DORA: DYNAMIC ONLINE REINFORCEMENT AGENT FOR TOKEN PRUNING IN VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Transformers (ViTs) incur quadratic self-attention cost in the number of tokens. Most token-reduction methods adapt token identities within a prescribed layer-wise compression schedule, or search a static mask offline, and thus limit online adaptation of when and how much to prune. We propose **DORA** (**D**ynamic **O**nline **R**einforcement **A**gent), which learns an input-adaptive pruning policy itself for frozen ViTs. At each eligible block, a hierarchical actor decides whether to prune, how many tokens to remove, and which tokens to remove from each image's evolving representation. Because early deletions change the states observed by later decisions, DORA formulates pruning as a finite-horizon Markov decision process. Complete-prefix shadow evaluations convert final-prediction fidelity into localized per-step credit, while closed-loop accuracy feedback adjusts the fidelity penalty toward a shared accuracy-drop target. A privileged critic and all shadow computations are training-only. Deployment retains the frozen backbone and a lightweight actor that applies hard deletion and packed variable-length FlashAttention, converting token reduction into measured speedups. On ImageNet-1K with DeiT-Base, DORA reduces FLOPs by 38.4% relative to the uncompressed backbone within one percentage point of accuracy loss. Averaged across four ViT-type backbones at matched accuracy, DORA uses 13.2% fewer FLOPs and achieves 32.4% higher throughput than the corresponding per-backbone baseline means. Under zero-shot transfer to ImageNet-A, these gains widen to 20.3% and 45.6%, respectively.

1 INTRODUCTION

Vision Transformers (ViTs) (Dosovitskiy et al., 2021) incur quadratic self-attention cost in the number of tokens, making token pruning or merging a direct route to lower inference cost (Bolya et al., 2023; Wang et al., 2024). Most existing methods improve how tokens are scored, matched, or aggregated within a prescribed layer-wise compression schedule: token identities may depend on the image, but the per-layer budget is shared across inputs (Bolya et al., 2023; Wang et al., 2024; Lee et al., 2024; Tran et al., 2024; Choi et al., 2025). This restriction is mismatched to input-dependent redundancy: a common schedule can retain unnecessary tokens for one image and remove useful detail from another. We argue that the compression budget itself should be input-adaptive, decided online from each image's evolving representations.

Token pruning is also not a sequence of independent local ranking problems. An early deletion saves more downstream computation, but it changes the representations on which later decisions are made; tokens that are individually expendable may be unsafe to remove together. We therefore treat pruning as online sequential control, formulated as a finite-horizon Markov decision process (MDP). This formulation lets the objective account for the long-term effect of discrete pruning actions rather than relying only on local attention or similarity proxies. V-Pruner also uses PPO, but searches for an offline mask shared across inputs (Yao et al., 2026); our learned policy remains active at inference.

We introduce **DORA** (**D**ynamic **O**nline **R**einforcement **A**gent), an RL framework that learns the pruning policy itself for frozen ViTs. A hierarchical actor decides *where* to prune, *how many* tokens to remove, and *which* tokens to remove (Figure 1, right), with rollouts that physically delete tokens so that later decisions observe the state created by earlier actions. Two training-only mechanisms make this policy learnable without backbone-specific manual tuning: complete-prefix shadow evaluations

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.

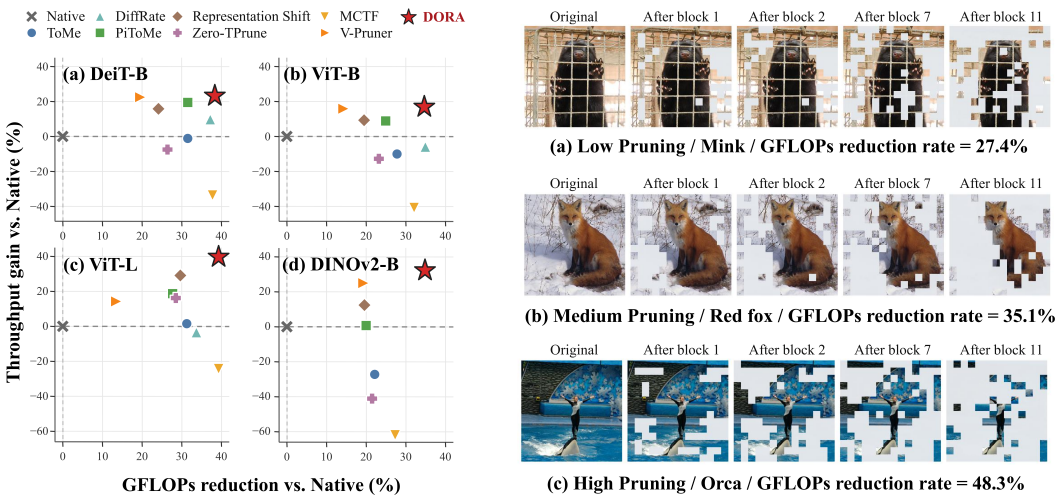

Figure 1: **Joint efficiency and input-dependent behavior.** Left: GFLOPs reduction and measured throughput gain at accuracy-matched operating points across four ViT-type backbones; DORA (star) achieves the strongest joint efficiency at comparable accuracy. Right: representative DeiT-B trajectories showing that the learned policy applies different pruning intensities and retains different spatial evidence across images.

convert final-prediction fidelity into localized credit for individual pruning actions, and closed-loop accuracy feedback adjusts the fidelity penalty toward a single accuracy-drop target shared across backbones. Deployment retains only the frozen backbone and a lightweight deterministic actor, whose hard deletion and packed variable-length attention turn token reduction into measured wall-clock speedups. At matched ImageNet-1K accuracy, DORA averages 13.2% fewer FLOPs and 32.4% higher throughput than the baseline means across four ViT-type backbones (Figure 1, left); under zero-shot transfer to ImageNet-A, these gains widen to 20.3% and 45.6% (Table 1).

Our contributions are:

- We formulate dynamic online token pruning as a finite-horizon MDP and propose a hierarchical policy consisting of a gate, a budget head, and a selector, which jointly decide where to prune, how much to prune, and which tokens to remove for each input.
- We introduce localized credit assignment based on complete-prefix shadow evaluations, together with closed-loop accuracy feedback for regulating the compression–fidelity trade-off, eliminating backbone-specific manual tuning.
- We couple adaptive hard pruning with packed variable-length FlashAttention and avoid attention-map materialization, token-size corrections, and batch-wide padding, all contributing to higher end-to-end throughput.

2 Related Work

Token pruning and merging. Token reduction shortens the sequence processed by a ViT through either pruning or merging. Pruning methods score and discard tokens using attention- or graph-based importance: ATS samples tokens from classification-token attention scores without additional parameters, keeping a per-image varying number of tokens up to a preset maximum (Fayyaz et al., 2022), while Zero-TPrune builds an attention graph and ranks tokens with Weighted PageRank (Wang et al., 2024). Merging methods instead aggregate tokens: ToMe matches keys via bipartite soft matching (Bolya et al., 2023), ATC clusters tokens bottom-up (Haurum et al., 2024), MCTF combines similarity, informativeness, and token size with one-step-ahead attention (Lee et al., 2024), and PiToMe protects informative tokens with a graph-energy criterion before matching (Tran et al., 2024). These methods improve token scoring, matching, or aggregation; yet while the compressed tokens may be chosen per image, the per-layer budget is prescribed and shared across inputs.

