RAJNI: IMPORTANCE-AWARE, DISPERSION-SCHEDULED TRAINING-FREE TOKEN PRUNING FOR VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Transformers incur inference cost that scales quadratically in the number of patch tokens, making token pruning an attractive training-free accelerator. We present RAJNI, built on two components: a forward-pass importance score that is exactly a token's contribution to its layer's CLS aggregate under ablation, and that we show upper-bounds (though does not guarantee) the token's effect on the model's output, under an alignment condition we validate empirically; and a closed-form scheduler that sets each layer's keep-ratio from the dispersion of that layer's own importance distribution, requiring no learned parameters, auxiliary networks, or fine-tuning. We validate the method with three kinds of evidence rather than a single accuracy table: agreement with exact gradient sensitivity across depth (up to $\rho = 0.98$ Spearman correlation), a controlled ablation showing the score's two factors are individually insufficient, and a large-scale (n=50,000) analysis showing the scheduler's adaptivity compounds systematically with depth. On ImageNet-1K, RAJNI reaches an approximate throughput gain of 40% at an accuracy cost of 0.7% on ViT-Base, remaining close to the Pareto front of stronger training-free baselines (ToMe, SAINT) in the moderate-compression regime, and consistently ahead of GTP-ViT. Main-text throughput numbers use batch-aggregated dispersion statistics for GPU efficiency; we state this explicitly, since per-image dynamic sequence lengths would require custom sparse-attention kernels in a batched production deployment.

1 INTRODUCTION

Self-attention's cost (Vaswani et al., 2017) grows quadratically with the number of patch tokens a Vision Transformer processes, and this has made token pruning a popular deployment-time accelerator (Dosovitskiy et al., 2021; Rao et al., 2021; Liang et al., 2022), particularly the training-free variants that need no backward pass and no fine-tuning stage to deploy. Building one well means answering two open design questions: how to score token importance without a backward pass, and how much to prune at each layer. Most existing methods answer the second question with a single, hand-tuned keep-ratio applied uniformly across depth, which implicitly assumes token redundancy is constant across layers and across inputs, an assumption we find no strong basis for. Figure 3 (Section 5.2) illustrates the resulting behavior qualitatively: surviving tokens concentrate on foreground structure as depth increases.

The intuition behind our design is simple. A photograph of a single bird against an empty sky carries most of its useful signal in a handful of patches, while a cluttered street scene needs many more tokens to represent reliably; an early layer, where features are still diffuse, is a different pruning decision than a late layer, where attention has already concentrated on the relevant region. A single global keep-ratio cannot tell any of these cases apart, and doing so would normally require either a backward pass or a trained predictor, both of which existing training-free methods try to avoid. RAJNI reads this structure directly from quantities the forward pass already computes: an importance score for each token that is, by construction, exactly that token's contribution to its layer's CLS aggregate, and a closed-form scheduler that sets each layer's keep-ratio from how concentrated that

Anonymized code and configuration files: `https://anonymous.4open.science/r/RAJNI-ViT`

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

layer's own importance distribution already is. Neither piece needs learned parameters, an auxiliary network, or fine-tuning, and we show the same importance score also upper-bounds, though does not guarantee, the token's effect on the model's final output, under a condition we test empirically rather than assume.

Rather than lean on a single accuracy table, we check this design against exact gradient sensitivity layer by layer, ablate the two halves of the score independently, and track the scheduler's behavior over the full n=50,000 ImageNet-1K validation set to confirm its adaptivity is content-driven rather than an artifact of a handful of examples.

Our contributions are:

- A forward-pass importance score with an exact interpretation as layer-local contribution (Proposition 1) and a provable, condition-dependent bound on output-level importance (Proposition 2), validated empirically where the analysis alone cannot close the gap.
- A dispersion-based adaptive scheduler requiring no learned parameters, shown via ablation to improve over an identical score paired with a fixed schedule.
- A large-scale ($n = 50{,}000$) validation that the scheduler's adaptivity is genuinely content-driven and compounds with depth, not an artifact of a handful of examples.

2 Related Work

Attention-based and learned-predictor pruning. EViT (Liang et al., 2022) and A-ViT (Yin et al., 2022) discard patches using CLS-attention weights directly, a signal whose relationship to causal output importance is itself debated in the interpretability literature (Jain & Wallace, 2019); Propositions 1–2 (Section 3) formalize this concern, showing attention alone is not the exact contribution term and that even the full product is only a bound on output effect. DynamicViT (Rao et al., 2021), SPOT (Schlesinger et al., 2025), and SPViT (Kong et al., 2022) instead train an auxiliary module or predictor to make this decision; Evo-ViT (Xu et al., 2022) and Zero-TPrune (Wang et al., 2024) remain training-free but rely on the attention signal alone. RAJNI's score is read from quantities the transformer already computes, requiring no such training.

