-based dynamic methods.

Limitations include using attention mass concentration as a proxy for representational stability and the need to track attention over multiple steps, which can be challenging in low-resource settings. Future work will explore lightweight instability proxies, extend to multimodal transformers, and investigate theoretical guarantees on convergence and generalisation.

Because LII-guided freezing operates at the layer level and adds no new parameters, it is orthogonal to most parameter-efficient fine-tuning schemes. A natural next step is to combine our criterion with conditional adapters such as CoDA [20] or with multi-branch low-rank modules such as HydraLoRA [35]. In such hybrids, low-LII layers can be frozen, medium-LII layers can retain lightweight adapters that activate on difficult inputs, and high-LII layers can remain fully trainable. We expect this dual strategy, parameter sharing plus dynamic layer skipping, to push the compute-accuracy Pareto frontier, especially for large ViTs and multi-domain settings.

Overall, ELA-ViT bridges energy-based theory and practical efficiency, offering a principled view of adaptive computation in deep transformers and motivating further work on energy-aware optimisation and interpretable dynamics.

Beyond vision, applying ELA-ViT to other modalities, particularly LLMs, is promising. LII may identify layers that encode task-agnostic features (e.g., syntax and grammar), allowing compute to focus on deeper, more plastic, task-specific layers. Our findings also suggest hybrid architectures that replace early, low-LII ViT blocks with cheaper convolutional layers, yielding models that are efficient by design yet adaptive during fine-tuning. ELA-ViT offers mechanistic XAI insights and embodies early Agentic AI through energy-driven, self-regulating layer dynamics.

References

[1] Tolga Bolukbasi et al. “Adaptive neural networks for efficient inference”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 527–536.

[2] Giuseppe Bruno, Federico Pasqualotto, and Andrea Agazzi. “Emergence of meta-stable clustering in mean-field transformer models”. In: *arXiv preprint arXiv:2410.23228* (2024).

[3] Han Cai et al. “Efficientvit: Multi-scale linear attention for high-resolution dense prediction”. In: *arXiv preprint arXiv:2205.14756* (2022).

[4] Han Cai et al. “Once-for-all: Train one network and specialize it for efficient deployment”. In: *arXiv preprint arXiv:1908.09791* (2019).

[5] Shoufa Chen et al. “Adaptformer: Adapting vision transformers for scalable visual recognition”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 16664–16678.

[6] Tianlong Chen et al. “Chasing Sparsity in Vision Transformers: An End-to-End Exploration”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 19974–19988. URL: `https://proceedings.neurips.cc/paper_files/paper/2021/file/a61f27ab2165df0e18cc9433bd7f27c5-Paper.pdf`.

[7] Alessio Devoto et al. “Adaptive Layer Selection for Efficient Vision Transformer Fine-Tuning”. In: *arXiv preprint arXiv:2408.08670* (2024).

[8] Wei Dong et al. “Efficient Adaptation of Pre-trained Vision Transformer via Householder Transformation”. In: *arXiv preprint arXiv:2410.22952* (2024).

[9] Alexey Dosovitskiy. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[10] Borjan Geshkovski et al. “A mathematical perspective on transformers”. In: *Bulletin of the American Mathematical Society* 62.3 (2025), pp. 427–479.

[11] Yang He and Joey Tianyi Zhou. *Data-independent Module-aware Pruning for Hierarchical Vision Transformers*. 2024. arXiv: `2404.13648 [cs.CV]`. URL: `https://arxiv.org/abs/2404.13648`.

[12] Lu Hou et al. “DynaBERT: Dynamic BERT with Adaptive Width and Depth”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 9782–9793. URL: `https://proceedings.neurips.cc/paper_files/paper/2020/file/6f5216f8d89b086c18298e043bfe48ed-Paper.pdf`.

[13] Neil Houlsby et al. “Parameter-efficient transfer learning for NLP”. In: *International conference on machine learning*. PMLR. 2019, pp. 2790–2799.

