LATER IS BETTER: TOKEN REDUCTION FOR VITS UNDER DISTRIBUTION SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Training-free token reduction accelerates vision transformers by removing redundant tokens across layers, recovering most of the original accuracy at a fraction of the compute. These methods, however, are designed and evaluated primarily on clean data, and under real-world distribution shift their accuracy gap to the uncompressed model widens with the removal rate. We show that this gap is governed by the reduction *schedule*, the depth profile of removal, usually left fixed as an implementation detail. Concretely, we introduce a one-parameter *late-concentrated* power-law schedule that consistently improves out-of-distribution accuracy over flat at no extra inference cost. On ImageNet-C with DeiT-S, the late schedule closes 83% of that gap at a 26% compute reduction ($+1.17$pp), and 99% of it at a lighter 7% reduction ($+0.26$pp). The gain cannot be attributed to retaining more tokens or using extra compute: held to flat's compute, the late schedule removes more tokens in total and leaves fewer tokens at the end, yet still wins. Single-layer probes point to a mechanism: earlier reductions perturb features that pass through more remaining layers, front-loading reduction error in depth. The effect is broad, holding across five token-reduction methods (ToMe, EViT, ATS, ATC, PiToMe), nine backbones, all ImageNet-C corruption types, eight further shift suites, and two further modalities, video and vision-language QA. It is also specific to shift, still positive on clean and rising monotonically to $\sim 4\times$ that at the highest severity 5. The schedule keeps its gain under six test-time adaptation methods, and needs no per-input or per-domain tuning.

1 INTRODUCTION

Vision transformers (ViTs) (Dosovitskiy et al., 2021) split an image into patches, embed each as a token, and process the resulting sequence through a stack of transformer layers. Their compute scales with the number of tokens, steeply so in self-attention, whose cost is quadratic in the sequence length. Because many of these tokens are redundant, token-reduction methods cut this cost by removing tokens as the image passes through the network, shrinking the sequence the deeper layers must process: pruning unimportant tokens (Liang et al., 2022), merging similar ones (Bolya et al., 2023; Tran et al., 2024), clustering them into groups (Haurum et al., 2024), or sampling a subset to keep (Fayyaz et al., 2022), all recovering most of the original accuracy at a fraction of the compute. Because they require no retraining and act at test time, with no extra data, they are widely studied as plug-in efficiency tools for existing ViT pipelines.

These methods, however, are designed and evaluated primarily on clean, in-distribution images, while the deployments they target rarely see such inputs. Sensor noise, motion blur, weather, and compression artifacts produce a distribution shift away from the training data (Hendrycks & Dietterich, 2019; Mintun et al., 2021; Kar et al., 2022). Whether token reduction, tuned for clean efficiency, retains its accuracy under such real-world shift is largely unexamined. Figure 1a,b answers this for the widely used ToMe (Bolya et al., 2023): the more aggressive the reduction, the more accuracy it sacrifices under shift, while clean accuracy is barely touched.

The remedy lies in a choice these methods leave implicit. A token-reduction method is defined not only by how many tokens it removes, but by *where* along the network's depth it removes them, that is, its *schedule*. Existing methods largely use a flat schedule that removes the same number at every layer (Bolya et al., 2023; Haurum et al., 2024), treated as a fixed implementation detail,

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.

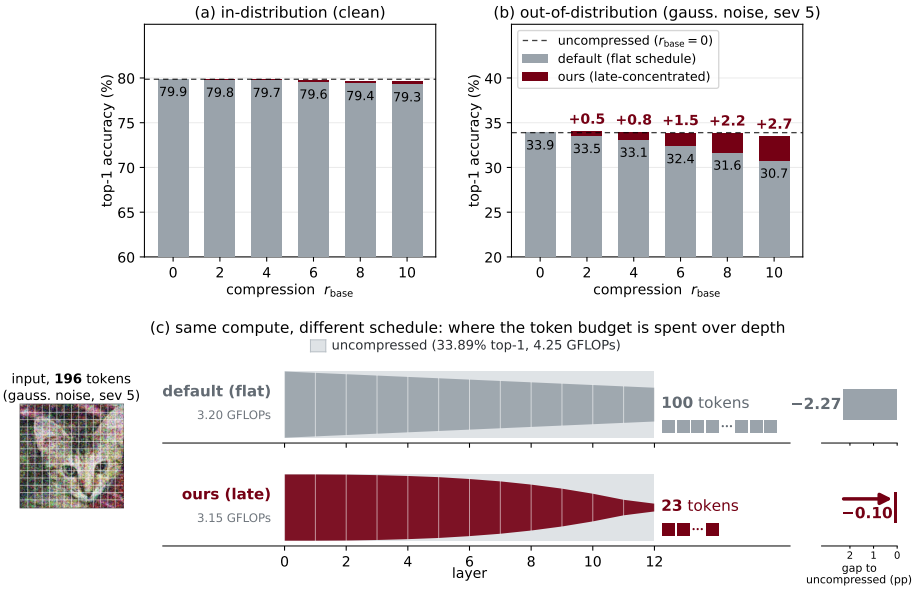


Figure 1: Token reduction increasingly sacrifices out-of-distribution accuracy; a *late-concentrated schedule* recovers it at *no extra compute*. **(a, b)** DeiT-S top-1 vs. the per-layer token removal rate r_{base} on clean and gaussian-noise ImageNet-C. **(c)** each ribbon's height is the tokens remaining, so the pale band above is what was removed; flat removes the same number at every layer, ours follows the late-concentrated power law. Ours ends with far fewer tokens, yet closer to uncompressed.

with tuning effort targeting the per-layer rate or input-adaptive rules rather than the depth profile of removal itself. The deployment shift is unknown in advance and unlabeled at test time. We therefore ask whether, in this training-free regime, the schedule is itself a lever for robustness at matched inference cost.

We find that it is, and that concentrating removal *late* is the direction that helps, while the mirror-image early-concentrated schedule is the most damaging of the three. The probes in Section 3 point to amplification: a token-reduction step perturbs the features it acts on, and the earlier it acts, the more layers amplify that perturbation before it reaches the classifier. Removing later therefore distorts the output less, and shift widens the gap by raising that amplification. Concretely, we propose a *late-concentrated* power-law schedule, a per-layer removal rate that increases with depth (Eq. 1, Figure 1c). We use a single fixed exponent at our main operating point, with no per-input or per-domain tuning. On DeiT-S and ImageNet-C this recovers 83% of the accuracy that token reduction loses under shift at a 26% compute reduction ($+1.17$pp over flat), and 99% of it at a lighter 7% reduction ($+0.26$pp). It also removes more tokens in total than flat, yet still improves accuracy, so neither a larger token budget nor extra compute can account for the gain.

The benefit holds across five training-free reducers and nine backbones from four pretraining recipes, all ImageNet-C corruption types, eight further shift suites (ImageNet-A/R/V2/Sketch, DomainNet-126, ImageNet-$\overline{\text{C}}$/3DCC, NaBirds-C), and two modalities beyond image classification, video on Kinetics and vision-language QA on GQA-C. The schedule also composes with six test-time adaptation methods, which adapt the model to the same shift. And the gain rises monotonically with corruption severity, $\sim 4\times$ from a small $+0.28$pp on clean images to $+1.17$pp at severity 5, so it is a robustness property rather than a general accuracy improvement.

We make the following contributions:

- We identify the reduction schedule (the depth profile of removal) as an under-characterized determinant of ViT robustness to distribution shift, and show that at matched compute a late-concentrated power law recovers most of the accuracy that reduction costs under shift.
- We provide a first-order diagnostic account of the gain by measuring per-layer distortion and perturbation amplification, finding both to be front-loaded in depth.
- We show the effect is broad, holding across five training-free reducers, nine backbones, all ImageNet-C corruptions, eight further shift suites, two other modalities, growing with shift severity, and composing with six test-time adaptation methods.

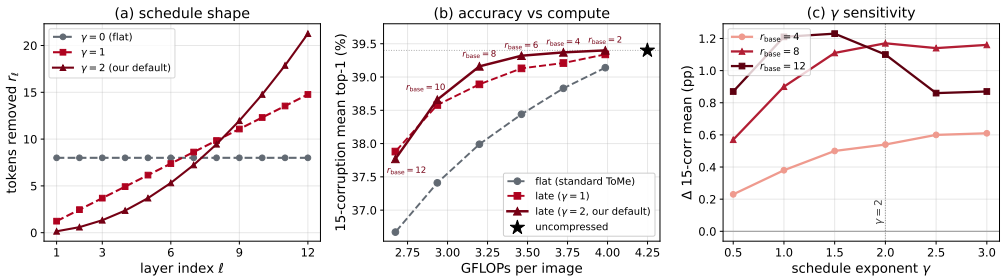


Figure 2: The late-concentrated schedule and its effect (DeiT-S, ImageNet-C severity 5). **(a)** schedule shape: tokens removed per layer $r_\ell \propto (\ell/L)^\gamma$ vs. layer index (at r_{base}=8), for γ=0 (flat), 1, and our default 2; larger γ defers removal to later layers. **(b)** 15-corruption-mean top-1 vs. compute (GFLOPs), each curve a schedule swept over compression (points labeled by r_{base}); $\star$ marks the uncompressed model. **(c)** gain over flat vs. γ, one curve per compression level $r_{\text{base}} \in \{4, 8, 12\}$.

2 Method: Late-Concentrated Schedule

Power-law schedule. A reduction schedule sets a removal count at each of the network's L layers. We seek a one-parameter family that spans schedules from flat to strongly late-concentrated and whose compute cost is analytic in that parameter, so a schedule can be matched to a baseline's compute without search. We therefore propose a power-law schedule over normalized depth,

$$r(\ell) = R \cdot \frac{(\ell/L)^\gamma}{\sum_{k=1}^{L} (k/L)^\gamma}, \qquad \ell = 1, \ldots, L, \tag{1}$$

where R is the total number of tokens removed across the network and the exponent γ is the only parameter our method sets. We use a power law because it recovers the flat schedule as the special case γ=0 (a uniform R/L removal at every layer, the default across training-free token reducers (Bolya et al., 2023; Liang et al., 2022; Fayyaz et al., 2022; Haurum et al., 2024)), stays monotone non-decreasing in depth for every γ>0 (no layer removes more than a later one), and redistributes the same budget R continuously toward the later layers as γ grows (Figure 2a). It parameterizes late-concentration rather than asserting an optimal schedule; likewise, γ is set to a robust default (γ=2), not tuned per task (Figure 2c). The finding is the late-concentrated *direction* and its intensity (Figure 6), and we expect other monotone late-weighted profiles to behave similarly; a check at the main operating point supports this, with an exponential profile and a non-smooth two-level step (each at iso-GFLOPs) recovering gains comparable to the power law (Appendix P). We round the schedule to integer per-layer counts and apply it identically across all five reducers, which then differ only in which tokens they select (rounding rule and per-layer cap in Appendix B).

Calibration to the flat baseline's compute (iso-GFLOPs). We label each operating point by r_{base}, the per-layer rate of the flat baseline it is matched against and the deployment choice of how much compute to save (R=$r_{\text{base}}L$ for the flat baseline only); the late schedule's own budget is set by the calibration below and is larger, with per-layer mean $\bar{r}$=$R/L > r_{\text{base}}$ (Appendix V). Later-layer removals save less compute (fewer tokens cascade through fewer remaining layers). At the same r_{base}, late-concentrated uses fewer GFLOPs (giga floating-point operations), permitting more (later) token removal to restore GFLOPs parity. All comparisons in this paper are at fair GFLOPs, not equal r_{base}. These iso-GFLOPs comparisons are also matched in wall-clock: at the main operating point the late (γ=2) and flat schedules differ by under 0.5% in measured throughput, so the comparison is fair in real latency and not only in analytic FLOPs (Appendix W).

We calibrate each $(r_{\text{base}}, \gamma)$ by choosing the integer per-layer removal budget whose GFLOPs match the corresponding flat baseline; because GFLOPs is analytic in the schedule, this calibration needs no extra evaluation. When an exact match is not attainable at integer budgets, we always round the budget so that the late-concentrated schedule uses no more GFLOPs than its flat baseline (the full list of calibrated schedules is given in Appendix V, Table 31). The late-versus-flat comparison is therefore strictly conservative, and the gains cannot be attributed to a larger compute budget. Figure 2b plots the resulting accuracy–compute frontier on the 15-corruption mean: the calibrated late-concentrated family Pareto-dominates the flat one across the entire compute range.

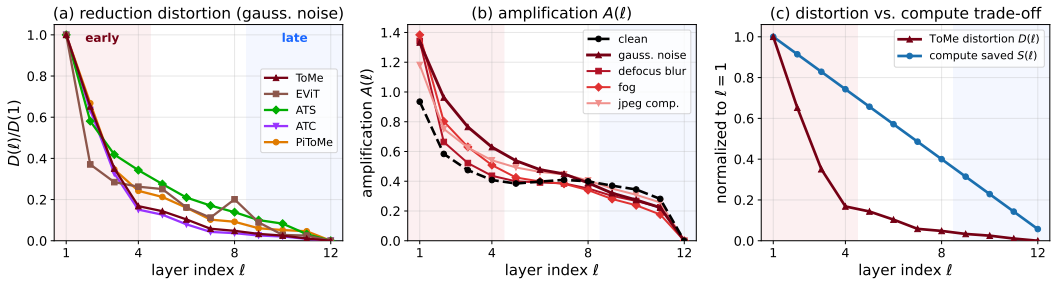


Figure 3: Per-layer reduction distortion and amplification (DeiT-S, ImageNet-C severity 5; means over $N{=}1024$ images, standard error $<2\%$). **(a)** The directly-measured distortion $D(\ell)$ of reducing at layer ℓ only (relative output-logit change, gaussian noise), normalized to layer 1, for all five reducers. **(b)** The amplification $A(\ell)$ (clean and one corruption per ImageNet-C group). **(c)** $D(\ell)$ against the compute a removal at layer ℓ saves, $S(\ell)$, both normalized; D falls far faster than S.

3 Why Late Concentration Improves Robustness

We provide a first-order account for why late concentration is preferred in our measurements under distribution shift, and why the empirical optimum is interior rather than maximally late. (i) A reduction's output distortion is front-loaded in depth, because earlier perturbations pass through more downstream layers. (ii) The compute saved by a reduction also falls with depth, but more slowly, so moving budget later lowers distortion per unit of compute saved. (iii) Per-layer capacity keeps the empirical optimum interior, because each layer can merge only a bounded number of similar tokens.

The reduction distortion is front-loaded. A reduction step (merging, pruning, clustering, or sampling) removes or distorts token information at the layer where it acts, and the perturbation propagates forward to the classification readout; we call the resulting output error the distortion of reducing at layer ℓ. We measure it directly and reducer-agnostically. For each training-free reducer we remove a fixed number of tokens at layer ℓ and nowhere else, so per-layer differences reflect depth alone, and read the relative logit change $D(\ell) = \mathbb{E}_x \, \|f^{(\ell)}(x) - f(x)\| / \|f(x)\|$ (f unreduced, $f^{(\ell)}$ reduced at ℓ only), a removal-based sensitivity measure in the tradition of ablation-importance criteria (LeCun et al., 1989; Michel et al., 2019; Lee & Choi, 2025). For all five reducers (ToMe, EViT, ATS, ATC, PiToMe) $D(\ell)$ is steeply front-loaded (Figure 3a), and it stays front-loaded under every corruption, with the early/late distortion ratio exceeding $6\times$ in all 20 reducer$\times$corruption cases (Spearman ≤ -0.90; per-cell values in Appendix M). The distortion is near zero at the last layer ($D(\ell{=}12) \approx 0$). Under the class-token readout a final-layer patch merge has no later layer through which to reach the class token, so it costs almost nothing, and the gain does not depend on this (Appendix T). The account rests on single-layer probes, whereas a deployed schedule composes many reductions whose effects interact, each merge changing what later ones see. It still holds for the composed schedules: at equal compute the late schedule's mean per-case KL to the unreduced model is 0.029 against 0.085 for flat, and it changes the unreduced model's top-1 on 10.8% of cases against 19.1%, lower on all 15 corruptions and both backbones (Appendix U, Figure 9).

Perturbation amplification induces the front-loading. The distortion is front-loaded because the network amplifies an early perturbation more: it passes through more downstream layers on its way to the classifier. We probe this gain directly. At layer ℓ we perturb each output patch token by $\delta_t = \alpha \, \|x_t\| \, u_t$, with u_t an isotropic unit vector and $\alpha{=}0.1$ (the class token is left untouched), let it propagate to the end, and read the pre-logit class-token representation z. The amplification $A(\ell) = \mathbb{E}\big[\|z_{\text{pert}} - z_{\text{clean}}\| / \|z_{\text{clean}}\|\big]/\alpha$, averaged over 5 draws shared across layers, is a Jacobian-norm sensitivity probe (Novak et al., 2018) anchored at a hidden layer. Linearizing the downstream layers (Lemma 1, Appendix N) sends a perturbation at layer ℓ to the readout through the composed Jacobian $\prod_{k>\ell} J_k$; the isotropic probe measures the average-case gain of that product, and its operator norm bounds that gain from above. Each deeper layer drops one factor, so the gain is non-increasing in ℓ whenever those layers are on average non-contractive, the familiar depth-compounding of perturbations in deep networks (Poole et al., 2016). Amplification is a property of the network, not of any reducer's choices, consistent with $D(\ell)$ being front-loaded for all five. Measured on clean inputs and one corruption from each ImageNet-C group (severity 5), $A(\ell)$ is front-loaded in every condition (Figure 3b; Spearman ≤ -0.92), with clean the flattest profile, so corruption steepens the amplification.