Similarity-based reduction. ToMe (Bolya et al., 2023) merges tokens judged similar rather than discarding them; SAINT (Jeddi et al., 2025) instead prunes tokens outright, using a bipartite similarity graph with batch-level voting to set pruning rates and redundancy thresholds per layer; GTP-ViT (Xu et al., 2024) propagates information from pruned tokens to their remaining neighbors rather than dropping it outright. Token similarity is a different signal from output sensitivity: near-duplicate tokens can both be uninformative, or dissimilar tokens both irrelevant. Section 5 shows RAJNI and these methods trade the Pareto lead depending on backbone and compression level rather than one uniformly dominating.

Adaptive sampling. ATS (Fayyaz et al., 2022) varies the retained token count per layer via inverse-transform sampling over cumulative attention distributions, making it the closest prior method in spirit to RAJNI's scheduler, though its sampling is calibrated jointly with fine-tuning rather than computed in closed form. Section 6 looks at what happens when that calibration step is removed. DVT (Wang et al., 2021) instead adapts token count per image, a coarser granularity than RAJNI's per-layer schedule.

3 Method

RAJNI inserts two sequential operations into the vision transformer block: a forward-pass proxy for token sensitivity, and an adaptive dispersion-based scheduler that dynamically sets sequence retention. Figure 1 illustrates the architectural workflow within a transformer block.

3.1 Token Contribution and the CLS Aggregate

3.1.1 Setup

Let $x_t^{(l)}$ denote the t-th patch token representation entering layer l, and let y denote the output logit corresponding to the target class. The ideal sensitivity score for ranking token importance is the

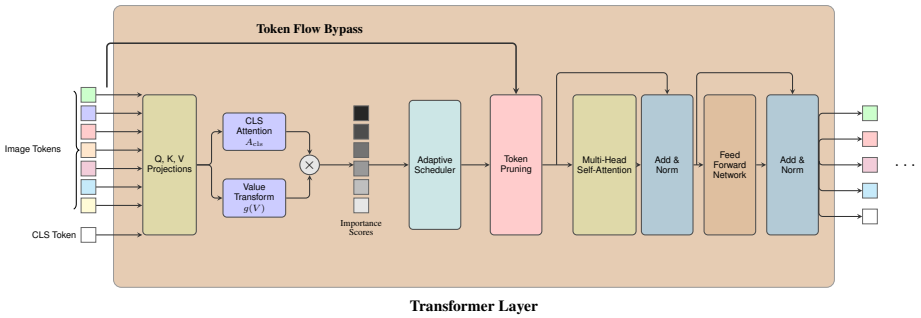

Figure 1: Integration of RAJNI into a transformer layer. Standard Q, K, V projections yield CLS attention weights and value magnitudes. These are combined into per-token importance scores, from which the adaptive scheduler determines layer-specific keep-ratios prior to self-attention. The CLS token bypasses pruning.

gradient norm:

$$s_t^{(l)} = \left\| \frac{\partial y}{\partial x_t^{(l)}} \right\|. \tag{1}$$

Evaluating Eq. (1) requires backward passes for each candidate pruning layer, making it unfeasible for real-time inference acceleration. We instead inspect how the CLS token, which feeds the final classifier head, aggregates information from spatial patch tokens at layer l:

$$p^{(l)} = \sum_{i=1}^{N} A_{\text{cls},i}^{(l)} V_i^{(l)}, \tag{2}$$

where $A_{\text{cls},i}^{(l)}$ is the attention weight from the CLS token to patch token i, and $V_i^{(l)}$ is the associated value vector.

3.1.2 Ablation, Not Perturbation

Token pruning removes a token entirely from the sequence; it is not an infinitesimal perturbation of that token's representation. The natural object to analyze is therefore the effect of *ablating* token t, comparing $p^{(l)}$ against $p^{(l)}$ recomputed with term t dropped from the sum, rather than a derivative of $p^{(l)}$ with respect to $x_t^{(l)}$. We write

$$p_{-t}^{(l)} \triangleq \sum_{i \neq t} A_{\text{cls},i}^{(l)} V_i^{(l)} \tag{3}$$

for this *soft* ablation: term t is zeroed but the remaining attention weights are left unnormalized (Section 3.1.5 returns to the gap between this and our hard-removal implementation).

3.1.3 Exact Layer-Local Contribution

Proposition 1 (Exact layer-local contribution)**.**

$$p^{(l)} - p_{-t}^{(l)} \;=\; A_{\text{cls},t}^{(l)}\, V_t^{(l)}.$$

Proof. Immediate from Eq. (2) and Eq. (3): the two sums differ by exactly the $i = t$ term. □

Taking norms, since $A_{\text{cls},t}^{(l)} \geq 0$ is a scalar,

$$\left\| p^{(l)} - p_{-t}^{(l)} \right\| \;=\; A_{\text{cls},t}^{(l)} \left\| V_t^{(l)} \right\|. \tag{4}$$

This is the load-bearing claim of this section: $A_{\text{cls},t}^{(l)} \|V_t^{(l)}\|$ is exactly the magnitude of token t's contribution to the layer's CLS aggregate: this is what our importance score measures, by construction, with no approximation.