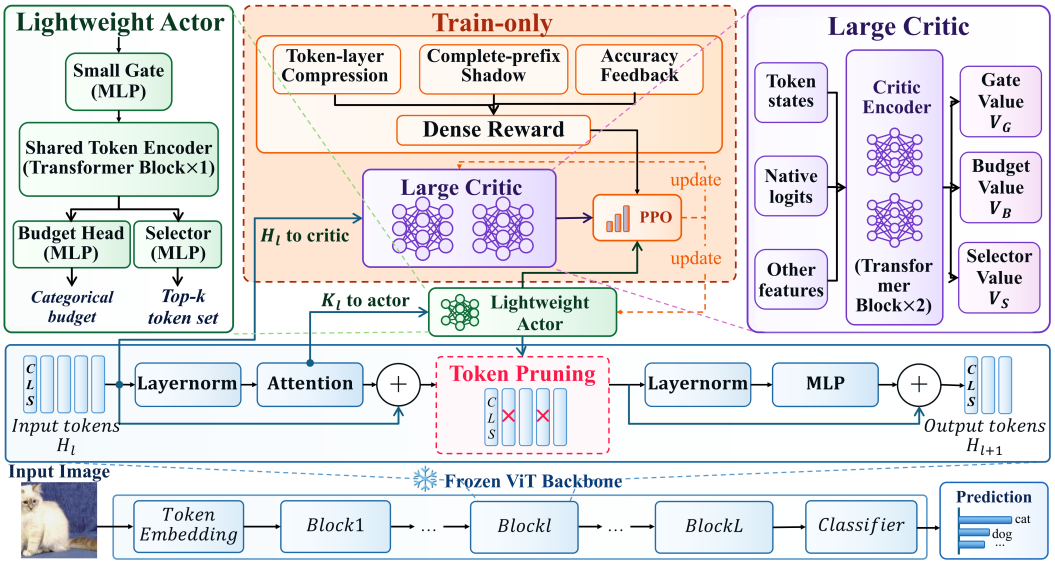

Figure 2: **Training–deployment separation in DORA.** Training uses a hierarchical actor with complete-prefix rewards, accuracy feedback, and a privileged critic; deployment retains only the frozen ViT and deterministic actor with hard pruning and variable-length FlashAttention.

Adaptive token reduction. DynamicViT predicts image-conditioned token masks but keeps a predefined token count at each sparsification stage and fine-tunes the backbone together with the prediction modules (Rao et al., 2021). DiffRate learns layer-wise compression rates under a computation constraint, but the learned rates are shared across images (Chen et al., 2023). RL4EViT formulates token pruning as a Markov game and uses MAPPO to learn input-dependent binary keep-or-prune decisions for individual tokens, but the pruning stages remain manually predefined and pruned tokens are suppressed through attention masking rather than physically removed from the sequence (Lu et al., 2025). V-Pruner applies PPO with a Fisher-information prior, but as an offline mask search whose pattern is shared across inputs (Yao et al., 2026). Across these methods, some element of the schedule—the stages, the token counts, or the mask pattern—remains fixed and shared across inputs.

Execution with fused attention. FlashAttention computes attention without materializing the full probability matrix (Dao et al., 2022), so methods that rank tokens by intermediate attention probabilities—ATS, Zero-TPrune, and MCTF among them—cannot use the standard fused operator directly (Fayyaz et al., 2022; Wang et al., 2024; Lee et al., 2024). Representation Shift restores compatibility with a training-free importance score derived from representation changes (Choi et al., 2025). Merging methods that add token-size terms to the attention logits, such as proportional attention, likewise require modifications to the operator (Bolya et al., 2023; Haurum et al., 2024).

3 Methodology

3.1 Overview and Formulation

DORA addresses the limitations above with a single learned policy: the compression budget is decided per image rather than prescribed, each decision accounts for its downstream effects through sequential rollouts, and hard deletion keeps execution compatible with the unmodified fused attention operator. Figure 2 gives an overview. A lightweight hierarchical actor is attached to a frozen ViT: its gate decides whether to prune, its budget head determines how many tokens to remove, and its selector chooses which tokens to remove.

Consider a pretrained ViT with L Transformer blocks. At block l, let $H_l = [x_{l,\mathrm{CLS}}; x_{l,1}, \ldots, x_{l,N_l}] \in \mathbb{R}^{(N_l+1)\times d}$, where $x_{l,\mathrm{CLS}}$ is the CLS classification-token representation, N_l is the number of surviving visual tokens, and d is the token dimension. N_0 denotes the initial visual-token count. DORA may prune at any $l \in \{0, \ldots, L-1\}$, while CLS is always retained.

Pruning operator. The actor makes its decision from the current attention keys, while deletion takes effect after the attention residual:

$$\begin{aligned} H_l^{\text{attn}} &= H_l + \text{Attention}_l(\text{LayerNorm}_l(H_l)), \\ H_{l+1} &= H_l^{\text{keep}} + \text{MLP}_l(\text{LayerNorm}'_l(H_l^{\text{keep}})). \end{aligned} \tag{1}$$

Here H_l^{keep} removes the selected visual-token rows while preserving CLS and survivor order. If k_l tokens are removed, $N_{l+1} = N_l - k_l$, with $k_l = 0$ when the gate is off. Removed tokens therefore participate in the current attention but not in the current MLP or subsequent blocks; the sequence is physically shortened without merging or zero masking.

MDP formulation. We formulate dynamic online token pruning as a finite-horizon MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ with one decision step per block and $\gamma = 1$. At block l, the policy observes $o_l = (K_l, h_l, l)$, where K_l contains the surviving attention keys and h_l summarizes the pruning history. Its hierarchical action $a_l = (g_l, k_l, S_l)$ specifies whether to prune, how many tokens to remove, and their indices. The transition is deterministic under the frozen backbone, while the reward balances token-layer compression and incremental prediction-fidelity loss (Section 3.3).

DORA learns a policy π that maximizes the undiscounted return $J(\pi) = \mathbb{E}_\pi\left[\sum_{l=0}^{L-1} r_l\right]$, so that intervention locations, pruning budgets, and token identities adapt to each image's evolving representation.

3.2 Hierarchical policy

The actor consists of a gate followed by a budget–selector controller that shares a token encoder. All layer widths and embedding sizes are listed in Appendix A.1.

Gate. The gate reads only the CLS key of K_l, concatenated with h_l and a learned block embedding, and predicts an activation probability p_l^G through a small MLP. When $g_l = 0$, no tokens are removed and the controller is skipped entirely; the gate is forced off at blocks where no positive removal count is legal. Reading only the CLS key keeps the gate's cost negligible relative to the block it guards.

Shared encoder. When the gate is active, the controller projects all surviving keys, adds a learned block embedding and the pruning history h_l, and processes the sequence with a single lightweight Transformer block, producing $Z_l \in \mathbb{R}^{(N_l+1)\times 128}$. This encoder models relations among surviving tokens, so scores can depend on the context shaped by preceding pruning decisions.

Budget head and selector. A budget head maps the encoded CLS token to a categorical distribution $\pi_B(\cdot \mid o_l)$ over a backbone-specific grid of candidate removal counts (Appendix A.2), masking counts that are invalid at the current block before the softmax.

Given the chosen budget k_l, the selector scores each visual token for removal:

$$s_{l,j} = \text{MLP}_S\left(Z_{l,j} + W_k \frac{k_l}{N_l}\right), \qquad j = 1, \ldots, N_l. \tag{2}$$

Where $Z_{l,j}$ is the encoded representation of token j and W_k is a learned projection of the removal fraction. A higher score indicates a stronger preference for removal; the CLS token is never selected. Conditioning on k_l/N_l allows token rankings to change with the requested pruning intensity.

Sampling and deployment. During training, the gate and budget are sampled from Bernoulli and categorical distributions, and exactly k_l distinct tokens are sampled without replacement using an ordered Plackett–Luce distribution over the scores (Plackett, 1975). Deployment uses a gate threshold of 0.5, the largest budget logit, and the k_l highest selector scores.