Capacity keeps the optimum interior, not maximally late. What a schedule spends is compute, not token count, so the governing quantity is not the raw distortion but the distortion per unit of compute saved, $\rho(\ell) = D(\ell)/S(\ell)$, where $S(\ell)$ is the marginal compute a removal at layer ℓ saves. ToMe, our default reducer, merges between attention and the MLP, so a removal saves that layer's own MLP and all downstream layers: $S(\ell) = c + (L - \ell)$ in units of a full layer, with $c \approx 0.61$ the own-MLP fraction and $S(L) = c > 0$. A THOP measurement confirms this near-linear profile, the last layer saving only $\approx 6\%$ of what the first does. Figure 3c contrasts them: D falls far faster than S, so late reductions are cheap in distortion yet save little compute, and ρ falls with depth. That is why late concentration pays at equal compute. The ratio alone does not bound how late: since $A(L) = 0$ the last layer reduces at near-zero distortion, $\rho(L) \to 0$, and unconstrained ρ-minimization would pile the budget onto the latest layers. We attribute the *interior* optimum to per-layer capacity instead. Each layer can merge only a bounded number of similar tokens, so the budget must spill beyond the few cheap late layers, and as a layer nears that limit the marginal distortion of each further merge rises, favoring a smooth spread over a few saturated layers. The account is first-order and does not pinpoint an optimal γ, but its predictions hold: the interior inverted-U in γ (Table 1, Figure 2c), its migration toward weaker concentration at larger removal budgets (Appendix S), and the late-concentrated, last-layer-heavy shape a schedule searched for clean accuracy takes (Appendix Q).

Why the advantage grows under shift. The gain grows monotonically with corruption severity (Section 4.2); this is an empirical property, which the probes explain in direction, not magnitude. Token reduction sacrifices more accuracy under a stronger shift (Figure 1), so a better allocation of the same budget has more to recover. The representational reason is that the network amplifies a reduction's perturbation more off the clean manifold. Self-attention is non-Lipschitz (Kim et al., 2021) and can develop larger Jacobians off-distribution (Appendix N); the amplification $A(\ell)$ rises under every corruption group (early $A(\ell)$ from 0.60 on clean to 0.74–0.92) and, compounding through more downstream layers, does so most in the early layers (Lemma 1). The early reductions the flat schedule spends its budget on are therefore the most amplified under shift, so the late schedule recovers more. We do not read the probes as predictors of the per-corruption gain, whose magnitude also reflects how close the corrupted inputs sit to the decision boundary.

The gain is not a readout or insertion-point artifact. The late advantage survives dropping the final layers from both schedules, and it reproduces under global-average-pooling readout, which removes the free last layer but leaves the front-loaded distortion intact: on MAE ViT-B and ViT-L every one of the 15 corruptions improves ($+1.28$ and $+1.22$). The strictly free final-layer saving is in any case only 0.2–0.7% of total GFLOPs (Appendix T).

Allocation outweighs the matching criterion. Our account concerns where tokens are removed; a distinct possible source of the OOD penalty is which tokens are merged, since corrupted similarities could worsen merge decisions (the matching criterion). In an oracle control, for each corrupted image we force the per-layer merge decisions its clean counterpart would make (captured offline, unavailable at test time), holding the schedule and the compute fixed. Even the flat schedule with these oracle decisions recovers less OOD accuracy than the late schedule at every operating point (Figure 4, gaussian noise; full ImageNet-C in Appendix O). Allocation therefore outweighs the gain from this oracle's improved matching; a genuinely corruption-robust criterion, which this oracle does not upper-bound, remains a complementary axis (Section 5).

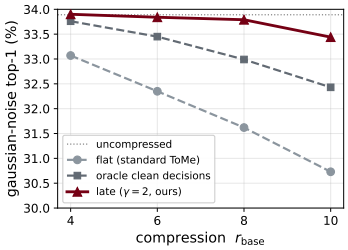


Figure 4: Oracle clean merge decisions recover less than the late schedule at every r_{base}, the gap widening with compression.

4 Experiments

Setup. We evaluate token-reduction schedules across image-classification backbones (DeiT-S/B (Touvron et al., 2021), ViT-S/B/L (Dosovitskiy et al., 2021), CLIP ViT-B/L (Radford et al., 2021), and MAE ViT-B/L (He et al., 2022)), a video model (VideoMAE ViT-B (Tong et al., 2022)), and a vision-language model (LLaVA-1.5-7B (Liu et al., 2024)). The schedule is applied to five training-free reducers; ToMe (Bolya et al., 2023) is the default because it is widely used and merges

Table 1: γ **sensitivity across operating points at iso-GFLOPs** (DeiT-S, ImageNet-C severity 5, 15-corruption mean top-1, %). The GFLOPs column is the default γ=2 schedule's compute and its reduction against the uncompressed model (39.40 at 4.25G; per-γ schedules in Table 31). flat is the const-r_{base} baseline in absolute top-1; the γ columns are Δ = late − flat. Shaded = the default γ=2; **bold** = main result (r_{base}=8); underline = per-row optimum $\gamma^\star$.

			late-concentrated γ (Δ vs. flat)					
r_{base}	GFLOPs (cut)	flat	0.5	1.0	1.5	2.0	2.5	3.0
2	3.97G (−7%)	39.14	+0.09	+0.20	+0.22	+0.26	+0.27	+0.28
4	3.69G (−13%)	38.83	+0.23	+0.38	+0.50	+0.54	+0.61	+0.60
6	3.34G (−21%)	38.44	+0.34	+0.69	+0.81	+0.88	+0.83	+0.69
8	3.15G (−26%)	37.99	+0.57	+0.90	+1.11	**+1.17**	+1.14	+1.16
10	2.90G (−32%)	37.41	+0.67	+1.17	+1.28	+1.25	+1.15	+0.92
12	2.67G (−37%)	36.67	+0.87	+1.21	+1.23	+1.10	+0.86	+0.87

tokens rather than discarding them, while EViT, ATS, ATC, and PiToMe verify the effect is not specific to merging. Our primary benchmark is ImageNet-C at severity 5, and we test generalization on ImageNet-$\overline{\text{C}}$/3DCC, ImageNet-R, ImageNet-Sketch, NaBirds-C, Mini-Kinetics-C, and GQA-C. To probe composability with test-time adaptation we run Tent, EATA, SAR, FOA, SPA, and NEO on top of the schedule. Comparisons are inference-only (and deterministic) unless a TTA method is stated; we report the severity-5 15-corruption mean top-1 unless noted. Our main operating point is r_{base}=8 with γ=2 (a 26% compute reduction); we sweep compression from r_{base}=2 to 12 (Table 1), with γ fixed at 2 rather than tuned per rate. Full implementation details are in Appendix B.

4.1 Late-Concentrated vs. Flat at iso-GFLOPs

The late-concentrated schedule at r_{base}=8, γ=2 improves the corrupted mean by +1.17pp over flat (Table 1). That closes 83% of the accuracy compression costs, leaving a −0.24pp gap to the uncompressed model against −1.41pp for flat, at a ∼26% compute reduction. The gain is recovered predictions, not a compensating mix: flat destroys 8.5% of what the unreduced model gets right and ours 3.9% (Appendix U). The gain grows with compression, from +0.26pp at the lightest r_{base}=2 to +1.25pp at r_{base}=10, and holds at +1.10pp at the aggressive r_{base}=12 (Figure 2b). The γ sweep is an inverted-U, positive for every tested $\gamma \in [0.5, 3.0]$ ($\geq$ +0.57pp), with the optimum migrating from $\gamma^\star \approx 3$ at r_{base}=2 to ≈ 1.5 at r_{base}=12; γ=2 stays within 0.13pp of the per-rate optimum. All 15 corruptions improve (Figure 5; +0.21pp on elastic transform to +2.45pp on impulse noise), as do the four outside ImageNet-C's standard set (Appendix J); a sign test rejects the no-effect null at $p < 10^{-4}$. Depth allocation is a deployment-neutral lever: it requires no retraining or extra compute, and the gain holds for every tested γ and every corruption rather than at a single tuned setting.

Early vs. flat vs. late schedule. To test the schedule's directionality, we evaluate the symmetric early-concentrated schedule (the same power law reflected to remove more early). The result is a monotone ordering, early < flat < late, at both operating points (Figure 6). Concentrating removal

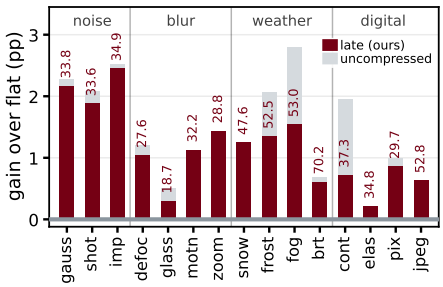


Figure 5: Per-corruption gain over flat at iso-GFLOPs (DeiT-S, r_{base}=8, γ=2, ImageNet-C sev=5); uncompressed in gray. All 15 corruptions improve; labels are late's top-1.

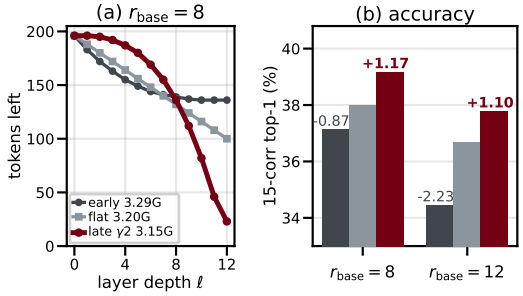


Figure 6: Schedule placement (DeiT-S, ImageNet-C sev=5). **(a)** tokens left per layer at r_{base}=8, with each schedule's GFLOPs; the r_{base}=12 profiles are in Appendix R. **(b)** top-1 at both rates, Δ vs. flat.

early hurts (-2.23 to -0.87pp vs. flat), late helps ($+1.10$ to $+1.17$pp). This is the ordering predicted by the front-loaded per-step distortion (Section 3). Early reductions are the most damaging, so concentrating removal early is the worst allocation. Crucially the early deficit is attributable to the schedule, not compute or token budget. We calibrate the early-concentrated budget to flat's compute; it comes out slightly higher: 2.8% more GFLOPs at r_{base}=8 and 1.3% more at r_{base}=12. The early schedule also removes fewer tokens in total (60 vs. 96 at r_{base}=8). More compute and less removal should both help, yet early is the worst of the three, losing to flat by 0.87–2.23pp, while the late schedule uses the least compute and wins. Thus the ordering cannot be explained by the simple resource variables of retained token count or compute alone; the depth allocation of removal is the decisive factor in these comparisons.

4.2 The Gain Across Shift Severities

The improvement grows with the shift, rising monotonically with corruption severity (Table 2). At late γ=2 it rises from $+0.28$pp on clean images to $+1.17$pp at severity 5, so the corrupted gain is $4.2\times$ the clean gain. The gain is therefore a robustness effect, not a general accuracy improvement. The γ=1 schedule traces the same monotone profile (Appendix C). The nonzero clean gain is not a separate effect: the per-step reduction distortion $D(\ell)$ is front-loaded even on clean inputs (Section 3), so deferring removal helps by a small margin with no corruption, and shift inflates the early-layer distortion further, widening that same margin. The clean and corrupted gains are thus two readings of one allocation principle, the clean gain being the floor that shift steepens.

Table 2: Per-severity 15-corruption mean (%), default γ=2, r_{base}=8, DeiT-S; severity 0 is clean ImageNet val. Δ = late − flat, Δ/Δ_0 its multiple of the clean gain.

Sev.	flat	late	Δ	Δ/Δ_0
0 (clean)	79.40	79.68	**+0.28**	1.0×
1	70.52	71.04	**+0.52**	1.9×
2	64.33	64.99	**+0.66**	2.4×
3	59.06	59.89	**+0.83**	3.0×
4	50.06	51.12	**+1.06**	3.8×
5	37.99	39.16	**+1.17**	**4.2**×

4.3 Generalization Across Token-Reduction Methods

At iso-GFLOPs the late-concentrated schedule (default γ=2) improves all five training-free methods, on both DeiT-S and DeiT-B, at every operating point (Table 3): all 30 cells are positive ($+0.23$ to $+2.52$pp), so the same schedule used for ToMe transfers unchanged to the other methods. This is what Section 3 predicts, since the distortion $D(\ell)$ is front-loaded for every one of the five reducers (Figure 3a). The benefit spans method families, from attention-guided pruning (EViT (Liang et al., 2022)) and merging (ToMe (Bolya et al., 2023), bipartite matching; PiToMe (Tran et al., 2024), spectrum-preserving) to agglomerative clustering (ATC (Haurum et al., 2024)) and adaptive sampling (ATS (Fayyaz et al., 2022)), with EViT gaining most. The same five methods at γ=1, including a more aggressive operating point, are in Appendix K, where the late schedule again beats flat in all cells, and Figure 7 plots each method's accuracy against compute.

Table 3: Late-concentrated schedule applied to five training-free token-reduction methods at the default γ=2. DeiT-S/B, ImageNet-C severity 5, 15-corruption mean top-1 (%), iso-GFLOPs; each header's % is the γ=2 compute reduction vs. the uncompressed model. For each operating point we report flat, **late** (bold), and Δ = late − flat (pp); The γ=1 and larger r_{base} results are in Appendix K.

	r_{base}=4 (−13%)			r_{base}=6 (−21%)			r_{base}=8 (−26%)		
Method	flat	late	Δ	flat	late	Δ	flat	late	Δ
DeiT-S									
ToMe	38.83	39.37	**+0.54**	38.44	39.32	**+0.88**	37.99	39.16	**+1.17**
EViT	37.50	38.97	**+1.47**	36.69	38.91	**+2.22**	35.98	38.50	**+2.52**
ATS	34.95	35.43	**+0.48**	34.50	35.21	**+0.71**	33.92	34.88	**+0.96**
ATC	38.28	38.73	**+0.45**	37.96	38.74	**+0.78**	37.55	38.65	**+1.10**
PiToMe	38.45	38.99	**+0.54**	38.14	38.67	**+0.53**	37.67	37.90	**+0.23**
DeiT-B									
ToMe	45.98	46.50	**+0.52**	45.27	46.17	**+0.90**	44.35	45.89	**+1.54**
EViT	44.52	45.80	**+1.28**	43.57	45.38	**+1.81**	42.45	44.62	**+2.17**
ATS	40.60	41.34	**+0.74**	39.79	40.75	**+0.96**	38.81	40.11	**+1.30**
ATC	45.41	45.96	**+0.55**	44.79	45.73	**+0.94**	43.97	45.47	**+1.50**
PiToMe	45.25	45.78	**+0.53**	44.56	44.84	**+0.28**	43.69	43.97	**+0.28**

Table 4: Generalization across backbones at the default r_{base}=8 rate, a 24–25% compute cut (ImageNet-C sev 5, 15-corruption mean top-1, %). Δ = late − flat, **bold** the default.

Backbone	flat	late γ=1 (Δ)	**late γ=2 (Δ)**
Supervised, distilled			
DeiT-S	37.99	38.89 (+0.90)	**39.16 (+1.17)**
DeiT-B	44.35	45.35 (+1.00)	**45.89 (+1.54)**
Supervised, AugReg			
ViT-S	38.05	39.02 (+0.97)	**39.46 (+1.41)**
ViT-B	53.39	54.38 (+0.99)	**54.96 (+1.57)**
ViT-L	58.82	60.32 (+1.50)	**60.62 (+1.80)**
Contrastive image-text			
CLIP ViT-B	43.68	44.72 (+1.04)	**45.07 (+1.39)**
CLIP ViT-L	57.29	58.10 (+0.81)	**58.41 (+1.12)**
Self-supervised, global pooling			
MAE ViT-B	42.81	43.77 (+0.96)	**44.09 (+1.28)**
MAE ViT-L	52.97	53.96 (+0.99)	**54.19 (+1.22)**

Table 5: Generalization across shift types and modalities (flat vs. late γ=2, iso-GFLOPs, accuracy %). Δ = late − flat (pp); the same schedule is applied to each system's ViT backbone.

Setting	flat	late	Δ
Natural shift			
ImageNet-A	25.31	26.55	**+1.24**
ImageNet-R	45.90	46.57	**+0.67**
ImageNet-V2	69.38	69.97	**+0.59**
ImageNet-Sketch	32.89	33.33	**+0.44**
DomainNet-126	65.14	65.84	**+0.70**
Synthetic corruption			
ImageNet-$\overline{\text{C}}$	47.28	48.74	**+1.45**
ImageNet-3DCC	41.47	42.38	**+0.91**
NaBirds-C	31.06	31.83	**+0.77**
Other modalities			
Mini-Kinetics-C (VideoMAE)	66.17	66.79	**+0.62**
GQA-C (LLaVA)	54.96	55.20	**+0.24**

4.4 Generalization Across Backbones and Pretraining

Table 4 applies the schedule at iso-GFLOPs to nine backbones spanning four pretraining recipes, three scales, and both readouts. The default γ=2 beats flat on all nine (+1.12 to +1.80pp) and γ=1 on all nine. The effect does not fade as the backbone grows larger or more accurate, ViT-L being both the most accurate (58.82 flat) and the one that gains most (+1.80). Nor does it depend on how the backbone was trained. The three ViT-B rows span 10.6pp of flat accuracy under three pretraining recipes yet their gains stay within 0.29pp, and the MAE rows keep theirs with neither supervised pretraining nor a class token. CLIP ViT-L carries 256 tokens rather than 196, and the schedule, defined over normalized depth, transfers to it unchanged. Appendix D adds the per-corruption results and the aggressive r_{base}=12 rate.