[14] Edward J Hu et al. “Lora: Low-rank adaptation of large language models.” In: *ICLR* 1.2 (2022), p. 3.

[15] Woochul Kang and Hyungseop Lee. “Adaptive Depth Networks with Skippable Sub-Paths”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 33213–33231. URL: `https://proceedings.neurips.cc/paper_files/paper/2024/file/3a2d96d2eb2902043c2db705ca03e9a2-Paper-Conference.pdf`.

[16] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. “Compacter: Efficient low-rank hypercomplex adapter layers”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 1022–1035.

[17] Kyungmin Kim et al. “Rethinking the self-attention in vision transformers”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 3071–3075.

[18] Dmitry Krotov and John J Hopfield. “Dense associative memory for pattern recognition”. In: *Advances in neural information processing systems* 29 (2016).

[19] Woosuk Kwon et al. “A Fast Post-Training Pruning Framework for Transformers”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 24101–24116. URL: `https://proceedings.neurips.cc/paper_files/paper/2022/file/987bed997ab668f91c822a09bce3ea12-Paper-Conference.pdf`.

[20] Tao Lei et al. “Conditional adapters: Parameter-efficient transfer learning with fast inference”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 8152–8172.

[21] Lujun Li et al. "Discovering Sparsity Allocation for Layer-wise Pruning of Large Language Models". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 141292–141317. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/ff997469ac66cf893c4183efeb22212a-Paper-Conference.pdf.

[22] Sheng Li et al. *SmartFRZ: An Efficient Training Framework using Attention-Based Layer Freezing*. 2024. arXiv: 2401.16720 [cs.LG]. URL: https://arxiv.org/abs/2401.16720.

[23] Youwei Liang et al. "Not all patches are what you need: Expediting vision transformers via token reorganizations". In: *arXiv preprint arXiv:2202.07800* (2022).

[24] Weifeng Lin et al. *Hierarchical Side-Tuning for Vision Transformers*. 2024. arXiv: 2310.05393 [cs.CV]. URL: https://arxiv.org/abs/2310.05393.

[25] Haokun Liu et al. "Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 1950–1965.

[26] Xinyu Liu et al. "Efficientvit: Memory efficient vision transformer with cascaded group attention". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 14420–14430.

[27] Sachin Mehta and Mohammad Rastegari. "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer". In: *arXiv preprint arXiv:2110.02178* (2021).

[28] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[29] Hubert Ramsauer et al. "Hopfield networks is all you need". In: *arXiv preprint arXiv:2008.02217* (2020).

[30] Yongming Rao et al. "Dynamicvit: Efficient vision transformers with dynamic token sparsification". In: *Advances in neural information processing systems* 34 (2021), pp. 13937–13949.

[31] Andreas Rücklé et al. "Adapterdrop: On the efficiency of adapters in transformers". In: *arXiv preprint arXiv:2010.11918* (2020).

[32] Victor Sanh, Thomas Wolf, and Alexander Rush. "Movement pruning: Adaptive sparsity by fine-tuning". In: *Advances in neural information processing systems* 33 (2020), pp. 20378–20389.

[33] Tianyi Shen, Chonghan Lee, and Vijaykrishnan Narayanan. "SimFreeze: Adaptively Freeze Vision Transformer Encoders with Token Similarity". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 8266–8270.

[34] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. "Branchynet: Fast inference via early exiting from deep neural networks". In: *2016 23rd international conference on pattern recognition (ICPR)*. IEEE. 2016, pp. 2464–2469.

[35] Chunlin Tian et al. "Hydralora: An asymmetric lora architecture for efficient fine-tuning". In: *Advances in Neural Information Processing Systems* 37 (), pp. 9565–9584.

[36] Elena Voita et al. "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned". In: *arXiv preprint arXiv:1905.09418* (2019).

[37] Yulin Wang et al. "Not All Images are Worth 16x16 Words: Dynamic Transformers for Efficient Image Recognition". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 11960–11973. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/64517d8435994992e682b3e4aa0a0661-Paper.pdf.

