D-STABLE: LEARNING RESIDUAL ALLOCATIONS UNDER A CONTROLLED DEPTH BUDGET

Anonymous authors
Paper under double-blind review

ABSTRACT

The sensitivity of a residual network—how much it amplifies a small input change across all layers—is a product over layers, so it is governed by one quantity: the total *residual budget*, the summed strength of the per-layer residual branches. We show that this budget and its allocation across layers are two separable levers. The budget's growth rate in depth sets the *stability class*: a linear budget makes worst-case sensitivity exponential, a square-root budget (Fixup, Stable ResNet) sub-exponential, and a *logarithmic* budget polynomial, so $\log d$ is the critical rate. The allocation leaves the class unchanged and sets accuracy within it. Prior stabilizers couple the two: fixed schedules freeze both, and learned gates (ReZero, LayerScale) let the budget drift. **d-STABLE** is a reparameterization that holds the total budget at exactly $\beta \log d$ at every optimization step while the optimizer learns the allocation. When every branch is norm-bounded this gives an a-priori polynomial bound on sensitivity; otherwise the pinned budget is an inductive bias whose amplification we measure. Our experiments separate the two levers. On `ogbn-arxiv` at 64 layers, unscaled residual GCN/SAGE collapse to 16–22% accuracy through gradient expansion, not oversmoothing. Keeping the residual mass small prevents the collapse: Fixup, ReZero, and a fixed logarithmic schedule all reach the same 69.5–69.8% plateau. The allocation supplies the margin above that plateau, reaching $71.4 \pm 0.2\%$ ($+1.7$ over a fixed $1/k$ profile at an identical budget), ahead of GCNII, LayerScale, and Gradient Gating. On this benchmark the margin comes from the allocation's shape, not its training; enforcing the a-priori bound costs 1.8 points. On `ImageNet-1k`, where residual networks already train stably, the budget costs 0.5 top-1 points on ResNet-50 and 1.8 on DeiT-Tiny, where vanilla, LayerScale, and a DeepNorm-style rule are more accurate.

1 INTRODUCTION

Depth is the most direct way to give a model more computation, but it destabilizes the end-to-end map. For a residual network $X^{(k+1)} = X^{(k)} + g^{(k)}(X^{(k)})$, the input–output Jacobian is a product over layers, $J_{\text{eff}}^{(d)} = \prod_k (I + J_g^{(k)})$, and with depth this product can drift to either extreme. When it *contracts*, representations lose their distinctions—*oversmoothing* in deep graph networks (Oono & Suzuki, 2020; Cai & Wang, 2020) and rank collapse in Transformers (Dong et al., 2021), features converging toward a constant. When it *expands*, the map turns hypersensitive: exploding gradients and brittle sensitivity in deep residual stacks. These are the same object at opposite ends. Residual connections are added precisely to escape the contracting regime—and they do—but at large depth they overshoot into the expanding one: the deep residual GNNs that collapse at d=64 fail by *gradient expansion*, not oversmoothing (Sec. 4.2). The product itself is therefore the right lens. Throughout, we use *stability* in one precise sense: the operator norm $\|J_{\text{eff}}^{(d)}\|$ grows at most *polynomially* in depth d, rather than exponentially—a worst-case sensitivity statement about the trained map, distinct from robustness to a particular attack or convergence of the optimizer (Sec. 4.3, Finding 3).

One scalar controls the product. With explicit residual scales $X^{(k+1)} = X^{(k)} + s_k\, g^{(k)}(X^{(k)})$, submultiplicativity gives $\|J_{\text{eff}}^{(d)}\| \le \prod_k (1 + \rho_k |s_k|)$ with $\rho_k \ge \|J_g^{(k)}\|$, and taking logs shows the entire product is governed by a single functional of the schedule—the weighted ℓ_1 *residual budget* $\sum_k \rho_k |s_k|$. What matters is its *growth rate in depth*. A budget growing linearly gives exponential sensitivity (the collapse of an unscaled residual); a $\sqrt{d}$ budget—Fixup and Stable ResNet's $1/\sqrt{d}$