4.5 Generalization Across Shift Types and Modalities

The schedule generalizes well beyond synthetic ImageNet-C. We apply the same iso-GFLOPs flat-versus-late comparison to eight further shift suites and two other modalities (Table 5): five natural shifts (ImageNet-A (Hendrycks et al., 2021b), ImageNet-R, ImageNet-V2 (Recht et al., 2019), ImageNet-Sketch, and DomainNet-126 (Peng et al., 2019) averaged over its three target domains), two harder synthetic suites (ImageNet-$\overline{\text{C}}$ (Mintun et al., 2021), ImageNet-3DCC (Kar et al., 2022)) with a fine-grained task (NaBirds-C), and two non-image modalities (video on Mini-Kinetics-C (Yi et al., 2021), vision-language QA on GQA-C (Hudson & Manning, 2019; Usama et al., 2025)). The image settings share a common ViT-S backbone, the non-image modalities their native models (VideoMAE, LLaVA). The gain is positive in all ten settings. Other backbones and per-corruption results are in Appendices F, H and I.

The gains are largest on ImageNet-$\overline{\text{C}}$ (+1.45) and ImageNet-A (+1.24) and smallest on the style shifts (ImageNet-R +0.67, Sketch +0.44); they do not track how much accuracy the shift costs, since near-clean ImageNet-V2 gains more than Sketch. ImageNet-A is where the schedule helps most across backbones, +2.01pp on average against the +1.42pp those same five gain on ImageNet-C, and on three of them the reduced model passes the uncompressed one. Where we dump per-image predictions, on ImageNet-A, ImageNet-V2 and each DomainNet domain, every gain in the table is significant under an exact McNemar test (McNemar, 1947), with the intervals in Appendix E, and NaBirds-C and DomainNet-126 show the effect surviving fine-tuning to a new label space. The benefit also crosses modality. On video we run VideoMAE (Tong et al., 2022) under the six spatial Mini-Kinetics-C corruptions with ToMe at iso-GFLOPs (+0.62 at severity 5 and +0.25 at severity 3, so the video gain grows with severity as well, and it again concentrates on the structure-degrading corruptions; per-corruption in Appendix G). On a vision-language model, the furthest transfer we test, the schedule acts only on the CLIP vision encoder that feeds LLaVA-1.5-7B (Liu et al., 2024), and corrupted GQA accuracy still improves at iso-GFLOPs (+0.24pp).

Table 6: Late-concentrated schedule layered under TTA methods. ImageNet-C severity 5, 15-corruption mean top-1, r_{base}=8, iso-GFLOPs; each method is run on the flat and the late (γ=2) schedule, which cuts GFLOPs by 26% against the same method on full tokens (Appendix L). Δ = late − flat (pp); ± = std over 3 seeds.

	ViT-S			ViT-B		
Method	flat	late	**Δ**	flat	late	**Δ**
Source (inference-only)	38.05	39.46	**+1.41**	53.39	54.96	**+1.57**
Tent (Wang et al., 2021)	48.47±0.00	52.04±0.00	**+3.57±0.00**	54.49±0.00	55.72±0.00	**+1.23±0.00**
EATA (Niu et al., 2022)	52.65±0.16	53.66±0.26	**+1.00±0.38**	59.09±0.62	60.11±0.37	**+1.01±0.43**
SAR (Niu et al., 2023)	51.52±0.00	51.96±0.00	**+0.43±0.00**	59.71±0.00	61.07±0.00	**+1.35±0.00**
FOA (Niu et al., 2024)	42.42±0.48	42.81±0.30	**+0.39±0.60**	59.80±0.10	60.41±0.14	**+0.61±0.09**
SPA (Niu et al., 2025)	51.22±0.09	52.98±0.12	**+1.76±0.21**	63.53±0.08	64.81±0.06	**+1.28±0.13**
NEO (Murphy et al., 2026)	40.97±0.00	41.97±0.00	**+1.00±0.00**	56.81±0.00	58.16±0.00	**+1.34±0.00**

4.6 Composability with Test-Time Adaptation

The late-concentrated schedule changes no parameters, so it can be layered under test-time adaptation (TTA), which addresses the same shift by updating the model on unlabeled test inputs (Wang et al., 2021). We ask whether its robustness gain persists when a TTA method runs on top, at iso-GFLOPs. We run six methods: three minimize entropy over normalization parameters (Tent (Wang et al., 2021), EATA (Niu et al., 2022), SAR (Niu et al., 2023)); the forward-only FOA (Niu et al., 2024) searches input prompts with an evolution strategy; the self-bootstrapping SPA (Niu et al., 2025) trains a predictor on two geometry-preserving views; and NEO (Murphy et al., 2026) optimizes nothing and re-centers the classifier's input. All six run on ViT-S and ViT-B, the backbones this line of adaptation work evaluates on, and on both the flat (γ=0) and the late schedule at the main operating point (r_{base}=8, γ=2); we report the 15-corruption mean over 3 seeds (Table 6). Learning rates are frozen and shared by the two schedules; these, the backbone choice and the rest of the protocol are in Appendix B, with r_{base}=4 in Appendix L.

At r_{base}=8, γ=2 the schedule improves over flat for every method on both backbones (+0.39 to +3.57pp, including the gradient-free FOA and NEO), and over both operating points it is positive in all 24 cells (Table 18). The gain is additive rather than absorbed: on top of the adapted baseline the schedule still adds +1.36pp on ViT-S and +1.14pp on ViT-B. SPA has the strongest flat baseline (63.53 on ViT-B) and still gains +1.28, so the effect is not an artifact of a weak baseline: at 26% fewer GFLOPs its late schedule comes within 0.1pp of full-token accuracy on ViT-S and 0.6pp on ViT-B, against 1.9pp for flat on both (Appendix L). Under test-time adaptation, the late-concentrated schedule therefore keeps the compute saving and improves on the flat schedule by about a point, leaving each method's mechanism untouched.

5 Limitations and Conclusion

Limitations. Our first-order amplification account predicts the direction and ordering of the effects, borne out in every test, but not their per-corruption magnitude, which also reflects proximity to the decision boundary (Section 3), though the amplification it predicts rises under every corruption group. The schedule is static by design, which keeps it tuning-free; adapting γ to the estimated shift is a refinement, not a requirement for the reported gains.

Conclusion. We showed that the token-reduction schedule, how the removal budget is distributed over depth, is a first-order determinant of a token-reduced vision transformer's robustness to distribution shift. At no extra inference cost, a simple late-concentrated power-law schedule recovers 83% of the accuracy gap that token reduction opens against the uncompressed model (+1.17pp at a 26% compute reduction), and 99% at a lighter 7% reduction (+0.26pp; DeiT-S, ImageNet-C). The gain comes from the schedule, not from retaining more tokens. We trace this to a front-loaded error budget: the earlier a reduction acts the more the network amplifies its error, and shift widens that gap. The effect is broad: nine backbones, eight further shift suites, two other modalities, a gain that grows ∼4× from clean to severity 5, and composability with six test-time adaptation methods, all from one static schedule applied at test time with no tuning.

AI USE STATEMENT

We used generative AI tools to help interpret results, edit the text for grammar and readability, draft the figure scripts, and clean up code. The authors checked every interpretation and every figure against the underlying data, reviewed every edited passage, and verified that all AI-edited code reproduces the reported numbers. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly available benchmarks and publicly released model checkpoints. It involves no human subjects, collects no new data, and introduces no new dataset. The contribution is methodological and inherits, and neither adds to nor removes, the known biases and limitations of the models and datasets it is applied to. We do not foresee ethical concerns beyond those general to machine learning research on public benchmarks.

REPRODUCIBILITY STATEMENT

Code and data. The code is available at `https://anonymous.4open.science/r/LCTR-0042`, with the schedule and its iso-GFLOPs calibration, the evaluation pipeline, and a launcher for every image-classification table and figure in the paper. It also lists the DomainNet-126 and NaBirds source checkpoints we finetuned with their SHA-256; those checkpoints and the video and vision-language harnesses will be released with the code. Every dataset used is a public benchmark: the ImageNet-1k validation set, ImageNet-C, ImageNet-A, ImageNet-R, ImageNet-V2, ImageNet-Sketch, DomainNet-126, ImageNet-$\overline{\text{C}}$, ImageNet-3DCC, NaBirds, Mini-Kinetics-C, and GQA. Nothing is collected or relabelled for this work beyond applying the published corruption suites.

Models. Every backbone but one set is a public checkpoint, named with its exact identifier in Appendix B: DeiT-S/B and ViT-S/B/L from `timm` pretrained on ImageNet-1k (DeiT-S is the default for the main analysis), CLIP ViT-B/16 and ViT-L/14 finetuned on ImageNet-1k, MAE-pretrained ViT-B/L that read out by global average pooling, VideoMAE ViT-B finetuned on Kinetics-400, and LLaVA-1.5-7B with a CLIP ViT-L/14 encoder. The exception is DomainNet-126, where no public source checkpoint covers all five backbones, so we finetune each on the *real* domain with one recipe given in Appendix B.

The schedule is specified in closed form. The only thing we change in a reducer is the per-layer removal budget of Eq. 1, which is a function of $(r_{\text{base}}, \gamma, L)$ alone: no training, no per-input decision, and no learned parameter. The continuous counts are rounded to integers by largest remainder and capped at $\lfloor n_\ell/2 \rfloor$, the same cap for every reducer (Appendix B). Because GFLOPs is analytic in the schedule, the iso-GFLOPs calibration needs no measurement: for each $(r_{\text{base}}, \gamma)$ we read off the integer budget whose GFLOPs match the flat baseline, always rounding so the late schedule uses no more GFLOPs than flat, with GFLOPs from a single THOP counter. Table 31 lists every calibrated schedule used in the paper, so the exact per-layer counts behind any reported number can be reconstructed without rerunning the calibration.

Determinism and reported variance. Inference-only evaluation is fully deterministic: we set `cudnn.deterministic`, enable deterministic algorithms, and use a deterministic scatter-mean in the merge. The flat-versus-late comparisons therefore carry no run variance, which is why the inference-only tables report single numbers rather than error bars. The test-time-adaptation experiments (Section 4.6) are the stochastic ones and are run over 3 seeds $(42, 43, 44)$, with seed standard deviations reported alongside the means.

Evaluation scope. The main results use the complete standard benchmarks, so there is no subsampling uncertainty: the full 50k images per ImageNet-C corruption at severity 5, the 50k ImageNet validation set for clean accuracy, the full ImageNet-A, ImageNet-R, ImageNet-V2 and ImageNet-Sketch sets, the four DomainNet-126 splits, and the 24,633 NaBirds-C test images. Two suites are exceptions, marked wherever they are reported: ImageNet-$\overline{\text{C}}$ and ImageNet-3DCC use the standard RobustBench (Croce et al., 2021) 5k-image-per-corruption subset, and Mini-Kinetics-C uses 2500

clips over its six spatial corruptions. In the vision-language experiment the encoder schedule is matched to within 0.03% of the flat schedule's GMACs (Appendix B).

Fairness of the compute matching. The iso-GFLOPs comparisons are also matched in wall-clock: at the main operating point the late ($\gamma{=}2$) and flat schedules differ by under 0.5% in measured throughput (batch 64, single GPU), so the comparison is fair in real latency and not only in analytic FLOPs (Appendix W).

Hyperparameters. Token reduction here is training-free and inference-only, so it adds no hyperparameter beyond the schedule itself. For test-time adaptation, each method's learning rate is chosen once per method and backbone from a flat-schedule stability protocol and then frozen before the flat-versus-late comparison, with no per-schedule tuning; every other setting is that method's official default. Appendix B lists them all.

REFERENCES

Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

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 (ICLR)*, 2023.

Maxim Bonnaerens and Joni Dambre. Learned thresholds token merging and pruning for vision transformers. *Transactions on Machine Learning Research (TMLR)*, 2023.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. RobustBench: a standardized adversarial robustness benchmark. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

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 (ICLR)*, 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 (ECCV)*, 2022.

Joakim Bruslund Haurum, Sergio Escalera, Graham W. Taylor, and Thomas B. Moeslund. Agglomerative token clustering. In *European Conference on Computer Vision (ECCV)*, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b.

Jung Hwan Heo, Seyedarmin Azizi, Arash Fayyazi, and Massoud Pedram. Training-free acceleration of ViTs with delayed spatial merging. In *Workshop on Efficient Systems for Foundation Models (ES-FoMo) at the International Conference on Machine Learning (ICML)*, 2024.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3D common corruptions and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. The Lipschitz constant of self-attention. In *International Conference on Machine Learning (ICML)*, 2021.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 2. Morgan-Kaufmann, 1989.

Jaeyeon Lee and Dong-Wan Choi. Lossless token merging even without fine-tuning in vision transformers. In *European Conference on Artificial Intelligence (ECAI)*, 2025.

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 (ICLR)*, 2022.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Alexander Murphy, Michal Danilowski, Soumyajit Chatterjee, and Abhirup Ghosh. NEO: No-optimization test-time adaptation through latent re-centering. In *International Conference on Learning Representations (ICLR)*, 2026.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning (ICML)*, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations (ICLR)*, 2023.

Shuaicheng Niu, Chunyan Miao, Guohao Chen, Pengcheng Wu, and Peilin Zhao. Test-time model adaptation with only forward passes. In *International Conference on Machine Learning (ICML)*, 2024.

Shuaicheng Niu, Guohao Chen, Peilin Zhao, Tianyi Wang, Pengcheng Wu, and Zhiqi Shen. Self-bootstrapping for versatile test-time adaptation. In *International Conference on Machine Learning (ICML)*, 2025.

Roman Novak, Yasaman Bahri, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: An empirical study. In *International Conference on Learning Representations (ICLR)*, 2018.

Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: Theory and practice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

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 (NeurIPS)*, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *International Conference on Machine Learning (ICML)*, 2019.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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 (ICML)*, 2021.

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 (NeurIPS)*, 2024.

Muhammad Usama, Syeda Aishah Asim, Syed Bilal Ali, Syed Talal Wasim, and Umair Bin Mansoor. Analysing the robustness of vision-language-models to common corruptions. *arXiv preprint arXiv:2504.13690*, 2025.

Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P. Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Zhenhailong Wang, Senthil Purushwalkam, Caiming Xiong, Silvio Savarese, Heng Ji, and Ran Xu. DyMU: Dynamic merging and virtual unmerging for efficient variable-length VLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Zixin Wang, Dong Gong, Sen Wang, Zi Huang, and Yadan Luo. Is less more? exploring token condensation as training-free test-time adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Yizhe Xiong, Zihan Zhou, Yiwen Liang, Hui Chen, Zijia Lin, Xinhao Xu, Tianxiang Hao, Fan Zhang, Jungong Han, and Guiguang Ding. Neutralizing token aggregation via information augmentation for efficient test-time adaptation. In *European Conference on Computer Vision (ECCV)*, 2026.

Chenyu Yi, Siyuan Yang, Haoliang Li, Yap-Peng Tan, and Alex C. Kot. Benchmarking the robustness of spatial-temporal models against corruptions. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

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

A Related Work

Token reduction for efficient vision transformers. A large body of work accelerates ViTs by discarding redundant tokens. Pruning methods drop low-importance tokens: EViT (Liang et al., 2022) reorganizes tokens by their attention to the class token and fuses the inattentive ones. Merging methods instead combine similar tokens: ToMe (Bolya et al., 2023) merges a fixed number of token pairs per layer via bipartite soft matching, and PiToMe (Tran et al., 2024) merges progressively using proportional attention. Adaptive schemes vary the number of tokens per input: ATC (Haurum et al., 2024) clusters tokens into groups, while ATS (Fayyaz et al., 2022) samples a subset to keep by significance. Dynamic-sparsification methods such as DynamicViT (Rao et al., 2021) and A-ViT (Yin et al., 2022) instead learn an input-dependent number of tokens to keep, but require training the reduction policy and so do not apply when the model cannot be retrained on the shift. We therefore scope this work to *training-free* token reduction, which operates purely at inference and is the relevant regime under distribution shift, and use the five methods above throughout.

Depth allocation has been considered mostly for clean-data efficiency or adaptive reduction policies, not as a controlled robustness lever under distribution shift. ToMe random-searches clean ImageNet merging schedules and reports throughput-dependent preferences, including late-heavy schedules at low throughput, flat schedules at intermediate throughput, and decreasing schedules at high throughput. Other recent methods introduce depth dependence in different ways: ATM (Lee & Choi, 2025) varies similarity thresholds with depth, DyMU (Wang et al., 2025a) studies dynamic merging for VLMs, DSM (Heo et al., 2024) delays global merging by skipping early blocks, and DiffRate (Chen et al., 2023)/LTMP (Bonnaerens & Dambre, 2023) learn layer-wise rates or thresholds with training-time clean objectives. Our question is different: for fixed training-free reducers, with no target labels, adaptation, or retraining, how should a static schedule allocate removal under distribution shift at matched inference cost? We answer this with a one-parameter late-concentrated family calibrated to use no more GFLOPs than the flat baseline; under that conservative cost constraint, late removes more tokens in total yet consistently improves OOD accuracy.