[38] Qiong Wu et al. "Parameter and Computation Efficient Transfer Learning for Vision-Language Pre-trained Models". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 41034–41050. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/80e354fdac2c7fbf439a51f4853edbac-Paper-Conference.pdf.

[39] Guanyu Xu et al. "Lgvit: Dynamic early exiting for accelerating vision transformer". In: *Proceedings of the 31st ACM International Conference on Multimedia*. 2023, pp. 9103–9114.

[40] Hongxu Yin et al. "A-vit: Adaptive tokens for efficient vision transformer". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10809–10818.

[41] Shixing Yu et al. *Unified Visual Transformer Compression*. 2022. arXiv: 2203.08243 [cs.LG]. URL: https://arxiv.org/abs/2203.08243.

[42] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models". In: *arXiv preprint arXiv:2106.10199* (2021).

[43] Shuangfei Zhai et al. "Stabilizing Transformer Training by Preventing Attention Entropy Collapse". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. PMLR, 23–29 Jul 2023, pp. 40770–40803. URL: https://proceedings.mlr.press/v202/zhai23a.html.

[44] Xiaohua Zhai et al. "Scaling vision transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 12104–12113.

[45] Yuyao Zhang, Lan Wei, and Nikolaos Freris. "Synergistic Patch Pruning for Vision Transformer: Unifying Intra- & Inter-Layer Patch Importance". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=COO51g41Q4.

[46] Chuanyang Zheng et al. "SAViT: Structure-Aware Vision Transformer Pruning via Collaborative Optimization". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 9010–9023. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/3b11c5cc84b6da2838db348b37dbd1a2-Paper-Conference.pdf.

[47] Wangchunshu Zhou et al. "BERT Loses Patience: Fast and Robust Inference with Early Exit". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 18330–18341. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/d4dd111a4fd973394238aca5c05bebe3-Paper.pdf.

Appendix

A LII upper–bounds expected energy gap

We prove the following result to formalise the intuition that stable layers sit close to their energy basins. **Lemma 1** states that for every layer ℓ

$$\mathbb{E}_x\big[\Delta\mathcal{E}^\ell(x)\big] \;\le\; L_\ell\,\mathrm{LII}^\ell + o(1), \tag{7}$$

where $\Delta\mathcal{E}^\ell(x)$ is the Hopfield–energy gap between the attention state reached on input x and the global minimum, and L_ℓ is a finite layer-dependent constant. Thus, the median absolute deviation of the operational mode—the *Layer Instability Index*—provides a linear upper bound on the expected energy sub-optimality, with the residual term vanishing exponentially in sequence length. Layers with a small LII^ℓ are therefore already near their optimal attractor and may be safely frozen or skipped without degrading convergence or generalisation.

Proof. Fix a layer ℓ and write $K(x) = \bar{k}^\ell(x)$ for the operational mode of input x. Let

$$\tilde{k} = \mathrm{median}_x\, K(x) \tag{8}$$

and recall that by Eq. (6) the expected per-head energy gap satisfies

$$f_\ell(k) := \mathbb{E}_{h,x}\left[\Delta\mathcal{E}_k^{\ell,h}(x)\right] \le L_\ell e^{-\gamma k}, \tag{9}$$

and is L-Lipschitz with constant $L = L_\ell\gamma$.

Step 1: Decompose around the median. Using Lipschitz continuity, we have, for each input x,

$$|f_\ell(K(x)) - f_\ell(\tilde{k})| \le L\,|K(x) - \tilde{k}|. \tag{10}$$

Taking expectation over x and rearranging,

$$\mathbb{E}_x\left[f_\ell(K(x))\right] \le f_\ell(\tilde{k}) + L\,\mathbb{E}_x|K(x) - \tilde{k}|. \tag{11}$$

Step 2: Median absolute deviation. By definition,

$$\mathrm{LII}^\ell = \mathrm{MAD}[K] = \mathrm{median}_x\,|K(x) - \tilde{k}| \le \mathbb{E}_x|K(x) - \tilde{k}|, \tag{12}$$

so the second term in (11) is bounded by $L\,\mathrm{LII}^\ell$.