3.1.4 BOUND ON OUTPUT-LEVEL IMPORTANCE

A natural further question is whether contribution to $p^{(l)}$ (Proposition 1) also bounds contribution to the eventual output y. The following gives a first-order bound, not an equality. Write $y(p^{(l)})$ for the output logit as a function of the CLS aggregate at layer l (rest of the sequence and downstream computation held fixed), with $\nabla y \triangleq \nabla_p y(p^{(l)})$ its gradient there.

Proposition 2 (First-order output bound)**.**

$$\begin{aligned}\left|y(p^{(l)}) - y(p^{(l)}_{-t})\right| &\approx \left|(\nabla y)^\top \left(p^{(l)} - p^{(l)}_{-t}\right)\right| \\ &= A^{(l)}_{\text{cls},t} \left|(\nabla y)^\top V^{(l)}_t\right| \\ &\le A^{(l)}_{\text{cls},t} \left\|\nabla y\right\| \left\|V^{(l)}_t\right\|. \end{aligned} \tag{5}$$

Equality holds in the last step if and only if $V^{(l)}_t$ is parallel to ∇y.

Proof. First-order Taylor approximation of y around $p^{(l)}$; the equality substitutes Proposition 1; the inequality is Cauchy–Schwarz on $(\nabla y)^\top V^{(l)}_t$. □

Since $\|\nabla y\|$ is constant across tokens within a layer, ranking tokens by $A^{(l)}_{\text{cls},t}\|\nabla y\|\|V^{(l)}_t\|$ is equivalent to ranking by $A^{(l)}_{\text{cls},t}\|V^{(l)}_t\|$. This ranking is exact for the bound; it coincides with the ranking of the true output effect $A^{(l)}_{\text{cls},t}|(\nabla y)^\top V^{(l)}_t|$ only to the extent that tokens' alignment with ∇y does not vary systematically across tokens, an assumption we cannot verify without a backward pass and do not claim to prove. This motivates our forward-pass proxy:

$$\text{Importance}^{(l)}_t = A^{(l)}_{\text{cls},t} \cdot g\left(V^{(l)}_t\right), \tag{6}$$

where $g(\cdot)$ (Section 3.2) normalizes $\|V^{(l)}_t\|$ across attention heads and sequence length for numerical stability; it does not alter the ranking within a layer.

3.1.5 WHAT THE BOUND DOES NOT RESOLVE

Propositions 1 and 2 together establish that $A^{(l)}_{\text{cls},t}\|V^{(l)}_t\|$ is an exact measure of layer-local contribution and a valid, but not necessarily tight, bound on output-level importance. Whether it tracks true output sensitivity in practice is an empirical question, which we answer directly in Section 5.3 by comparing our forward-pass score against exact gradient sensitivity $\|\partial y/\partial x_t\|$ (Eq. 1), computed via backpropagation, on a held-out subset of the validation set. We find agreement improves monotonically with depth (Spearman ρ from 0.73 at layer 1 to 0.983 at layer 11), which we interpret as evidence that the alignment condition underlying Proposition 2's bound holds increasingly well in later layers, precisely where the scheduler (Section 3.3) also chooses to prune most aggressively. Separately, Proposition 1 is exact only for the soft ablation of Eq. (3); our implementation performs hard removal with softmax renormalization, introducing a second-order correction that vanishes as $A^{(l)}_{\text{cls},t} \to 0$, the regime in which a token is a plausible pruning candidate in the first place.

3.2 VALUE TRANSFORMATION

Raw value vector magnitudes exhibit variance across different attention heads and layers. To normalize these values for consistent score computation, we average across H attention heads, subtract the sequence-wide mean to isolate token-specific deviations, and map the normalized result through a sigmoid activation gate:

$$V^{(l)}_{\text{agg}} = \frac{1}{H}\sum_{h=1}^{H} V^{(h,l)}, \tag{7}$$

$$V^{(l)}_{\text{norm},t} = \text{mean}\left(\left|V^{(l)}_{\text{agg},t} - \mu^{(l)}_{\text{seq}}\right|\right), \tag{8}$$

$$G^{(l)}_t = \sigma\left(\frac{V^{(l)}_{\text{norm},t} - \mu^{(l)}_V}{\sigma^{(l)}_V + \epsilon}\right). \tag{9}$$

The final score is expressed as $\text{Importance}_t^{(l)} = A_{\text{cls},t}^{(l)} \cdot G_t^{(l)}$.