Robustness to distribution shift. ImageNet-C (Hendrycks & Dietterich, 2019) measures robustness to common corruptions at graded severities and is the standard distribution-shift benchmark. ViTs (Dosovitskiy et al., 2021; Touvron et al., 2021) exhibit nontrivial robustness to such corruptions, and a line of work studies how their architecture and training affect this robustness (Bhojanapalli et al., 2021; Paul & Chen, 2022). The robustness of token-reduced ViTs, however, is largely unexamined: efficiency methods are tuned for clean accuracy, and whether, and how, token reduction should differ under shift is open. We show that the schedule specifically governs shift robustness, with gains that grow with corruption severity and stay small on clean data.

Test-time adaptation, and adaptation for token reduction. Test-time adaptation (TTA) improves robustness by updating model parameters on unlabeled test inputs (Wang et al., 2021). Most related to us, recent work combines token reduction with TTA: NAViA (Xiong et al., 2026) adds learnable class-token biases to recover the mutual information that token aggregation discards, and TCA (Wang et al., 2025b) performs VLM-specific token condensation with a CLIP/SigLIP test-time cache of domain-anchor tokens and logit self-correction. Both report that token reduction costs accuracy under existing TTA and answer it by adding to the model, via backpropagation or a maintained cache, while leaving the reduction schedule itself unchanged. We instead reshape the schedule alone, with no gradients, cache, or added parameters, and gain over the flat schedule both use; being inference-only, it is orthogonal to TTA and composes with it (Section 4.6). A direct comparison is not straightforward. TCA is coupled to CLIP/SigLIP zero-shot inference, so porting it to the supervised DeiT/ViT classifiers we study would define a new TCA-inspired method rather than a drop-in baseline, and evaluating it natively would need a separate zero-shot protocol; NAViA, closer to our setting, has no official implementation, so any comparison would rest on our reimplementation, not the authors' method. We therefore treat both as complementary, not competing: because they reduce tokens on a flat schedule by default, our reshaping could in principle be applied within either, changing where tokens are merged across depth, not how adaptation operates.

B Implementation Details

Backbones. Image classification uses DeiT-S/B (`deit_small/base_patch16_224`) and ViT-S/B/L (`vit_small/base/large_patch16_224`) from `timm`, pretrained on ImageNet-1k; DeiT-S is the default for the main analysis. Table 4 adds two contrastively pretrained backbones finetuned on ImageNet-1k (`vit_base_patch16_clip_224.openai_ft_in1k` and `vit_large_patch14_clip_224.openai_ft_in1k`, the latter carrying 256 tokens rather than 196). The twelve-layer backbones there run at r_{base}=8; on the twenty-four-layer ones that per-layer rate would cut nearly twice the compute, so we match the compute cut instead, at r_{base}=4 for the ViT-L/16 models and 5 for CLIP ViT-L/14, a 24–25% reduction in every case, with the late budget calibrated exactly as elsewhere. Video uses VideoMAE ViT-B (Tong et al., 2022) finetuned on Kinetics-400 (Kay et al., 2017), tokenizing a clip into 1568 space-time tokens (tubelet embedding, no class token). Vision-language QA uses LLaVA-1.5-7B (Liu et al., 2024) with a CLIP ViT-L/14 vision encoder (576 visual tokens). The schedule acts inside this encoder; of prior VLM token reduction, PruMerge (Shang et al., 2025) reduces the encoder's output and DyMU (Wang et al., 2025a) merges inside it with a constant per-layer target. MAE-pretrained ViT-B and ViT-L (He et al., 2022) finetuned on ImageNet-1k (official checkpoints) classify by global average pooling over the patch tokens rather than a class token; they appear both in Table 4 and as the readout control of Appendix T. For DomainNet-126 no public source checkpoint covers all five backbones, so we finetune each one on the *real* domain with a single recipe (5 epochs, AdamW, lr 5×10^{-5} cosine, label smoothing 0.1, each model's own normalization), reaching 91.6 to 94.3 on the held-out real split. The recipe is fixed once and shared by all five, and the schedule comparison that follows is inference-only on the frozen weights, so the flat and the late arm see the identical model and no part of the finetuning can prefer one to the other. The held-out real split is the control for the finetuning itself, and the gain is absent there. We release these checkpoints with the code.

Token reduction and iso-GFLOPs. The power-law schedule (Eq. 1) is realized with ToMe bipartite soft matching by default, and with EViT (Liang et al., 2022), ATS (Fayyaz et al., 2022), ATC (Haurum et al., 2024), and PiToMe (Tran et al., 2024) for the cross-method study (the same schedule, each method's own scoring). The per-layer budget reaches each reducer through its native interface (a keep-rate for EViT and ATS, a merge count for ToMe and PiToMe, a cluster count for ATC), so the schedule is identical and only the scoring rule differs; ToMe and PiToMe use proportional attention, the merge is reweighted by a token-survival correction, and ATC clusters with average linkage. The continuous per-layer counts of Eq. 1 are rounded to integers by largest remainder, and, because ToMe's bipartite matching removes at most half the tokens present at a layer, each count is capped at $\lfloor n_\ell/2 \rfloor$; we apply the same cap to every reducer so the realized schedule stays identical across methods. At strongly late-concentrated settings this cap lowers the deepest layers, where few tokens remain, below the nominal power law (the faster late drop in Figure 6), so removal cannot be pushed arbitrarily late. GFLOPs are computed with a single THOP counter; since GFLOPs is analytic in the schedule, each $(r_{\text{base}}, \gamma)$ is calibrated by reading off the integer per-layer budget whose GFLOPs match the flat baseline, rounded so the late schedule never exceeds the flat GFLOPs (Table 31). Inference is fully deterministic (`cudnn.deterministic`, deterministic algorithms, and a deterministic scatter-mean in the merge), so schedule comparisons carry no seed variance. Tent, SAR and NEO have no stochastic step, so their three seeds coincide (std 0.00).

Benchmarks. ImageNet-C (Hendrycks & Dietterich, 2019): 15 corruptions, severity 5, 50k images per corruption; clean = the 50k ImageNet validation set. ImageNet-A (Hendrycks et al., 2021b) (7,500 natural adversarial images, evaluated through its 200-class mask), ImageNet-R (Hendrycks et al., 2021a) (renditions), the matched-frequency split of ImageNet-V2 (Recht et al., 2019) (10,000 images) and ImageNet-Sketch (Wang et al., 2019) use their full sets. DomainNet-126 (Peng et al., 2019): models finetuned on *real* and evaluated on sketch (24,147), clipart (18,523) and painting (30,042), with the held-out real split (6,962) as an in-domain control. ImageNet-$\overline{\text{C}}$ (Mintun et al., 2021) and ImageNet-3DCC (Kar et al., 2022) use the standard RobustBench (Croce et al., 2021) 5k-image-per-corruption subset at severity 5. NaBirds-C: DeiT-S and ViT-S finetuned on NaBirds (Van Horn et al., 2015) (555 species), evaluated under the 15 ImageNet-C corruptions at severity 5 (24,633 test images). Mini-Kinetics-C (Yi et al., 2021): the six *spatial* corruptions at severity 5 over 2500 clips spanning 400 classes (the six temporal corruptions are out of scope). GQA-C (Usama et al., 2025): GQA (Hudson & Manning, 2019) testdev-balanced under corruption; because VLM inference is costly we evaluate the three corruptions that benchmark flags as hardest

(frost, impulse noise, glass blur) at severity 5, with the encoder schedule matched to within 0.03% of the flat schedule's GMACs.

Test-time adaptation (Section 4.6). Tent, EATA, SAR, FOA, SPA and NEO are run over 3 seeds $(42, 43, 44)$ at full 50k (FOA is gradient-free, optimizing input prompts with CMA-ES; NEO performs no optimization at all). We use ViT rather than the DeiT backbones of the earlier sections because these adaptation methods were developed and tuned primarily on ViT (SAR and FOA report on ViT-Base), so evaluating them in their intended regime avoids confounding the comparison with per-backbone re-tuning. The learning rate is selected once per method/backbone from the const-schedule stability protocol and then frozen before the flat-versus-late comparison (no per-schedule tuning): 5×10^{-4}, except EATA at its official rate (1×10^{-4} on ViT-S, 1×10^{-3} on ViT-B) and Tent on ViT-B at 1×10^{-4}. SPA also runs at 5×10^{-4} on both backbones: its official rate (1×10^{-2}, stated for ViT-B) diverges here, taking the ViT-S flat baseline to near-chance after the first $1{,}280$ images and collapsing the ViT-B one in 8 of the 15 corruption$\times$seed cells it was checked on; 5×10^{-4} is the largest rate whose flat baseline survives every corruption and seed. NEO has no learning rate and no optimizer, so nothing is selected for it. EATA estimates Fisher information from $2{,}000$ samples; Tent, SAR and NEO have no seed-dependent randomness and so are deterministic across seeds. All methods take a single update step with SGD (momentum 0.9); SAR uses sharpness-aware minimization (ρ=0.05). The reliable-sample entropy threshold is $E_0 = 0.4 \ln 1000 \approx 2.76$ (EATA, SAR), EATA's redundancy threshold is 0.05. FOA optimizes 3 input prompts with CMA-ES. SPA uses its official ViT settings, a 0.2 low-frequency amplitude mask and an average high-frequency noise fraction of 0.4, with the predictor at $5\times$ the norm-layer rate; NEO has no settings. These are each method's official defaults, apart from SPA's learning rate, noted above.

C Per-Severity Detail: γ=1 vs. γ=2

Table 7 gives the full per-severity breakdown for both exponents (companion to the γ=2 body Table 2). Both schedules' gains grow monotonically with severity; γ=1 rises from $+0.27$ (clean) to $+0.90$pp (severity 5), γ=2 from $+0.28$ to $+1.17$pp. The sev=0 (γ=2) value matches the clean improvement in Table 1.

Table 7: Per-severity 15-corruption mean (%), DeiT-S, r_{base}=8, iso-GFLOPs. flat = const (γ=0); Δ = late $-$ flat for γ=1 and the default γ=2; sev=0 is clean ImageNet val.

Sev	flat	γ=1	$\Delta_{\gamma 1}$	γ=2	$\Delta_{\gamma 2}$
0 (clean)	79.40	79.67	+0.27	79.68	+0.28
1	70.52	70.95	+0.43	71.04	+0.52
2	64.33	64.82	+0.49	64.99	+0.66
3	59.06	59.71	+0.65	59.89	+0.83
4	50.06	50.88	+0.82	51.12	+1.06
5	37.99	38.89	+0.90	39.16	+1.17

D Cross-Backbone Per-Corruption Results

At the aggressive r_{base}=12 rate (a 37% compute reduction) the default γ=2 still beats flat on all four backbones evaluated there, by $+0.85$ to $+1.30$pp, and γ=1 is the stronger exponent, reaching $+1.78$pp (Table 8). This is the migration toward weaker concentration at larger removal budgets that Section 3 predicts and Figure 2c shows.

E Natural Shift: ImageNet-A, ImageNet-V2, and DomainNet-126

Table 5 reports these settings on ViT-S. Table 10 gives all five backbones with paired statistics: for every setting we dump per-image correctness for the flat and the late run on the identical images, then compare the two with a 95% paired bootstrap interval (10^4 resamples) and an exact McNemar test. ImageNet-A (Hendrycks et al., 2021b) uses its 200-class mask and ImageNet-V2 (Recht et al., 2019) the matched-frequency split. For DomainNet-126 (Peng et al., 2019) each backbone is fine-tuned

Table 8: Cross-backbone generalization at the aggressive rate r_{base}=12 (ImageNet-C sev 5, 15-corruption mean top-1, %, iso-GFLOPs). Δ = late $-$ flat.

Backbone	flat	late γ=1 (Δ)	late γ=2 (Δ)
DeiT-S	36.67	37.88 (+1.21)	37.77 (+1.10)
DeiT-B	41.44	42.96 (+1.52)	42.74 (+1.30)
ViT-S	36.07	37.48 (+1.41)	36.92 (+0.85)
ViT-B	50.93	52.71 (+1.78)	52.15 (+1.22)

Table 9: Per-corruption top-1 (%), ViT-B and ViT-S, sev=5, 50k samples per corruption, at the main operating point r_{base}=8. flat = const (γ=0); late = γ=2. Δ = late $-$ flat. The late schedule uses no more GFLOPs than flat on each backbone, and all 15 corruptions improve on both.

	ViT-B			ViT-S		
Corruption	flat	late	Δ	flat	late	Δ
gaussian_noise	53.64	56.30	+2.66	31.51	33.38	+1.87
shot_noise	53.85	56.16	+2.31	30.93	32.54	+1.61
impulse_noise	54.11	56.92	+2.82	31.08	32.93	+1.85
defocus_blur	44.74	46.98	+2.23	30.26	31.78	+1.52
glass_blur	35.04	35.57	+0.53	22.61	23.00	+0.39
motion_blur	50.99	52.55	+1.56	39.64	41.01	+1.37
zoom_blur	42.48	44.17	+1.69	30.78	31.92	+1.14
snow	60.54	61.48	+0.94	41.90	43.17	+1.27
frost	58.78	60.39	+1.60	41.28	43.26	+1.98
fog	62.44	64.53	+2.08	44.51	46.40	+1.90
brightness	76.74	77.27	+0.53	69.16	70.19	+1.03
contrast	29.92	32.44	+2.52	14.45	16.32	+1.87
elastic_transform	45.46	45.81	+0.35	35.50	36.80	+1.30
pixelate	65.17	66.39	+1.22	52.42	53.78	+1.36
jpeg_compression	66.96	67.42	+0.47	54.78	55.44	+0.66
Mean	53.39	54.96	**+1.57**	38.05	39.46	**+1.41**

on the *real* domain with one recipe and evaluated on the three target domains, plus the held-out *real* split as an in-domain control. Uncompressed accuracy matches the published checkpoints within 0.1pp everywhere.

The late schedule wins on ImageNet-A on all five backbones by +1.24 to +2.60pp, the largest gains we measure, and on ViT-B, DeiT-S and CLIP ViT-B the reduced model also passes the uncompressed one (49.83 vs 49.67, 19.73 vs 18.68, 48.13 vs 47.31) while using 74% of its compute. The DomainNet target domains are positive in 12 of 15 cells, the exceptions being DeiT-S on sketch and clipart, which are the two most abstract domains; painting, the most photographic, is positive on all five. The in-domain control is within $\pm$0.24pp everywhere and significant nowhere, so the schedule neither helps nor hurts when the test distribution matches the training one.

F Fine-Grained Shift: NaBirds-C

To test transfer beyond ImageNet-scale classification, we apply the same iso-GFLOPs const-versus-late comparison to NaBirds-C: a DeiT-S and a ViT-S finetuned on NaBirds (Van Horn et al., 2015) (555 fine-grained bird species), evaluated under the 15 ImageNet-C corruptions at severity 5 (24,633 test images), inference-only. Operating point: const r_{base}=8 (3.199G) vs. late γ=2 (r_{base}=8, $\bar{r}$=16, 3.147G $\leq$ flat). At the default γ=2 late beats flat on all 15 corruptions for both backbones (Table 11), mean +0.92pp (DeiT-S) and +0.77pp (ViT-S); for DeiT-S contrast is near-degenerate at $\sim$2%.

Table 10: Natural shift across backbones, Δ = late γ=2 $-$ flat (pp) at iso-GFLOPs, r_{base}=8. * marks an exact McNemar $p < 0.05$; those cells also have a 95% paired bootstrap interval excluding zero. DN = DomainNet-126, whose *real* column is the in-domain control.

	ImageNet		DN target domains			DN in-domain
Backbone	A	V2	sketch	clipart	painting	real
ViT-S	$+1.24^*$	$+0.59^*$	$+0.63^*$	$+0.56^*$	$+0.90^*$	$+0.04$
ViT-B	$+2.60^*$	$+0.09$	$+0.35^*$	$+0.28^*$	$+0.46^*$	-0.04
DeiT-S	$+1.67^*$	$+0.38^*$	-0.10	-0.01	$+0.62^*$	$+0.11$
DeiT-B	$+1.96^*$	$+0.55^*$	$+0.47^*$	$+0.33^*$	$+0.82^*$	$+0.24$
CLIP ViT-B	$+2.56^*$	$+0.16$	$+0.29^*$	$+0.19^*$	$+0.56^*$	-0.03
mean	$+2.01$	$+0.35$	$+0.33$	$+0.27$	$+0.67$	$+0.06$

Table 11: NaBirds-C per-corruption top-1 (%), DeiT-S and ViT-S (NaBirds-finetuned, 555 classes), sev=5, 24,633 images. flat = const (r_{base}=8); late = γ=2 (the default, $\bar{r}$=16, 3.147G $\leq$ flat). Δ = late $-$ flat.

