BANDSTOP: EMERGENT COMPRESSION VIA RESIDUAL FLOW THROTTLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Residual connections turn network depth into a sequence of refinements to a shared representation. However, treating residual branches uniformly provides no structured way to exploit redundancy among these transformations. We introduce Bandstop, a gated residual architecture that imposes an inductive bias toward non-uniform residual branch scaling across depth. Without an auxiliary compression objective or prescribed pruning budget, gradient-based training learns two scalar cutoffs that delimit a contiguous intermediate region in which residual contributions are attenuated. Across architectures, this structure yields emergent compression: the attenuated region can be removed with limited loss in task performance. This learned depth structure improves robustness to post-hoc pruning by up to 8.6 percentage points in mean accuracy over the matched Residual network, with the advantage persisting even after layer replacement. In vision, Bandstop learns lower-rank representations while improving classification accuracy and robustness to input corruption. In language models, Bandstop counters the curse of depth reported for standard residual networks and shows the reasoning-primitive gains associated with depth-grown models. Our results demonstrate that non-uniform layer-wise throttling of residual flow is a simple mechanism for inducing compression during standard training, and establish layer-wise information flow as an important architectural degree of freedom for building efficient and adaptable deep networks.

1 INTRODUCTION

Deep networks build representations through successive transformations. Adding layers gives a model more opportunities to refine those representations, but very deep plain networks can be difficult to optimize (He et al., 2016b). Residual connections address this difficulty by dividing each block into two routes: a skip connection that carries the current representation forward and a residual branch that contributes a learned update (He et al., 2016b;a). Stacking such blocks turns depth into a sequence of updates to a shared representation, with each block determining how much to preserve and how much to transform.

Once greater depth becomes trainable, a different question arises: how much of it performs distinct computation? Comparisons of language models at different depths suggest that additional layers can distribute similar computations more finely across the network (Csordás et al., 2025). Experiments on convolutional residual networks and pretrained Transformers also find that some blocks can be skipped or reordered with limited performance degradation (Veit et al., 2016; Sun et al., 2025a). These observations suggest that deep residual networks distribute learned computation unevenly across depth.

Every layer retained at inference adds to a model's computational and memory costs, so redundant depth offers an opportunity to build smaller, faster models. Post-training pruning pursues this opportunity by finding blocks that can be removed from an already trained network with limited damage to its predictions (Gromov et al., 2025; Chen et al., 2025; Lu et al., 2025). Removing blocks can nevertheless reduce performance, so these methods often fine-tune the remaining network or train a lightweight replacement to recover some of what was lost (Gromov et al., 2025; Chen et al., 2025; Lu et al., 2025). In this approach, a shallower model is obtained by locating and repairing redundant depth.

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.

An alternative is to train for robustness to reduced depth. Stochastic depth exposes a model to bypassed blocks during training, while LayerDrop uses layer dropping to produce transformers that can operate at different depths (Huang et al., 2016; Fan et al., 2020). Auto-Compressing Networks take a different route: their modified connections encourage information to concentrate in earlier layers, allowing substantial depth reduction while retaining performance (Dorovatas et al., 2025). These results suggest that a network's tolerance to layer removal can be shaped during training, providing a structured starting point for post-hoc pruning, layer replacement, or adaptive computation.

Residual networks offer a natural mechanism for shaping this behavior, since the balance between the residual branch and the skip path controls how much each block changes its input. Learned gates have primarily set this balance to improve optimization in deep networks. Highway networks learn transform and carry gates (Srivastava et al., 2015). Residual-gated networks and ReZero learn scalar controls over individual blocks (Savarese, 2016; Bachlechner et al., 2021), while LayerScale learns per-channel residual-branch scales (Touvron et al., 2021). These methods show that a network can learn how strongly each block updates its representation. Combined with the evidence that some trained layers can be removed with little loss, these findings raise a structural question: can residual scaling be constrained so that a network learns not only how strongly its blocks contribute, but also a coherent region of depth whose contribution is dispensable?

To study this, we introduce Bandstop, a gated architecture that scales residual branches non-uniformly according to an initial depth profile. The profile retains an active prefix of early layers and an active suffix of later layers while attenuating a contiguous middle band. Two learned scalar cutoffs determine the boundaries of this band. As the cutoffs shift during training, the model can recruit additional middle layers and adapt its effective depth profile to the task (Figure 2).

By attenuating residual updates in the middle while preserving the identity path, Bandstop biases where changes to the representation occur across depth. The cutoffs are optimized jointly with the network weights under the task loss, without a separate compression objective or fixed pruning budget. After training, the lowest coefficients identify a contiguous candidate for removal without a validation-based search over layer importance. We test whether that candidate is genuinely dispensable by removing it and measuring the resulting performance.

Appendix A provides a broader discussion of prior work on how network depth is parameterized, utilized, and prepared for subsequent compression.

Our contributions are as follows:

- We introduce Bandstop, a two-parameter, non-uniform residual depth profile that enables compression by removing contiguous blocks of depth. At matched retained depth, Bandstop produces pruned subnetworks that outperform the oracle-best contiguous subnetworks of matched ungated residual baselines. We further establish that the structured, non-uniform gating of Bandstop is essential to this compression benefit, as previously proposed residual gating approaches fail to provide comparable gains.
- We demonstrate that Bandstop's learned gate localizes removable depth: it identifies a contiguous removal window without validation-set search, and the resulting subnetworks closely match Bandstop's best contiguous subnetwork at each evaluated removal budget.
- We show that Bandstop learns lower-effective-rank representations in vision models while improving full-depth classification accuracy and increasing robustness to input corruption.
- We find evidence that Bandstop makes greater use of later layers in language models, exhibiting greater measured later-layer influence and higher aggregate reasoning-primitive accuracy while retaining full-depth validation loss comparable to matched Residual models.

2 METHOD

2.1 CONSTRUCTING THE BANDSTOP GATE

Our first task is to enforce residual flow throttling in a typical residual network. Starting from a standard pre-activation residual network of L blocks, $x_\ell = x_{\ell-1} + f_\ell(x_{\ell-1}; W_\ell)$, we scale each

residual branch by a depth-dependent coefficient $\beta_\ell \in (0, 1)$:

$$x_\ell \;=\; x_{\ell-1} \;+\; \beta_\ell \, f_\ell(x_{\ell-1}; W_\ell), \qquad \ell = 1, \ldots, L. \tag{1}$$

Equation 1 shows that β_ℓ controls the magnitude of the residual-branch contribution while leaving the identity path unchanged. When β_ℓ is small, the update is attenuated and the block is biased toward passing its input forward with little change. When β_ℓ is close to one, the block has approximately the same residual scaling as an ungated block. These two regimes motivate a smooth depth profile with regions of stronger and weaker residual flow.

Rather than learning an independent coefficient for every block, we generate the depth profile from two oppositely oriented sigmoid fronts. Let $\sigma(z) = (1 + e^{-z})^{-1}$ and define

$$s_\ell^L = \sigma\left(\frac{c_L - \ell}{\tau_L}\right), \qquad s_\ell^R = \sigma\left(\frac{\ell - c_R}{\tau_R}\right). \tag{2}$$

The left front assigns larger values to early layers, while the right front assigns larger values to late layers. We initialize the cutoffs with $c_L < c_R$ and fix the positive sharpness parameters τ_L and τ_R within each experiment.

We combine the fronts and apply a small positive floor to obtain the coefficient used in Equation 1:

$$g(\ell; \theta) \equiv s_\ell^L + s_\ell^R - s_\ell^L s_\ell^R. \tag{3}$$

The coefficient applied to the residual branch is

$$\beta_\ell = \beta_{\min} + (1 - \beta_{\min}) g(\ell; \theta). \tag{4}$$

When the cutoffs are sufficiently separated, this profile assigns large residual scales to a prefix and suffix and smaller scales to the layers between them. As the cutoffs approach one another, the attenuated region contracts. By analogy with a band-stop filter in signal processing, which attenuates an intermediate frequency band while passing frequencies on either side, we call this profile *Bandstop*.

The cutoffs c_L and c_R are the only learned gate parameters. They are optimized jointly with the network weights under the task loss, without an auxiliary compression objective or a prescribed pruning budget. Moving a cutoff changes which layers receive strong or weak residual scaling. The profile therefore encourages an attenuated middle region, whose actual removability we test after training.

2.2 Why band-stop

The same sigmoid fronts can define several profiles that differ in where they preserve residual capacity.[1] We choose Bandstop because it attenuates a contiguous interior region while retaining active blocks at both ends. This choice reflects the hypothesis that blocks near the input and output have more distinct, less substitutable roles than intermediate blocks.

This hypothesis is consistent with evidence that layers at different depths perform different functions. In ResNets, early blocks have been observed to emphasize representation learning, while later blocks tend to refine features (Jastrzebski et al., 2018). Transformer studies describe distinct phases of processing across depth (Queipo-de-Llano et al., 2025; Lad et al., 2025). Layer-wise interventions find that early and final layers are most sensitive to deletion, whereas middle layers are comparatively robust (Lad et al., 2025). Contiguous-pruning studies similarly identify removable blocks among deeper non-final layers (Gromov et al., 2025).

A related precedent comes from methods that grow networks in the middle. MIDAS and LIDAS expand a model between its existing early and late blocks (Saunshi et al., 2024; Kapl et al., 2025). Bandstop applies the same idea of reserving the middle of the network for adaptable capacity, but within a fixed-depth network: its cutoffs adjust the residual contribution of middle blocks while keeping an active prefix and suffix. Unlike staged-growth approaches, however, Bandstop does not impose an external growth schedule or duplicate layers. Its effective depth profile is learned during training by shifting two scalar cutoffs. This produces a contiguous candidate region for removal, whose actual removability we test after training.

[1] For example such profiles could be: Low-pass, High-pass, and Band-pass profiles, which retain an active prefix, suffix, or middle region, respectively. Appendix D provides their definitions and empirical comparisons; Table 5 summarizes their functional forms.

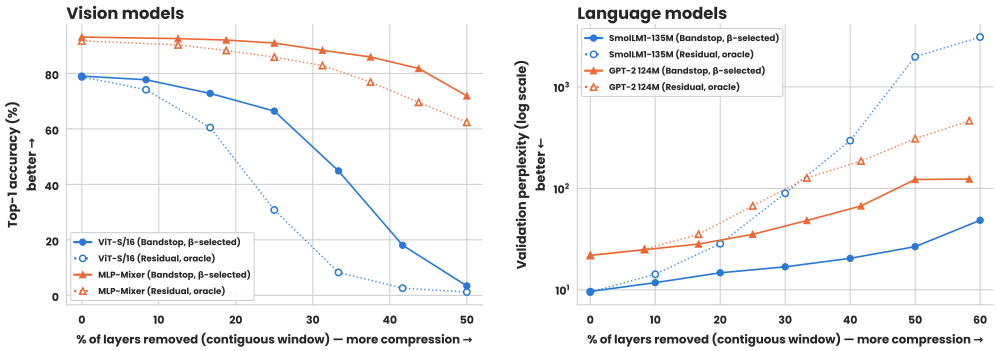


Figure 1: **Bandstop induces robust subnetworks under contiguous layer removal.** We compare Bandstop's β-selected window against the matched Residual model's exhaustive-search oracle. **Left:** vision top-1 accuracy for ViT-S/16 and MLP-Mixer. **Right:** language-model validation perplexity for SmolLM1 and GPT-2 124M. At matched retained depths, the β-pruned Bandstop subnetworks consistently outperform the Residual oracle subnetworks despite requiring no search.

