Dyna-ViT: Parameter-Free Pre-Encoder Token Pruning for Efficient Vision Transformers

Supplementary Material

Supplementary Material

This document provides supplementary material for the paper titled *"Dyna-ViT: Parameter-Free Pre-Encoder Token Pruning for Efficient Vision Transformers"*, accepted for publication in the *CVPR 2026 Findings*. It contains additional experimental results, ablation studies, qualitative analyses, and implementation details that complement the main manuscript.

Contents

- Additional results and ablation studies
- Qualitative visualizations
- Implementation details and reproducibility notes
- Scalability protocol and additional notes

This supplementary document is organized as follows. Section 1 provides additional quantitative comparisons and ablation plots. Section 2 presents qualitative visualizations and token-selection masks. Section 3 summarizes implementation details and reproducibility notes. Section 4 describes the scalability protocol and additional notes.

1. Additional Results and Ablation Studies

This section provides extended quantitative results and ablation analyses referenced in the main paper. We report K-sweep results (accuracy versus retained sequence length), average training time per epoch, dataset-specific breakdowns, and additional ImageNet-1k evaluation results.

Table 1 summarizes results across datasets. To provide a more detailed view of large-scale classification performance, Table 2 reports ImageNet-1k results for ViT-B/16 and Dyna-ViT.

We evaluate whether Dyna-ViT is competitive with DynamicViT and ToMe while introducing no additional parameters.

Compute accounting (analytic vs. tool). Analytic FLOPs derived from the effective sequence length S closely match tool estimates where operator support is available, consistent with the compute analysis reported in the main paper. Additional ablation results for different keep ratios K are also provided in the main paper.

2. Qualitative Visualizations

This section provides qualitative visualizations referenced in the main manuscript. We present saliency heatmaps, token masks, selection overlays, prediction examples, and LIME-based explanations to illustrate the behavior of Dyna-ViT under different token-retention settings and saliency proxies.

Table 1. Main comparison under matched backbones, resolution, training schedules, and comparable token budgets. Dyna-ViT uses L2 saliency with K=70%. Times are average seconds per epoch on a single T4 GPU with AMP. For ImageNet-1k, time per epoch is not reported in this table.

Dataset	Method	Acc. (%)	T/epoch	SeqLen
VOC-val	ViT-B/16 (baseline)	96.8	16.7s	197
	DynamicViT	46.0	12.2s	<197
	ToMe	87.1	11.5s	<197
	Dyna-ViT (ours)	**97.1**	**12.5s**	**1+137**
CIFAR-100 (test)	ViT-B/16 (baseline)	92.0	163s	197
	DynamicViT	88.30	100s	<197
	ToMe	90.49	112s	<197
	Dyna-ViT (ours)	**91.71**	**117s**	**1+137**
Tiny-ImageNet	ViT-B/16 (baseline)	85.5	515s	197
	DynamicViT	83.40	1632s	>197
	ToMe	85.64	413s	<197
	Dyna-ViT (ours)	**88.31**	**407s**	**1+137**
ImageNet-1k	ViT-B/16 (baseline)	81.8	6101s	197
	Dyna-ViT (ours)	**82.3**	4093s	**138**

2.1. Scoring Proxy Heatmaps

We compare L2, Sobel, and entropy-based saliency maps to show how different parameter-free scoring functions emphasize different image structures. Although the highlighted regions vary slightly, the resulting token selections are broadly consistent. Fig. 1

2.2. Dynamic Selection Overlay

We visualize the retained patches after top-K selection to show which image regions are preserved by Dyna-ViT. At aggressive pruning levels, the selected regions remain concentrated around semantically relevant content.

Table 2. ImageNet-1k classification results with ViT-B/16 at 224×224. Throughput is measured on a single NVIDIA T4 GPU with AMP and includes end-to-end pruning overhead.