	DeiT-S			ViT-S		
Corruption	flat	late	Δ	flat	late	Δ
gaussian_noise	23.41	24.88	$+1.47$	14.85	16.04	$+1.19$
shot_noise	23.78	24.95	$+1.17$	16.23	17.83	$+1.60$
impulse_noise	26.02	27.47	$+1.45$	15.60	16.76	$+1.16$
defocus_blur	16.49	16.70	$+0.21$	6.98	7.03	$+0.05$
glass_blur	16.92	17.53	$+0.61$	17.29	17.40	$+0.11$
motion_blur	25.16	25.87	$+0.71$	20.76	20.98	$+0.22$
zoom_blur	33.78	34.75	$+0.97$	36.74	37.34	$+0.60$
snow	44.46	45.83	$+1.37$	46.56	47.18	$+0.62$
frost	46.81	48.22	$+1.41$	35.83	36.15	$+0.32$
fog	45.56	46.13	$+0.57$	38.26	39.41	$+1.15$
brightness	63.37	63.87	$+0.50$	67.09	67.38	$+0.29$
contrast	2.17	2.18	$+0.01$	18.11	19.38	$+1.27$
elastic_transform	45.00	46.45	$+1.45$	47.89	49.21	$+1.32$
pixelate	33.15	34.28	$+1.13$	38.26	39.17	$+0.91$
jpeg_compression	44.98	45.76	$+0.78$	45.50	46.24	$+0.74$
Mean	32.74	33.66	$\mathbf{+0.92}$	31.06	31.83	$\mathbf{+0.77}$

G VIDEO: PER-CORRUPTION RESULTS

Table 12 gives the per-corruption Mini-Kinetics-C results behind Table 5 (VideoMAE ViT-B, six spatial corruptions, severities 3 and 5). The gain concentrates on structure-degrading corruptions (shot noise, fog, rain) and is near-neutral on mild photometric shifts (contrast, brightness, saturate), which leave accuracy near its clean value. Clean single-clip, single-crop top-1 is 76.8% on this 2500-clip subset (the published 80.9% uses multi-view inference). Mini-Kinetics-C also defines six temporal corruptions; these probe a redundancy the spatial-token schedule does not act on, so we leave them out of scope.

H IMAGENET-R / IMAGENET-SKETCH: ALL BACKBONES

Table 13 gives the full ImageNet-R / ImageNet-Sketch results behind Table 5, for the four supervised backbones at r_{base}=8, γ=2. All eight cells are positive (mean $+0.28$pp); the small backbones (DeiT-S, ViT-S) gain more than the base backbones (DeiT-B, ViT-B). Inference-only, iso-GFLOPs (late $\leq$ flat), full sets.

Table 12: Cross-modality, per-corruption. VideoMAE ViT-B on Kinetics-400 with ToMe, inference-only, iso-GFLOPs (flat r=48 vs. late γ=2, both ≈142.6 GFLOPs, ∼20% reduction), six spatial corruptions of Mini-Kinetics-C (Yi et al., 2021), 2500 clips. Δ = late − flat.

	Severity 3			Severity 5		
Corruption	flat	late	Δ	flat	late	Δ
Shot noise	67.40	68.28	+0.88	50.04	51.68	+1.64
Fog	70.92	71.36	+0.44	66.80	67.96	+1.16
Contrast	76.08	76.08	+0.00	73.20	73.08	−0.12
Brightness	75.84	75.52	−0.32	71.72	71.80	+0.08
Saturate	75.64	75.76	+0.12	65.60	65.52	−0.08
Rain	73.48	73.88	+0.40	69.68	70.68	+1.00
Mean	73.23	73.48	+0.25	66.17	66.79	+0.62

Table 13: ImageNet-R / ImageNet-Sketch, all backbones. r_{base}=8, late γ=2, iso-GFLOPs. Δ = late − flat (top-1 pp). R = 30,000 images, Sketch = 50,889.

	ImageNet-R			ImageNet-Sketch		
Backbone	flat	late	Δ	flat	late	Δ
DeiT-S	42.14	42.45	+0.31	29.40	29.68	+0.28
DeiT-B	44.56	44.63	+0.07	32.21	32.42	+0.21
ViT-S	45.90	46.57	+0.67	32.89	33.33	+0.44
ViT-B	58.77	58.85	+0.08	44.92	45.09	+0.17

I IMAGENET-$\overline{\text{C}}$ AND IMAGENET-3DCC: PER-CORRUPTION

Tables 14 and 15 give the per-corruption results behind Table 5 for the two harder synthetic corruption suites, ImageNet-$\overline{\text{C}}$ (Mintun et al., 2021) (10 corruptions) and ImageNet-3DCC (Kar et al., 2022) (12 corruptions), for DeiT-S and ViT-S at r_{base}=8, late γ=2, iso-GFLOPs (late ≤ flat), severity 5, on the standard RobustBench 5k-image subset. The body table reports the stronger-gaining ViT-S; both backbones improve on average on both suites.

Table 14: ImageNet-$\overline{\text{C}}$ per-corruption top-1 (%), DeiT-S and ViT-S, r_{base}=8, late γ=2, iso-GFLOPs, severity 5, 5k-image subset. Δ = late − flat.

	DeiT-S			ViT-S		
Corruption	flat	late	Δ	flat	late	Δ
blue noise	50.36	50.62	+0.26	53.08	55.10	+2.02
brownish noise	66.40	67.36	+0.96	60.42	62.34	+1.92
caustic refr.	53.60	54.46	+0.86	51.32	53.12	+1.80
checkerboard	62.66	63.58	+0.92	64.06	65.20	+1.14
sine waves	31.42	31.70	+0.28	24.62	26.42	+1.80
inv. sparkles	18.68	19.22	+0.54	18.04	18.96	+0.92
perlin noise	69.42	70.22	+0.80	65.64	66.88	+1.24
plasma noise	51.74	52.88	+1.14	33.48	36.08	+2.60
single-freq. grey	31.20	30.66	−0.54	34.00	34.00	+0.00
sparkles	67.14	68.46	+1.32	68.18	69.28	+1.10
Mean	50.26	50.92	**+0.65**	47.28	48.74	**+1.45**

Table 15: ImageNet-3DCC per-corruption top-1 (%), DeiT-S and ViT-S, r_{base}=8, late γ=2, iso-GFLOPs, severity 5, 5k-image subset. Δ = late − flat.

	DeiT-S			ViT-S		
Corruption	flat	late	Δ	flat	late	Δ
bit error	15.50	16.04	+0.54	16.70	16.98	+0.28
color quant	59.82	59.80	−0.02	41.56	41.44	−0.12
far focus	50.96	51.62	+0.66	53.30	54.14	+0.84
flash	39.28	39.84	+0.56	38.98	39.70	+0.72
fog 3d	38.72	38.52	−0.20	31.34	31.96	+0.62
h265 abr	43.12	43.46	+0.34	45.94	46.44	+0.50
h265 crf	46.58	46.72	+0.14	50.60	51.32	+0.72
iso noise	44.04	46.46	+2.42	42.06	44.00	+1.94
low light	49.22	50.68	+1.46	44.30	45.74	+1.44
near focus	59.30	60.44	+1.14	63.04	64.50	+1.46
xy motion	28.22	29.22	+1.00	34.44	35.62	+1.18
z motion	31.24	32.34	+1.10	35.38	36.78	+1.40
Mean	42.17	42.93	**+0.76**	41.47	42.38	**+0.91**

J The Four ImageNet-C Corruptions Outside the Standard Fifteen

ImageNet-C ships four further corruptions that the standard benchmark, and so the rest of this paper, leaves out: speckle noise, Gaussian blur, spatter and saturate. We run them at the main operating point to check that the fifteen reported throughout are not a favorable subset. The late schedule wins on all four, by more on average (+1.66pp) than on the fifteen (+1.17pp), so the schedule improves every corruption ImageNet-C defines.

Table 16: The four ImageNet-C corruptions outside the standard fifteen (DeiT-S, severity 5, 50k images each, r_{base}=8, iso-GFLOPs, inference-only). Δ = late − flat.

Corruption	flat	late	**Δ**
Speckle noise	41.67	43.98	**+2.30**
Gaussian blur	23.38	24.09	**+0.71**
Spatter	47.15	49.72	**+2.57**
Saturate	62.10	63.17	**+1.08**
Mean	43.58	45.24	**+1.66**

K Baseline Generalization at γ=1

Table 17 is the γ=1 counterpart of the body Table 3: the same five training-free reducers, but with the late schedule at γ=1 (the previous default) and operating points $r_{\text{base}} \in \{6, 8, 10\}$ that include the aggressive r_{base}=10. Late beats flat in all 30 cells here as well, so the schedule benefit is robust to the exponent and persists at the aggressive rate (where γ=1 is the operating-point optimum, Figure 2c). Figure 7 plots these cells together with the body table's as the accuracy-versus-compute frontier of Figure 2b, one panel per reducer; for every reducer the late curves lie above flat at every budget.

L Composability with TTA: Full Results

This appendix gives the full γ=2 composability experiment of Section 4.6 on both backbones: per-rate means at $r_{\text{base}} \in \{4, 8\}$ (Table 18) and the per-corruption breakdown at r_{base}=8 (Table 19), and what each method costs on the two token budgets (Table 20). The late schedule is the default γ=2

Table 17: Late-concentrated at γ=1 across five training-free reducers (companion to the γ=2 body Table 3). DeiT-S/B, ImageNet-C sev=5, 15-corruption mean top-1 (%), full 50k, iso-GFLOPs (late $\leq$ flat). Δ = **late** − flat (pp); all 30 cells positive. Operating points $r_{\text{base}} \in \{6, 8, 10\}$.

	r_{base}=6			r_{base}=8			r_{base}=10		
Method	flat	late	Δ	flat	late	Δ	flat	late	Δ
DeiT-S									
ToMe	38.44	**39.13**	+0.69	37.99	**38.89**	+0.90	37.41	**38.58**	+1.17
EViT	36.69	**38.10**	+1.41	35.98	**37.87**	+1.89	35.16	**37.62**	+2.46
ATS	34.50	**34.99**	+0.49	33.92	**34.61**	+0.69	33.35	**34.38**	+1.03
ATC	37.96	**38.53**	+0.57	37.55	**38.35**	+0.80	37.04	**38.26**	+1.22
PiToMe	38.14	**38.62**	+0.48	37.67	**38.23**	+0.56	37.08	**37.75**	+0.67
DeiT-B									
ToMe	45.27	**46.00**	+0.73	44.35	**45.35**	+1.00	43.10	**44.46**	+1.36
EViT	43.57	**44.91**	+1.34	42.45	**44.02**	+1.57	41.11	**43.45**	+2.34
ATS	39.79	**40.50**	+0.71	38.81	**39.66**	+0.85	37.64	**39.11**	+1.47
ATC	44.79	**45.48**	+0.69	43.97	**44.95**	+0.98	42.81	**44.55**	+1.74
PiToMe	44.56	**45.06**	+0.50	43.69	**43.96**	+0.27	42.43	**43.06**	+0.63

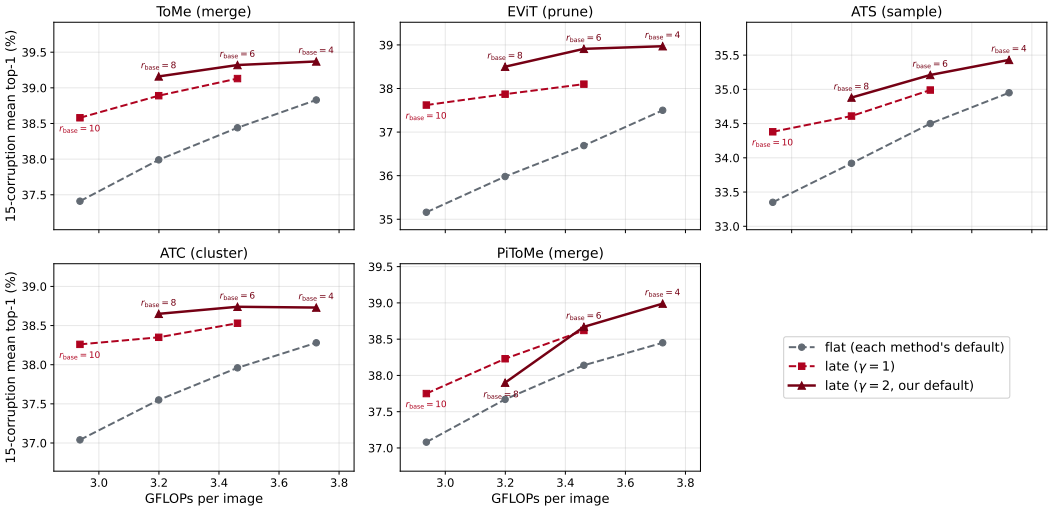


Figure 7: 15-corruption-mean top-1 vs. compute for each of the five training-free reducers (DeiT-S, ImageNet-C severity 5), each curve a schedule swept over compression (points labeled by r_{base}). Flat and late γ=2 at $r_{\text{base}} \in \{4, 6, 8\}$ are the cells of Table 3, late γ=1 at $r_{\text{base}} \in \{6, 8, 10\}$ those of Table 17; each late point is drawn at the GFLOPs of the flat schedule it is calibrated to (its own cost is at or below that).

throughout. Learning rate is selected once per method/backbone from the const-schedule stability protocol and then applied identically to both schedules (no per-schedule tuning): 5×10^{-4}, except EATA at its official rate (1×10^{-4} ViT-S, 1×10^{-3} ViT-B) and ViT-B Tent at 1×10^{-4}; FOA is gradient-free (CMA-ES prompt optimization) and NEO optimizes nothing, so neither has a rate to select.

At both rates late beats flat for every method on both backbones (all 24 cells positive over 3 seeds); the residual seed variance is confined to the const baselines of the gradient methods, where adaptation occasionally collapses on a single corruption×seed run. SPA and NEO carry none of it, SPA because no run of it collapses and NEO because it is deterministic. Table 19 breaks the r_{base}=8 result down by corruption.

Knife-edge collapses. Per corruption, the late schedule wins on the large majority of cells (166/180). For the three gradient methods the only failures under adaptation are *knife-edge* cor-

Table 18: TTA × schedule, per-rate means (γ=2). ImageNet-C sev=5, 15-corruption mean top-1, 3 seeds. full = the same method with no reduction (197 tokens in every block); flat = const (γ=0); late = γ=2, iso-GFLOPs. Δ = late − flat; small ± values are seed stds. (r_{base}=8 also appears in the body, Table 6.)

		$r_{\text{base}} = 4$			$r_{\text{base}} = 8$		
Method	full	flat	late	Δ	flat	late	Δ
ViT-S							
Tent	50.19±0.00	49.27±0.00	50.84±0.00	+1.57±0.00	48.47±0.00	52.04±0.00	**+3.57**±0.00
EATA	55.20±0.02	54.32±0.11	54.84±0.13	+0.52±0.14	52.65±0.16	53.66±0.26	+1.00±0.38
SAR	53.39±0.00	52.77±0.00	53.22±0.00	+0.45±0.00	51.52±0.00	51.96±0.00	+0.43±0.00
FOA	45.27±0.23	43.97±0.28	44.68±0.15	+0.71±0.36	42.42±0.48	42.81±0.30	+0.39±0.60
SPA	53.09±0.29	52.38±0.11	53.12±0.06	+0.74±0.15	51.22±0.09	52.98±0.12	**+1.76**±0.21
NEO	42.66±0.00	42.10±0.00	42.55±0.00	+0.45±0.00	40.97±0.00	41.97±0.00	+1.00±0.00
ViT-B							
Tent	56.32±0.00	55.86±0.00	56.24±0.00	+0.38±0.00	54.49±0.00	55.72±0.00	+1.23±0.00
EATA	59.81±1.78	59.37±2.76	60.50±0.51	+1.13±2.56	59.09±0.62	60.11±0.37	+1.01±0.43
SAR	57.91±0.00	60.95±0.00	61.51±0.00	+0.56±0.00	59.71±0.00	61.07±0.00	+1.35±0.00
FOA	62.16±0.13	61.80±0.02	61.84±0.07	+0.04±0.09	59.80±0.10	60.41±0.14	+0.61±0.09
SPA	65.41±0.03	64.77±0.03	65.27±0.03	+0.50±0.04	63.53±0.08	64.81±0.06	+1.28±0.13
NEO	59.00±0.00	58.23±0.00	58.77±0.00	+0.54±0.00	56.81±0.00	58.16±0.00	+1.34±0.00

Table 19: TTA × schedule, per-corruption top-1 (%) at r_{base}=8, γ=2 (sev=5, 3-seed mean). For each method, flat = const, late = γ=2 at iso-GFLOPs. Knife-edge collapses (< 20%) are where adaptation, not the schedule, fails; the schedule shifts them between methods (e.g. γ=2 repairs Tent's ViT-S zoom, 18.2→45.1). NEO's ViT-S contrast cell is low but is not one of them: it optimizes nothing and so cannot diverge, and it improves that cell over the 14.4 of no adaptation.