3.3 Localized rewards from complete-prefix evaluation

To localize prediction degradation, we ask, for every block l in a sampled rollout: what would the backbone predict if pruning stopped right after block l? We answer by replaying the recorded actions at blocks 0 through l and running all remaining blocks with no further pruning, producing logits $z^{(l)}$ from the original classifier. We call each such replay a *complete-prefix shadow evaluation*, as it branches off the realized trajectory without affecting it. Its fidelity cost relative to the unpruned

logits z^{N} uses a temperature-scaled soft-target KL divergence with the conventional τ^2 scaling from knowledge distillation (Hinton et al., 2015):

$$D_l = \tau^2 \,\mathrm{KL}\Big(\mathrm{softmax}(z^{\mathrm{N}}/\tau) \,\|\, \mathrm{softmax}(z^{(l)}/\tau)\Big), \qquad \tau = 3. \tag{3}$$

Because consecutive prefixes share all earlier actions and contain no later ones, $D_l - D_{l-1}$ attributes the change in final-prediction fidelity to the action at block l alone. The difference is signed: positive when the action moves the prediction away from the reference, negative when it moves closer.

Rewarding fidelity alone would discourage pruning altogether, so we pair it with a compression term measured across the whole backbone, since removing a token earlier saves computation in more downstream blocks. Let N_0 denote the initial number of visual tokens and N_l^{post} the number remaining after pruning at block l. The token-layer compression, and the increment we assign to block l, are

$$C = \frac{1}{LN_0} \sum_{l=0}^{L-1} \left(N_0 - N_l^{\mathrm{post}}\right), \qquad \Delta C_l = \frac{k_l(L-l)}{LN_0}, \tag{4}$$

so that $C = \sum_{l=0}^{L-1} \Delta C_l$: each removed token contributes in proportion to the number of blocks it skips.

Combining the two terms, the reward at training update u is

$$r_l = 25\,\Delta C_l - a_u\,(D_l - D_{l-1}), \qquad D_{-1} = 0, \tag{5}$$

where the shared factor 25 sets the scale of the compression term relative to the fidelity cost, and a_u is the fidelity coefficient adjusted by accuracy feedback (Section 3.4). The incremental fidelity terms telescope to the final trajectory-level divergence, distributing the end-to-end cost across decisions; we add no separate terminal or classification-event reward. The derivation and shadow-evaluation procedure are provided in Appendix B.

3.4 Closed-loop accuracy feedback

A fixed fidelity coefficient does not produce the same quality regime across backbones. DORA therefore evaluates the deterministic deployment policy periodically on a held-out shard of the training set. Let d_u be its paired Top-1 drop relative to the Native model on that shard, i.e., degraded minus improved prediction fractions. The coefficient is updated toward a one-percentage-point target:

$$a_{u+1} = \mathrm{clip}\left(a_u + 5\ \mathrm{clip}\left(\frac{d_u - 0.01}{0.01}, -2, 2\right), 0, 300\right). \tag{6}$$

The penalty increases when the measured drop exceeds the target and is relaxed otherwise; labels are used only to compute feedback accuracy, and the target regulates training without being a hard constraint on unseen validation data. Cadence, shard construction, and initialization are given in Appendix C.

3.5 Optimization and inference

Training-only critic. A separate, larger critic is available only during training. Beyond the current token state, it reads privileged information unavailable to the actor: the Native logits of the same image and scalar features describing the block position, token count, pruning history, reference confidence, and the current fidelity coefficient (Appendix D.1). It predicts value baselines for the gate, budget, and selector decisions; the gate and budget values are computed without the current removal count, while the selector value additionally receives the chosen budget and is evaluated before token selection, so no baseline observes which tokens will be removed or the resulting prediction.

Policy optimization. We use GAE (Schulman et al., 2016) and separate clipped PPO objectives (Schulman et al., 2017), one per decision type. A common λ-return is shared by all three decision types, each subtracting its own value baseline, and the advantages are normalized separately over their eligible transitions. The critic architecture, value targets, PCGrad handling (Yu et al., 2020), and hyperparameters are specified in Appendix D.

Inference. At inference, only the frozen backbone and deterministic actor are retained. Surviving sequences are compacted into a packed buffer with explicit sequence boundaries that prevent attention

between different images, and processed with variable-length FlashAttention (Dao et al., 2022). Compaction preserves per-image token order, and only gate-active examples enter the controller. The actor requests neither attention probabilities nor token-size corrections, so the backbone uses the unmodified fused attention operator; reported throughput includes the actor, token selection, packing, and compaction.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Backbones. We evaluate four frozen pretrained backbones: DeiT-B/16, ViT-B/16 and ViT-L/16 with AugReg checkpoints, and DINOv2-B/14 with its official linear readout (Touvron et al., 2021; Steiner et al., 2022; Oquab et al., 2024). The patch-16 backbones process 196 visual tokens at 224×224 resolution, while DINOv2-B/14 processes 256. All methods under a backbone use the same weights and the checkpoint's official evaluation preprocessing.

Data and checkpoint selection. DORA is trained only on mutually disjoint subsets of the ImageNet-1K training set (Russakovsky et al., 2015); checkpoints are selected on a held-out development split by minimizing GFLOPs subject to a one-percentage-point Native accuracy drop. The DORA checkpoint is selected on the development split and frozen before evaluation on ImageNet-1K validation and ImageNet-A. Baseline operating points are matched to DORA's realized accuracy under the protocol in Appendix E.2.

Baselines. We compare with seven frozen-backbone token-reduction methods: ToMe, PiToMe, MCTF, DiffRate, Representation Shift, Zero-TPrune, and V-Pruner. For Zero-TPrune, we use a paper-based reimplementation. For V-Pruner, the per-budget offline mask search is repeated independently on each backbone; this search is an offline cost and is excluded from deployment measurements.

Metrics and runtime measurement. All final numbers are produced under a single BF16 inference contract, so each backbone has exactly one Native reference. We report full-set Top-1 accuracy, complete-forward GFLOPs ($1\,\mathrm{MAC} = 2\,\mathrm{FLOPs}$, including patch embedding and the classifier), and GPU-resident end-to-end throughput on one RTX 4090 at batch size 320. Timing includes each method's online selection, merging or pruning, packing, and compaction overhead, and excludes data loading and host-to-device transfer. DORA uses packed variable-length FlashAttention; other methods use the fastest attention path compatible with their semantics. Exact data splits, checkpoint selection, implementations, and timing protocol are given in Appendix E.

4.2 ACCURACY–COMPUTE–THROUGHPUT TRADE-OFFS

ImageNet-1K results. At matched accuracy, DORA has the highest measured throughput on every backbone and, with two small exceptions, also the lowest FLOPs. The pairwise comparisons show that lower FLOPs do not necessarily imply higher throughput: on DeiT-B, MCTF nearly matches DORA's computation (21.9 vs. 21.7 GFLOPs) yet reaches only 2,122 images/s against DORA's 3,944, while V-Pruner comes within 0.7% of DORA's throughput but requires 28.3 GFLOPs to DORA's 21.7. The two exceptions undercut DORA's computation only marginally—DiffRate on ViT-B by 2.2% and MCTF on ViT-L by 2.0%—while DORA's throughput is 23.4% and 84.7% higher, respectively, showing that FLOPs and wall-clock performance are not interchangeable. Averaged equally across the four backbones, DORA reduces FLOPs by 13.2% and raises throughput by 32.4% relative to the baseline means, indicating that the actor and dynamic-sequence operations do not erase the savings from hard deletion.