2.3 Experimental Setup

Across all four architectures, we compare Bandstop with a matched ungated residual baseline. For vision, we train a 16-block MLP-Mixer (Tolstikhin et al., 2021) on CIFAR-10 and CIFAR-100, and a 12-block ViT-S/16 following Beyer et al. (2022) on ImageNet-100 and ImageNet-1k. For language modeling, we train a 12-layer, 124M-parameter GPT-2 (Radford et al., 2019; Karpathy, 2024) on approximately 10B tokens of FineWeb-Edu and a 30-layer, 135M-parameter SmolLM1 (Allal et al., 2024) on approximately 200B tokens of the SmolLM corpus. We initialize the cutoffs symmetrically around the network midpoint, giving an initial gate mass $\sum_\ell \beta_\ell \approx 4$ for the 12- and 16-layer models and approximately 9 for SmolLM1. We fix $\tau_L = \tau_R = 0.5$ for the vision models and 0.75 for the language models. The gate floor is $\beta_{\min} = 10^{-4}$ except for GPT-2, where it is 10^{-2}. Full architecture, data-processing, and training details are provided in Appendix C. Sensitivity to the gate hyperparameters and a learned-versus-frozen cutoff ablation are reported in Appendix E. For contiguous layer removal we utilize the two following approaches:

Oracle subnetworks For a network of L layers and a removal budget r, we define the oracle as the contiguous window of r layers whose removal costs the least on the held-out validation split. We find it by exhaustive search: every one of the $L - r + 1$ candidate windows is evaluated on the validation set, and the best-scoring one (highest accuracy, or lowest loss/perplexity) is kept. This search is repeated separately for each architecture, each budget r, and each model variant.

β pruning At a given removal budget r, the β-guided window is read directly off the trained gate: we take the r layers with the lowest β_ℓ, with no search and no access to validation performance. For Bandstop this window is automatically contiguous, since the suppressed band is itself the interval between the two learned cutoffs.

3 Results

We begin by testing Bandstop's central compression claim: whether contiguous layers can be removed and whether the learned gate identifies where to remove them. We then examine accuracy, robustness, and representations in vision models before pruning. Finally, we study depth utilization and downstream behavior in language models.

3.1 Compressing along the depth axis

Bandstop induces contiguous removable depth Across every architecture we test, dropping a contiguous block of layers (window) costs Bandstop substantially less than it costs the matched Residual network. We make this comparison deliberately conservative: the Bandstop window is

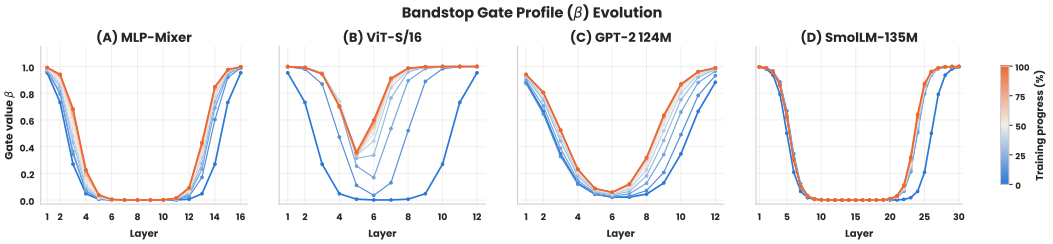


Figure 2: **Progressive depth recruitment across architectures.** All models begin with a broad suppressed middle band. During training, the two cutoffs move toward one another, activating layers from the edges of the band. Where recruitment stalls, a stable low-β interval survives to convergence; where it continues, the profile approaches a fully active network.

selected directly from the learned β profile, whereas the Residual window is selected by exhaustive oracle search. As shown in Figure 1, **the β-pruned Bandstop subnetworks outperform the Residual oracle subnetworks at matched retained depths across all architectures**, demonstrating that Bandstop training induces contiguous removable depth. The gap appears in both modalities: at 25% layer removal, ViT exhibits a 35.6-point accuracy gap (66.4% vs. 30.8%), while removing half of SmolLM1 increases perplexity to only 2.7× its full-depth value for Bandstop, compared with 208× for Residual. Bandstop training therefore produces networks that are substantially more tolerant to contiguous layer removal, even without searching for the best removal window. Additional comparisons with stochastic layer-removal methods and different types of networks are reported in Appendix F.1.

Progressive depth recruitment. Figure 2 shows a common outside-in recruitment trajectory. All models begin with a broad suppressed middle band, which contracts as the cutoffs recruit adjacent layers. Recruitment varies across architectures: MLP-Mixer and SmolLM1 remain close to their compressed initialization, whereas ViT and GPT-2 activate most of their depth. Throughout training, the structured profile biases functional adaptation across depth: a small β_ℓ attenuates both a block's residual contribution and the effect of its parameter updates on the residual stream. Cutoff motion adaptively relaxes this bias by increasing the participation of adjacent layers when favored by the task loss (Appendix B). This depthwise allocation resembles the layer-wise dynamics of ACNs (Dorovatas et al., 2025) and contrasts with the unstructured scaling of ReZero (Bachlechner et al., 2021) and LayerScale (Touvron et al., 2021) (Appendix F.2).

The converged profile localizes removable depth We next test whether the converged profile reliably identifies which layers can be removed. At each removal budget, we compare the β-pruned window with an exhaustive-search oracle. We measure both performance regret relative to the oracle and window overlap, which captures the fraction of selected layers shared by the two windows. As shown in Table 1, the windows overlap by 78%–91% on average, while mean regret is at most 0.95 accuracy points for the vision models and 0.14 nats for the language models. The relative gate profile therefore identifies the same general depth region while achieving near-oracle validation performance. SmolLM1 illustrates why exact boundary recovery is unnecessary: despite having the lowest mean overlap (78%), its mean regret is only 0.02 nats, suggesting a broader region containing several similarly removable windows. Thus, the converged profile provides an effective, search-free pruning signal without needing to reproduce the oracle window exactly.

Table 1: Mean regret and window overlap rate (shared layers / window size) between each Bandstop model's β-selected window and its own exhaustive-search oracle, averaged over the evaluated removal budgets of Figure 1.

Model	Regret	Overlap rate
ViT-S/16	0.95 pp	85%
MLP-Mixer	0.72 pp	91%
SmolLM1-135M	0.02 nats	78%
GPT-2 124M	0.14 nats	86%

Bandstop remains superior after post-hoc healing Having established that Bandstop induces and localizes removable depth, we next ask whether its advantage persists under post-hoc repair.

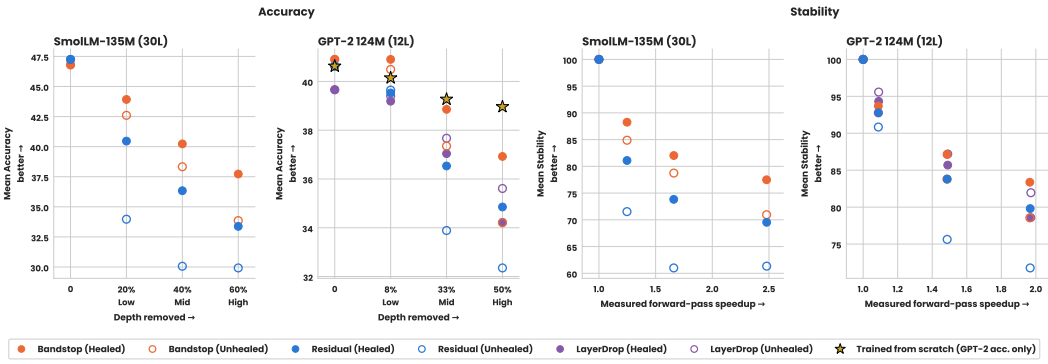


Figure 3: **Comparison under direct removal, post-hoc healing, and training from scratch.** Mean accuracy vs. contiguous depth removed (left two panels) and stability (confidence-weighted agreement with the unpruned model's predictions; Chen et al. (2025)) vs. measured forward-pass speedup (right two panels) for SmolLM1-135M and GPT-2 124M. Unhealed models remove the selected layers directly, whereas healed models replace them with a lightweight MLP under the LLM-Streamline protocol. The GPT-2 panels additionally include LayerDrop, and gold stars in the GPT-2 accuracy panel denote models trained from scratch at the corresponding retained depths. Low/Mid/High denote removal of 6/12/18 layers for SmolLM1 (of its 30 layers) and 1/4/6 layers for GPT-2 (of its 12 layers). Accuracy is averaged over HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), ARC-Easy, ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), and LAMBADA (Paperno et al., 2016).

We apply the same LLM-Streamline pruning-and-repair protocol (Chen et al., 2025) to the matched Residual and Bandstop checkpoints and, for GPT-2, to LayerDrop. Figure 3 compares direct layer removal (*unhealed*) with lightweight post-hoc replacement (*healed*) using mean downstream accuracy and stability. Full methodological and metric details are provided in Appendix C.

Bandstop outperforms Residual after direct removal at every evaluated budget, by up to 8.6 accuracy points and 17.7 stability points. Unhealed Bandstop also matches or outperforms healed Residual except at the most aggressive GPT-2 setting. On GPT-2, healed Bandstop is more accurate than both healed and unhealed LayerDrop throughout. It surpasses the corresponding model trained from scratch at the lowest removal budget and remains the closest pruned model to the from-scratch reference thereafter. For stability, unhealed LayerDrop remains stronger at $1.09\times$ and $1.49\times$ speedup, while healed Bandstop leads at $1.97\times$. Notably, healing improves both accuracy and stability for Bandstop at every evaluated setting but degrades both metrics for LayerDrop, demonstrating Bandstop's greater adaptability under post-pruning reconstruction.

3.2 Representation structure and full-depth generalization

The preceding results establish compression along the depth axis: Bandstop concentrates useful computation into a smaller effective subnetwork and identifies a contiguous span that can be removed with limited loss. Computational redundancy, however, does not by itself reveal how the retained network organizes information. We therefore ask whether residual-flow throttling also compresses the representations propagated through depth, and whether this representational change is associated with improved generalization.