		gauss	shot	imp	defoc	glass	motn	zoom	snow	frost	fog	brt	cont	elas	pix	jpeg	mean
ViT-S																	
Tent	flat	42.4	44.3	44.5	46.5	45.2	51.3	18.2	10.9	55.8	60.8	73.4	48.9	57.5	64.2	63.3	48.5
	late	45.5	47.1	47.2	47.7	46.1	52.2	45.1	21.7	56.8	61.3	73.5	51.0	57.4	64.5	63.5	52.0
EATA	flat	42.1	44.1	44.2	46.3	45.6	50.5	46.8	55.5	54.9	59.8	73.6	41.6	57.8	63.9	63.1	52.7
	late	45.0	46.5	46.6	46.7	46.1	51.1	47.4	55.8	55.8	60.6	73.8	44.1	57.7	64.1	63.5	53.7
SAR	flat	40.9	42.5	43.1	45.1	43.9	50.0	45.0	53.5	53.8	58.1	71.9	45.1	56.0	62.5	61.5	51.5
	late	43.7	45.2	45.5	46.2	44.9	50.8	45.7	54.2	54.7	58.8	72.2	37.0	55.8	63.0	61.7	52.0
FOA	flat	31.4	31.9	32.4	36.5	30.2	40.5	34.4	45.4	45.8	50.2	68.4	38.8	40.9	51.4	57.9	42.4
	late	33.2	32.6	34.7	35.8	29.6	40.7	34.4	45.5	46.1	51.2	67.6	39.5	41.7	51.4	58.1	42.8
SPA	flat	43.0	45.2	45.0	41.6	36.6	28.7	49.2	58.2	57.6	60.3	73.7	43.0	60.1	63.8	62.4	51.2
	late	45.6	47.5	47.9	43.0	38.6	35.8	50.0	58.9	58.6	61.5	74.0	45.0	60.8	64.6	62.8	53.0
NEO	flat	33.2	32.8	32.9	34.3	25.3	41.5	34.2	46.1	45.0	49.4	70.4	17.6	41.2	54.1	56.6	41.0
	late	34.8	34.2	34.5	35.3	25.5	42.7	34.8	46.8	46.2	50.4	71.1	19.2	41.6	55.3	57.2	42.0
ViT-B																	
Tent	flat	54.8	55.7	56.3	52.5	44.3	55.9	48.8	61.3	56.1	12.9	77.1	56.2	50.3	67.1	68.0	54.5
	late	57.2	58.2	59.0	54.0	45.1	56.9	50.3	62.6	57.3	11.8	77.6	58.9	50.4	68.0	68.5	55.7
EATA	flat	56.1	58.3	56.1	50.7	54.9	60.3	57.9	67.3	62.7	67.1	77.4	6.5	67.4	72.6	71.2	59.1
	late	57.8	59.4	59.7	52.1	55.6	60.9	58.3	67.6	63.5	68.0	79.1	5.6	68.0	73.8	72.2	60.1
SAR	flat	55.1	56.1	56.3	53.6	50.7	57.7	52.3	62.8	58.4	64.3	77.2	56.2	57.4	69.0	68.6	59.7
	late	57.5	58.5	58.9	54.7	51.4	58.7	53.7	63.9	60.6	66.6	77.8	57.6	57.8	69.6	69.0	61.1
FOA	flat	56.6	57.4	57.6	53.5	43.8	55.8	50.6	64.9	60.5	66.2	77.0	59.7	55.0	68.9	69.4	59.8
	late	58.1	59.4	59.6	54.5	43.4	56.3	49.5	65.1	61.5	65.5	77.5	61.6	53.3	69.9	70.8	60.4
SPA	flat	59.3	60.4	60.9	54.3	52.0	61.9	57.4	68.5	65.9	65.0	79.1	61.0	63.1	73.7	70.7	63.5
	late	61.4	62.7	63.2	55.6	53.0	63.1	58.3	69.4	66.6	66.2	79.4	64.0	63.9	74.3	71.0	64.8
NEO	flat	54.0	54.4	54.6	47.3	37.1	52.5	44.9	62.9	61.4	69.0	77.2	54.1	48.7	66.0	68.1	56.8
	late	56.7	56.6	57.3	49.1	37.6	53.8	46.2	63.7	62.7	70.0	77.6	57.1	48.5	67.0	68.5	58.2

ruptions that collapse to near-chance (< 20%). At γ=2 these shifts are favorable on balance: Tent's ViT-S zoom rises from 18.2% to 45.1%, while Tent's ViT-B fog and EATA's ViT-B contrast are structural, staying near-chance on both schedules at every rate. The schedule thus *moves* these collapses between methods rather than creating them, which is exactly the regime merge-aware sample selection targets (Section 5). The two methods that take no gradient step on the network never collapse. FOA's few late-versus-flat regressions are modest (≤ 2.1pp, e.g. ViT-S defocus and ViT-B elastic) and do not concentrate in any corruption family, and NEO's single regression is ViT-B elastic at −0.2pp. NEO's lowest cell, ViT-S contrast at 17.6 flat and 19.2 late, reads as a collapse only against the 20% line: with nothing to optimize there is nothing to diverge, and no adaptation

at all scores 14.4 there, so NEO improves it and the late schedule improves it again. SPA collapses nowhere and regresses nowhere, the only method in the table with a positive cell for every corruption on both backbones.

What the two token budgets cost. Appendix W measures the schedule at inference. Under adaptation the accounting is per method, since each one wraps the backbone differently. Every cell below is the same method at the same learning rate; only the token budget changes. GFLOPs are counted with PyTorch's flop counter, so a method's extra forward passes, its backward pass and its auxiliary modules are all included, and every cell was identical across ten batches. Times are medians over repeated runs at batch 64 on one GPU, 20 warmup and 100 timed steps, one process per cell, with deterministic kernels disabled; the repeats agreed to within 0.06% at the median.

The saving is the same for every method. GFLOPs fall by 26% throughout, within 0.9pp, because a method's mechanism multiplies the reduced and the unreduced model alike and only the token profile survives: SAR costs exactly twice Tent at either budget, SPA about two and a half times, FOA thirteen to fourteen times, and all of them shed the same fraction. Wall-clock follows at 17.5 to 24.7%. So the composability in Table 6 is not bought back in compute: whichever method is layered on top, the late schedule runs it at about three quarters of the cost of full tokens.

Table 20: Cost of one adaptation step on full tokens against the late schedule (r_{base}=8, γ=2), ImageNet-C severity 5, batch 64. Each percentage is the fraction the quantity shrank. Full tokens = 197 in every block.

	GFLOPs / step			ms / step		
Method	full	late	cut	full	late	cut
ViT-S						
Tent	1216	890	26.8%	156.9	124.5	20.7%
EATA	1216	890	26.8%	171.5	139.2	18.8%
SAR	2431	1781	26.8%	318.5	256.1	19.6%
FOA	15555	11533	25.9%	1927.4	1590.8	17.5%
SPA	3028	2226	26.5%	394.3	319.0	19.1%
NEO	589	435	26.2%	67.5	54.2	19.8%
ViT-B						
Tent	4573	3359	26.5%	459.1	353.4	23.0%
EATA	4573	3359	26.5%	524.4	394.7	24.7%
SAR	9146	6718	26.5%	934.1	719.4	23.0%
FOA	63943	47383	25.9%	6378.6	4984.7	21.9%
SPA	11410	8401	26.4%	1155.8	892.0	22.8%
NEO	2248	1658	26.2%	209.7	161.5	23.0%

M Reduction Distortion Is Front-Loaded for Every Reducer

The *reduction distortion* $D(\ell)$, the output error a single reduction at layer ℓ produces, is the per-step quantity the schedule trades off. We measure it directly for each training-free reducer: starting from the full 197-token DeiT-S, at layer ℓ (and only there) we let the reducer remove a fixed 16 tokens and read the relative change in the output logits, $D(\ell) = \mathbb{E}_x \, \|f^{(\ell)}(x) - f(x)\| / \|f(x)\|$ over N=1024 images (f is the unreduced model). This reads the model output rather than operation-specific merged features, so it applies uniformly to merging, pruning, sampling, and clustering reducers.

The main-text Figure 3a plots $D(\ell)$ under gaussian noise (severity 5) for all five reducers, every curve front-loaded, an early reduction distorting far more than a late one. This holds across the corruption taxonomy: Table 21 gives the early/late ratio per reducer for one corruption from each ImageNet-C group (noise, blur, weather, digital), $\geq 6\times$ in every cell (Spearman against depth ≤ -0.90 throughout). Because the amplification $A(\ell)$ is operation-agnostic (Section 3), this front-loaded $D(\ell)$ is what makes late concentration beneficial for every reducer, not only the merging case in the main text, generalizing the account to the full set of reducers in Table 3.

Table 21: Front-loading of the per-layer reduction distortion $D(\ell)$ under each ImageNet-C group (DeiT-S, N=1024, severity 5): early/late ratio (early = $\ell \in \{1, \ldots, 4\}$, late = $\ell \in \{9, \ldots, 12\}$ means) for one representative corruption per group (noise = gaussian, blur = defocus, weather = fog, digital = jpeg). Every cell is strongly front-loaded (ratio $\geq 6\times$; Spearman against depth ≤ -0.90 throughout).

Reducer	Family	noise	blur	weather	digital
ToMe	merging	31.5	21.1	35.2	20.8
EViT	pruning	13.4	10.5	15.8	13.5
ATS	sampling	10.9	8.9	9.5	10.1
ATC	clustering	38.2	22.6	38.2	21.6
PiToMe	merging	14.2	6.6	9.0	10.4

N Per-Layer Probe: Summary Values

Table 22 gives the exact per-condition summary behind Figure 3b (the amplification curves of Section 3): early- and late-layer means of the amplification $A(\ell)$, with the Spearman rank correlation against layer index over all 12 layers. The strongly negative Spearman values (≤ -0.92) confirm $A(\ell)$ is non-increasing in ℓ (front-loaded). The full per-layer efficiency $\rho(\ell) = D(\ell)/S(\ell)$, from the directly-measured per-layer reduction distortion $D(\ell)$ (Appendix M) and the marginal-saving denominator $S(\ell)$ of Appendix S, is reported in Table 27.

Why $A(\ell)$ is front-loaded (linearized view).

Lemma 1 (Linearized amplification)**.** *To first order, linearizing the layers above ℓ with input–output Jacobians J_k, a perturbation δ at layer ℓ's output reaches the readout as $\left(\prod_{k>\ell} J_k\right)\delta$; its gain is therefore at most the operator norm $\left\|\prod_{k>\ell} J_k\right\|$, attained only along the top singular direction, and the isotropic probe measures the average-case gain $A(\ell) \leq \left\|\prod_{k>\ell} J_k\right\|$.*

In words, an error introduced earlier passes through more layers on its way to the classifier, so it has more opportunity to be amplified. Two consequences follow. The bound $\left\|\prod_{k>\ell} J_k\right\|$ is a product of $L-\ell$ downstream factors, one fewer at each deeper layer, so it is non-increasing in ℓ whenever those layers are on average non-contractive, matching the measured front-loading of $A(\ell)$. And since $\left\|\prod_{k>\ell} J_k\right\| \leq \prod_{k>\ell}\|J_k\|$, a shift that scales the per-layer norms by $(1+\epsilon)$ scales this bound by $(1+\epsilon)^{L-\ell}$, which compounds over more factors at small ℓ, inflating the *early*-layer amplification most and leaving the latest layers almost unchanged (the realized inflation also depends on singular-vector alignment). This depth-compounding of a perturbation through a product of layer Jacobians is the mechanism studied in mean-field analyses of deep networks (Poole et al., 2016; Pennington et al., 2017), here measured in a trained ViT. This reduces the question of why shift raises $A(\ell)$ to the local, measurable claim that corruption raises the downstream Jacobian norms, which Table 22 confirms (early $A(\ell)$ inflates under corruption while late $A(\ell)$ barely moves).

Why the per-layer norms inflate off the clean distribution. Self-attention can amplify off-manifold perturbations in a way a local operator cannot. Standard dot-product self-attention is not Lipschitz (Kim et al., 2021): its output token $o_i = \sum_j p_{ij}\, v_j$ is a convex combination of value vectors with input-dependent weights $p_i = \mathrm{softmax}(q_i K^\top/\sqrt{d})$, and a perturbation reroutes these weights through $\partial p_i/\partial(\text{logits}) = \mathrm{diag}(p_i) - p_i p_i^\top$ scaled by $K/\sqrt{d}$, a term the unbounded bilinear form $q_i K^\top$ leaves unbounded. The coupling is all-to-all, so a change to one token can shift every token's weights, unlike the fixed, bounded Jacobian of a convolution's local receptive field. On the clean manifold this rerouting is organized by training; off it, where the network was never optimized, it is unconstrained, consistent with the early-layer inflation in Table 22.

We stress the scope. This is a capacity argument: it explains why a transformer can amplify off-manifold shifts, not that a given shift must, which component (attention, LayerNorm, or the MLP) dominates, or a signed magnitude per corruption; those the probe measures. The theory supplies only the structural reason amplification is available, while the depth profile (early-most) follows from Lemma 1 under the on-average non-contractive condition and is confirmed empirically (Table 22).

Table 22: Amplification $A(\ell)$ summary (DeiT-S, N=1024, compute-neutral), for clean inputs and one corruption from each ImageNet-C group (severity 5). $A(\ell)$ is front-loaded (early $\gg$ late) in every condition, with clean the flattest, so the network's amplification concentrates a reduction's distortion in the early layers. early $= \ell \in \{1, \ldots, 4\}$, late $= \ell \in \{9, \ldots, 12\}$ (means); Spearman is over all 12 layers.

Condition	early $A(\ell)$	late $A(\ell)$	Spearman
clean	0.60	0.25	−0.92
gaussian noise (noise)	0.92	0.20	−1.00
defocus blur (blur)	0.74	0.20	−1.00
fog (weather)	0.83	0.17	−1.00
jpeg compression (digital)	0.77	0.23	−1.00

O Clean Merge-Path Control: Is the Matching Criterion the Lever?

We test whether the OOD token-reduction penalty is driven by degraded merge *decisions* (which tokens are merged) rather than by the schedule. For each ImageNet-C image we capture, layer by layer, the merge path (the token groupings) that ToMe takes on the image's *clean* counterpart, and force the corrupted image to follow that same "oracle" path, ignoring its own corrupted features when choosing what to merge. The clean path is unavailable at test time (it requires the clean image); it upper-bounds what restoring the uncorrupted matching could recover (a clean-decision oracle, not a proven global optimum). We compare, at iso-GFLOPs, against the standard own-path baseline and against our late schedule. DeiT-S, the 15-corruption ImageNet-C mean (severity 5), full 50k; the own-path column reproduces the canonical flat numbers exactly, a correctness check on the replay. The uncompressed model scores 39.40.

Table 23: Clean merge-path control (15-corruption ImageNet-C mean, severity 5, DeiT-S, iso-GFLOPs). Forcing the clean (oracle) merge decisions recovers some OOD accuracy, but less than the late schedule at every r_{base}. Parentheses give the gain over the own-path baseline; the late schedule uses its own (corrupted) decisions and needs no clean reference.

r_{base}	own-path (flat)	oracle clean-decision	late schedule (γ=2)
4	38.83	39.22 (+0.39)	**39.37** (+0.54)
6	38.44	39.03 (+0.59)	**39.32** (+0.88)
8	37.99	38.79 (+0.80)	**39.16** (+1.17)
10	37.41	38.36 (+0.95)	**38.66** (+1.25)

Even an oracle that uses the uncorrupted image's merge decisions recovers less than the late schedule, which uses its own decisions and needs no clean reference. Relative to this clean-decision oracle, the schedule, not the matching criterion, is the stronger lever, which motivates holding the matching rule fixed while studying allocation. A more corruption-robust matching criterion is a separate design axis that we do not pursue here, and leave to future work (Section 5).

P Robustness to the Schedule's Functional Form

The power law (Eq. 1) is a convenient single-parameter probe of the late-concentrated axis, not a shape we claim is optimal (Section 3). To check that the gain reflects the late-concentrated *direction* rather than the power-law form, we replace the power law with two other monotone late-weighted families at the main operating point (r_{base}=8, DeiT-S, ToMe, ImageNet-C severity 5), each calibrated to the same iso-GFLOPs rule (at or below the flat baseline's GFLOPs) and with its shape parameter set so that its removal center-of-mass matches the γ=2 schedule's: an *exponential* profile $r_\ell \propto e^{\beta \ell / L}$ (β=3; per-layer removals $[2, 3, 4, 6, 7, 9, 12, 16, 21, 28, 36, 48]$), and a non-smooth *two-level step* that removes about 2 tokens per layer in the first half and about 30 per layer in the second half ($[2, 2, 2, 2, 2, 2, 30, 30, 30, 30, 30, 30]$).

Both recover gains on par with the power law (Table 24): the exponential adds $+1.05$pp and the step $+1.18$pp over flat, against the power law's $+1.17$pp, and all 15 corruptions improve under each shape (Table 25). The benefit is thus a property of the late-concentrated direction, not of the power-law parameterization; we keep the power law because its single scalar γ admits the analytic iso-GFLOPs calibration and the continuous concentration sweep used in the mechanism analysis (Figure 2c), not because it is the most accurate shape. The small spread across shapes is itself consistent with the front-loaded-error account (Section 3): the step, which defers early removal most ($\sim$2 tokens per layer in the first half), matches the power law, while the exponential, which removes slightly more in the earliest layers, gives back a little. This is a check at one operating point on one backbone, not an exhaustive search over schedule families.

Table 24: Late-concentration is robust to the schedule's functional form (DeiT-S, ToMe, ImageNet-C sev=5, 15-corruption mean top-1, %; r_{base}=8, inference-only, iso-GFLOPs $\leq$ flat). Each alternative shape is center-of-mass-matched to the γ=2 schedule. Δ = shape $-$ flat.