3.3 Adaptive Token Scheduling

Applying a fixed keep-ratio assumes that importance concentration remains static across transformer depth and input content. When an image contains a prominent foreground object against a uniform background, token importance accumulates on a small fraction of patches, enabling aggressive token removal. Conversely, when image content is visually complex or contextually distributed, importance scores remain spread across many tokens, requiring conservative retention to avoid dropping necessary spatial features.

To assess this spatial distribution during inference, we project importance scores into the log domain, $s_t^{(l)} = \log(\text{Importance}_t^{(l)} + \epsilon)$, and identify the sequence 75th percentile, $q_l = \text{percentile}_{75}(s^{(l)})$. For tokens falling below this threshold, we compute distance $d_t^{(l)} = \max(q_l - s_t^{(l)}, 0)$ and summarize layer dispersion as:

$$D_l = \frac{\text{mean}_t\left(d_t^{(l)}\right)}{|q_l|}. \tag{10}$$

A high dispersion value D_l signals strong contrast between foreground and background tokens, indicating that lower keep-ratios can be safely applied. We compute the layer keep-ratio using an exponential function bounded by a minimum retention threshold $k_{\min}$:

$$\text{KeepRatio}_l = \text{clamp}(\exp(-\gamma D_l),\, k_{\min},\, 1.0)\,, \tag{11}$$

where γ controls global pruning intensity. If $D_l < \tau_{\text{warmup}}$, pruning is bypassed for layer l to ensure feature representation stabilizes before token removal begins. The CLS token bypasses pruning across all layers. Algorithm 1 summarizes the full procedure.

Algorithm 1 RAJNI Layer-Wise Pruning

Require: Tokens $X^{(l)} \in \mathbb{R}^{N \times d}$, $\gamma, k_{\min}, \tau_{\text{warmup}}$
1: $Q, K, V \leftarrow$ LinearProjections$(X^{(l)})$
2: $\text{Importance}^{(l)} \leftarrow A_{\text{cls}}^{(l)} \odot g(V^{(l)})$
3: $s^{(l)} \leftarrow \log(\text{Importance}^{(l)} + \epsilon)$
4: $q_l \leftarrow \text{percentile}_{75}(s^{(l)})$
5: $D_l \leftarrow \text{mean}(\max(q_l - s^{(l)}, 0))/|q_l|$
6: **if** $D_l \leq \tau_{\text{warmup}}$ **then**
7: **return** $\text{MHSA}(X^{(l)})$ {Skip Pruning}
8: **end if**
9: $k \leftarrow \lceil \max(\exp(-\gamma D_l), k_{\min}) \cdot N \rceil$
10: $X_{\text{pruned}}^{(l)} \leftarrow$ Top-$k(X^{(l)}$ based on $s^{(l)})$
11: **return** $\text{MHSA}(X_{\text{pruned}}^{(l)})$

4 Experimental Setup

- **Datasets.** ImageNet-1K validation set, full $50{,}000$ images, 224×224 resolution, standard center-crop preprocessing.
- **Backbones.** ViT-B/16 (Dosovitskiy et al., 2021) and DeiT-S/16 (Touvron et al., 2021), pretrained `timm` checkpoints, weights unmodified. RAJNI operates strictly post-hoc, with no fine-tuning.
- **Baselines.** ToMe (Bolya et al., 2023), SAINT (Jeddi et al., 2025), and GTP-ViT (Xu et al., 2024), run from each method's official open-source codebase.
- **Hardware.** Single NVIDIA Tesla P100 GPU (16GB), FP32 precision.
- **Protocol.** All methods, including the unpruned baseline, are benchmarked in the same runtime session and batch size (32 for main-text throughput numbers, 256 for the full accuracy–speedup sweeps underlying Tables 1–2, released in the supplementary material) to control for session-to-session variance; speedups are relative to the same-session baseline.
- **RAJNI hyperparameters.** γ, τ_{warmup}, and $k_{\min}$ (Eq. 11) are selected once on a held-out calibration split and held fixed thereafter; exact values are in the Reproducibility Statement.

- **Image-complexity metrics** (Section 5.6). Shannon entropy of pixel intensities, Canny edge density, and Laplacian variance, computed per-image on the raw RGB input prior to normalization.

5 RESULTS

5.1 ACCURACY–THROUGHPUT PARETO COMPARISON

RAJNI sits at or above every competing method's curve on ViT-Base through roughly $1.35\times$, and remains close to the front on DeiT-S across the same range; GTP-ViT trails throughout on both backbones (Figure 2). Beyond approximately $1.4\times$, ToMe and SAINT begin to pull ahead on both backbones; we report this rather than crop the figure to hide it. Read together with Table 1 and Table 2, this marks out RAJNI's useful range: pruning recovers throughput cheaply while token redundancy, not model capacity, is the binding constraint, which is the moderate-compression regime reported here. Past that point, a smaller backbone is typically a more efficient way to buy further throughput than pushing pruning harder on a larger one, so we see this less as a limitation than as evidence that RAJNI's operating range is exactly where pruning is the right lever to pull. Both tables remain competitive with methods that require training or fine-tuning (e.g. DynamicViT and EViT, cited for reference in Appendix A.4) without paying their training cost.