Lower effective rank in vision models We measure the effective rank of the hidden representation h_ℓ throughout training, using the entropy of its covariance spectrum. Figure 4 reports the difference between the matched Residual and Bandstop models, so positive values indicate that Bandstop uses a lower-rank representation. In both architectures, Bandstop develops substantially lower effective rank over a contiguous range of depths. For the MLP-Mixer, the difference extends across much of the middle and late network, while for ViT-S/16 it is concentrated in the latter half and is strongest near the final blocks. The difference emerges during training and persists to convergence. Thus, Bandstop does not simply make layers removable: it changes the geometry of the solution, concentrating the learned representation into fewer effective directions.

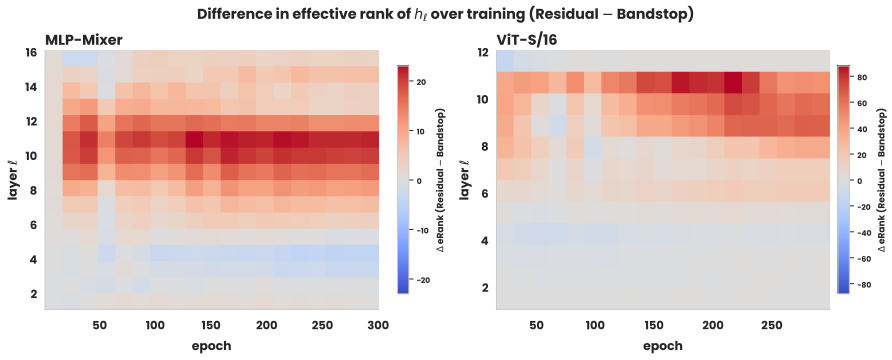


Figure 4: **Bandstop learns lower-rank hidden representations.** Effective-rank difference (Residual minus Bandstop) across training for MLP-Mixer (left) and ViT-S/16 (right). Positive values indicate lower rank under Bandstop.

Compressed representations, improved accuracy and robustness A lower-rank representation need not be more useful, since excessive compression could discard task-relevant variation and reduce predictive performance. We therefore first evaluate the full-depth models, before removing any layers. Table 2 shows that Bandstop improves accuracy across all settings, including a 4.9-point gain on CIFAR-100, while also reducing variation across seeds. We next examine how the two ImageNet-1k models respond to input noise. Following the random-pixel protocol of Sanyal et al. (2018), each spatial pixel is selected with probability $p \in \{0.4, 0.6, 0.8, 1.0\}$ and perturbed by additive Gaussian noise sampled from $\mathcal{N}(0, 128/255)$. We also evaluate both variants on ImageNet-C (Hendrycks & Dietterich, 2019). With comparable clean ImageNet-1k accuracy across three seeds (Table 2), Bandstop improves random-pixel corruption accuracy by 1.0–1.2 points across the tested probabilities and improves mean ImageNet-C accuracy by 0.46 points (Table 3). These robustness gains occur alongside the lower effective rank in Figure 4, suggesting a relationship between representational compression and reduced sensitivity to input perturbations.

Table 2: **Full-depth classification performance.** Top-1 accuracy (%) of matched models before pruning. We report mean and standard deviation over three seeds.

Setting	Residual	Bandstop
Mixer / CIFAR-10	91.7 ± 0.2	$\mathbf{93.1 \pm 0.1}$
Mixer / CIFAR-100	66.6 ± 1.0	$\mathbf{71.5 \pm 0.3}$
ViT / ImageNet-100	80.5 ± 0.5	$\mathbf{81.5 \pm 0.1}$
ViT / ImageNet-1k	78.8 ± 0.2	$\mathbf{79.0 \pm 0.1}$

Table 3: **Robustness to input corruptions on ImageNet-1k.** Final-checkpoint top-1 accuracy (%), each averaged over three independent training runs. Random-pixel results also average over three corruption noise draws per run; ImageNet-C averages 15 corruptions and 5 severities per run. Full corruption-protocol details are provided in Appendix C

Model	$p = .4$	$p = .6$	$p = .8$	$p = 1$	IN-C
Residual	48.94	36.89	25.62	15.58	55.58
Bandstop	**49.97**	**38.12**	**26.78**	**16.57**	**56.04**

3.3 Depth use and behavior in language models

The results on vision models show that Bandstop changes both the depth and the representations used by the network. We next examine how it allocates computation across depth in a language model. We first test whether later layers have greater influence on predictions, then compare its reasoning-primitive performance and use of in-context examples with patterns reported for depth-grown models. Further language model performance and evaluation metrics are presented in Appendix H.

Bandstop increases later-layer influence The *Curse of Depth* describes the limited influence that late layers of standard Pre-LN Transformers can have on final predictions (Sun et al., 2025b; Csordás et al., 2025). Sun et al. (2025b) propose growth in residual-stream variance as one explanation. We

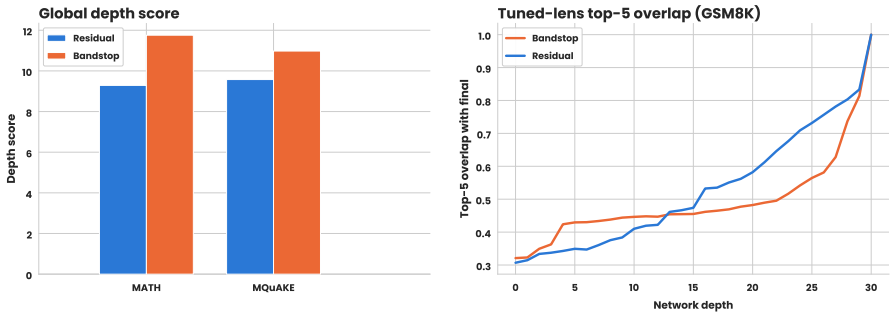


Figure 5: **Later-layer influence in SmolLM1-135M. Left**: Logit-effect depth score on MATH and MQuAKE; higher values place greater weight on later-layer effects. **Right**: Top-5 overlap between Tuned Lens predictions at each depth and the final prediction on 100 GSM8K examples. Bandstop has lower overlap through much of the second half of the network.

ask whether Bandstop changes this depth-use profile: attenuating middle-layer residual updates while retaining active later layers may encourage training to use those later layers more.

Following Csordás et al. (2025), we compute the logit-effect depth score on MATH (Hendrycks et al., 2021) and MQuAKE (Zhong et al., 2023). We also measure top-5 overlap between intermediate Tuned Lens predictions and the final prediction on GSM8K (Cobbe et al., 2021). The Tuned Lens adapters (Belrose et al., 2023) are trained on a held-out split of FineWeb-Edu following Kapl et al. (2025). Figure 5 shows higher depth scores for Bandstop on both MATH (11.76 vs 9.29) and MQuAKE (10.98 vs 9.57), indicating increased computation at deeper layers of the network. Across most of the second half of the network, Bandstop's intermediate Tuned Lens predictions have lower top-5 overlap with its final-layer predictions than those of the Residual model. This indicates that Bandstop's predictions become aligned with the final prediction later in depth, consistent with later layers making larger prediction-relevant changes. Together, these results show that Bandstop's progressive depth recruitment is associated with greater utilization of deeper layers, with later computation playing a more substantial role in the final prediction.

Similarities with depth-grown models Middle-depth growth methods such as MIDAS (Saunshi et al., 2024) and LIDAS (Kapl et al., 2025) likewise exhibit increased later-layer utilization while expanding the network by inserting copies of intermediate layers during staged training. Bandstop instead operates at fixed depth, assigning larger learned residual scales to input- and output-side blocks than to the suppressed middle, which is progressively recruited during training (Figure 2). Thus, newly inserted intermediate layers in depth-grown models contribute for a shorter portion of training, whereas Bandstop's middle layers contribute more weakly throughout training because of residual attenuation.

Table 4: **Five-shot reasoning-primitives accuracy** (%, mean ± std over three seeds). Chance is 10% for copying and 20% for depth-0 variable assignment; assignment depths 1–2 are reported in Appendix G.1.

SmolLM1-135M	**Copying**		**Variable assignment**			**Mean**
Model	Random words	Real words	Basic	Code	Math	
Residual	**12.07** ± 1.31	12.57 ± 1.46	20.53 ± 0.83	24.83 ± 1.96	25.47 ± 1.68	19.09
Bandstop	8.40 ± 0.08	**13.40** ± 1.18	**24.23** ± 1.87	**29.40** ± 0.85	**29.30** ± 0.50	**20.95**

We next ask whether Bandstop also shares the aggregate reasoning-primitive gains reported for depth-grown models. Following Saunshi et al. (2024); Kapl et al. (2025), we evaluate copying and variable assignment. Across the five tasks in Table 4, Bandstop raises mean accuracy from 19.09% to 20.95%, a gain of 1.86 points. It improves all three depth-0 variable-assignment formats and real-word copying, while scoring lower on random-word copying. Taken together, these results show that Bandstop improves aggregate reasoning-primitive accuracy over the matched Residual model.

This gain is achieved through a structured learned residual scaling, closely related to the scheduled depth increase of growing models, but without layer duplication.

In-context learning Depth-grown and middle-looped models have also shown gains from additional in-context examples (Kapl et al., 2026). We test whether Bandstop shows a similar pattern on reasoning primitives by varying the number of examples in the prompt. Figure 6 tests this hypothesis on both types of the reasoning-primitive tasks. On variable assignment, both models start from the same zero-shot performance; however, as additional in-context examples are introduced, Residual's performance decreases while Bandstop gains a notable advantage. At copying (real words) both models benefit from additional few-shot examples up to around 5, but after that point additional examples do not improve performance. Overall, additional examples benefit Bandstop especially at variable assignment, while for copying Bandstop is on par with Residual or slightly worse. Due to space limitations, results for the remaining reasoning-primitive tasks are reported in Appendix G.2.

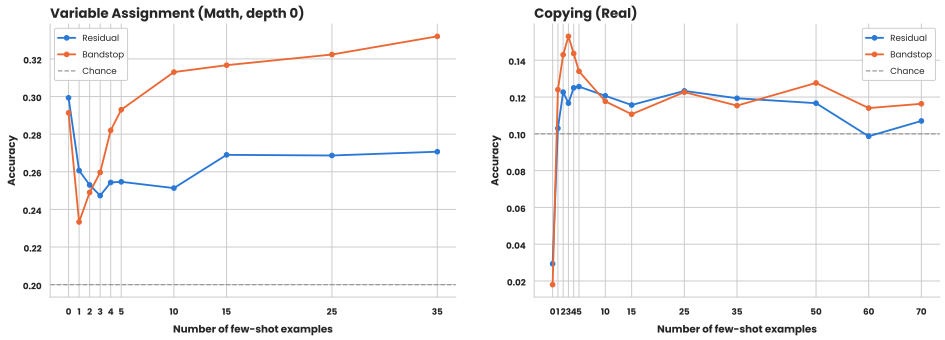


Figure 6: **In-context adaptation, Residual vs. Bandstop (SmolLM1-135M)** Accuracy vs. number of few-shot examples. **Left**: variable assignment (math, depth 0). **Right**: copying (real words). The curves show the mean across 3 evaluation seeds.