Schedule	Family	GFLOPs	mean	Δ vs. flat
flat (const r_{base}=8)	–	3.199	37.99	–
power law (γ=2)	power law (convex)	3.147	39.16	+1.17
exponential (β=3)	exponential (convex)	3.194	39.04	+1.05
two-level step	step (non-smooth)	3.178	39.17	+1.18

Table 25: Per-corruption top-1 (%) for the alternative late shapes (DeiT-S, ToMe, sev=5, r_{base}=8, iso-GFLOPs). uncompressed, flat and the power law (γ=2) are the same runs as Figure 5. All 15 corruptions improve over flat under every late shape.

	gauss	shot	imp	defoc	glass	motn	zoom	snow	frost	fog	brt	cont	elas	pix	jpeg	mean
uncompressed	33.89	33.79	34.92	27.73	18.92	32.23	28.41	47.53	53.16	54.22	70.25	38.55	34.75	29.87	52.72	39.40
flat	31.62	31.70	32.40	26.53	18.42	31.11	27.31	46.37	51.10	51.43	69.57	36.59	34.58	28.88	52.19	37.99
power law (γ=2)	33.79	33.58	34.85	27.57	18.72	32.24	28.75	47.62	52.45	52.97	70.18	37.31	34.79	29.74	52.82	39.16
exponential	33.43	33.28	34.53	27.42	18.72	32.11	28.36	47.49	52.42	52.85	70.05	37.43	35.07	29.63	52.77	39.04
two-level step	33.58	33.45	34.59	27.61	18.73	32.20	28.67	47.71	52.53	52.94	70.09	37.87	35.13	29.72	52.79	39.17

Q A Searched Schedule Is Also Late-Concentrated

DiffRate (Chen et al., 2023) searches a per-layer compression rate for a given architecture and FLOP target by differentiable optimization on labeled ImageNet, and is therefore the learned counterpart of the schedule Eq. 1 specifies in closed form; LTMP (Bonnaerens & Dambre, 2023) learns per-layer thresholds whose removal varies per input, a different object. DiffRate reports its searched per-layer rates and shows that they outperform constant schedules on clean ImageNet; it does not characterize the depth profile of those rates or evaluate under distribution shift. Section 3 predicts what an accuracy-maximizing search at fixed compute should find, namely removal concentrated in late layers, heaviest at the last layer where a removal costs almost no accuracy (Appendix T), yet spread over several layers by per-layer capacity.

We ran DiffRate's released search with its default recipe (full ImageNet, three epochs) at our main operating point for the two backbones its code supports. Table 26 lists the searched per-layer removal next to the γ=2 power law at the same budget. Both searched schedules have the predicted shape. The last three layers take 57% of all removal on DeiT-S and 69% on DeiT-B, against 56% for the power law, and the last layer alone takes 38% and 29%, against 22%, so the searched schedules are late-concentrated to a similar degree and heavier still at the final layer. For accuracy we also evaluate a flat schedule at the same budget; all three run in DiffRate's own prune-and-merge implementation, so their numbers are comparable with each other but not with Table 1. At matched THOP GFLOPs on ImageNet-C (severity 5, 15-corruption mean; within 0.4% on DeiT-S, and with 4.8% less compute for the power law on DeiT-B), the fixed γ=2 schedule reaches 36.30 on DeiT-S against 36.82 for the searched schedule and 34.33 for flat, and 43.27 on DeiT-B against 43.59 and 41.17. The one-parameter formula thus delivers most of the gain that a labeled, gradient-based search obtains over flat, with no search, labels, or gradients.

The searched schedule is also tied to the setting it was searched for. It comes from a differentiable search that needs the labeled training set and gradient access to the model, is run separately for each

architecture, token count, and FLOP budget (3.4 and 7.2 GPU-hours for DeiT-S and DeiT-B with the default recipe, paid again for every budget), and yields rates specific to DiffRate's own prune-and-merge operator; the released schedules cover the DeiT and MAE families at 197 tokens and a few budgets. None exists for the ViT-S/B backbones, the video model (1568 space-time tokens), the vision-language encoder (576 tokens), or the four further reducers of this paper. The closed-form schedule needs none of this. Given only the layer count and the budget, it is evaluated in under a millisecond on a CPU and applied unchanged in every setting above.

Table 26: Per-layer tokens removed at the main operating point (DeiT, 197 tokens). DiffRate's searched schedules, from a labeled search run per architecture and budget, and the closed-form γ=2 schedule; all three concentrate removal in the last layers, the searched ones most at the final layer, as Section 3 predicts.

layer	1	2	3	4	5	6	7	8	9	10	11	12	total	last 3
DeiT-S, searched	0	2	6	12	11	12	10	12	18	5	31	72	191	57%
DeiT-B, searched	0	1	2	2	7	8	11	11	17	49	29	55	192	69%
power law, γ=2 (both)	0	1	3	5	7	11	14	19	24	30	36	42	192	56%

R Schedule Placement at the Aggressive Rate

Figure 8 gives the token profiles at r_{base}=12 that accompany the r_{base}=8 panel of Figure 6. The ordering is the same at both rates: early-concentrated front-loads its removals and still uses the most compute (2.710G versus flat's 2.675G and late's 2.666G), yet it loses to flat by 2.23pp, while the late schedule uses the least compute and wins by 1.10pp.

The early-concentrated control is the same power law reflected, so its removal rate falls with depth, and we calibrate its per-layer budget against flat's compute as we do for the late schedules. Because a removal made early frees capacity in every later block, matching flat's cost forces a much smaller total removal: 60 tokens versus flat's 96 at r_{base}=8, and 96 versus 144 here. The calibrated control also ends up slightly above flat rather than below it, 3.290G versus 3.199G at r_{base}=8 and 2.710G versus 2.675G here, so the placement comparison runs against our own claim on both the compute and the token axis.

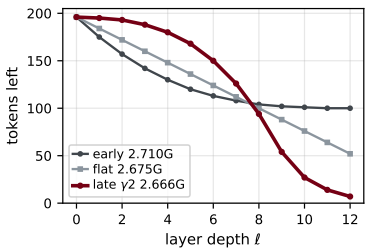


Figure 8: Tokens remaining after each layer at r_{base}=12 (DeiT-S), for the early-concentrated, flat and late-concentrated schedules; the legend gives each schedule's GFLOPs.

S Why the Optimum is Interior: an iso-GFLOPs Account of the Sweet Spot

At a fixed token budget $B = \sum_\ell r_\ell$, the front-loaded per-step distortion (Section 3) is minimized by pushing all removals to the latest layers. Taken alone this would predict ever-more-extreme concentration. The experiments instead show an inverted-U in γ (Table 1, Table 32). The resolution is that our comparisons are at *iso-GFLOPs*, under which the budget B is not a free constant.

The iso-GFLOPs constraint. Because ToMe merges between a layer's attention and its MLP, a removal at layer k shrinks the token stream seen by that layer's own MLP *and* by every subsequent layer. To first order the per-token cost of a layer is constant, so the marginal compute saved by one removal at layer k is

$$S(k) \;=\; c \;+\; (L-k), \tag{2}$$

in units of a full layer, where $(L-k)$ counts the downstream layers and the constant $c = c_{\text{MLP}}/c_{\text{layer}} \approx 0.61$ is the layer's own MLP saving (DeiT-S: MLP $8d^2$ MACs per token over a full layer's $12d^2+2Nd$, with d=384, N=197). Crucially $S(L) = c > 0$: a removal at the last layer still saves that layer's MLP. (The self-attention term, quadratic in the token count, only sharpens the depth dependence: early removals shrink the sequence where attention is most expensive.) The same

form holds for any reducer that removes tokens inside a block, with c the fraction of that block's per-token cost downstream of the removal point; the account uses the shape of $S(k)$, not the value of c. The total compute saved is $C_{\text{saved}} = \sum_k r_k\, S(k)$, and matching a flat baseline's GFLOPs fixes $\sum_k r_k\, S(k) = S_{\text{tot}}$, *not* $\sum_k r_k$.

Optimal allocation is governed by distortion-per-compute. Writing $D(k)$ for the per-layer reduction distortion (Section 3), minimizing the total distortion $\sum_k r_k\, D(k)$ subject to $\sum_k r_k\, S(k) = S_{\text{tot}}$ and $0 \le r_k \le \kappa_k$ (per-layer capacity) is, as a first-order diagnostic relaxation (not an optimization of the actual network), a linear program whose solution fills layers in increasing order of the ratio

$$\rho(k) \;=\; \frac{D(k)}{S(k)}, \tag{3}$$

the distortion incurred per unit of compute saved, until the constraint S_{tot} is met (a water-filling solution). We evaluate $D(k)$ with its directly-measured values (Appendix M); Table 27 reports the resulting ρ. This ratio falls with depth (later reductions are cheaper), and at the last layer it *vanishes* ($D \approx 0$, i.e. $A(L) = 0$): the readout sees only the class token, so a perturbation to the final layer's patch tokens never reaches it (the class token is never merged), while $S(L) = c > 0$. The last layer therefore reduces at near-zero distortion, $\rho(L) \to 0$, and unconstrained the water-filling solution would pour the entire budget into the latest layers. What makes the optimum interior is the per-layer capacity κ_k: only so many mutually similar tokens exist to merge at any one layer, so once the cheap late layers are filled the budget must spill into earlier, higher-ρ layers. The optimal schedule is thus late-concentrated but not maximally so. This is the opposite of a spurious blow-up at the last layer: the endpoint is the cheapest place to reduce, and it is capacity, not efficiency, that bounds the concentration.

The efficient region is late, in clean and corrupted inputs alike. Table 27 evaluates $\rho(\ell)$ with the corrected denominator $S(\ell)$ from the directly-measured per-layer reduction distortion $D(\ell)$ (the probe of Appendix M; DeiT-S, N=1024, L=12). Layer L is the degenerate free reduction ($D \approx 0$, confirming $A(L)$=0, $\rho(L)\approx$0). Among the capacity-bounded interior layers the efficient region is late in both conditions ($\arg\min_{\ell<L} \rho$ at ℓ=11): the distortion is steeply front-loaded already on clean (ρ at ℓ=1 about 11× that at ℓ=11), so late removal is the efficient choice with or without corruption. Corruption sharpens this rather than relocating it: the early-layer distortion inflates (early D rising about 40% from clean to gaussian noise while the late layers stay low), raising ρ at ℓ=1 to about 13× the late value. The efficiency diagnostic thus identifies *where* the budget is efficient (the late, capacity-bounded layers, in every condition); the growth of the *gain* with severity is established separately by the gains themselves (Section 4.2).

Capacity and saturation. Two features keep the optimum interior, at different levels. The hard capacity κ_k in the linear program above already prevents collapse onto the free late layers (the water-filling spill noted above), but with the per-unit distortion held fixed it would fill each layer bang-bang to its cap. A refinement of that fixed-distortion assumption smooths the solution. Treating the per-layer distortion $D(k)$ as independent of r_k is first-order; in practice a layer has a limited supply of mutually similar tokens, so as r_k grows toward κ_k, bipartite matching is forced onto less-similar pairs and the marginal distortion rises, i.e. $D(k)$ increases (and becomes convex) in r_k. We treat this as a modeling assumption rather than a directly measured quantity. This convexity penalizes dumping the entire budget into a few late layers (exactly what large γ does, where the removal budget balloons: its per-layer average rises from 12 at γ=0 to 44 at γ=3 for r_{base}=12), and favors spreading removal smoothly across the efficient late-but-interior band, i.e. a moderate power law.

Dependence on r_{base}. A larger overall budget (higher r_{base}, hence larger S_{tot}) reaches the capacity limits of the most efficient late layers sooner, forcing some removal into earlier, higher-ρ layers; the optimal concentration γ thus *decreases* with r_{base}. Conversely a smaller budget leaves more slack to concentrate, so the optimal γ increases. This matches our measurements: on the 15-corruption mean the optimum migrates from $\gamma \approx 3$ at the lightest r_{base}=2 to $\gamma \approx 1.5$ at r_{base}=12 (Table 1), and on gaussian noise (Table 32) over-concentration (pushing γ past the optimum) already hurts at the most aggressive setting we report (r_{base}=14, where γ=1.5 falls below γ=1.0). The account is first-order; it predicts the *shape* (an interior optimum that shifts with budget), which is what we observe, not a precise $\gamma^\star$.

Table 27: Distortion per unit of compute saved, $\rho(\ell) = D(\ell)/S(\ell)$ ($\times 10^3$), from the directly-measured per-layer reduction distortion $D(\ell)$ (relative output-logit change from removing tokens at layer ℓ only; DeiT-S, N=1024; Appendix M) and the exact marginal saving $S(\ell) = (L - \ell) + c$, c=0.61 (the layer's own-MLP fraction, since ToMe merges before the MLP). The distortion is steeply front-loaded in both conditions and steepens under noise. Layer L=12 is the degenerate free reduction: $D \approx 0$ confirms $A(L)$=0, giving $\rho(L)\approx 0$ despite $S(L)$=c>0. Among the interior layers the minimum (**bold**) is late (ℓ=11) in both conditions; it is per-layer capacity, not a blow-up of ρ, that keeps the optimum off the free last layer. The table is a directional diagnostic, showing the cheap-to-reduce region sits in the late (but capacity-bounded) layers, not a predictor of the exact optimal γ.

layer ℓ	clean	gaussian
1	16.4	25.8
2	12.7	18.4
3	8.3	10.9
4	6.3	5.9
5	4.1	5.7
6	3.4	4.7
7	2.8	3.1
8	2.4	3.1
9	2.7	2.8
10	2.3	2.9
11	**1.5**	**2.0**
12	0.0	0.0

T THE LATE GAIN IS NOT A FINAL-LAYER OR READOUT ARTIFACT

ToMe inserts its merge between a layer's attention and its MLP and protects the class token. At the final layer the attention has already run, so the class token has finished aggregating patch information; merging patch tokens afterward passes only through the token-wise MLP and LayerNorm, with no further patch-to-class attention. A final-layer patch merge therefore cannot change the logits (Section 3: $A(L) = 0$), while still saving that layer's MLP on the discarded patch tokens. Because the late-concentrated schedule places its largest removals at the deepest layers, one might worry that part of its iso-GFLOPs advantage comes for free from this degenerate reduction, or more generally from the class-token readout. We rule this out with three controls.

The final layer is a structurally free reduction. Merging tokens at only the last layer leaves the logits *bit-identical* to the unreduced model (max $|\Delta\text{logit}| = 0$ over 300 gaussian-noise images at severity 5, for every merge count up to the per-layer clamp; a 1.9×10^{-6} rounding residue appears only at an extreme r=40). The same merge at layer 1 changes 29% of predictions (max $|\Delta\text{logit}| = 5.5$), and at layer 10 it is already tiny (0.076). This confirms $A(L) = 0$ is structural (no patch-to-class path remains), not merely small. The strictly-free final-layer saving is itself small (only that layer's MLP on the discarded patches, 0.2–0.7% of total GFLOPs, versus the full downstream stack that early removals save); more to the point, the late gain survives after this endpoint is removed entirely (next paragraph).

Excluding the final layer(s) leaves the gain intact. We drop the last layer (and the last two) from *both* schedules and re-evaluate on the full ImageNet-C validation set at the same operating point as the main comparison (DeiT-S, flat const r_{base}=8 and late γ=2, each recalibrated over the remaining layers so that flat stays at its baseline compute and late at $\leq$ flat; 15 corruptions, severity 5, 50,000 images each; Table 28). The late-over-flat gain is essentially unchanged: +1.17 (full, reproducing the main result) versus +1.15 (exclude last) and +1.09 (exclude last two), positive on all 15 corruptions in every condition. The advantage thus comes from the front-loaded amplification across the interior layers, not from the free final reduction. This matches the distortion account, in which the last layer contributes $r_L\, D(L) = 0$ to the direct error (since $A(L) = 0$), so only the small compute it frees could matter, and the recalibrated control absorbs that.

The effect reproduces under global-average-pooling readout. The free-final-layer phenomenon is specific to class-token readout. Under global average pooling (GAP) the final layer's patch tokens are read out, so $A(L) \neq 0$ and no free reduction exists (merging at only the last layer now shifts the logits, max $|\Delta\text{logit}| = 0.13$). We evaluate two MAE-pretrained models with GAP readout, ViT-B and ViT-L, the setting ToMe itself studied, reusing ToMe's size-proportional pooling. At a matched $\sim 25\%$ GFLOPs reduction (ViT-B r_{base}=8; ViT-L r_{base}=4 over its 24 layers; γ=2 late calibrated to the closest schedule at or below each model's flat GFLOPs), the late schedule beats flat on all 15 corruptions for both models, by $+1.28$ (ViT-B) and $+1.22$ (ViT-L) mean (Table 29, which also lists the compute and token budgets), reproducing the class-token DeiT-S result ($+1.17$) in direction and magnitude. The schedule effect is therefore not specific to the class-token readout.