Step 3: Bounding the residual term. Applying Eq. (6) at $k = \tilde{k}$ gives

$$f_\ell(\tilde{k}) \le L_\ell e^{-\gamma\tilde{k}}. \tag{13}$$

Because $\tilde{k}$ is a median of token counts, it grows at least logarithmically with sequence length; hence $L_\ell e^{-\gamma\tilde{k}} = o(1)$ and can be absorbed into the $o(1)$ term of the lemma.

Step 4: Combine. Substituting these bounds into Eq. (11) yields

$$\boxed{\mathbb{E}_x\left[\Delta\mathcal{E}^\ell(x)\right] \le L\,\mathrm{LII}^\ell + o(1).} \tag{14}$$

Thus, the desired inequality follows immediately. □

B Information-geometric bound: LII^ℓ upper-bounds the fisher trace

We establish a theoretical bound showing that the Layer Instability Index (LII) upper-bounds the trace of the Fisher Information Matrix (FIM) for transformer layers. The FIM trace characterises the sensitivity of the loss to parameter updates, providing insights into learning dynamics. Through careful analysis of gradients via softmax logits and exponential tail bounds of attention probabilities, we derive a direct relationship between the Fisher trace and the dispersion of operational modes as measured by LII.

Preliminaries. Let $p_i = \mathrm{softmax}(\beta\, q^\top k_i)$ with inverse temperature β. Denote the per-example loss by $\mathcal{L} = \mathcal{L}(p)$ and define the layer-wise FIM

$$F^\ell \;=\; \mathbb{E}_x\Big[\nabla_{\theta_\ell}\mathcal{L}(x)\,\nabla_{\theta_\ell}\mathcal{L}(x)^\top\Big], \quad \theta_\ell \in \{W_Q^\ell, W_K^\ell, K^\ell\}. \tag{15}$$

Our goal is to bound $\operatorname{tr} F^\ell$.

Step 1: Gradient of the loss w.r.t. logits. For the logits $z_i := \beta q^\top k_i$,

$$\frac{\partial \mathcal{L}}{\partial z_i} \;=\; \sum_j \frac{\partial \mathcal{L}}{\partial p_j}\frac{\partial p_j}{\partial z_i} \;=\; \beta\big(p_i - \hat{y}_i\big), \tag{16}$$

where $\hat{y}_i$ is the "effective" target (one-hot for CE).

Step 2: Fisher trace through the chain rule. Let $J_z := \partial z / \partial \theta_\ell$. Then

$$\operatorname{tr} F^\ell = \mathbb{E}_x \big\|J_z^\top \nabla_z \mathcal{L}(x)\big\|_2^2 \;\le\; \frac{1}{\lambda_{\min}^2}\,\mathbb{E}_x \big\|\nabla_z \mathcal{L}(x)\big\|_2^2, \tag{17}$$

where $\lambda_{\min}$ is the smallest singular value of J_z (assumed layer-dependent but strictly positive for typical initialisation).

Using (16),

$$\big\|\nabla_z \mathcal{L}\big\|_2^2 = \beta^2 \sum_i (p_i - \hat{y}_i)^2 \;\le\; \beta^2 \sum_i p_i^2. \tag{18}$$

Step 3: Relate $\sum_i p_i^2$ to residual mass r_k. Let $k = \bar{k}^\ell(x)$ be the operational mode and $r_k := 1 - \sum_{i \le k} p_i \le 0.1$ (by definition of $\rho = 0.9$). Then

$$\sum_i p_i^2 = \sum_{i \le k} p_i^2 + \sum_{i > k} p_i^2 \;\le\; \sum_{i \le k} p_i + \max_{i > k} p_i\, r_k \;\le\; 0.9 + r_k^2. \tag{19}$$

Assuming an exponential tail $p_{i>k} \le p_k e^{-\gamma(i-k)}$, $r_k \le p_k/(e^\gamma - 1) \le C\, e^{-\gamma k}$.