4 Discussion & Limitations

We introduced Bandstop, a two-parameter residual depth profile that attenuates a contiguous middle band while keeping the blocks near the input and output active. Training moves the two cutoffs under the task loss alone, recruiting middle blocks from the band's edges only when they reduce the loss. The learned profile then points to a contiguous region to test for removal.

Across all four tested architectures, subnetworks selected from Bandstop's gate outperform the best validation-selected contiguous subnetworks of matched Residual models at the same retained depth. The selected windows also perform close to Bandstop's own best contiguous windows, and the advantage on language models persists after lightweight layer replacement. In vision, Bandstop learns lower-rank representations and improves clean accuracy, with additional gains under input corruption. In SmolLM1, later layers have greater measured influence and aggregate reasoning-primitive accuracy is higher. This combination makes Bandstop useful when the desired inference depth is unknown during training. A single trained checkpoint can yield subnetworks at different removal budgets without searching the validation set for a window each time. The extracted model uses fewer blocks and can be deployed directly or given a lightweight replacement for the removed region. Because the gate adds only two learned parameters and retains the usual residual connections, the approach can be applied to established architectures with little change to their structure.

While we demonstrate the effectiveness of Bandstop across selected vision and language architectures, several limitations remain. Our evaluation is restricted to relatively small models, and we do not directly measure hardware-specific performance or deployment-level cost savings. Future work should investigate whether the observed robustness to depth pruning extends to larger and more diverse architectures, as well as quantify the resulting gains on real hardware. If these results persist at scale, Bandstop could potentially reduce deployment costs and broaden access to capable, adaptive models in resource-constrained settings.

AI USE STATEMENT

Generative AI tools were used to help edit and polish the manuscript, including improvements to clarity, grammar, style, and readability. Their use was limited to the presentation of the work and did not replace the authors' judgment about its scientific content or conclusions. All AI-assisted changes were reviewed and approved by the authors, who take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

The Bandstop formulation and pruning procedures are described in Section 2, while architecture, data-processing, training, and evaluation details are provided in Appendix C. Further examination of gate hyperparameters and ablations is provided in Appendix E.

ETHICS STATEMENT

This work does not involve human subjects or personally identifiable data, and relies solely on publicly available benchmark datasets. We do not identify any ethical issues arising from this work.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. Smollm-blazingly fast and remarkably powerful. *Hugging Face Blog*, 16, 2024.

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Gary Cottrell, and Julian J. McAuley. Rezero is all you need: fast convergence at large depth. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1352–1361. AUAI Press, 2021. URL https://proceedings.mlr.press/v161/bachlechner21a.html.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *CoRR*, abs/2303.08112, 2023. URL https://doi.org/10.48550/arXiv.2303.08112.

Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Better plain vit baselines for imagenet-1k. *CoRR*, abs/2205.01580, 2022. URL https://doi.org/10.48550/arXiv.2205.01580.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6239. URL https://doi.org/10.1609/aaai.v34i05.6239.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. Streamlining redundant layers to compress large language models. In *International Conference on Learning Representations*, volume 2025, pp. 30362–30383, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/4b00a351b41358965613c118e87dc28c-Paper-Conference.pdf.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Róbert Csordás, Christopher D. Manning, and Christopher Potts. Do language models use their depth efficiently? In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/eae3af0f5868f0a2eceb74208966d55b-Abstract-Conference.html.

Evangelos Dorovatas, Georgios Paraskevopoulos, and Alexandros Potamianos. Auto-compressing networks. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/9ff1ac9a659085fed0735362cafe5e53-Abstract-Conference.html.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020. URL https://openreview.net/forum?id=SylO2yStDr.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers. In *International Conference on Learning Representations*, volume 2025, pp. 81906–81920, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/cbabc2f70de2dd09f491a8715ec3e80f-Paper-Conference.pdf.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908, pp. 630–645, 2016a. URL https://doi.org/10.1007/978-3-319-46493-0_38.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778, 2016b. URL https://doi.org/10.1109/CVPR.2016.90.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The*

Netherlands, October 11-14, 2016, Proceedings, Part IV, volume 9908, pp. 646–661, 2016. URL https://doi.org/10.1007/978-3-319-46493-0_39.

Stanislaw Jastrzebski, Devansh Arpit, Nicolas Ballas, Vikas Verma, Tong Che, and Yoshua Bengio. Residual connections encourage iterative inference. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=SJa9iHgAZ.

Ferdinand Kapl, Emmanouil Angelis, Tobias Höppe, Kaitlin Maile, Johannes von Oswald, Nino Scherrer, and Stefan Bauer. Do depth-grown models overcome the curse of depth? an in-depth analysis. *CoRR*, abs/2512.08819, 2025. URL https://doi.org/10.48550/arXiv.2512.08819.

Ferdinand Kapl, Emmanouil Angelis, Kaitlin Maile, Johannes von Oswald, and Stefan Bauer. From growing to looping: A unified view of iterative computation in llms. *CoRR*, abs/2602.16490, 2026. URL https://doi.org/10.48550/arXiv.2602.16490.

Andrej Karpathy. build-nanogpt: Reproducing GPT-2 (124m) from scratch. https://github.com/karpathy/build-nanogpt, 2024.

Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. Remarkable robustness of LLMs: Stages of inference? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Wxh5Xz7NpJ.

Yao Lu, Yuqi Li, Wenbin Xie, Shanqing Yu, Qi Xuan, Zhaowei Zhu, and Shiping Wen. The structural scalpel: Automated contiguous layer pruning for large language models. *CoRR*, abs/2510.23652, 2025. URL https://doi.org/10.48550/arXiv.2510.23652.

Wojciech Masarczyk, Mateusz Ostaszewski, Ehsan Imani, Razvan Pascanu, Piotr Milos, and Tomasz Trzcinski. The tunnel effect: Building data representations in deep neural networks. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/f249db9ab5975586f36df46f8958c008-Abstract-Conference.html.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pp. 20192–20204. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-ACL.1035. URL https://doi.org/10.18653/v1/2025.findings-acl.1035.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/V1/P16-1144. URL https://doi.org/10.18653/v1/p16-1144.

Enrique Queipo-de-Llano, Alvaro Arroyo, Federico Barbero, Xiaowen Dong, Michael M. Bronstein, Yann LeCun, and Ravid Shwartz-Ziv. Attention sinks and compression valleys in llms are two sides of the same coin. *CoRR*, abs/2510.06477, 2025. URL https://doi.org/10.48550/arXiv.2510.06477.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sashank J. Reddi, Sobhan Miryoosefi, Stefani Karp, Shankar Krishnan, Satyen Kale, Seungyeon Kim, and Sanjiv Kumar. Efficient training of language models using few-shot learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14553–14568. PMLR, 2023. URL https://proceedings.mlr.press/v202/j-reddi23a.html.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *15th European Signal Processing Conference, EUSIPCO 2007, Poznan, Poland, September 3-7, 2007*, pp. 606–610. IEEE, 2007. URL https://ieeexplore.ieee.org/document/7098875/.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models. *Comput. Speech Lang.*, 77:101429, 2023. doi: 10.1016/J.CSL.2022.101429. URL https://doi.org/10.1016/j.csl.2022.101429.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8732–8740. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6399. URL https://doi.org/10.1609/aaai.v34i05.6399.

Amartya Sanyal, Varun Kanade, and Philip H. S. Torr. Low rank structure of learned representations. *CoRR*, abs/1804.07090, 2018. URL http://arxiv.org/abs/1804.07090.

Nikunj Saunshi, Stefani Karp, Shankar Krishnan, Sobhan Miryoosefi, Sashank Jakkam Reddi, and Sanjiv Kumar. On the inductive bias of stacking towards improving reasoning. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/837bc5db12f3d394d220815a7687340c-Abstract-Conference.html.

Pedro H. P. Savarese. Learning identity mappings with residual gates. *CoRR*, abs/1611.01260, 2016. URL http://arxiv.org/abs/1611.01260.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015. URL http://arxiv.org/abs/1505.00387.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. Transformer layers as painters. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 25219–25227, 2025a. URL https://doi.org/10.1609/aaai.v39i24.34708.

Wenfang Sun, Xinyuan Song, Pengxiang Li, Lu Yin, Yefeng Zheng, and Shiwei Liu. The curse of depth in large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025b. URL http://papers.nips.cc/paper_files/paper/2025/hash/eeb57fdf745eb31a3c7ef22c59a4661d-Abstract-Conference.html.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, volume 12356 of *Lecture Notes in Computer Science*, pp. 776–794. Springer, 2020. URL https://doi.org/10.1007/978-3-030-58621-8_45.

Ilya O. Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 24261–24272, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/cba0a4ee5ccd02fda0fe3f9a3e7b89fe-Abstract.html.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 32–42. IEEE, 2021.

doi: 10.1109/ICCV48922.2021.00010. URL https://doi.org/10.1109/ICCV48922.2021.00010.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Andreas Veit, Michael J. Wilber, and Serge J. Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 550–558, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/37bc2f75bf1bcfe8450a1a41c200364c-Abstract.html.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 15686–15702. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.971. URL https://doi.org/10.18653/v1/2023.emnlp-main.971.

A Related Work

Bandstop draws on four lines of work: residual gating and depth parameterization, depth redundancy and post-hoc layer pruning, training-time preparation for reduced depth, and studies of depth utilization, including progressive model growth. Together, these areas examine how computation is distributed across network depth, why some layers become weakly used or redundant, and how training can reshape that distribution. We review each in turn and position Bandstop as a fixed-depth method that learns a structured residual profile, preserves active endpoint layers, and creates a contiguous middle interval that can be removed after training.

Residual gating and depth parameterization. Residual gates control how strongly each block updates its input, and often improve optimization by keeping the network close to an identity mapping early in training. This idea was introduced in Highway Networks through complementary transform and carry gates, which allow information to bypass transformations that are not yet useful (Srivastava et al., 2015). It was later simplified to a single scalar gate per block in Gated Residual Networks, making identity mappings easier to learn and increasing independence between blocks (Savarese, 2016). A similar block-wise parameterization underlies ReZero, with each residual scale initialized to zero so that training begins at dynamical isometry with improved conditioning (Bachlechner et al., 2021). Finer-grained control is provided by LayerScale, which learns a small initial scale for each channel and thereby stabilizes very deep vision Transformers (Touvron et al., 2021). While these methods learn largely independent residual scales, Bandstop imposes a structured profile across depth. Two learned cutoffs preserve an active prefix and suffix while attenuating a contiguous middle interval, making that interval a natural candidate for removal.