Zero-shot transfer to ImageNet-A. ImageNet-A consists of naturally occurring adversarial examples with atypical object appearances, backgrounds, textures, poses, scales, and partial occlusions, providing a more challenging distribution-shift setting than ImageNet-1K for preserving recognition accuracy under token reduction (Hendrycks et al., 2021). We transfer the selected DORA policies directly from ImageNet-1K to ImageNet-A without retraining, checkpoint re-selection, or any adjustment of the DORA operating point (Section 4.1). For accuracy-matched comparison, each baseline is evaluated at its available ImageNet-A operating points, and we report the point whose Top-1 accuracy is closest to that of the transferred DORA policy. Under this protocol, DORA's advantage widens under distribution shift: averaged across the four backbones, it uses 20.3% fewer FLOPs and achieves

Table 1: Accuracy-matched comparison with seven token-reduction baselines on ImageNet-1K and ImageNet-A. Baseline operating points match DORA within 0.1 pp Top-1. *Avg. baselines* is the unweighted mean over available baselines; "–" denotes no matching point. Bold marks the best efficiency value in each backbone block and the DORA row.

Backbone	Method	ImageNet-1K			ImageNet-A		
		Top-1 Acc (%)	GFLOPs	Throughput (images/s)	Top-1 Acc (%)	GFLOPs	Throughput (images/s)
DeiT-B	Native	81.8	35.2	3230	27.9	35.2	3230
	ToMe	80.9	24.1	3163	26.6	29.6	2433
	DiffRate	80.8	22.1	3512	–	–	–
	PiToMe	80.8	24.1	3821	26.7	23.6	3844
	MCTF	80.8	21.9	2122	26.7	26.6	1647
	Zero-TPrune	80.8	25.9	2960	26.7	23.2	3216
	Representation Shift	80.8	26.7	3705	26.7	28.5	3480
	V-Pruner	80.8	28.3	3917	26.6	32.5	3441
	Average of baselines	80.8	24.7	3314	26.7	27.3	3010
	DORA	**80.8**	**21.7**	**3944**	**26.7**	**21.0**	**4233**
ViT-B	Native	84.6	35.2	3211	44.0	35.2	3209
	ToMe	83.7	25.2	2887	43.1	29.6	2432
	DiffRate	83.8	**23.1**	3018	–	–	–
	PiToMe	83.8	26.2	3494	43.2	26.5	3458
	MCTF	83.8	23.7	1900	43.2	26.6	1647
	Zero-TPrune	83.8	26.8	2800	43.2	23.3	3201
	Representation Shift	83.8	28.1	3508	43.2	28.6	3457
	V-Pruner	83.8	29.9	3719	43.2	34.3	3278
	Average of baselines	83.8	26.1	3046	43.2	28.2	2912
	DORA	**83.8**	23.6	**3725**	**43.2**	**23.2**	**3831**
ViT-L	Native	85.8	123.4	998	55.3	123.4	997
	ToMe	85.0	84.8	1013	54.6	115.7	726
	DiffRate	85.1	81.8	963	54.6	77.2	1070
	PiToMe	85.1	89.2	1184	54.7	102.9	1035
	MCTF	85.1	**73.4**	755	54.7	109.4	464
	Zero-TPrune	85.1	88.2	1159	54.7	75.1	1304
	Representation Shift	85.1	86.8	1288	54.7	94.3	1182
	V-Pruner	85.1	106.7	1140	54.6	121.5	1012
	Average of baselines	85.1	87.3	1071	54.7	99.4	970
	DORA	**85.1**	74.9	**1394**	**54.7**	**74.3**	**1456**
DINOv2	Native	84.5	46.5	1842	57.3	46.5	1842
	ToMe	83.6	36.2	1340	55.9	40.8	1176
	DiffRate	–	–	–	56.0	33.5	1640
	PiToMe	83.7	37.2	1854	56.0	37.4	1843
	MCTF	83.7	33.8	701	56.0	36.8	628
	Zero-TPrune	83.7	36.5	1085	56.0	33.9	1282
	Representation Shift	83.7	37.4	2070	56.0	41.3	1879
	V-Pruner	83.7	37.5	2301	55.9	44.0	1992
	Average of baselines	83.7	36.4	1559	56.0	38.2	1491
	DORA	**83.7**	**30.3**	**2466**	**56.0**	**32.5**	**2389**

45.6% higher throughput than the baseline means. DORA also has the lowest FLOPs and the highest measured throughput among the accuracy-matched operating points on all four backbones.

4.3 DORA LEARNS BACKBONE- AND INPUT-ADAPTIVE PRUNING

Where to intervene. Figure 3 analyzes where DORA intervenes and how much it prunes. On the first axis, the gate converges to backbone-specific intervention schedules rather than inheriting a common hand-designed placement: DeiT-B intervenes at blocks $\{1, 2, 7, 11\}$, ViT-B at $\{1, 4, 9, 11\}$, ViT-L at $\{3, 4, 16, 23\}$, and DINOv2-B at $\{1, 2, 3, 6\}$, where block 3 is conditional and activates on only 66.5% of validation images. This learned placement contrasts with hand-crafted patterns: ToMe merges tokens at every block (Bolya et al., 2023), and Zero-TPrune prunes only at predefined

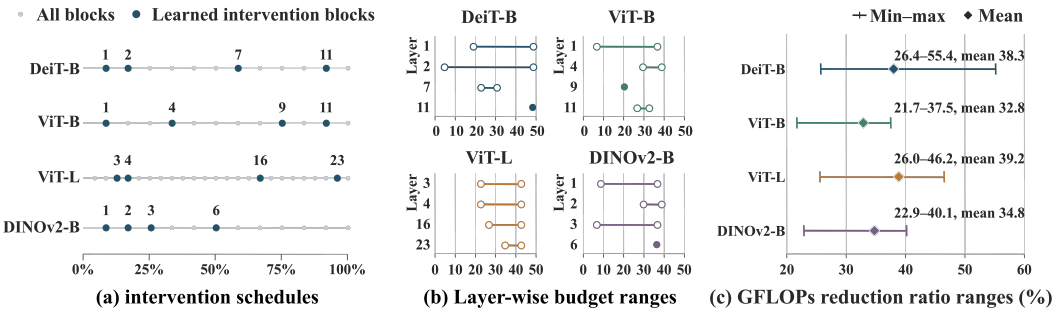

Figure 3: **Learned adaptivity in where and how much to prune.** (a) Intervention locations learned for each backbone. (b) Min–max deletion budgets over gate-active validation images at each intervention block; filled points denote collapsed single-budget decisions. (c) The resulting per-image GFLOPs-reduction ranges and means.

layers (Wang et al., 2024), whereas DORA discovers where to intervene through training, including input-conditioned activation at individual positions.

How much to prune. Several intervention blocks span broad budget ranges across images, while others converge to narrow or deterministic budgets (Figure 3(b)). Consequently, per-image GFLOPs reductions range from 26.4–55.4% on DeiT-B (mean 38.3%), 21.7–37.5% on ViT-B (mean 32.8%), 26.0–46.2% on ViT-L (mean 39.2%), and 22.9–40.1% on DINOv2-B (mean 34.8%), in line with the average reductions in Table 1. The qualitative trajectories in Figure 1 illustrate the corresponding differences in retained spatial evidence.

Which tokens to prune: selector quality. We ask whether the learned selector identifies the right tokens. Holding DORA's per-image gate decisions and deletion budgets fixed, we replace only the token-identity rule with random, similarity-based (ToMe-style), or attention-importance (Zero-TPrune-style) hard-pruning rules, each operating on its own evolving features. Under the fixed trajectory, the learned selector preserves higher Top-1 accuracy: improvements of 1.6–1.8 points over Random, 0.3–0.9 points over Similarity, and 0.9–3.5 points over Attention-WPR across the four backbones. This conclusion is conditional on the jointly learned gate and budget trajectories and does not claim universal optimality under arbitrary pruning budgets. The controlled protocol and full results are in Appendix F.1.