Step 4: From $e^{-\gamma k}$ to LII$^\ell$. Taking expectation over inputs and using Jensen,

$$\mathbb{E}_x\big[e^{-\gamma k(x)}\big] \;\le\; e^{-\gamma\ \mathrm{median}(k)}\big(1 + \gamma\, \mathrm{LII}^\ell\big), \tag{20}$$

hence

$$\sum_i p_i^2 \;\le\; 0.9 + C'\, e^{-\gamma\ \mathrm{median}(k)}\big(1 + \gamma\, \mathrm{LII}^\ell\big). \tag{21}$$

Step 5: Final bound. Substituting into (17),

$$\operatorname{tr} F^\ell \;\le\; \frac{\beta^2(0.9 + C')}{\lambda_{\min}^2}\big(1 + \gamma\, \mathrm{LII}^\ell\big) \;=\; C_\ell\, \mathrm{LII}^\ell + C_{0,\ell}, \tag{22}$$

where C_ℓ and $C_{0,\ell}$ are layer-dependent constants. For practical purposes $C_{0,\ell}$ is negligible once LII exceeds 10^{-2}, yielding

$$\boxed{\operatorname{tr} F^\ell \;\lesssim\; C_\ell\, \mathrm{LII}^\ell} \tag{23}$$

as claimed.

Connection to the 1-Wasserstein distance. Sort the attention vector of layer ℓ and head h at step t, $a^{\downarrow}_{1:N}(t)$, and define its empirical cumulative distribution function (CDF) $F_t(m) = \sum_{i=1}^m a_i^{\downarrow}(t)$. Because tokens are indexed by their rank, the earth-mover (1-Wasserstein) distance between two attention snapshots is simply

$$W_1\big(a(t), a(t')\big) \;=\; \sum_{m=1}^N \big|F_t(m) - F_{t'}(m)\big|. \tag{24}$$

Let $\rho = 0.9$ and let k_t be the minimal m such that $F_t(m) \geq \rho$ (operational mode). Then any deviation $|k_t - k_{t'}|$ shifts *at least* a residual mass $r = |F_t(k_{t'}) - \rho| \leq 1 - \rho = 0.1$ across $|k_t - k_{t'}|$ token positions, so

$$W_1\big(a(t), a(t')\big) \;\leq\; r\,|k_t - k_{t'}| \;\leq\; 0.1\,|k_t - k_{t'}|. \tag{25}$$

Taking the median over t' in the sliding window and then the median over t gives

$$\boxed{W_1^{\text{med}}(\ell) \;\leq\; 0.1\,\text{LII}^{\ell}} \tag{A.1}$$

where $W_1^{\text{med}}(\ell)$ is the median Wasserstein distance between successive attention snapshots of layer ℓ. Equation (A.1) shows that **LII controls the earth-mover distance between attention distributions**: A low LII implies the layer's attention landscape hardly moves in Wasserstein space and is therefore safe to freeze. Via the Kantorovich–Rubinstein dual, the same bound controls the difference of *all* 1-Lipschitz observables of the attention measure, linking the energy and Fisher-flatness views in a common optimal- transport metric.

C Energy landscape–aware fine-tuning

Algorithm 1 details the complete training routine used in all experiments. After a short warm-up that estimates the Layer Instability Index (LII) for every block, layers whose instability falls below a user-defined threshold τ_{freeze} are frozen (`requires_grad=False`). Fine-tuning then proceeds on the remaining adaptive layers, incurring no further LII overhead.