Depth redundancy and post-hoc layer pruning. Layer-removal studies show that substantial portions of pretrained Transformers can sometimes be removed with limited downstream degradation (Sajjad et al., 2023; Men et al., 2025; Gromov et al., 2025). Early approaches typically assess the importance of individual layers. For example, ShortGPT (Men et al., 2025) ranks blocks by their influence on hidden states. More recent methods exploit the structure of contiguous depth. LLM-Streamline (Chen et al., 2025) selects a consecutive low-importance interval and can learn a

lightweight module to replace it, while Structural Scalpel (Lu et al., 2025) optimizes a two-sigmoid contiguous mask of a specified width over frozen weights before tuning the layers adjacent to the removed interval. These approaches, together with the contiguous layer-removal strategy studied by Gromov et al. (2025), are complementary to Bandstop because its learned gate exposes the same form of structured removal opportunity. The key distinction is when that structure is identified: post-hoc methods locate a removable interval in a pretrained network, whereas Bandstop jointly learns the network weights and two free endpoints under the task loss, asking whether training itself can create a removable interval.

Training-time preparation for reduced depth. Rather than identifying redundant layers only after training, several methods prepare networks to tolerate reduced depth during training. Stochastic Depth (Huang et al., 2016) randomly bypasses residual blocks, improving optimization and regularizing the resulting network. LayerDrop (Fan et al., 2020) applies structured dropout to Transformers so that fixed shallower subnetworks can be extracted at inference time. Auto-Compressing Networks (Dorovatas et al., 2025) take a more directed approach, modifying the network's connections to encourage information to concentrate in earlier layers and thereby support static truncation. Bandstop shares the goal of shaping representations during training so that a static reduced-depth network can be extracted afterward. Instead of preparing the network for prefix truncation or arbitrary layer dropping, however, it learns two endpoints that organize the residual updates around a contiguous removable middle interval while preserving active computation at both ends.

Depth utilization and model growth. Increasing model depth does not necessarily produce proportionally deeper computation. In conventionally trained language models, layers in the second half can contribute substantially less to the residual stream, while larger models often distribute similar computations more finely across additional layers rather than learning qualitatively deeper compositions (Csordás et al., 2025). This phenomenon, termed the *curse of depth* (Sun et al., 2025b), has been linked in Pre-LN Transformers to the growth of residual-stream variance, which drives later blocks toward identity-like behavior and reduces their influence during training. Complementary intervention studies find that middle layers are comparatively uniform and that many can be deleted, exchanged, or reordered with limited degradation, while early and final layers remain more functionally distinct (Lad et al., 2025; Sun et al., 2025a).

Progressive stacking offers an alternative training procedure: it constructs deeper models in stages from previously trained shallower models, reducing pretraining cost (Reddi et al., 2023). MIDAS further shows that growing a model by duplicating middle blocks induces gains on reasoning benchmarks and reasoning primitives despite similar or slightly worse language-modeling perplexity (Saunshi et al., 2024). Subsequent analysis connects these benefits directly to depth utilization: gradual middle-layer growth increases the influence of later layers, changes the organization of the residual stream, and encourages the formation of more modular computational units, thereby mitigating the curse of depth (Kapl et al., 2025). Bandstop addresses the same depth-utilization problem without staged growth. It trains a fixed-depth model while learning how residual contribution is distributed across depth, preserving an active prefix and suffix while attenuating a contiguous middle region.

B Optimization Dynamics and Emergent Redundancy

This section provides an optimization-based interpretation of the depth profiles in Figure 2. We separate the persistent effect of the gate values from the adaptive effect of cutoff motion. Small gates reduce the functional participation of suppressed blocks throughout training; when the cutoffs move, they selectively relax this attenuation near the transition fronts. This distinction explains how the structured profile shapes functional development across depth. A small gate alone does not certify removability, which is established empirically in Section 3.

B.1 Attenuation of residual writes

For block ℓ, define its contribution to the residual stream as

$$r_\ell = \beta_\ell f_\ell(x_{\ell-1}; W_\ell). \tag{5}$$

The block weights, gate, and block input can all change during an optimization step. A first-order expansion therefore gives

$$\Delta r_\ell \approx \underbrace{\beta_\ell \frac{\partial f_\ell}{\partial W_\ell} \Delta W_\ell}_{\text{weight-induced change}} + \underbrace{f_\ell(x_{\ell-1}; W_\ell)\Delta\beta_\ell}_{\text{gate-induced change}} + \underbrace{\beta_\ell \frac{\partial f_\ell}{\partial x_{\ell-1}} \Delta x_{\ell-1}}_{\text{input-induced change}}. \tag{6}$$

The first and third terms are explicitly multiplied by β_ℓ. Consequently, for a fixed parameter or input displacement, changes to a suppressed block have a smaller direct effect on the residual stream than corresponding changes to an active block. The second term describes changes caused by movement of the shared depth profile.

The gate also appears in the gradient received by the block weights. Writing $\delta_\ell = \partial\mathcal{L}/\partial r_\ell$, the chain rule gives

$$\frac{\partial \mathcal{L}}{\partial W_\ell} = \beta_\ell \left(\frac{\partial f_\ell}{\partial W_\ell}\right)^\top \delta_\ell. \tag{7}$$

Thus, small gates attenuate both the block's current residual contribution and the task gradient transmitted to its branch parameters. Under plain gradient descent, $\Delta W_\ell = O(\beta_\ell)$, so the weight-induced term in Equation 6 is $O(\beta_\ell^2)$. Under AdamW, however, the first- and second-moment estimates scale approximately as $m_\ell = O(\beta_\ell)$ and $\sqrt{v_\ell} = O(\beta_\ell)$ when β_ℓ varies slowly over the moment-estimation window. Their ratio therefore approximately cancels the gate factor in the parameter gradient, giving $\Delta W_\ell = O(1)$. The explicit factor of β_ℓ in the parameter-to-function map remains, so the resulting weight-induced residual update is $O(\beta_\ell)$, and hence $O(\beta_{\min})$ in the suppressed interior.

This creates two functional regimes. Blocks in the active prefix and suffix contribute directly to the representation and receive comparatively strong task gradients. Blocks in the suppressed middle remain closer to the identity path and have less opportunity to develop task-specific residual writes. This effect follows from the depth profile itself and persists unless their gate values increase.

B.2 Front-local recruitment

Although the two cutoffs affect every gate coefficient, their influence is localized around the two transition fronts. Differentiating the Bandstop profile gives

$$\frac{\partial \beta_\ell}{\partial c_L} = \frac{1-\beta_{\min}}{\tau_L} s_\ell^L (1-s_\ell^L)(1-s_\ell^R), \qquad \frac{\partial \beta_\ell}{\partial c_R} = -\frac{1-\beta_{\min}}{\tau_R} s_\ell^R (1-s_\ell^R)(1-s_\ell^L). \tag{8}$$

Because $s(1-s)$ decays rapidly away from a sigmoid transition, an individual cutoff update changes primarily the gates of blocks near its current front. A block in the interior changes appreciably only when a front approaches it.

The task-loss gradients of the cutoffs inherit the same localization:

$$\frac{\partial \mathcal{L}}{\partial c_q} = \sum_{\ell=1}^{L} \frac{\partial \mathcal{L}}{\partial \beta_\ell} \frac{\partial \beta_\ell}{\partial c_q}, \qquad q \in \{L, R\}. \tag{9}$$

The contribution of a block to a cutoff gradient is therefore weighted by its proximity to that cutoff's current transition. When the task loss favors additional residual capacity, gradient descent can move a front into the suppressed interval, increasing the gates of the adjacent blocks. Those blocks then contribute more strongly to the residual stream and receive a stronger task-gradient signal. Recruitment consequently proceeds from the two edges of the band toward its interior.

This coupling explains how a learned profile adapts its initial allocation of residual computation. The active prefix and suffix form a provisional reduced-depth network. When the task loss favors greater participation from the suppressed region, a cutoff can increase the gates of adjacent blocks; otherwise, a low-gate interval remains at convergence. That interval is a structured pruning proposal rather than a theoretical certificate of redundancy. Its correspondence with genuinely removable depth is evaluated directly in Section 3.

Connection to progressive depth recruitment. Together, the two effects give a training-dynamics interpretation of Figure 2. The gate profile determines the relative functional participation of blocks

at every step, while cutoff motion adaptively changes that allocation near the two fronts. At the level of unequal layer utilization, this resembles the layer-wise training dynamics of ACNs (Dorovatas et al., 2025), although Bandstop induces them through multiplicative depthwise scaling rather than additive output connections.

C Experimental Setup and Details

This appendix collects the architecture, training, and evaluation details omitted from Sections 2 and 3 for space, together with exact definitions of the metrics used in Figures 4 and 5. We also describe the pruning and adaptation procedure for Figure 3. Code will be released soon.

Gate application. The gate is applied once per block. For a block with two residual sublayers, let

$$r_\ell^{(1)} = f_\ell^{(1)}(x_{\ell-1}), \qquad u_\ell = x_{\ell-1} + r_\ell^{(1)}, \qquad r_\ell^{(2)} = f_\ell^{(2)}(u_\ell).$$

We use the combined update

$$x_\ell = x_{\ell-1} + \beta_\ell\big(r_\ell^{(1)} + r_\ell^{(2)}\big).$$

For Transformers (Vaswani et al., 2017), $r_\ell^{(1)}$ and $r_\ell^{(2)}$ are the attention and MLP updates, respectively; for MLP-Mixer, they are the token- and channel-mixing updates. Thus, the second sublayer receives the standard intermediate state u_ℓ, while β_ℓ scales the combined update written to the residual stream. Setting $\beta_\ell = 1$ exactly recovers the corresponding standard block.

Architectures and training configurations. All four architecture/dataset pairings share the same gate construction from Equation (4). Cutoffs are initialized symmetrically about the network midpoint, giving initial gate mass $\sum_\ell \beta_\ell \approx 4.0$ for the 12- and 16-layer models and ≈ 9.0 for SmolLM1's 30 layers; both cutoffs c_L, c_R are learned, while $\tau_L = \tau_R = 0.5$ for the vision models and 0.75 for the language models are fixed. The gate floor is $\beta_{\min} = 10^{-4}$ except for GPT-2, where it is 10^{-2}; the cutoff parameters use a $5\times$ multiplier on the base learning rate in all four settings.

MLP-Mixer / CIFAR-10, CIFAR-100. We stack 16 Mixer blocks, each 128 channels wide. We split each 32×32 image into 4×4 patches (64 patches total) and skip the class token. Each block mixes tokens through a 64-wide MLP, then mixes channels through a 512-wide MLP. We optimize with AdamW, set the learning rate to 1e-3 and weight decay to 1e-2, and clip gradients to norm 1.0. We warm up the learning rate linearly for 10 epochs, then decay it with cosine annealing down to 1e-6. We train for 300 epochs with batch size 64. We augment every training image: we crop it (random 32×32 crop, 4-pixel padding), then apply CIFAR-10 AutoAugment. We apply CutMix with probability 0.5, drawing the mixing ratio from Beta(1.0, 1.0). We smooth labels with factor 0.1. For CIFAR-100 we keep the architecture and recipe identical and only swap the output layer to 100 classes.