Method	Speedup	Top-1	ΔAcc
Baseline	1.00×	81.10	–
RAJNI	**1.07**×	**81.11**	+0.01
SAINT	1.04×	80.96	−0.14
ToMe	1.05×	80.95	−0.15
RAJNI	**1.24**×	**80.87**	−0.23
ToMe	1.23×	80.51	−0.59
SAINT	1.32×	80.40	−0.70
GTP-ViT	1.14×	80.19	−0.91
RAJNI	**1.39**×	**80.37**	−0.73
ToMe	1.48×	79.70	−1.40

Table 1: ViT-Base/16.

Method	Speedup	Top-1	ΔAcc
Baseline	1.00×	79.73	–
RAJNI	**1.07**×	**79.64**	−0.08
ToMe	0.99×	79.60	−0.20
SAINT	1.00×	79.52	−0.28
RAJNI	**1.36**×	**78.75**	−0.97
ToMe	1.39×	78.92	−0.88
SAINT	1.27×	78.78	−1.02
GTP-ViT	1.32×	78.41	−1.39

Table 2: DeiT-S/16.

Representative operating points on ImageNet-1K (Top-1 and ΔAcc in %, pp). Full operating-point sweeps for all methods are released in the supplementary material.

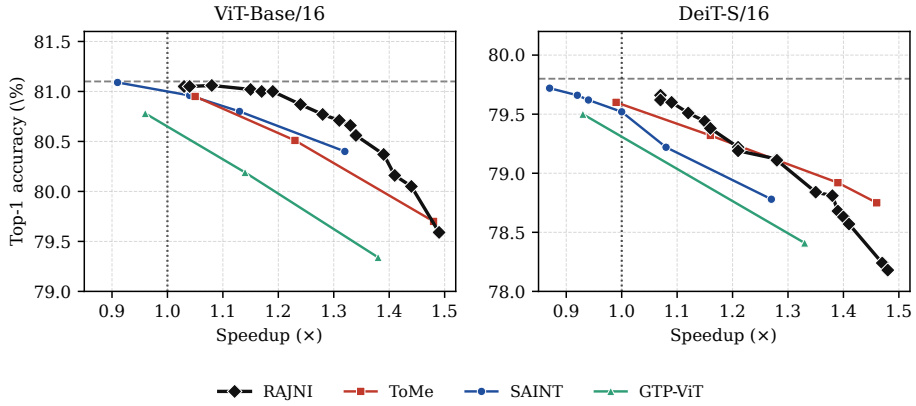

Figure 2: Accuracy versus throughput on ImageNet-1K, ViT-Base/16 (left) and DeiT-S/16 (right). Dashed lines mark unpruned baseline performance. Full numeric operating points underlying this figure are released in the supplementary material.

5.2 QUALITATIVE ANALYSIS

Figure 3 shows this qualitatively: by layer 11, RAJNI's surviving tokens concentrate on each image's foreground subject while background and out-of-focus patches are discarded first, matching what Section 3 predicts. This is a sanity check on that prediction, not evidence on its own; the score is validated statistically in Section 5.3, not image by image.

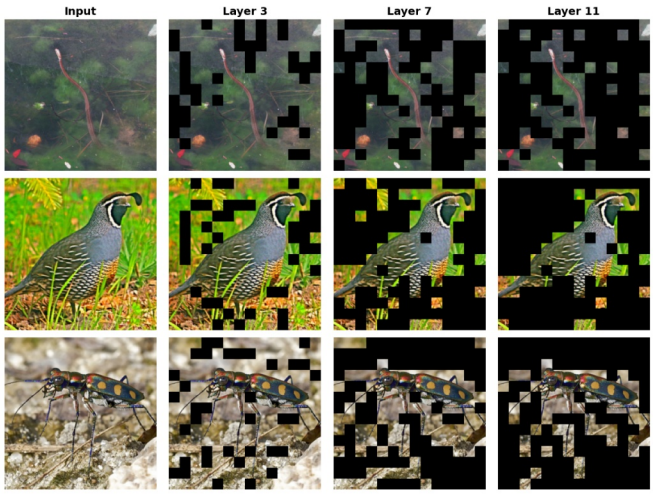

Figure 3: Token retention across transformer layers under RAJNI. As layer depth increases, surviving tokens concentrate progressively on semantically relevant foreground structures while redundant background patches are discarded.

5.3 Correlation with Exact Gradient Sensitivity

Proposition 2's bound is tight only under an alignment condition ($V_t^{(l)} \parallel \nabla y$) unverifiable without a backward pass, so we test it directly. Agreement between the forward-pass score and exact gradient sensitivity rises monotonically with depth, reaching $\rho = 0.983$ by layer 11 (Figure 4), and it is strongest in exactly the layers where the scheduler's dispersion statistic D_l (Eq. 10) is largest. In other words, RAJNI concentrates its most aggressive pruning precisely where the underlying signal is most trustworthy, rather than pruning uniformly regardless of how much confidence the score actually warrants at that depth.