4.4 Accuracy feedback calibrates fidelity across backbones

The same update rule yields different coefficient regimes: by 300K training images, a reaches 186.2, 55.0, 85.0, and 32.4 for DeiT-B, ViT-B, ViT-L, and DINOv2-B, respectively (Figure 4, top). All runs share the initialization $a = 30$, so these endpoints are not backbone-specific settings; they emerge from the measured behavior of each policy. The endpoint values describe the ends of the training trajectories and need not coincide with the coefficients of the checkpoints selected for final evaluation. At the same time, the recent feedback averages move toward and then fluctuate around the shared 1-pp target (Figure 4, bottom). The feedback signal is measured on a quality-control split that is disjoint from both the PPO rollout pool and the development split used for checkpoint selection, so the target is a training signal rather than a guarantee on the final validation set.

The shared initialization is benign only because the loop is closed. Fixing a at 30 for the entire run is catastrophic on DeiT-B: no saved checkpoint satisfies the development-set quality constraint, and the terminal checkpoint reaches only 78.65% Top-1 with a 3.14-pp Native drop (Table 2). This failure is consistent with the trajectory in Figure 4: holding the 1-pp target on DeiT-B requires the coefficient to rise sixfold, from 30 to 186.2. A fixed low coefficient under-penalizes fidelity loss and the policy over-prunes; the closed loop removes this failure mode without backbone-specific tuning.

The feedback target can also control the operating point. Retraining DORA on DeiT-B with target drops of 0.5, 1.0, and 1.5 percentage points produces three distinct accuracy–efficiency regimes without otherwise changing the method (Figure 5): at every target, DORA maintains competitive GFLOPs and the highest measured throughput among accuracy-matched methods, occupying the

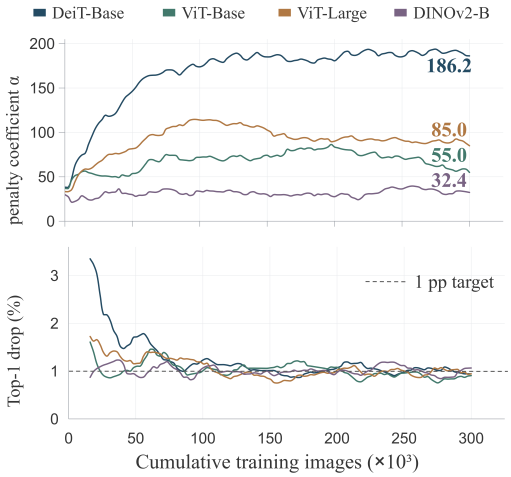

Figure 4: **Closed-loop accuracy feedback.** Top: learned fidelity coefficient a across four backbones. Bottom: recent eight-check mean Top-1 drop.

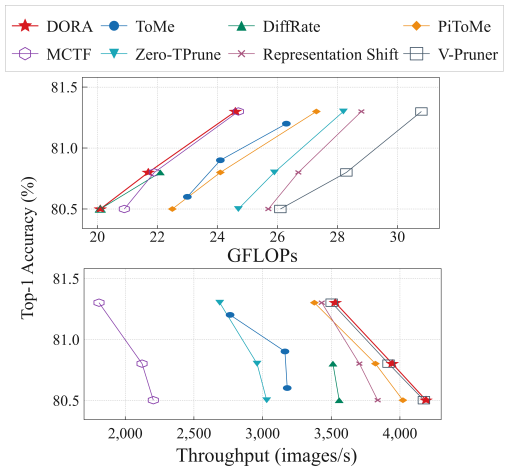

Figure 5: **Feedback-target sweep on DeiT-B.** Top: ImageNet-1K Top-1 accuracy versus GFLOPs per image. Bottom: Top-1 accuracy versus throughput.

Table 2: **Training-mechanism ablations on DeiT-B.**

Setting	Top-1(%)	GFLOPs	Native drop (pp)
Full DORA	**80.79**	**21.66**	**1.00**
Terminal-only reward	80.81	22.95	0.98
Fixed coefficient (a=30)	78.65	17.78	3.14

top-left of the accuracy–GFLOPs plane and the top-right of the accuracy–throughput plane. The wall-clock advantage thus persists across operating regimes rather than being tied to the default 1-pp target. Details are given in Appendix F.2.

4.5 Ablations and Stability

Replacing localized complete-prefix credit with a terminal-only reward produces a nearby but weaker selected operating point (80.81% Top-1, 22.95 GFLOPs versus 80.79%, 21.66 GFLOPs; Table 2), supporting the use of localized per-step credit. Repeating the full training procedure with three additional seeds yields closely clustered selected operating points: 80.82 ± 0.07% Top-1 and 21.76 ± 0.18 GFLOPs across four runs, indicating that the selected accuracy–efficiency regime is stable across the evaluated training seeds. Full results are reported in Appendix F.3.

5 Conclusion

We presented DORA, a reinforcement learning framework that formulates token pruning in vision transformers as a finite-horizon Markov decision process. A hierarchical actor separates whether, how much, and which tokens to prune, while complete-prefix shadow evaluations provide localized credit for these coupled decisions. A privileged critic and a closed-loop accuracy-feedback mechanism stabilize training, and the backbone remains frozen with a deterministic actor at deployment. Across four ViT backbones, DORA consistently improves the accuracy-efficiency trade-off among the evaluated frozen-backbone methods, and hard deletion combined with packed variable-length FlashAttention converts the reduced computation into measured acceleration.

DORA targets throughput-oriented ViT inference where input redundancy varies across images. Our evaluation is limited to image classification on four ViT-type backbones, with runtime benefits measured on GPU execution with variable-length FlashAttention; training also incurs additional cost from shadow evaluations and the privileged critic.

AI USE STATEMENT

We used AI assistants only for manuscript drafting and polishing, first-draft figure generation, discussion of experimental results, and reference checking. The authors reviewed and edited all AI-assisted material and take full responsibility for the final paper.

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. URL https://arxiv.org/abs/2210.09461.

Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. DiffRate: Differentiable compression rate for efficient vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2305.17997.

Joonmyung Choi, Sanghyeok Lee, Byungoh Ko, Eunseo Kim, Jihyung Kil, and Hyunwoo J. Kim. Representation shift: Unifying token compression with FlashAttention. In *IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2508.00367.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html.

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. URL https://arxiv.org/abs/2010.11929.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Juergen Gall. Adaptive token sampling for efficient vision transformers. In *European Conference on Computer Vision*, 2022. URL https://arxiv.org/abs/2111.15667.

Joakim Bruslund Haurum, Sergio Escalera, Graham W. Taylor, and Thomas B. Moeslund. Agglomerative token clustering. In *Computer Vision – ECCV 2024*, 2024. URL https://arxiv.org/abs/2409.11923.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. URL https://arxiv.org/abs/1907.07174.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Sanghyeok Lee, Joonmyung Choi, and Hyunwoo J. Kim. Multi-criteria token fusion with one-step-ahead attention for efficient vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2403.10030.

Chenglong Lu, Shen Liang, Xuewei Wang, and Wei Wang. Reinforcement learning-based token pruning in vision transformers: A Markov game approach. In *IEEE International Conference on Multimedia and Expo*, 2025. URL https://arxiv.org/abs/2503.23459.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick

Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975. doi: 10.2307/2346567. URL https://academic.oup.com/jrsssc/article/24/2/193/6953554.

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. URL https://arxiv.org/abs/2106.02034.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL https://link.springer.com/article/10.1007/s11263-015-0816-y.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? Data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=4nPswr1KcP.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357, 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

Hoai-Chau Tran, Duy M. H. Nguyen, Duy M. Nguyen, Trung-Tin Nguyen, Ngan Le, Pengtao Xie, Daniel Sonntag, James Y. Zou, Binh T. Nguyen, and Mathias Niepert. Accelerating transformers with spectrum-preserving token merging. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.16148.

Hongjie Wang, Bhishma Dedhia, 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. URL https://arxiv.org/abs/2305.17328.