ViT-S/16 / ImageNet-100, ImageNet-1k. We use the typical ViT architecture (Dosovitskiy et al., 2021) and follow the training recipe of Beyer et al. (2022) for 300 epochs. Bandstop adds only the two learned cutoffs to the 22.12M-parameter baseline. ImageNet-1k uses a fixed contiguous 99%/1% train/minival split; ImageNet-100 uses the standard Tian et al. (2020)-style 100-class subset of ImageNet-1k.

GPT-2 124M / FineWeb-Edu. We match the training recipe of Karpathy (2024)'s GPT-2 (124M) reproduction on FineWeb-Edu unchanged. Bandstop adds only the two learned cutoffs to the 124.48M-parameter baseline. Trained on the FineWeb-Edu sample-10BT split with the GPT-2 BPE tokenizer.

SmolLM1-135M / SmolLM-Corpus. Architecture, tokenizer, and optimizer settings match the public HuggingFaceTB/SmolLM-135M configuration (Allal et al., 2024) unchanged. Bandstop adds only the two learned cutoffs. The one deviation is the training data: rather than the official five-source mixture, we use three of the five sources (FineWeb-Edu-dedup, Cosmopedia-v2, and OpenWebMath), re-weighted to $75.3\%/18.3\%/6.4\%$ and drawn without replacement under a fixed per-source quota for ≈ 2 epochs over the ≈ 100B materialized tokens.

Post-hoc healing: LLM-Streamline implementation. Following LLM-Streamline (Chen et al., 2025), for a budget of n layers we select the contiguous window beginning at

$$\ell^*(n) = \arg\max_{\ell} \cos\Big(\boldsymbol{x}^{(\ell)}, \boldsymbol{x}^{(\ell+n)}\Big),$$

where cosine similarity is averaged over tokens and calibration examples. Thus, the selected window is the one that changes its input representation least. Unhealed models remove this window directly. For healed models, a lightweight replacement h_ϕ is trained to approximate the window's output:

$$\phi^* = \arg\min_{\phi} \mathbb{E}_{i\in\mathcal{D}} \left[\mathrm{MSE}\left(h_\phi(\boldsymbol{x}_i^{(\ell^*)}), \boldsymbol{x}_i^{(\ell^*+n)}\right)\right].$$

We use an MLP-only replacement: RMSNorm followed by a SwiGLU MLP for SmolLM1, and LayerNorm followed by a GELU MLP with $4\times$ expansion for GPT-2. We train it with batch size 16 and perform no subsequent end-to-end post-training. These settings produce the results in Figure 3 and Table 10.

Stability metric. Figure 3's "stability" score follows the definition of Chen et al. (2025): task-level agreement between the pruned and source model's per-example correctness, weighted toward examples where the source model's answer was confident, then averaged unweighted across the six tasks.

Language-modeling benchmarks and mean accuracy. Figure 3's "mean accuracy," results are computed zero-shot with the lm-evaluation-harness (Gao et al., 2024). Reasoning-primitive results in Tables 4 and 12 are evaluated 5-shot. For Figure 3 we use the normalized accuracy where a task defines it and raw accuracy otherwise (HellaSwag, ARC-Easy, and ARC-Challenge use normalized accuracy; PIQA, WinoGrande, and LAMBADA-OpenAI use raw accuracy), and report the unweighted mean of the six per-task scores.

Regret. We define regret as the difference in held-out performance between removing the β-guided window and removing the oracle window at the same removal budget r, reported in accuracy points for the vision models and nats of held-out loss for the language models, with positive regret meaning the β-guided window under-performs the oracle.

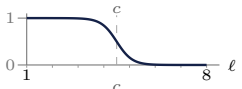
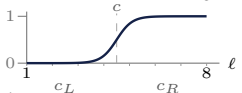
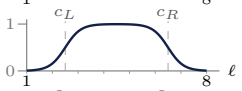
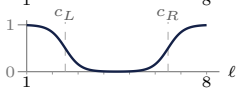
Effective rank (Figure 4). We use the entropy-based effective rank of Roy & Vetterli (2007), computed over the covariance eigenspectrum of h_ℓ, the residual-stream representation immediately after block ℓ, following Masarczyk et al. (2023). The covariance is estimated from a fixed probe batch held out from training (512 ImageNet validation images for ViT-S/16, 256 CIFAR test images for MLP-Mixer; the same batch is reused at every measurement so that changes in effective rank reflect training rather than sampling noise).

Depth score and Tuned Lens top-5 overlap (Figure 5). The logit-effect depth score is the global (full-network) residual-erasure depth score of Csordás et al. (2025), evaluated on 100 examples of MATH and MQuAKE as described in Section 3.

For the Tuned Lens we use the per-depth affine translator of (Belrose et al., 2023), trained on a held-out split of FineWeb-Edu following (Kapl et al., 2025). For the training objective we use next-token cross-entropy rather than Belrose et al.'s KL-to-final-logits distillation.

Noise-robustness protocol. We follow the random-pixel perturbation protocol of Sanyal et al. (2018). After resizing and center-cropping each ImageNet-1k validation image to 224×224, we sample an independent Bernoulli(p) mask over spatial pixel locations (shared across the three color channels), for $p \in \{0.4, 0.6, 0.8, 1.0\}$; at selected pixels we add independent Gaussian noise with standard deviation $128/255$ in $[0, 1]$ pixel space, applied before the model's own input normalization, and clip the result back to $[0, 1]$. We additionally evaluate both final checkpoints on ImageNet-C (Hendrycks & Dietterich, 2019) using its standard 15 corruption types (blur, digital, weather, and noise categories; the 4 held-out corruptions in extra.tar are not used) at all five severity levels, and report the arithmetic mean of top-1 accuracy over all 75 corruption$\times$severity combinations.

Table 5: Depth profiles obtained from the sigmoid fronts. The plots show the gate value as a function of layer depth; high values denote active residual branches.

Profile	Gate $g(\ell;\theta)$	Active region	Shape
Low-pass	s_ℓ^L	Prefix	
High-pass	s_ℓ^R	Suffix	
Band-pass	$(1-s_\ell^L)(1-s_\ell^R)$	Middle	
Bandstop	$s_\ell^L + s_\ell^R - s_\ell^L s_\ell^R$	Prefix and suffix	

D Analysis of Different Depth Profiles

This appendix complements the motivation in the main paper by comparing the four depth profiles directly. We examine how profile geometry affects the allocation of computation across depth, the resulting full-depth performance, and whether the inactive regions remain removable.

Adaptation across depth. Table 5 summarizes the structural constraint imposed by each profile. Low-pass and High-pass preserve an active prefix or suffix, respectively, while Band-pass concentrates residual computation in the middle and Bandstop retains both endpoint regions around a suppressed interior band. Together, the four profiles provide complementary views of how allocating capacity to different regions of network depth affects layer adaptation and task performance.

The trajectories in Figure 7 make the recruitment process visible: as a cutoff moves, adjacent layers receive larger gate values and become active participants in the residual computation. For Low-pass, recruitment toward the output turns the former last active layer from a task-facing endpoint into an intermediate transformation. High-pass induces the analogous change at the input-facing boundary. Band-pass can move both boundaries and therefore reassign both endpoint roles. Bandstop instead recruits capacity along the internal edges of its suppressed band while keeping the active prefix and suffix in place, allowing effective depth to adapt without reassigning the network's input- and output-facing regions.

Table 6: **Depth profiles, β-guided layer removal.** Validation performance vs. layers removed. Gated profiles (Low-pass, High-pass, Band-pass, Bandstop) are removed using β pruning, while Residual uses oracle search. MLP-Mixer columns are the mean over 3 seeds; GPT-2 124M columns are a single seed.

	MLP-Mixer/CIFAR-10 (top-1 %)			GPT-2 124M (PPL)		
Profile	0%	25%	50%	0%	25%	50%
Residual	91.72	85.88	62.38	**21.79**	67.22	308.61
Low-pass	90.73	90.75	**90.21**	23.11	**34.72**	**91.21**
High-pass	91.17	89.91	85.17	23.23	195.39	3025.03
Band-pass	91.35	57.67	11.88	22.45	15401.82	129408.67
Bandstop	**93.11**	**90.95**	71.90	21.87	35.26	122.26

D.1 Full-Depth Performance and Retained Prunability

The primary comparison in Table 6 is full-depth performance. Despite using closely related parameterizations, the profiles converge to different task performance, showing that profile geometry influences the learned solution. Bandstop improves over the residual MLP-Mixer and provides the strongest GPT-2 performance among the gated variants while remaining close to the residual model.

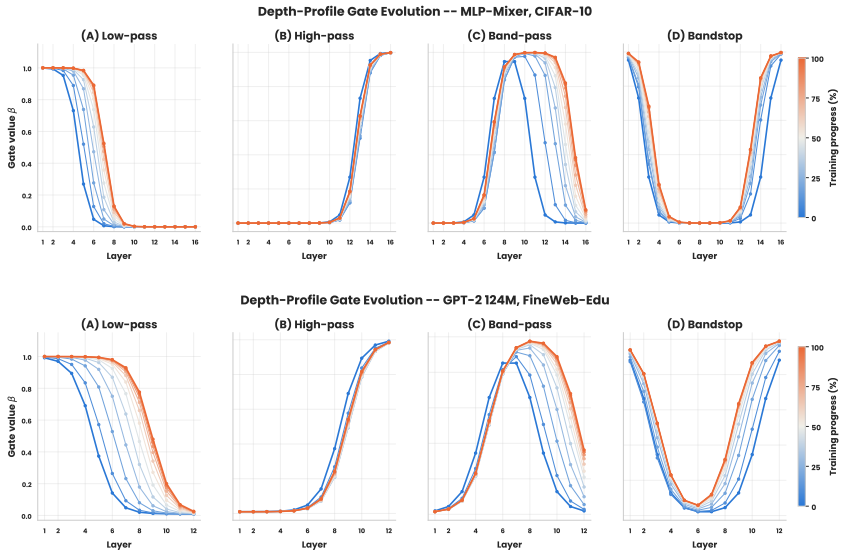


Figure 7: **Gate evolution per depth profile. Top: MLP-Mixer/CIFAR-10** (seed 7); Band-pass (active middle, suppressed ends) recruits a contiguous interior band from both edges, the mirror image of Bandstop's edge-recruitment pattern. **Bottom: GPT-2 124M**; all four profiles are trained on FineWeb-Edu (10B tokens). Low-pass and High-pass each converge to a single monotone front rather than a band, consistent with their single-cutoff parameterization in Table 5.

In contrast, suppressing one or both endpoint regions generally reduces full-depth quality. These results support the importance of retaining an active prefix and suffix when allocating capacity across depth.