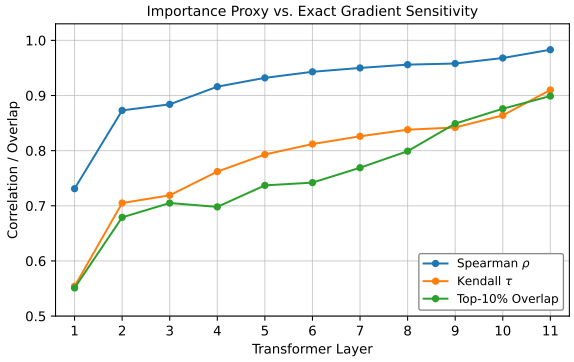

Figure 4: Layer-wise correlation between RAJNI's forward-pass importance proxy and exact gradient sensitivity (Eq. 1).

5.4 Importance Score Component Ablation

Neither factor in Eq. (6) alone matches the combined score on either backbone (Figure 5): magnitude alone is consistently the weaker signal, and combining both closes most of the gap either factor leaves open. This confirms Proposition 1 directly: it is the product form, not either factor in isolation, that tracks layer-local contribution. We use DeiT-Base rather than DeiT-S/16 here since the larger backbone widens the gap between variants; only the qualitative conclusion, not the exact percentage-point gaps, should be assumed to transfer across backbones.

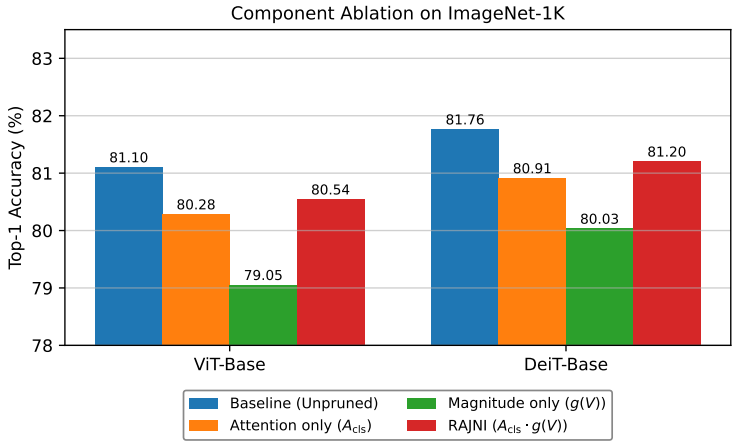

Figure 5: Importance score component ablation on ImageNet-1K, both backbones, at matched throughput ($\sim$ 240 img/s).

5.5 Scheduler Ablation

Scheduler	Top-1 (%)	ΔAcc (pp)	Speedup
Fixed heuristic schedule	80.27	−0.83	1.25×
RAJNI adaptive (ours)	**80.87**	−0.23	**1.24**×

Table 3: Adaptive versus fixed-ratio scheduling on ViT-Base, identical importance score (Eq. 6) in both rows.

At matched throughput, the adaptive scheduler outperforms an identical importance score paired with a hand-tuned fixed per-layer schedule (Table 3), isolating the scheduler's own contribution from the score it operates on: content-driven, per-layer budgets beat a single hand-tuned schedule even when both start from the same underlying signal. This comparison is run at a different operating point than Figure 5 (matched relative speedup here, rather than a fixed throughput target), which is why RAJNI's own accuracy differs slightly between the two.

5.6 Validating Scheduler Adaptivity at Scale

We use three complexity proxies, each capturing a different notion of visual complexity: Shannon entropy of pixel intensities, a global measure of how much information the image's tonal distribution carries; Canny edge density, the fraction of pixels flagged as object or texture boundaries; and Laplacian variance, a local measure of high-frequency detail and focus. Across the full $n = 50{,}000$ ImageNet-1K validation set, token retention correlates most strongly with global Shannon entropy and only weakly with the two local texture proxies (full table in Appendix A.1), consistent with D_l responding to how importance is distributed across the whole sequence rather than to local high-frequency detail.

Stratifying into complexity quartiles (Figure 6) makes the practical consequence of this concrete: images in the simplest quartile (Q1) are pruned the most aggressively, ending with the fewest retained tokens by the final layer, while images in the most complex quartile (Q4) are pruned the most conservatively and retain the most tokens, with the two middle quartiles falling in between in the same order. All four quartiles start identically through the warmup phase (layers 0–2) and separate monotonically afterward under all three metrics, so this is not an aggregate artifact averaged over many images: the scheduler is measurably more conservative on complex images and measurably more aggressive on simple ones, which is exactly the behavior the dispersion statistic D_l (Section 3.3) was designed to produce, now confirmed at the scale of the full validation set rather than a handful of examples.