Guangzhen Yao, Jiayun Zheng, Zezhou Wang, Wenxin Zhang, Renda Han, Chuangxin Zhao, Zeyu Zhang, and Runhao Liu. V-Pruner: A fast and globally-informed token pruning framework for vision transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(40):34396–34404, 2026. doi: 10.1609/aaai.v40i40.40737. URL https://ojs.aaai.org/index.php/AAAI/article/view/40737.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2001.06782.

A Actor Architecture and Action-Space Details

A.1 Gate, shared encoder, budget head, and selector

Each backbone has an independently trained actor whose parameters are shared across blocks, with block-specific learned embeddings. At block $l \in \{0, \dots, L-1\}$, the actor reads the backbone keys $K_l \in \mathbb{R}^{(N_l+1)\times d}$, with attention-head channels concatenated. Token counts exclude CLS. The history vector is

$$h_l = \left[\frac{N_l}{N_0},\ \rho_l,\ \frac{E_{<l}}{L-1} \right], \tag{7}$$

where $\rho_0 = 0$, $\rho_l = k_{l-1}/N_{l-1}$ for $l > 0$, and $E_{<l}$ counts earlier pruning interventions. We take $k_j = 0$ at gate-off steps.

The gate concatenates $K_{l,\mathrm{CLS}}$, a learned block embedding $e_l^G \in \mathbb{R}^{16}$, and h_l. A $(d+19)\to 64 \to 1$ MLP with a SiLU hidden activation and sigmoid output produces the activation probability p_l^G. The controller runs only when the gate is active.

The shared encoder constructs

$$X_{l,j} = W_{\mathrm{in}} K_{l,j} + b_{\mathrm{in}} + e_l^C + \mathbf{1}[j = \mathrm{CLS}] W_h h_l, \tag{8}$$

where $W_{\mathrm{in}} : d \to 128$, $e_l^C \in \mathbb{R}^{128}$, and $W_h : 3 \to 128$. The gate and controller use separate block embeddings. The controller adds its block embedding to every token and injects history only into CLS. A single pre-LayerNorm Transformer block, with width 128, four attention heads, and a $128 \to 256 \to 128$ GELU feed-forward sublayer, produces Z_l.

The budget head applies LayerNorm to $Z_{l,\mathrm{CLS}}$ followed by a $128 \to 128 \to M$ MLP with GELU, where M is the output dimension in Table 3. Given k_l, the selector applies a shared $128 \to 64 \to 1$ SiLU MLP to $Z_{l,j} + W_k(k_l/N_l)$, where $W_k : 1 \to 128$. The resulting score $s_{l,j}$ indicates deletion preference; higher scores favor removal, and CLS is excluded.

A.2 Backbone-specific budget grids

All L blocks are eligible for pruning. The budget grid is $\mathcal{B} = \{2, 4, \dots, N_0 - 4\}$, with $M = (N_0 - 4)/2$ outputs. At block l, the feasible budgets preserve at least four visual tokens:

$$\mathcal{B}_l = \{k \in \mathcal{B} : k \le N_l - 4\}. \tag{9}$$

The same rule applies at every depth. No pruning is represented by gate-off rather than a zero budget; if $\mathcal{B}_l$ is empty, the gate remains off.

Table 3: Backbone dimensions and initial budget grids. M is the budget-head output dimension. Token counts exclude CLS.

Backbone	d	L	N_0	M	$\mathcal{B}_0$
DeiT-B/16	768	12	196	96	$\{2, 4, \dots, 192\}$
ViT-B/16	768	12	196	96	$\{2, 4, \dots, 192\}$
ViT-L/16	1024	24	196	96	$\{2, 4, \dots, 192\}$
DINOv2-B/14	768	12	256	126	$\{2, 4, \dots, 252\}$

A.3 Training-time sampling and deterministic inference

When $\mathcal{B}_l \neq \varnothing$, training samples $g_l \sim \mathrm{Bernoulli}(p_l^G)$. Conditional on $g_l = 1$, k_l is sampled from the categorical distribution $\pi_B(\cdot \mid o_l)$ over $\mathcal{B}_l$. The selector computes its scores once and samples an ordered sequence $\boldsymbol{i}_l = (i_1, \dots, i_{k_l})$ without replacement. With $R_t = \{1, \dots, N_l\} \setminus \{i_1, \dots, i_{t-1}\}$, the Plackett–Luce log-probability is

$$\log \pi_S(\boldsymbol{i}_l \mid o_l, k_l) = \sum_{t=1}^{k_l} \left[s_{l,i_t} - \log \sum_{j \in R_t} \exp(s_{l,j}) \right]. \tag{10}$$

Although deletion depends only on the selected set, PPO retains the sampled order to evaluate the same action's likelihood under the old and new policies.

At inference, the gate opens when $p_l^G \geq 0.5$ and $\mathcal{B}_l \neq \varnothing$. The budget selects the highest-logit feasible budget, and the selector removes the k_l highest-scoring visual tokens. Deletion occurs after the attention residual and before the MLP, preserving CLS and the relative order of surviving tokens.

B Complete-Prefix Shadow Evaluation and Reward Derivation

B.1 Token-layer compression

Let L be the backbone depth and N_0 the initial number of visual tokens. Token counts exclude CLS, and $k_l = 0$ at gate-off steps. The token counts before and after deletion satisfy

$$N_l = N_0 - \sum_{j=0}^{l-1} k_j, \qquad N_l^{\text{post}} = N_l - k_l. \tag{11}$$

Since deleted tokens remain absent from subsequent blocks, exchanging the order of summation gives

$$\begin{aligned} C &= \frac{1}{LN_0} \sum_{l=0}^{L-1} \big(N_0 - N_l^{\text{post}}\big) \\ &= \frac{1}{LN_0} \sum_{l=0}^{L-1} \sum_{j=0}^{l} k_j = \sum_{l=0}^{L-1} \underbrace{\frac{k_l(L-l)}{LN_0}}_{\Delta C_l}. \end{aligned} \tag{12}$$

The factor $L - l$ counts the post-pruning block outputs in which the deleted tokens are absent. Because deletion follows the attention residual and precedes the MLP, these tokens still participate in current-block attention. Thus C serves as a normalized token-layer training surrogate.

B.2 Shadow evaluation procedure

For each image, let z^N be the logits of the unpruned Native model. A complete-prefix shadow produces $z^{(l)}$ by replaying the sampled rollout's recorded deletions at blocks $0, \ldots, l$ and completing all remaining blocks without further pruning. No actor decisions are resampled. Each shadow uses the same frozen model and preprocessing as the Native reference.

Define $p^N = \mathrm{softmax}(z^N/\tau)$ and $p^{(l)} = \mathrm{softmax}(z^{(l)}/\tau)$. The per-image fidelity cost is

$$D_l = \tau^2 \sum_c p_c^N \log \frac{p_c^N}{p_c^{(l)}}, \qquad \tau = 3. \tag{13}$$

The empty prefix is the Native forward, giving $D_{-1} = 0$. A gate-off step introduces no additional pruning, so $D_l = D_{l-1}$ when $k_l = 0$; the full prefix reproduces the rollout's final prediction. Shadow evaluation and reward construction are gradient-free.

B.3 Telescoping reward derivation

The coefficient a_u remains fixed throughout a rollout and its associated PPO update, with feedback changes applied to subsequent rollouts. The local reward is

$$r_l = 25\Delta C_l - a_u(D_l - D_{l-1}). \tag{14}$$

For the undiscounted objective ($\gamma = 1$), Eq. (12) and cancellation of adjacent fidelity terms yield

$$\begin{aligned} \sum_{l=0}^{L-1} r_l &= 25C - a_u \sum_{l=0}^{L-1} (D_l - D_{l-1}) \\ &= 25C - a_u \big(D_{L-1} - D_{-1}\big) \\ &= 25C - a_u D_{\text{final}}, \end{aligned} \tag{15}$$

where $D_{\text{final}} = D_{L-1}$ is the complete rollout's fidelity cost. The signed increment $D_l - D_{l-1}$ measures the current action's marginal effect on final-prediction fidelity, rather than repeatedly penalizing previously accumulated divergence. No additional terminal reward is added, since the final fidelity cost is already included.