Table 28: Excluding the final layer(s) from both schedules (DeiT-S, full ImageNet-C, severity 5, 50k images; flat const r_{base}=8 at 3.20 G, late γ=2 at 3.15 G, late $\leq$ flat). flat/late columns are top-1 (%) for the full schedule; Δ = late $-$ flat for the full, exclude-last, and exclude-last-two variants, each recalibrated to the same compute. The full column reproduces Table 2; the gain persists when the (free) final layer(s) are removed.

corruption	flat	late	Δ full	Δ excl-1	Δ excl-2
gaussian noise	31.62	33.79	+2.17	+2.16	+1.89
shot noise	31.70	33.58	+1.88	+1.92	+1.70
impulse noise	32.40	34.85	+2.45	+2.33	+2.21
defocus blur	26.53	27.57	+1.04	+1.11	+1.13
glass blur	18.42	18.72	+0.30	+0.27	+0.29
motion blur	31.11	32.24	+1.13	+1.16	+0.98
zoom blur	27.31	28.75	+1.44	+1.42	+1.12
snow	46.37	47.62	+1.25	+1.15	+1.12
frost	51.10	52.45	+1.36	+1.19	+1.16
fog	51.43	52.97	+1.54	+1.54	+1.46
brightness	69.57	70.18	+0.61	+0.58	+0.53
contrast	36.59	37.32	+0.73	+0.78	+1.16
elastic transform	34.58	34.79	+0.21	+0.24	+0.26
pixelate	28.88	29.74	+0.86	+0.86	+0.69
jpeg compression	52.19	52.82	+0.64	+0.53	+0.60
mean	**37.99**	**39.16**	**+1.17**	**+1.15**	**+1.09**

U What the Late Schedule Recovers: a Two-Channel Decomposition

The body (Table 1) reports that late concentration closes 83% of the accuracy the flat schedule loses to compression. That is an aggregate accounting: in principle a schedule could close the same gap by getting a *different* set of images right. This appendix shows it does not.

Setup. We split every image-corruption case by what the same model does with no reduction at all. In region A the unreduced model is correct, so any error there was manufactured by the reduction and an advantage in A is literally less reduction damage. In region B the unreduced model is already wrong, so an advantage there is the schedule arriving at a new correct answer, which is the "it got lucky" explanation. The two regions partition the flat-to-late accuracy difference exactly,

$$\underbrace{\Pr(A)\big[\text{acc}_{\text{late}}(A) - \text{acc}_{\text{flat}}(A)\big]}_{\text{recovery}} + \underbrace{\Pr(B)\big[\text{acc}_{\text{late}}(B) - \text{acc}_{\text{flat}}(B)\big]}_{\text{new correctness}} = \text{acc}_{\text{late}} - \text{acc}_{\text{flat}},$$

so no part of the measured difference is left unattributed. All pairs are the calibrated iso-GFLOPs schedules of Table 31 (late GFLOPs $\leq$ flat), full ImageNet-C at severity 5, 50,000 images $\times$ 15 corruptions $=$ 750,000 cases per backbone.

The gain is recovery, not new correctness. Table 30 gives the decomposition. In every one of the twelve cells the recovery channel exceeds the net difference and the new-correctness channel is

Table 29: Global-average-pooling readout (MAE ViT-B and ViT-L, full ImageNet-C, severity 5, 50k images). Each model's flat / late columns carry, in the upper rows, its budget (GFLOPs, with the reduction from the uncompressed model in parentheses, 16.9G for ViT-B and 59.7G for ViT-L; tokens removed and final, out of 196 patches) and, in the lower rows, its per-corruption top-1 (%); late is γ=2 calibrated to the closest schedule at or below the flat GFLOPs. At iso-GFLOPs (late $\leq$ flat) late removes more tokens in total yet keeps fewer at the end, and still beats flat on all 15 corruptions for both scales (mean +1.28 / +1.22): the gain is not from a larger token budget and reproduces the class-token result.

	MAE ViT-B			MAE ViT-L		
	flat	late	Δ	flat	late	Δ
Budget						
GFLOPs	12.67 (−25%)	12.62 (−25%)	–	44.98 (−25%)	44.90 (−25%)	–
tokens removed	96	169	–	96	180	–
final tokens	100	27	–	100	16	–
Top-1 accuracy (%) by corruption						
gaussian noise	40.64	44.50	+3.86	54.14	57.08	+2.94
shot noise	41.62	44.57	+2.95	56.91	59.37	+2.46
impulse noise	42.13	45.88	+3.75	56.47	59.24	+2.77
defocus blur	27.74	29.15	+1.41	36.75	37.57	+0.82
glass blur	18.38	18.42	+0.05	24.42	24.78	+0.36
motion blur	35.38	36.23	+0.85	50.83	52.00	+1.16
zoom blur	29.24	30.27	+1.02	43.32	44.16	+0.84
snow	53.04	53.92	+0.87	64.19	65.21	+1.02
frost	54.47	55.12	+0.65	65.04	65.84	+0.80
fog	56.54	57.89	+1.36	60.48	63.22	+2.74
brightness	73.55	73.88	+0.32	78.18	78.35	+0.17
contrast	46.69	46.96	+0.26	54.58	54.61	+0.03
elastic transform	26.19	26.30	+0.10	38.96	39.48	+0.53
pixelate	43.61	45.09	+1.48	49.98	51.34	+1.35
jpeg compression	52.93	53.17	+0.24	60.29	60.55	+0.25
mean	**42.81**	**44.09**	**+1.28**	**52.97**	**54.19**	**+1.22**

Table 30: Two-channel decomposition of the late-minus-flat accuracy difference (pp), full ImageNet-C severity 5. *recovery* = cases the unreduced model already got right; *new correctness* = cases it got wrong. In all twelve cells the recovery channel is positive and larger than the net, and the new-correctness channel is negative: the gain comes from not destroying predictions the model already had, not from finding new ones.

backbone	pair (r_{base}, γ)	net	recovery	new correctness
DeiT-S	4, γ=1	+0.375	**+0.740**	−0.366
DeiT-S	4, γ=2	+0.542	**+1.169**	−0.627
DeiT-S	8, γ=1	+0.902	**+1.257**	−0.355
DeiT-S	8, γ=2	+1.173	**+1.798**	−0.625
DeiT-S	12, γ=1	+1.213	**+1.417**	−0.204
DeiT-S	12, γ=2	+1.103	**+1.331**	−0.228
ViT-B	4, γ=1	+0.348	**+0.685**	−0.337
ViT-B	4, γ=2	+0.576	**+1.238**	−0.662
ViT-B	8, γ=1	+0.991	**+1.325**	−0.334
ViT-B	8, γ=2	+1.567	**+2.111**	−0.544
ViT-B	12, γ=1	+1.774	**+1.780**	−0.006
ViT-B	12, γ=2	+1.224	**+1.249**	−0.025

negative, so late concentration actually concedes ground on cases the unreduced model was already getting wrong and wins purely by preserving the ones it was getting right. At the main operating point (r_{base}=8, γ=2) the unreduced DeiT-S is correct on 295,479 of the 750,000 cases. The flat schedule destroys 8.5% of those predictions (25,017) and ours destroys 3.9% (11,531), at no extra

compute; the intermediate γ=1 sits between them at 5.3%. On ViT-B the rates are 8.0% and 4.2% of 416,401 correct predictions.

The damage is a subpopulation, not a uniform drift. The same fact is visible per image. Figure 9 plots, for each case, the probability the unreduced model assigns to its own predicted class $c^\star$ against the probability the reduced model assigns to that same class, so exact agreement lies on the diagonal. Most cases sit on the diagonal under either schedule; what distinguishes them is a subpopulation that the flat schedule scatters off it. Averaged over the 15 corruptions the mean per-case $\mathrm{KL}(p_{\text{unreduced}} \,\|\, p_{\text{reduced}})$ is 0.085 for flat and 0.029 for ours, and the flat schedule changes the unreduced model's top-1 prediction on 19.1% of cases against ours on 10.8% (ViT-B: 17.4% and 11.0%), with ours lower on 15/15 corruptions for both backbones. The panels also show that this is specific to the corrupted condition: on clean input the two schedules are equally close to the unreduced model, consistent with the clean-to-corrupted ratio of Section 4.2.

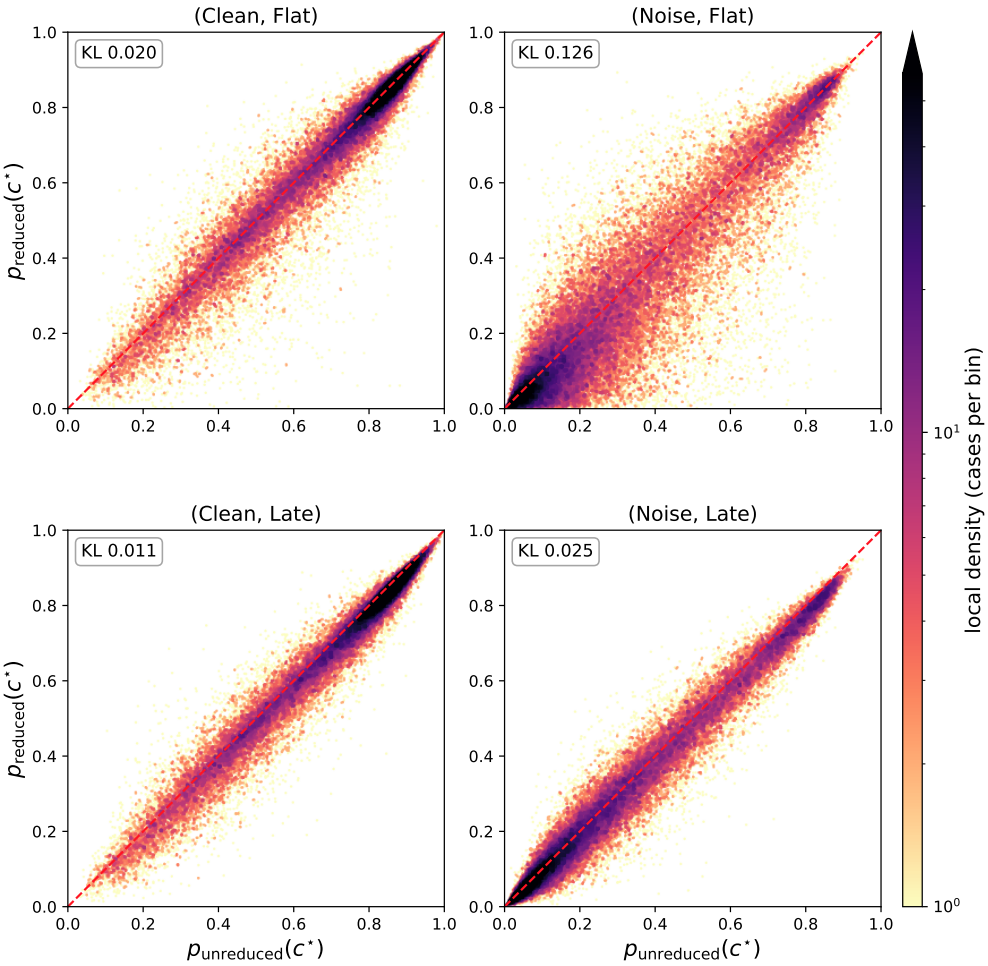


Figure 9: Per-image agreement with the unreduced model (DeiT-S, r_{base}=8, iso-GFLOPs, 50,000 images per panel). Horizontal axis: the probability the unreduced model assigns to its own predicted class $c^\star$; vertical axis: the probability the reduced model assigns to that same class; exact agreement lies on the dashed line. Color is the local density of cases, on a shared scale across panels, saturating above its 75th percentile. Reduction is nearly harmless on clean input under either schedule and remains so for ours under corruption; the flat schedule departs from the unreduced model only where reduction and corruption meet. KL is the mean per-case $\mathrm{KL}(p_{\text{unreduced}} \,\|\, p_{\text{reduced}})$ over the full class distribution on the same input. Gaussian noise at severity 5 is shown for legibility; the aggregate over all 15 corruptions is quoted above.

V CALIBRATION: PER-CELL DETAIL (GAUSSIAN NOISE)

Figure 2 summarizes the iso-GFLOPs calibration; the tables below list the per-cell numbers. Table 32 gives the gaussian-noise sweep; the 15-corruption mean sweep for all six operating points is the body Table 1. Δ = late $-$ flat. The gain is an inverted-U in γ whose optimum shifts to higher γ as r_{base} decreases, and over-concentration begins to hurt (the optimum drops below γ=1.5) at the most aggressive settings we report (r_{base}=14). Table 31 lists the exact calibrated schedules used throughout the paper, confirming that every late-concentrated operating point sits at or below its flat baseline's GFLOPs.

Table 31: Calibrated iso-GFLOPs schedules used throughout the paper. For each operating point r_{base} and exponent γ we list the late-concentrated per-layer-mean removal budget $\bar{r}$, chosen so the schedule's GFLOPs do not exceed the flat (const r_{base}) baseline. Parentheses give the GFLOPs deviation from flat (DeiT-S, THOP): *every entry is* ≤ 0, so every late schedule is at most as expensive as its flat counterpart (strictly conservative). $\bar{r}$ is backbone-independent (197 tokens, 12 layers at 224px), so the same schedules apply to DeiT-S/B and ViT-S/B; the flat column shows DeiT-S GFLOPs. The larger under-shoots at small r_{base} with high γ (e.g. r_{base}=6, γ=3) are an integer-granularity effect and remain conservative.

r_{base}	flat (GFLOPs)	γ=0.5	γ=1.0	γ=1.5	γ=2.0	γ=2.5	γ=3.0
2	2 (3.99G)	3 (-1.6%)	3 (-0.0%)	4 (-1.4%)	4 (-0.4%)	5 (-1.3%)	5 (-0.6%)
4	4 (3.72G)	5 (-0.7%)	6 (-1.1%)	7 (-0.7%)	8 (-0.9%)	9 (-1.1%)	10 (-1.4%)
6	6 (3.46G)	8 (-2.7%)	9 (-1.2%)	10 (-0.2%)	13 (-3.6%)	16 (-5.2%)	20 (-7.6%)
8	8 (3.20G)	10 (-1.2%)	12 (-2.0%)	14 (-1.8%)	16 (-1.6%)	18 (-0.5%)	21 (-0.7%)
10	10 (2.94G)	13 (-3.6%)	15 (-2.2%)	17 (-0.6%)	21 (-1.2%)	26 (-1.8%)	33 (-2.9%)
12	12 (2.67G)	15 (-2.2%)	18 (-1.6%)	22 (-0.4%)	27 (-0.3%)	34 (-0.1%)	44 (-0.4%)

Table 32: iso-GFLOPs calibration on gaussian noise (sev=5, 50k), DeiT-S. const rows are flat baselines; late rows are calibrated to the matching flat GFLOPs. Δ = late $-$ flat.

r_{base}	schedule	gaussian top-1	Δ
6	const	32.35	–
6	late γ=2.0	33.84	+1.49
8	const	31.62	–
8	late γ=1.0	33.12	+1.50
8	late γ=1.5	33.49	+1.86
8	late γ=2.0	33.79	+2.17
10	const	30.73	–
10	late γ=1.0	32.96	+2.23
10	late γ=1.5	33.25	+2.52
10	late γ=2.0	33.44	+2.71
10	late γ=2.5	33.36	+2.63
12	const	29.68	–
12	late γ=0.5	31.41	+1.72
12	late γ=0.75	32.15	+2.47
12	late γ=1.0	32.33	+2.65
12	late γ=1.5	32.64	+2.96
12	late γ=2.0	32.72	+3.04
12	late γ=2.5	32.62	+2.93
12	late γ=3.0	32.44	+2.75
14	const	28.51	–
14	late γ=1.0	31.35	+2.84
14	late γ=1.5	30.99	+2.48

Across operating points the 15-corruption mean optimum migrates from $\gamma \approx 3$ at the lightest r_{base}=2 to $\gamma \approx 1.5$ at the aggressive r_{base}=12 (Table 1), exactly as in Figure 2c. The default γ=2 is the per-point optimum at the main r_{base}=8 (and r_{base}=6) and stays within 0.13pp of the per-point optimum at every operating point we test, which is why a single static γ=2 suffices without per-shift tuning.

W WALL-CLOCK LATENCY AT ISO-GFLOPS

We verify that the iso-GFLOPs match translates into matched real latency, addressing the concern that analytic FLOPs need not predict wall-clock speed. Table 33 reports throughput at the main operating point (r_{base}=8, batch 64) on a single GPU, averaged over 200 timed iterations after 30 warmup iterations. Latency is data-independent (the per-layer token counts are fixed by the schedule, not by the input). The late (γ=2) and flat schedules are wall-clock-indistinguishable (within 0.5%), so every iso-GFLOPs comparison in the paper is also an iso-latency comparison; token reduction itself speeds up inference over the uncompressed model at this throughput-oriented batch size.

Table 33: Wall-clock latency at the main operating point (r_{base}=8), batch 64, single GPU (200 timed iterations, 30 warmup). At iso-GFLOPs the late (γ=2) and flat schedules match in throughput to within 0.5%. The 384-dim row covers DeiT-S and ViT-S; the 768-dim row covers DeiT-B and ViT-B (identical architecture, hence identical latency).

backbone	schedule	GFLOPs	ms / batch	img / s
d=384 (DeiT-S, ViT-S)	uncompressed	4.25	67.1	954
	flat (const r=8)	3.20	54.6	1172
	late (γ=2)	3.15	54.6	1172
d=768 (DeiT-B, ViT-B)	uncompressed	16.86	212.0	302
	flat (const r=8)	12.67	167.0	383
	late (γ=2)	12.46	167.5	382