As a check against silently trading accuracy for adaptivity, keep-ratios for correctly and incorrectly classified samples are statistically indistinguishable (median 0.39 vs. 0.40): the scheduler is not systematically over-pruning the images it gets wrong.

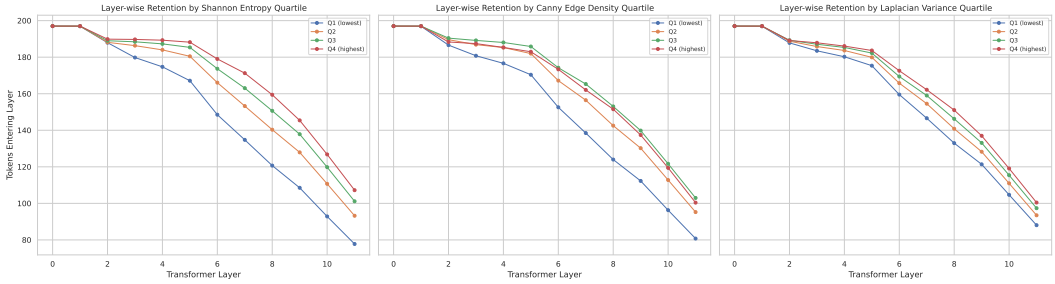

Figure 6: Mean sequence length entering each transformer layer, grouped by quartiles (Q1 = simplest 25%, Q4 = most complex 25%) of three complexity metrics: Shannon entropy (left), Canny edge density (middle), Laplacian variance (right); $n = 50{,}000$. Retention rates diverge systematically beyond layer 2 under all three metrics.

6 ATS Without Its Fine-Tuning Stage

ATS (Fayyaz et al., 2022) is the closest prior method in spirit to RAJNI's scheduler, since it too varies retained-token count per layer at inference time, though its sampling threshold is calibrated jointly with fine-tuning rather than computed in closed form. Running it under our protocol (Section 4) without that fine-tuning stage gives a concrete sense of what the calibration is doing: accuracy drops to 49.63% (from a 79.8% DeiT-S baseline) at a throughput of $0.16\times$, below the unpruned baseline. We read this as evidence that ATS's inverse-transform-sampling mechanism leans on calibration learned jointly during fine-tuning: without it, stochastic token dropping degrades downstream feature representations, and the cumulative-distribution computation itself adds overhead that the reduced token count does not recoup. RAJNI's scheduler has no equivalent calibration step to lose, since $\mathrm{KeepRatio}_l$ (Eq. 11) is a deterministic closed-form function of quantities already computed in the forward pass. We did not attempt to reproduce ATS's fine-tuned performance; the comparison illustrates what a closed-form, training-free scheduler avoids, not a claim that ATS is weak in its intended, fine-tuned setting.

7 Limitations

- **Bound, not guarantee, on output-level importance.** The score is exact as layer-local contribution (Proposition 1) but only an upper bound on output-level importance (Proposition 2), tight only under an unverifiable alignment condition; the correlation study (Section 5.3) is indirect evidence, not a closed-form guarantee.
- **Batch-aggregated deployment.** Main-text throughput uses batch-aggregated dispersion statistics for GPU efficiency; variable per-image sequence lengths would require custom sparse-attention kernels to realize the same speedups in production.

Reproducibility Statement

All hyperparameters (γ, τ_{warmup}, $k_{\min}$; Eq. 11) and the calibration-split selection procedure are described in Section 4. Pretrained backbone weights are standard `timm` checkpoints, used unmodified with no fine-tuning. Code and configuration files to reproduce our tables and figures are at the anonymized repository linked on the title page (`https://anonymous.4open.science/r/RAJNI-ViT`). All throughput numbers are measured on a single NVIDIA Tesla P100 GPU at the batch size stated in Section 4, with each method and its baseline benchmarked in the same runtime session to control for session-to-session variance (an explicit caveat in Section 7). The importance-score derivation (Section 3) combines an exact layer-local claim (Proposition 1) with a separately validated output-level bound (Proposition 2); Appendix A.2 discloses which comparison numbers are self-measured versus cited from prior literature.

Ethics Statement

This work uses only the public ImageNet-1K benchmark and publicly released pretrained checkpoints; it involves no human subjects, no new data collection, and no personally identifiable information. As an inference-time efficiency method, RAJNI's primary societal effect is a reduction in

the compute and energy cost of deploying existing Vision Transformer models, without altering their underlying training data or the biases already present in those pretrained weights.