Method	K (%)	SeqLen	Top-1 Acc. (%)	GFLOPs	Throughput (img/s)	Speedup
ViT-B/16 (Baseline)	100	197	81.8	19.90	210	1.00×
Dyna-ViT (Ours)	**70**	**138**	**82.3**	**13.40**	**313**	**1.49×**

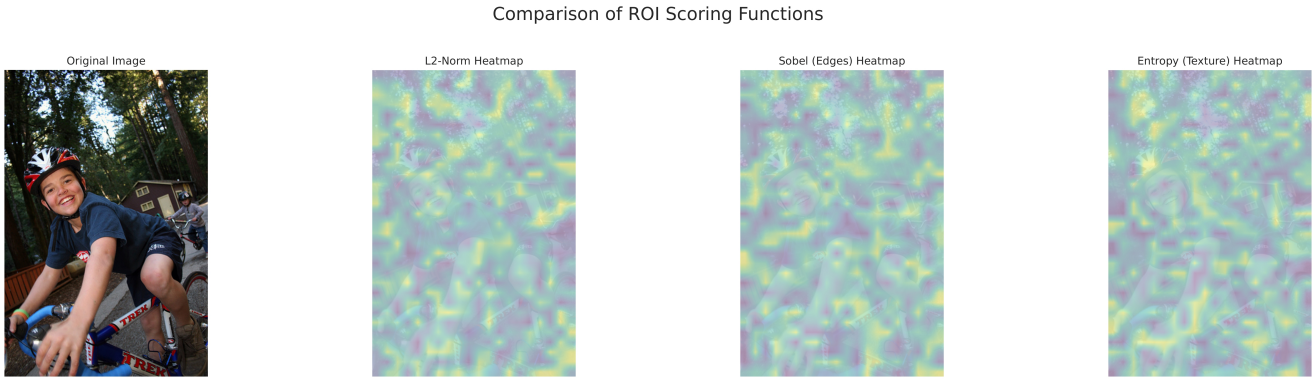


Figure 1. Saliency proxy behavior. L2, Sobel, and entropy emphasize slightly different structures, but produce broadly similar token selections at K=70%.

2.3. Binary Selection Mask at K=25%

This visualization shows the binary token-selection mask corresponding to the retained patches. White regions indicate selected patches, while black regions denote pruned patches.

2.4. Selection Mask Overlay at K=50%

We further visualize token retention at a less aggressive pruning level. Compared with K=25%, more contextual regions are preserved while still reducing the effective sequence length substantially. Qualitative visualizations of the selection process are presented in Fig. 2a, Fig. 2b, and Fig. 2c.

2.5. Sample Predictions

We present representative prediction examples from the test split to provide a qualitative sanity check of model behavior under sparse token selection.

2.6. LIME-Based Explanations

We compare LIME visualizations for Dyna-ViT, ToMe, and DynamicViT to examine which image regions each method relies on for prediction.

3. Implementation Details and Reproducibility

Our experiments use a single NVIDIA T4 GPU (16GB) with CUDA 12.x, PyTorch 2.x, and AMP FP16 enabled for all timing and throughput measurements. Models are based on `vit_base_patch16_224` from the `timm` library [1], pretrained on ImageNet-21k and fine-tuned for each dataset.

Datasets and Preprocessing:

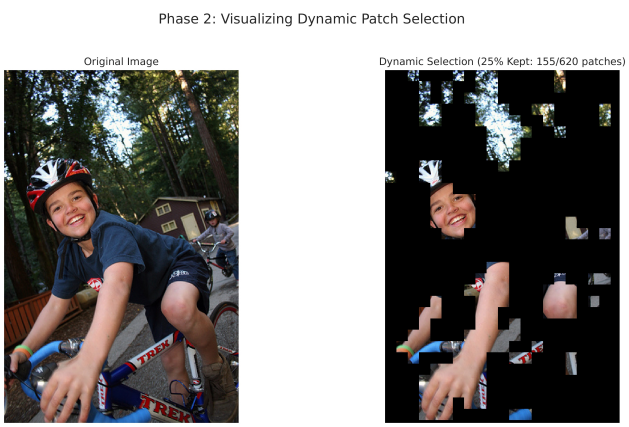
- **PASCAL VOC 5-class:** 3875/647/653 train/validation/test images (224×224, ImageNet normalization), filtered for person, car, cat, dog, bicycle.
- **CIFAR-100:** Official 50k/10k train/test split, resized from 32×32 to 224×224, CIFAR normalization.
- **Tiny-ImageNet:** Subset of 200 ImageNet classes, 100k train and 10k validation images (from 64×64 upsampled to 224×224, ImageNet normalization).