At nonzero removal budgets, layers are removed in ascending order of their converged gate values. The results show that prunability depends strongly on which depth regions remain active. Profiles that preserve an active prefix are generally more robust. Low-pass is particularly effective at moderate removal budgets, retaining nearly all of its full-depth MLP-Mixer accuracy after removing 50% of the layers and achieving the lowest GPT-2 perplexity at both 25% and 50% removal. In contrast, High-pass places its inactive region at the beginning of the network, so gate-guided removal eliminates early representation-building layers. Band-pass is the least robust because its inactive regions lie at both endpoints, so gate-guided removal disrupts both the input-facing and task-facing parts of the network. These differences show that retaining the same number of layers is insufficient: their positions and functional roles also determine whether the resulting subnetwork remains effective. Bandstop is preferred because it preserves both endpoint regions, removes capacity from the interior, and combines retained prunability with the strongest overall full-depth performance.

Overall, the comparison shows that the depth profile determines more than the location of removable layers: it shapes how computation and functional roles are organized throughout the network. Prunability is a consequence of this organization, while the preference for Bandstop follows primarily from its ability to preserve both endpoint regions without sacrificing full-depth performance.

E Sensitivity to Gating Hyperparameters - Frozen profiles

E.1 Ablations on hyperparameters

The Bandstop profile is controlled by two fixed shape hyperparameters, the transition width τ and the gate floor $\beta_{\min}$, plus the initial placement of the two learned cutoffs c_L, c_R. Figure 8 illustrates their qualitative effects through the converged profile, robustness to contiguous layer removal, and full-depth performance. Although the sweep is conducted with an MLP-Mixer, its purpose is to clarify the general role of each hyperparameter rather than to identify an architecture-specific optimum.

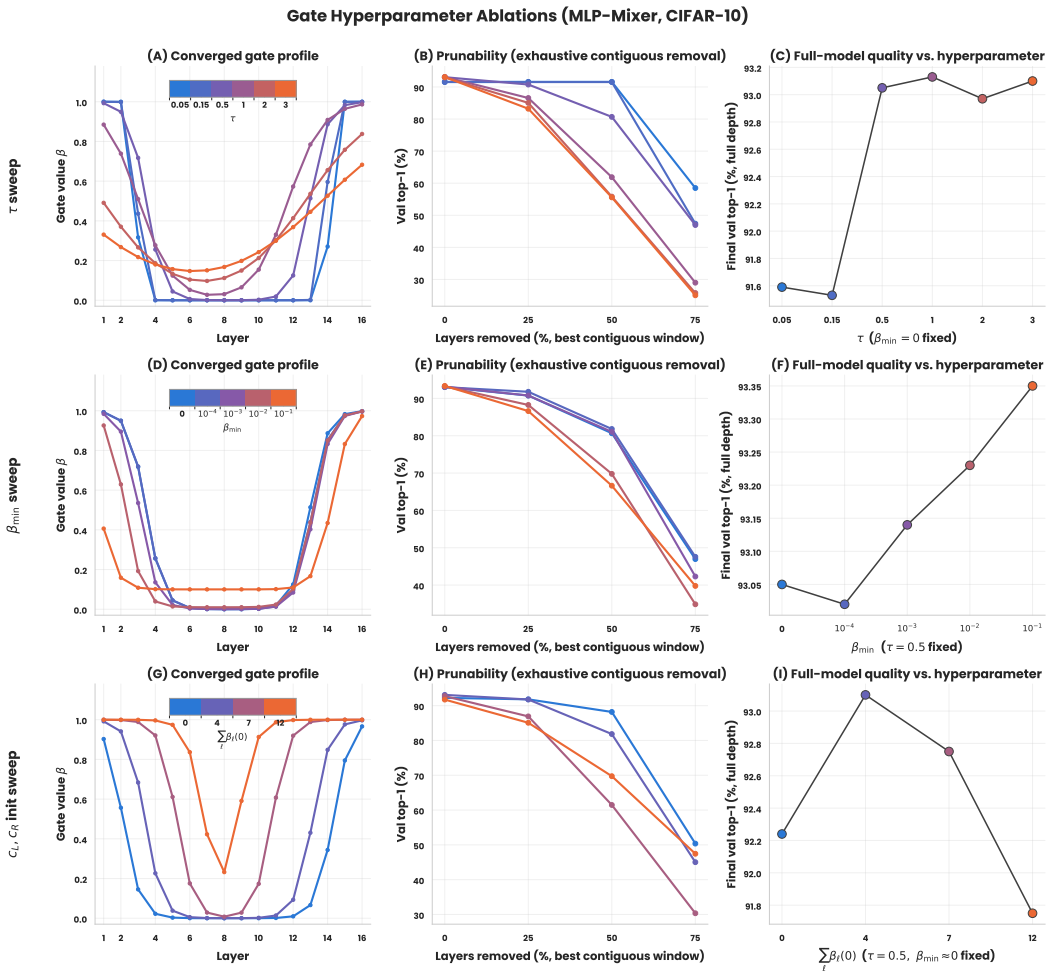


Figure 8: **Sensitivity to the Bandstop gate hyperparameters. Top row (A–C):** transition-width sweep with $\beta_{\min} = 0$. **Middle row (D–F):** gate-floor sweep with $\tau = 0.5$. **Bottom row (G–I):** cutoff-initialization sweep with $\tau = 0.5$, $\beta_{\min} = 0$, varying the initial gate mass $\sum_\ell \beta_\ell(0)$ via c_L, c_R's starting values. Columns show, from left to right, the converged gate profile, robustness to exhaustive contiguous-window removal, and full-depth validation performance.

Transition width τ. The transition width determines how distinctly the profile separates active and suppressed regions. Small values produce a sharp, nearly rectangular band in which the middle layers converge close to zero and the endpoints remain active. As τ increases, the transitions spread across a larger fraction of the network, producing smoother and more diffuse profiles with less separation between the middle and endpoint regions. Sharp profiles make the suppressed interval easier to remove, whereas smoother profiles distribute computation across more layers and are therefore less robust to pruning. This increased participation of intermediate layers can benefit the full-depth model, yielding a tradeoff between a clearly localized removable interval and unconstrained full-network performance.

Gate floor $\beta_{\min}$. The gate floor determines how inactive a suppressed block can become. When the floor is small, the converged profiles are almost identical: the learned cutoffs determine the band, while the precise minimum value has little effect on either prunability or full-depth performance. Once the floor becomes appreciable, the middle of the profile is lifted and suppressed blocks continue to make non-negligible residual updates. Their removal then becomes more damaging, even though retaining their contribution can slightly improve the full-depth model. The floor therefore controls how strongly Bandstop prioritizes removable structure over continued use of the suppressed

region. A small non-zero value preserves the behavior of a zero floor while avoiding exact deactivation.

Cutoff initialization. The initial cutoff positions determine the effective residual capacity available at the start of training, summarized by $\sum_\ell \beta_\ell(0)$. The converged profiles retain a clear dependence on this initialization: starting near the identity produces the widest suppressed band, whereas larger initial gate mass leads to progressively narrower and less pronounced suppressed regions. The cutoffs therefore adapt during training without entirely erasing the initial capacity bias. A broad initial band encourages the model to concentrate computation near the endpoints and consequently provides the strongest robustness to layer removal, but can restrict full-depth performance. Conversely, initializing too many layers as active weakens the resulting removable structure without improving the full model. An intermediate initialization provides the most balanced outcome, allowing additional capacity to be recruited when useful while preserving a well-defined compressible middle region.

E.2 Frozen ablations

Freezing the cutoffs turns Bandstop from an adaptive depth profile into a fixed compression prior. This ablation separates the effect of maintaining a suppressed middle region from the benefit of allowing its boundaries to adapt to the task. We compare learned and frozen cutoffs under the same training setup, evaluating both full-depth performance and robustness to contiguous layer removal.

Table 7 reveals a consistent tradeoff across architectures. Learning the cutoffs improves full-depth performance, yielding higher accuracy for both vision models and lower perplexity for GPT-2. Freezing them instead improves eight of the nine pruned settings, with the largest gains at moderate and high removal rates. This pattern is consistent with the optimization analysis in Appendix B: fixed cutoffs maintain the same depthwise attenuation throughout training, limiting both the residual contribution and functional development of the suppressed blocks. Learned cutoffs can relax this bias near the transition fronts, increasing the participation of adjacent blocks when favored by the task loss. Adaptation therefore benefits the full model but can make the recruited region more costly to remove, whereas freezing preserves a stronger compression prior at a small full-depth cost.

Table 7: **Freezing the Bandstop cutoffs strengthens pruning robustness at a small full-depth cost.** Validation performance before and after contiguous layer removal on ImageNet-1k (ViT-S/16), CIFAR-10 (MLP-Mixer), and FineWeb-Edu (GPT-2 124M). Pruned results report each checkpoint's oracle-best contiguous window under exhaustive search. Vision columns show top-1 accuracy (%, ↑), and GPT-2 columns show perplexity (PPL, ↓). Bold marks the better variant in each column. MLP-Mixer columns are the mean over the same 3 seeds used in Tables 8 and 11. ViT-S/16 and GPT-2 124M columns are each a single matched seed shared by the learned and frozen rows.

	ViT-S/16				MLP-Mixer				GPT-2 124M			
Variant	0%	25%	50%	75%	0%	25%	50%	75%	0%	25%	50%	75%
Bandstop (learned)	**78.99**	66.53	5.10	**0.78**	**93.11**	90.95	75.95	36.42	**21.87**	35.26	75.54	568.56
Bandstop (frozen)	78.57	**71.71**	**34.26**	0.68	92.77	**91.59**	**84.79**	**45.58**	22.20	**32.26**	**66.34**	**525.78**

F Comparison with other Training strategies

This appendix compares Bandstop training with three families of related approaches: stochastic layer-removal methods, different connectivity architectures, and residual-scaling methods. Our objective is not to compare general-purpose compression pipelines, but to evaluate how well each training strategy preserves performance when layers are removed after training. Unless stated otherwise, results report the best subnetwork under the pruning rule specified in each table.

F.1 Comparison with stochastic layer-removal and connectivity methods

Tables 8 and 9 compare Bandstop with Stochastic Depth, LayerDrop, and Auto-Compressing Networks (ACNs) (Huang et al., 2016; Fan et al., 2020; Dorovatas et al., 2025), which expose models

to reduced-depth computation through block dropping or modified connectivity during training. For the following comparisons we examine a steeper variant of Bandstop called Bandstop-sharp with $\tau = 0.25$ and $\beta_{\min} = 0$. Stochastic Depth and LayerDrop use $p_L = 0.5$ and $p = 0.2$ respectively.