AI USE STATEMENT

In this work, we used generative AI tools (Claude) to assist with the paper's design and structure, drafting and editing manuscript prose, formatting the draft into ICLR style, and verifying internal consistency of cross-references, citations, and notation; to explore and summarize related literature; and to audit and implement supplementary evaluation and baseline code from author-provided specifications. This assistance was structural and execution-level: the underlying mathematical claims, proofs, hypotheses, research methodology, and experimental design are the authors' own and were not formulated or proposed by generative AI. We did not use generative AI tools to generate synthetic data or to run the experiments or hyperparameter sweeps underlying our reported numbers, all of which were produced by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI; all technical claims, mathematical derivations, and reported numbers have been reviewed and verified by the authors.

REFERENCES

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *International Conference on Learning Representations*, 2023.

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

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European Conference on Computer Vision*, 2022.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2019.

Ahmadreza Jeddi, Negin Baghbanzadeh, Elham Dolatabadi, and Babak Taati. Similarity-aware token pruning: Your vlm but faster. *arXiv preprint arXiv:2503.11549*, 2025.

Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, Minghai Qin, and Yanzhi Wang. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European Conference on Computer Vision*, 2022.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, 2021.

Oded Schlesinger, Amirhossein Farzam, J. Matias Di Martino, and Guillermo Sapiro. Spot: Sparsification with attention dynamics via token relevance in vision transformers. *arXiv preprint arXiv:2511.10488*, 2025.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Hongjie Wang, Bhishma Dhedhia, and Niraj K. Jha. Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic vision transformers with adaptive sequence length. *arXiv preprint arXiv:2105.15075*, 2021.

Xuwei Xu, Sen Wang, Yudong Chen, Yanping Zheng, Zhewei Wei, and Jiajun Liu. Gtp-vit: Efficient vision transformers via graph-based token propagation. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024.

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

A Appendix

A.1 Image-Complexity Correlation Table

Complexity Metric	**Pearson** r	**Spearman** ρ	R^2
Shannon Entropy	0.411 [0.403, 0.418]	0.429	0.169
Canny Edge Density	0.210 [0.202, 0.218]	0.268	0.044
Laplacian Variance	0.117 [0.108, 0.126]	0.159	0.014

Table 4: Correlation between image complexity metrics and token retention ($n = 50{,}000$), underlying Figure 6 and Section 5.6.

A.2 Sourcing Disclosure

All numbers in Tables 1 and 2 (main text) are self-measured under the shared protocol described in Section 4 (single NVIDIA Tesla P100 GPU, batch size 256, full ImageNet-1K validation set, 50,000 images) rather than drawn from the cited papers, except where a row is explicitly marked "(cited)", e.g. the DynamicViT/EViT rows in Appendix A.4. Two sourcing notes for transparency:

- **DeiT-S baseline.** We use 79.73% top-1 as the canonical DeiT-S/16 baseline for computing ΔAcc throughout Table 2, matching our own measured no-pruning run.
- **RAJNI DeiT-S sweep.** Our source records contain two overlapping DeiT-S hyperparameter search exports. We report the 17-point export whose own baseline row ($1.00\times$, 79.73%) matches the canonical baseline above as "the dedicated sweep" referenced in Appendix A.3; we did not additionally attempt to recompute a merged non-dominated frontier across both exports for this submission.

A.3 Hyperparameter Search Summary

RAJNI's operating points are obtained by a Tree-structured Parzen Estimator (TPE) search over $(\tau_{\text{warmup}}, \gamma)$, run independently per backbone. Table 5 summarizes the search. The reported operating points in Tables 1 and 2 are representative, non-dominated points selected from these full Pareto fronts (23 retained points for ViT-Base and 17 for DeiT-S, released in full in the supplementary material), not points cherry-picked outside that set.

	ViT-Base/16	**DeiT-S/16**
Search method	TPE (Optuna)	TPE (Optuna)
τ_{warmup} range explored	0.060–0.156	0.051–0.176
γ range explored	0.706–1.921	0.712–2.243
Non-dominated points retained	23	17
Number of trials	not logged in retained sweep records	
Accuracy-drop constraint	$\sim$2.0pp soft budget (penalized, not hard-filtered, in the search objective)	

Table 5: Hyperparameter search summary. The τ_{warmup}/γ ranges are the range spanned by the non-dominated points retained in the full sweep (released in the supplementary material), not a separately logged search-space configuration; the true configured bounds may have been wider. The number of trials evaluated (as opposed to retained) was not recorded alongside these results, so we report it as unavailable rather than estimate it.

A.4 Trained-Method Reference Comparison

Table 6 lists two representative methods that require training or fine-tuning, cited for reference only: they are not directly comparable to the training-free methods in Tables 1 and 2, since they pay an offline training cost RAJNI does not.

Method	**Training**	**Top-1 (%)**	**Speedup**	**GFLOPs**
DynamicViT (cited)	30 epochs, keep_rate= 0.7	79.3	1.5×	3.0
EViT (cited)	30 epochs, keep_rate= 0.7	79.5	1.5×	3.0

Table 6: Trained token-pruning baselines on DeiT-S/16, cited for reference; not directly comparable to Tables 1/2.