Training:

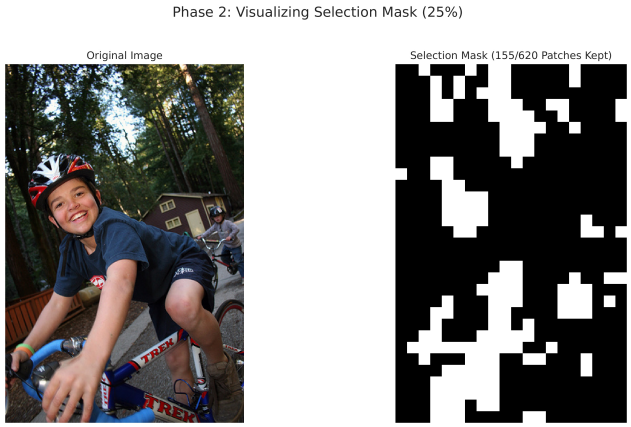
- Optimizer: AdamW (`lr`=3×10^{-5}, weight decay=0.05), StepLR (factor 0.1 every 3 epochs).
- Batch size: 32 for VOC, Tiny-ImageNet; 64 for CIFAR-100.
- Epochs: 5–10, no early stopping.
- Data augmentations: RandomResizedCrop(224), RandomHorizontalFlip.

Token Selection and Ablations:

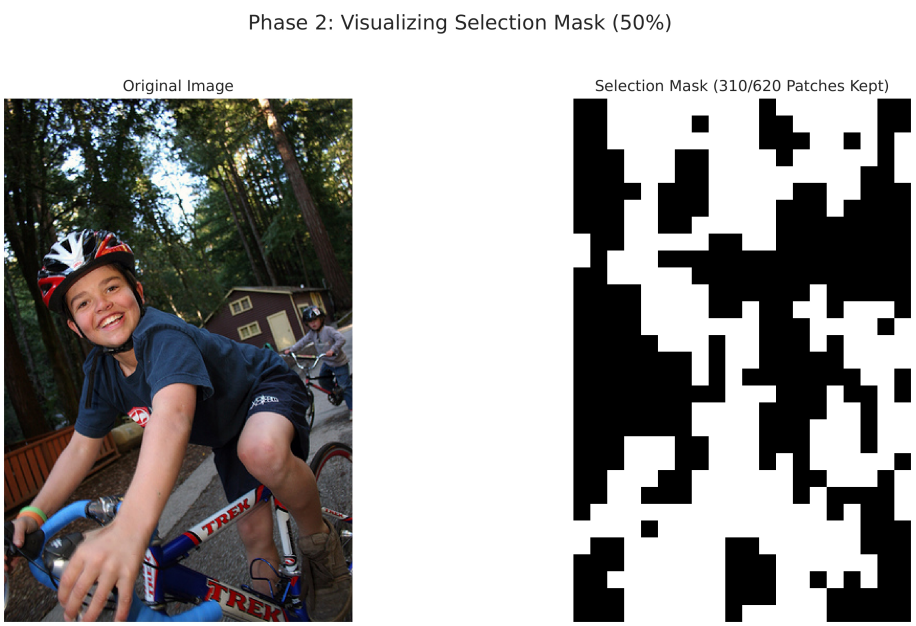
- Saliency proxies: L2 (default), Sobel, and entropy, as described in Sec. 3.
- Keep ratio K: Sweeps over $K = 30, 50, 70, 90, 100$; $K = 70$ used for main results (chosen by VOC validation).



(a) Dynamic selection visualization at $K{=}25\%$. Only the selected patches are retained.