At full depth, Bandstop retains a clear advantage. On MLP-Mixer, standard Bandstop reaches 93.11% accuracy and Bandstop-sharp reaches 92.24%, compared with 90.75% for the strongest removal-trained baseline. Both variants also retain useful prunability under mild removal: standard Bandstop is strongest through 14 layers retained, while Bandstop-sharp is strongest from 12 to 8 retained layers. On GPT-2, standard Bandstop obtains the lowest full-depth perplexity, while Bandstop-sharp is strongest with 10 and 8 layers retained. Thus, Bandstop combines substantially stronger full-model performance with robustness over the mild-to-moderate pruning regime, while gate sharpness provides an additional operating point when greater compression is desired.

As pruning becomes more aggressive, the specialized baselines progressively overtake standard Bandstop. On MLP-Mixer, ACN surpasses standard Bandstop once 10 or fewer layers are retained, but remains below Bandstop-sharp through 6 layers. At 4 layers, ACN surpasses both Bandstop variants and Stochastic Depth, while LayerDrop is strongest at both 6 and 4 layers. On GPT-2, Bandstop-sharp remains strongest with 10 and 8 layers retained, after which LayerDrop becomes strongest at 6 or fewer layers. The comparison therefore shows a division of strengths: Bandstop prioritizes full-model quality and mild-to-moderate contiguous depth reduction, whereas LayerDrop, Stochastic Depth and ACN become increasingly competitive when extracting very shallow subnetworks using their native procedures.

Table 8: **MLP-Mixer/CIFAR-10, 16 layers.** Best validation top-1 (%) vs. layers kept, in steps of 2. Mean over 3 seeds. Stochastic Depth (Huang et al., 2016) and Bandstop are evaluated using oracle search. LayerDrop (Fan et al., 2020) is evaluated under its prescribed Every-Other rule. ACN reports its own progressive readout (trained head applied to the running sum of the first k layers) rather than an oracle- or rule-selected subnetwork, since it requires no pruning search.

	Layers kept						
Variant	16	14	12	10	8	6	4
Bandstop	**93.11**	**92.73**	90.95	86.02	75.95	56.75	36.42
Bandstop-sharp	92.24	92.24	**92.24**	**91.80**	**89.52**	71.97	39.57
Stochastic Depth	90.75	90.82	90.90	90.61	88.53	77.85	55.10
LayerDrop (Every Other)	90.66	90.17	89.39	88.93	86.37	**81.84**	**72.09**
ACN	88.22	88.41	88.49	88.17	85.86	71.68	61.17

Table 9: **GPT-2 124M/FineWeb-Edu, 12 layers, vs. layers kept.** Perplexity (↓) of Bandstop and Bandstop-sharp under their own oracle-best contiguous window, against LayerDrop under its prescribed Every-Other rule.

	Layers kept						
Variant	12	10	8	6	4	2	1
Bandstop	**21.87**	28.26	43.98	75.54	275.27	1322.83	2769.89
Bandstop-sharp	24.11	**24.11**	**25.96**	35.67	101.13	1558.68	3633.06
LayerDrop	23.59	24.95	26.57	**31.18**	**45.83**	**300.83**	**1113.84**

Compatibility with structured post-hoc pruning and healing. We repeat this comparison with Bandstop-sharp included, under the same LLM-Streamline protocol as Figure 3 (Table 10).

Bandstop-sharp obtains the highest healed accuracy at 33% and 50% removal and the highest stability at every budget, and unhealed it retains essentially its full-depth accuracy after removing half the layers (39.41 vs. 39.31); healing still raises its stability at the highest budget even where healed accuracy dips slightly below unhealed (37.76 vs. 39.41), indicating the lightweight replacement continues to help adaptation on its own terms. The contiguous structure learned during Bandstop training therefore remains useful under a selection-and-repair pipeline.

Table 10: **GPT-2 124M/FineWeb-Edu under shared post-hoc pruning.** Mean downstream accuracy and stability after removing 8%, 33%, or 50% of layers using LLM-Streamline contiguous-window selection, with and without Stage-1 healing. Bold indicates the best training variant within each setting.

		Avg. Accuracy (%) ↑				Avg. Stability (%) ↑		
Method	Variant	0%	Low	Mid	High	Low	Mid	High
Healed	Bandstop	**40.90**	**40.72**	38.79	36.73	93.41	86.34	82.15
	Bandstop-sharp	39.31	39.48	**39.08**	**37.76**	**99.12**	**93.75**	**88.28**
	LayerDrop	39.66	39.19	37.04	34.22	94.32	85.69	78.58
Unhealed	Bandstop	**40.90**	**40.49**	37.35	34.21	92.77	83.79	78.55
	Bandstop-sharp	39.31	39.39	**38.84**	**39.41**	**99.01**	**91.01**	**87.70**
	LayerDrop	39.66	39.40	37.67	35.61	95.58	87.19	81.95

F.2 Comparison with residual-scaling methods

Table 11 compares Bandstop with ReZero and LayerScale, which modify residual-branch scaling primarily to improve optimization. At full depth, Bandstop remains competitive with both methods: ReZero is slightly better on CIFAR-10, while Bandstop is best on CIFAR-100. The distinction becomes substantially larger after layer removal. Across both datasets, Bandstop retains markedly higher accuracy at 12, 8, and 4 retained layers. These results indicate that the pruning robustness induced by Bandstop does not come at the expense of full-depth performance and is not reproduced by residual scaling alone.

Table 11: **MLP-Mixer, 16 layers, CIFAR-10 and CIFAR-100, vs. layers kept.** Best validation top-1 (%) under exhaustive contiguous-window-removal search. Mean over 3 seeds.

	Layers kept							
	CIFAR-10				CIFAR-100			
Variant	16	12	8	4	16	12	8	4
Bandstop	93.11	**90.95**	**75.95**	**36.42**	**71.53**	**67.34**	**34.34**	**4.65**
ReZero	**93.64**	82.30	34.23	17.73	71.20	48.72	8.86	3.11
LayerScale	92.97	73.37	29.63	14.24	70.54	56.91	12.05	2.34

G Reasoning-Primitives and In-Context Learning

G.1 Reasoning-primitives: additional depths

Table 4 in the main text reports variable-assignment results at assignment depth 0. Table 12 gives the corresponding results at assignment depths 1 and 2 (chance = 20%), reported as mean ± std over 3 evaluation seeds. Both models remain near chance.

Table 12: Variable-assignment accuracy (%, mean ± std over 3 evaluation seeds, SmolLM1-135M) at assignment depths 1 and 2 (chance = 20%).

	Depth-1			Depth-2		
	Basic	Math	Code	Basic	Math	Code
Residual	19.22 ± 0.65	20.12 ± 1.10	19.82 ± 0.83	19.36 ± 1.11	19.02 ± 1.99	19.64 ± 0.85
Bandstop	19.98 ± 1.08	20.88 ± 1.44	20.60 ± 0.73	19.98 ± 2.10	19.60 ± 1.35	19.86 ± 1.94

G.2 In-context learning: additional tasks

Figure 6 in the main text reports the few-shot sweep for variable assignment (math, depth 0) and copying (real words). Figure 9 gives the corresponding sweeps for the remaining depth-0 reasoning-

Table 13: **Full-depth validation loss and perplexity.** Results for the uncut Residual and Bandstop checkpoints. Bold marks the better variant for each backbone.

Backbone	Method	Val. Loss (nats) ↓	Perplexity ↓
SmolLM1-135M	Residual	**2.2501**	**9.49**
	Bandstop	2.2644	9.63
GPT-2 124M	Residual	**3.0813**	**21.79**
	Bandstop	3.0849	21.87

Table 14: **Full-depth six-task zero-shot accuracy (%).** Results for the uncut Residual and Bandstop checkpoints. Bold marks the better variant in each column.

Backbone	Method	HellaSwag	PIQA	ARC-E	ARC-C	WinoGrande	LAMBADA	Mean ↑
SmolLM1-135M	Residual	**42.02**	**67.85**	**55.09**	**30.12**	52.80	35.63	**47.25**
	Bandstop	41.76	67.14	54.04	27.82	**53.75**	**36.29**	46.80
GPT-2 124M	Residual	**32.35**	**63.60**	46.97	**26.45**	48.30	**26.08**	40.63
	Bandstop	32.04	62.46	**47.52**	26.28	**51.22**	25.91	**40.90**

primitive tasks: copying (random words), variable assignment (basic), and variable assignment (code). The number of few-shot examples extends up to the context length limit. Basic and code follow the same pattern as math: Bandstop starts at or behind Residual for the first few shots, then separates decisively from roughly 5 shots onward. Copying (random words) is the one task where this does not hold — Residual and Bandstop track closely through 2 shots, after which Residual pulls ahead and stays above Bandstop for the remainder of the sweep.

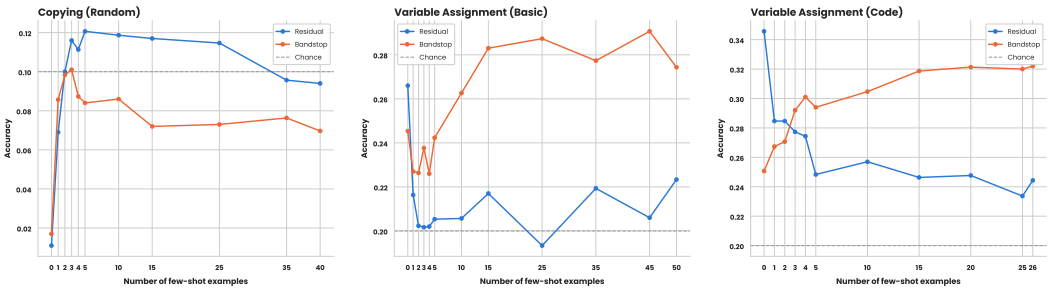


Figure 9: **In-context adaptation, Residual vs. Bandstop (SmolLM1-135M, mean across 3 seeds), additional tasks.** Accuracy vs. number of few-shot examples. **Left**: copying (random words). **Center**: variable assignment (basic). **Right**: variable assignment (code).

H Full-Depth Language-Model Results

This section compares the uncut Residual and Bandstop checkpoints before any layers are removed. Table 13 shows that Residual achieves slightly lower validation loss and perplexity on both SmolLM1-135M and GPT-2 124M. The differences are small, indicating that the two variants have comparable full-depth language-modeling performance.

Table 14 shows a mixed task-level comparison. On SmolLM1-135M, Residual performs better on HellaSwag, PIQA, ARC-Easy, and ARC-Challenge and achieves the higher mean accuracy, while Bandstop performs better on WinoGrande and LAMBADA. On GPT-2 124M, Residual leads on HellaSwag, PIQA, ARC-Challenge, and LAMBADA, whereas Bandstop leads on ARC-Easy and WinoGrande and achieves the higher mean accuracy. Overall, Residual has a small advantage in full-depth language-modeling loss and on the SmolLM1 downstream mean, while Bandstop has a small advantage on the GPT-2 downstream mean. These differences are modest compared with Bandstop's advantage after layer removal reported in Section 3.