C Closed-Loop Accuracy Feedback

C.1 Feedback split and cadence

We reserve 16,384 ImageNet-1K training images as a feedback split, disjoint from the PPO rollout pool and development split (Appendix E.1). The feedback split is used only to update the fidelity coefficient; development and validation data are not used for feedback.

The QC split contains eight fixed shards of 2,048 images, evaluated in round-robin order. Each rollout contains 256 images, and feedback is evaluated every eight PPO updates. Each check uses one shard and the deterministic actor defined in Appendix A.3, after rollout collection and before the corresponding policy update.

C.2 Coefficient initialization and update

Let $u = 1, 2, \ldots$ index PPO updates, and let a_u be the fidelity coefficient used in update u. All four backbones use $a_1 = 30$. At feedback checks, d_u denotes the current shard's paired Top-1 drop as an accuracy fraction. For target $d^\star$, the update is

$$a_{u+1} = \begin{cases} \mathrm{clip}_{[0,300]}\left(a_u + 5\,\mathrm{clip}_{[-2,2]}\left(\dfrac{d_u - d^\star}{0.01}\right)\right), & u \equiv 0 \pmod 8, \\ a_u, & \text{otherwise.} \end{cases} \tag{16}$$

The default target is $d^\star = 0.01$, corresponding to one percentage point. The target sweep in Appendix F.2 uses $d^\star \in \{0.005, 0.01, 0.015\}$; only the target changes, while the normalization scale remains 0.01 and all other feedback parameters remain fixed.

The coefficient a_u remains fixed throughout rollout collection and the associated PPO update (Appendix B); a_{u+1} is applied from the next rollout onward. The target regulates training but does not guarantee the same accuracy drop on development or validation data.

C.3 Paired accuracy-drop computation

Native and DORA predictions are compared on the same QC images, using the same frozen backbone weights and input preprocessing. For a shard of $M = 2{,}048$ images with labels y_i, let $c_i^N = \mathbf{1}[\hat{y}_i^N = y_i]$ and $c_i^D = \mathbf{1}[\hat{y}_i^D = y_i]$ indicate correct Native and DORA Top-1 predictions. The paired drop is

$$d_u = \frac{1}{M}\sum_{i=1}^{M}(c_i^N - c_i^D) = \mathrm{Acc}_N - \mathrm{Acc}_D, \tag{17}$$

where both accuracies are fractions evaluated on that shard. Equivalently, d_u is the fraction of Native-correct/DORA-wrong predictions minus the fraction of Native-wrong/DORA-correct predictions. It retains its sign, and $100 d_u$ gives the drop in percentage points.

D Privileged Critic and PPO Optimization

D.1 Critic inputs and architecture

The critic is used only during training. At block $l \in \{0, \ldots, L-1\}$, it reads the pre-action residual tokens H_l, the Native logits $z^N \in \mathbb{R}^{1000}$ of the same image, and a compact scalar state. Let $p^N = \mathrm{softmax}(z^N)$, $m^N = p^N_{(1)} - p^N_{(2)}$, and

$$\phi^N = \left[p^N_{(1)},\ m^N,\ \mathcal{H}(p^N),\ \mathbf{1}[m^N \le \tau_b]\right],$$

where $p^N_{(1)}$ and $p^N_{(2)}$ are the two largest Native probabilities and τ_b is a fixed backbone-specific low-margin threshold. The scalar input is

$$q_l = \left[\frac{N_l}{N_0},\ \frac{l}{L-1},\ C_{<l},\ \frac{E_{<l}}{L-1},\ D_{l-1},\ \phi^N,\ \frac{a_u}{300}\right], \tag{18}$$

where $C_{<l} = \sum_{j<l} \Delta C_j$, $E_{<l}$ counts earlier pruning interventions, and $D_{-1} = 0$ as defined in Appendix B.

The token branch projects H_l to width 256, adds learned block and CLS/visual-type embeddings, prepends a learned value token, and processes the sequence with two pre-LayerNorm Transformer blocks (width 256, four heads, feed-forward width 512). The value-token representation is concatenated with separate 64-dimensional projections of q_l and z^N, yielding $u_l \in \mathbb{R}^{384}$.

Independent $384 \to 128 \to 1$ MLPs predict the gate and budget values V_l^G and V_l^B. The selector value additionally conditions on the chosen budget:

$$V_l^S = f_S([u_l;\, e(k_l);\, k_l/N_l]),$$

where $e(k_l) \in \mathbb{R}^{32}$ is a learned budget embedding and f_S is a $417 \to 128 \to 1$ MLP. No value head observes selected token identities, post-action states, or ground-truth labels.

D.2 Value targets and PPO optimization

For each L-step trajectory, we use GAE with $\gamma = 1$ and $\lambda_{\mathrm{GAE}} = 0.95$. A common λ-return is constructed from the pre-update gate values:

$$\begin{aligned} \delta_l &= r_l + V_{l+1}^{G,\mathrm{old}} - V_l^{G,\mathrm{old}}, \\ A_l^G &= \delta_l + \lambda_{\mathrm{GAE}} A_{l+1}^G, \\ \widehat{R}_l^\lambda &= A_l^G + V_l^{G,\mathrm{old}}, \\ A_l^X &= \widehat{R}_l^\lambda - V_l^{X,\mathrm{old}}, \qquad X \in \{G, B, S\}, \end{aligned} \tag{19}$$

with $V_L^{G,\mathrm{old}} = A_L^G = 0$. Thus all three decision types share the same return while using their own value baselines.

Gate, budget, and selector are optimized with separate clipped PPO objectives with clip parameter $\epsilon = 0.2$. Their advantages are standardized separately over the corresponding eligible rollout transitions. Gate decisions require a nonempty feasible budget set, while budget and selector objectives are defined only for gate-on transitions. All steps contribute to GAE and V_G regression; V_B and V_S are trained only on gate-on transitions.

The three value heads regress the shared λ-return using running-return normalization and a PPO-style clipped Huber loss. Normalization is used only for value regression and does not alter the rewards or GAE computation.

D.3 PCGrad

Budget and selector share the controller encoder and can induce conflicting gradients. Let g_B and g_S denote the gradients of their respective PPO losses, including entropy regularization, with respect to the shared controller parameters, and define $c = \min(g_B^\top g_S, 0)$. We use symmetric PCGrad:

$$\widetilde{g}_B = g_B - \frac{c\, g_S}{\|g_S\|_2^2 + 10^{-12}}, \qquad \widetilde{g}_S = g_S - \frac{c\, g_B}{\|g_B\|_2^2 + 10^{-12}}. \tag{20}$$

The projected gradients are summed on the shared controller, while the budget and selector heads retain their original private gradients. The resulting controller gradient is clipped to norm 0.5 before optimization. Gate and critic updates do not use PCGrad.

D.4 Training hyperparameters

The optimization settings in Table 4 are shared across all four backbones.

Table 4: Shared optimization settings for DORA training.

Setting	Value
Rollout / total training	256 images / 299,776 images (1,171 updates)
Actor / critic epochs	4 / 4 per update
Gate / controller / critic minibatch	512 / 64 / 128 transitions
Optimizer / gradient clipping	Separate Adam optimizers / max norm 0.5
Gate / critic learning rate	5×10^{-5} / 10^{-4}
Controller learning rates	Shared: 2×10^{-5}; budget: 5×10^{-5}; selector: 2×10^{-5}
Learning-rate schedule / main-run seed	Constant, no warm-up / 42

E Experimental Protocol and Implementation Details

E.1 Data splits

We use mutually disjoint ImageNet-1K training subsets of 299,776, 16,384, and 10,240 images for PPO rollouts, accuracy feedback (Appendix C), and checkpoint selection, respectively. Final accuracy is evaluated on all 50,000 ImageNet-1K validation images and all 7,500 ImageNet-A images. ImageNet-A predictions are restricted to its official 200-class mapping.