(b) Binary selection mask at $K{=}25\%$. White patches are retained and black patches are pruned.



(c) Selection mask visualization at $K{=}50\%$. More patches are retained, providing additional contextual information.

Figure 2. Visualization of dynamic patch selection and selection masks at different retention ratios.

Figure 3. Sample predictions on the test split.

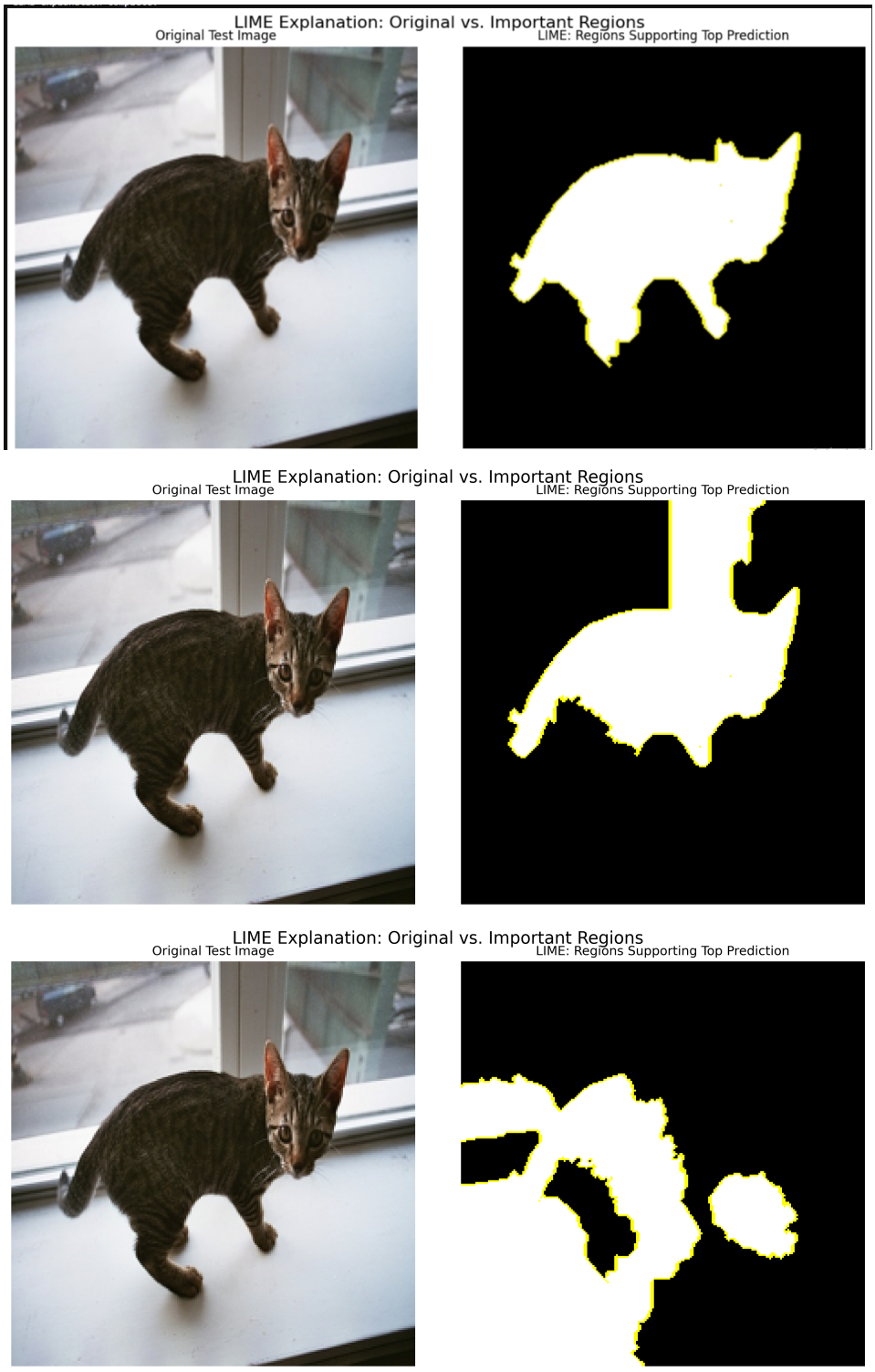


Figure 4. LIME-based explanations for Dyna-ViT, ToMe, and DynamicViT. From top to bottom, the panels show explanations for Dyna-ViT, ToMe, and DynamicViT, respectively. The visualizations highlight the image regions that most strongly influence the model predictions.