Algorithm 1: Energy Landscape–Aware ViT Fine-Tuning

```
Input: pre-trained weights $\Theta^{(0)}$; dataset $\mathcal{D}$; freeze threshold $\tau_{\text{freeze}}$; warm-up steps $T$; LII window $W$
Warm-up phase: ;                                          // estimate layer instability
for $t = 0$ to $T-1$ do                                   // collect $\bar{k}$ statistics
    sample mini-batch $(x, y) \sim \mathcal{D}$;
    forward and backward pass; update $\Theta^{(t+1)}$ with AdamW;
    update the circular buffer of size $W$ and compute $\widehat{\text{LII}}^{\ell}$ for all layers $\ell$;
Freeze decision: ;                                        // one-shot pruning of stable layers
foreach layer $\ell$ do
    if $\widehat{\text{LII}}^{\ell} < \tau_{freeze}$ then
        freeze($\ell$)                                    // disable gradient updates
Consolidation phase: ;                                    // train only adaptive layers
for $t = T$ to max_steps do                               // until convergence
    mini-batch $(x, y) \sim \mathcal{D}$;    forward + backward pass on unfrozen layers only;
```

The algorithm runs in three stages: *(i) Warm-up* gathers a robust estimate of each layer's variability via the median absolute deviation of its operational mode $\bar{k}$. *(ii) Freeze decision* is executed once, turning off gradient flow for layers whose LII indicates convergence to a low-energy basin. *(iii) Consolidation* fine-tunes the remaining layers, yielding substantial savings in memory and computation with no extra learnable parameters.

D Online update of the LII circular buffer

During warm-up we compute $\widehat{\text{LII}}_t^{\ell}$ for every layer on the fly. Algorithm 2 shows an eight-line Python reference implementation; it relies only on a layer-indexed `deque` of fixed capacity W (the sliding window size, default $W = 20$).

At each mini-batch, we compute the layer's operational mode $\bar{k}_t^{\ell}$ (Sec. 3.2) and call `update_lii`. The deque acts as a circular buffer: the newest value is appended, the oldest is popped when the buffer overflows, and both operations are $\mathcal{O}(1)$. We then take the median of the window, followed by the median absolute deviation—exactly Eq. (2) but restricted to the latest W steps. The result is the online estimate $\widehat{\text{LII}}_t^{\ell}$ used in Alg. 1.

Algorithm 2: Update of layer instability index (LII) buffer

```
Input: layer ID layer_id; new value $\bar{k}$; buffer buf; window size $W$
Output: current $\mathrm{LII}_t^\ell$
buf[layer_id].append($\bar{k}$) ;                    // 1.  push newest value
if len(buf[layer_id]) > W then
    buf[layer_id].popleft() ;                        // 2.  drop oldest (FIFO)
med = median(buf[layer_id]) ;                        // 3.  running median
abs_dev = [abs(x - med) for x in buf[layer_id]];
return median(abs_dev) ;                             // 4.  current $\mathrm{LII}_t^\ell$
Note: One dictionary, one deque per layer (maxlen = $W$).
Complexity: $\mathcal{O}(1)$ per call, $\mathcal{O}(W)$ memory per layer.
```

Cold-start. For $t < W$ the deque contains fewer than W elements; the function still returns a valid LII based on the available prefix, ensuring that no additional initialisation logic is required.

Efficiency. The routine consumes negligible resources: $\mathcal{O}(L\,W)$ memory for L layers and $\mathcal{O}(1)$ extra time per iteration, contributing less than 1% overhead in all experiments (see App. B).

E Code Availability

The source code supporting this study is available at `https://github.com/rxailab/ELA-ViT`.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Can be found in the section 1.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Can be found in the section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Can be found in the section 3.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Can be found in the section 4.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Additional details and code repository URL are included in the supplemental material.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Can be found in the section 4.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [NA]

   Justification: Given that our experiments focus on training and fine-tuning Vision Transformer models to demonstrate improvements in accuracy and efficiency rather than testing hypotheses, statistical significance through error bars or formal significance testing is not directly applicable. Instead, we clearly report model performance metrics, including accuracy and computational improvements, consistently across multiple datasets and settings. These metrics reliably indicate performance differences without requiring traditional statistical significance testing.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Can be found in section 4

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: I confirm the research conform with the Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [NA]

    Justification: No potential societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: The research is not a high risk research.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets and models used are explicitly mentioned and included in the reference.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: Not introduced new assets.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: Not require human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: Not require human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: No LLM used for original, or non-standard component of the core methods.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.