All splits use deterministic bicubic resizing and 224×224 center cropping, without stochastic augmentation. The resize short side is 248 for ViT-L/16 and 256 otherwise. ViT-B/16 and ViT-L/16 use RGB mean and standard deviation $(0.5, 0.5, 0.5)$; DeiT-B/16 and DINOv2-B/14 use mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$.

E.2 Checkpoint selection

Let $\mathcal{C}$ be the checkpoints saved during training. The default checkpoint minimizes development-set GFLOPs subject to a Native-relative Top-1 drop of at most one percentage point:

$$c^\star \in \operatorname*{arg\,min}_{c \in \mathcal{C}:\, A_{\mathrm{N}}^{\mathrm{dev}} - A_c^{\mathrm{dev}} \le 1.0} F_c^{\mathrm{dev}}, \tag{21}$$

where accuracies are percentages and F_c^{dev} is mean GFLOPs per image. Neither validation accuracy nor throughput is used for DORA checkpoint selection. The selected checkpoint transfers to ImageNet-A unchanged. If an ablation has no feasible checkpoint, its final checkpoint is reported as constraint-violating.

Baseline operating points are matched separately on each dataset: we select the measured point closest to DORA's Top-1 within 0.1 percentage points, without interpolation. Missing matches are omitted. Baseline means are unweighted over available matching methods, excluding Native and DORA.

E.3 Baseline implementations

All methods share the frozen backbone weights, classifier, and preprocessing within each backbone setting. We use non-distilled DeiT-B/16, AugReg ViT-B/16 and ViT-L/16, and register-free DINOv2-B/14 with its official LC1 readout. The latter preserves LayerScale and concatenates normalized CLS with the mean of normalized surviving visual tokens.

Zero-TPrune is a paper-based reimplementation. DiffRate uses author schedules as candidates, including ViT-L-MAE schedules transferred to ViT-L and DeiT-B schedules scaled by $256/196$ for DINOv2 token counts. All candidate GFLOPs are recomputed under the common accounting below. V-Pruner's offline mask search is repeated independently for each backbone and budget; its exported mask is fixed across images and applied before the first block. Offline search is excluded from deployment measurements.

E.4 GFLOPs ACCOUNTING

We count one multiply–accumulate as two FLOPs and include the complete online forward: patch embedding, backbone, classifier, and each method's token-reduction computations, including DORA's actor. With F_i denoting an image's floating-point operation count, the dataset average is

$$F_{\mathcal{D}} = \frac{1}{10^9|\mathcal{D}|}\sum_{i\in\mathcal{D}} F_i \quad \text{GFLOPs/image.} \tag{22}$$

Counts follow each image's realized token trajectory before averaging. At a DORA intervention block, attention uses $N_l + 1$ tokens and the MLP uses $N_l - k_l + 1$, including CLS. Non-floating-point selection and memory operations contribute to runtime rather than GFLOPs. Training-only critic and shadow computations are excluded.

E.5 RUNTIME AND THROUGHPUT MEASUREMENT

Timing uses the same RTX 4090, PyTorch 2.11, and CUDA 12.6, with batch size 320 and BF16 matrix operations. Score comparisons use FP32; TF32 and `torch.compile` are disabled. In our evaluated implementations, DORA uses packed variable-length FlashAttention, while Native, PiToMe, Representation Shift, and V-Pruner also use FlashAttention. Zero-TPrune uses dense attention in blocks requiring explicit attention graphs and FlashAttention in the remaining blocks. ToMe, MCTF, and DiffRate retain dense attention to preserve their proportional-attention and token-size weighting operations.

ImageNet-1K timing uses 8,000 GPU-resident, preprocessed validation images in 25 batches. Each of five repeats cycles through these batches for 200 warm-up and 200 timed forwards, with CUDA synchronization immediately before and after the timed loop. For elapsed wall time T_r in seconds, repeat r yields

$$t_r = \frac{1000T_r}{200}\ \text{ms/batch}, \qquad q_r = \frac{320\cdot 200}{T_r}\ \text{images/s.} \tag{23}$$

We report the median latency and throughput across repeats.

For ImageNet-A, we use a fixed random subset of 7,360 images, arranged into 23 full batches of 320 without padding. The same GPU-resident procedure, warm-up, repeats, and aggregation apply.

Timed forwards include the model, online token selection, merging/pruning, packing, and compaction. DORA recomputes its decisions on every forward. Image decoding, preprocessing, host-to-device transfer, and offline training or search are excluded.

F ADDITIONAL ABLATIONS AND ANALYSES

F.1 SELECTOR SUBSTITUTION STUDY

We fix DORA's per-image gate decisions and deletion budgets and replace only token selection. *Random* samples k_l visual tokens uniformly; *Similarity* removes the k_l most redundant tokens using a ToMe-style key-similarity rule (Bolya et al., 2023); and *Attention-WPR* removes the k_l lowest-importance tokens under a Zero-TPrune-style attention graph (Wang et al., 2024). Each substitute uses its own evolving pruned representation while following the recorded gate and budget decisions. These are component substitutions, not evaluations of the complete baseline methods.

Table 5: Selector substitution on ImageNet-1K (Top-1, %). Random reports mean ± standard deviation over five sampling seeds. Values are rounded to one decimal place.

Backbone	DORA selector	Random	Similarity	Attention-WPR
DeiT-B/16	**80.8**	79.1 ± 0.1	80.5	77.3
ViT-B/16	**83.8**	82.0 ± 0.2	83.1	81.6
ViT-L/16	**85.1**	83.3 ± 0.0	84.2	84.2
DINOv2-B/14	**83.7**	82.1 ± 0.1	83.3	82.2

The learned selector preserves higher Top-1 accuracy than all three substitutes on every backbone (Table 5). This result is conditional on DORA's learned gate and budget decisions.

F.2 ACCURACY-FEEDBACK TARGET SWEEP

We train DORA on DeiT-B with feedback targets $d^\star \in \{0.005, 0.01, 0.015\}$, corresponding to accuracy drops of 0.5, 1.0, and 1.5 percentage points. Only $d^\star$ changes: the feedback normalization scale remains 0.01 and all other training settings are fixed (Appendices A–D). Each operating point comes from a separate training run.

For each run, we select the lowest-GFLOPs checkpoint satisfying the corresponding Native-relative development-set accuracy constraint, following Appendix E.2. Baselines are matched to each DORA point's realized validation accuracy within 0.1 percentage points. The resulting operating points are shown in Figure 5 (Section 4.4).

F.3 FOUR-SEED STABILITY

We repeat DeiT-B training with three additional seeds, keeping the data and optimization protocol fixed. Each run's checkpoint is selected independently under the 1-pp development-set constraint in Eq. (21).

Table 6: Four-seed DeiT-B results on ImageNet-1K validation. Checkpoints are selected on the development split. The final row reports mean $\pm$ sample standard deviation across the four training runs.

Run	Top-1 (%)	GFLOPs/image	Native drop (pp)
Original seed	80.79	21.66	1.00
Additional seed 1	80.83	21.90	0.96
Additional seed 2	80.75	21.57	1.04
Additional seed 3	80.90	21.93	0.89
Mean $\pm$ std.	80.82 ± 0.07	21.76 ± 0.18	0.97 ± 0.07

The selected operating points remain closely clustered across the four runs (Table 6), with $80.82 \pm 0.07\%$ Top-1 and 21.76 ± 0.18 GFLOPs per image. The development-set constraint is applied independently to each run; the reported Native drops are measured on ImageNet-1K validation.