- For each split and dataset, Top-K indices are precomputed and reused for inference speed.
- For fair comparison, Baseline, DynamicViT, and ToMe use the same backbone, input resolution, and training schedule. The mapping from DynamicViT thresholds and ToMe merge ratios to effective sequence length is provided in Section 4.

Evaluation and Reporting:
- Mean and standard deviation are reported over 3 seeds: 42, 43, 44.
- Efficiency is reported as average wall-clock *time per epoch* on a single T4 GPU with AMP.
- Analytic FLOPs are computed from the effective sequence length S using a fitted S^2 model, and closely match the `fvcore` estimates reported in the main paper.
- Full K-sweep tables and scorer ablations are provided in Section 1; LIME visualizations and related qualitative analyses are provided in Section 2.

Scaling/Transfer Protocols. For ImageNet scaling, we provide an evaluation script (`scripts/scaling_protocol_imagenet.sh`) and precomputed patch indices for common grid sizes. Section 4 provides additional implementation details and usage notes.

Reproducibility Checklist.
- **Data and splits:** All datasets are public, and the train/validation/test splits follow the standard release protocols.
- **Hardware/software:** All experiments are conducted on a single NVIDIA T4 GPU with CUDA 12.x, PyTorch 2.x, and AMP enabled throughout.
- **Determinism:** Each experiment is repeated with three random seeds (42, 43, and 44), using deterministic settings in `torch`, `cudnn`, and the data-loading pipeline whenever possible.
- **Code and configs:** Configuration files and evaluation scripts for the baseline, DynamicViT, ToMe, and Dyna-ViT are included to support reproducibility.

Ethics statement: All experiments are conducted on standard, public datasets. No new data are collected or human subjects involved. See section **??** for potential limitations under domain shift and low-texture images.

4. Scalability Protocol and Additional Notes

This section details the protocol for scaling Dyna-ViT to large-scale benchmarks, specifically ImageNet, and outlines reproducibility steps. Training and evaluation for ImageNet-1k use the same backbone and token selection configuration as described above. Reference FLOPs estimation matches practical tool counts.

ImageNet Setup:
- Backbone: ViT-Base (patch16, 224)
- Training patches: Precomputed indices for multiple K values ($K = 70\%, 50\%, 25\%$) with 224×224 inputs
- Training schedule: Batch size 128, 10 epochs, AdamW optimizer (`lr`=1×10^{-4}, weight decay 0.05)
- Evaluation: Top-1 and Top-5 accuracy on official ImageNet validation set
- FLOPs calculation: Analytic formulas based on the effective sequence length are matched to the tool-based estimates reported in the main paper.

Additional Notes: ImageNet-1k results are included in Tables 1 and 2 to assess large-scale classification scalability.

Reproducibility. For all experiments:
- Random seeds: 42, 43, and 44.
- Hardware: Single NVIDIA T4 GPU with PyTorch 2.x, CUDA 12.x, and AMP enabled.
- Software/scripts: `timm`, custom Dyna-ViT modules, and ablation scripts for K-sweep and saliency-proxy analysis.
- Data splits: PASCAL VOC, CIFAR-100, Tiny-ImageNet, and ImageNet-1k official splits.
- Visualization: Scripts for saliency maps, overlays, selection masks, and confusion matrices.

The supplementary material is organized to support verification and replication of the reported results.

Ethics Statement: All data used in this work is public and non-sensitive. No human or personally identifying data is used.

References

[1] Ross Wightman. Pytorch image models. 2019. 2