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

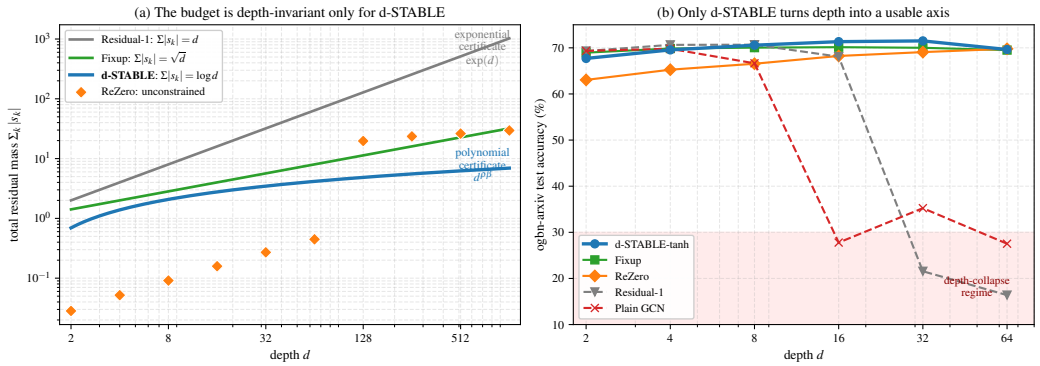


Figure 1: **The d-STABLE budget makes depth a usable axis. (a)** Total residual mass $\Sigma_k|s_k|$ vs. depth d. Residual-1 grows linearly ($\Sigma|s_k| = d$, exponential class), Fixup as $\sqrt{d}$ (sub-exponential), ReZero is unconstrained and drifts (orange scatter: ogbn-arxiv $d{\leq}64$ left, synthetic chain $d{\geq}128$ right), while d-STABLE pins $\Sigma|s_k| = \beta \log d$—the critical budget—giving a polynomial bound $d^{\rho\beta}$ (a priori under full-branch normalization; the per-branch ρ is otherwise measured, Sec. 3). **(b)** Test accuracy on `ogbn-arxiv` GCN. Plain GCN collapses by d=16, residual-1 by d=32, and Fixup/ReZero plateau at 69–70%. d-STABLE-tanh is the only method that improves monotonically with depth, reaching $\mathbf{71.4 \pm 0.2}\%$ (10 seeds) at d=64.

per-branch rule (Zhang et al., 2019; Hayou et al., 2021)—gives sub-exponential, $e^{\Theta(\sqrt{d})}$; and a *logarithmic* budget gives *polynomial*, $\|J_{\text{eff}}^{(d)}\| \leq d^{\beta\rho}$. Thus $\log d$ is the **critical budget**: the largest total residual mass whose worst-case sensitivity is still polynomial in depth (Sec. 3). The budget's rate, on its own, selects the stability class.

Stability is a budget, not a schedule. This reframes depth control. The budget's rate fixes the *stability class*; the *allocation*—where along the depth the mass is spent, and with which sign and magnitude—is orthogonal to it and controls *expressivity*. Prior stabilizers tie the two together. Fixed schedules (Fixup, harmonic decay (Zhang et al., 2019)) freeze both the budget and the allocation before training; learned gates (ReZero, LayerScale (Bachlechner et al., 2020; Touvron et al., 2021b)) learn the allocation but leave the cumulative mass $\sum_k |s_k|$ free to drift—a learned per-branch gate reaches $\sum_k |s_k|$=24.9 on ResNet-50 (Table 5). The missing regime is simple to state: **fix the budget, learn the allocation.**

d-STABLE. We pin the budget as an *exact invariant of optimization*. For residual updates $X^{(k+1)} = X^{(k)} + s_k\, g^{(k)}(X^{(k)})$ we parameterize

$$s_k \;=\; \beta \log d \cdot \frac{\tanh(u_k)}{\sum_j |\tanh(u_j)|}, \qquad \text{so that} \quad \sum_{k=0}^{d-1} |s_k| \;=\; \beta \log d \tag{1}$$

for any value of the learned $\{u_k\}$, at *every* gradient step (up to a 10^{-8} stabilizer that can only lower the sum, Sec. 3; a non-negative softplus variant gives $s_k \geq 0$). The optimizer can move mass across depth, drive a layer to zero, or flip signs, but it cannot create budget. The multiplier β is not learned; it sets the residual budget and, under full-branch normalization ($\rho \approx 1$), the polynomial exponent d^{β}—an explicit knob that trades certificate tightness for residual capacity, with a task-dependent optimum (Sec. 4.2). This is not a fixed schedule (the allocation is free) and not a penalty on ReZero-style gates (it is a reparameterization): it is the one cell prior methods leave empty—budget fixed, allocation learned.

What the evidence shows. Our experiments separate the two levers, and separate the certificate from both. *The budget sets the class.* In synthetic residual chains, dialing the budget moves the measured amplification from exponential to polynomial as predicted, and up to d=1024 Fixup's $\sqrt{d}$ budget degrades while the logarithmic one holds (Sec. 4.1). On `ogbn-arxiv`, deep residual GCN/SAGE collapse to 16–22% at d=64 by *gradient expansion*, not oversmoothing: the measured amplification J_{p95} explodes while Dirichlet energy *rises* (Sec. 4.2, App. E.1). Keeping the residual mass small prevents this collapse, and Fixup, ReZero, and a fixed logarithmic schedule all stop at the same 69.5–69.8% plateau. *The allocation sets accuracy within the class.* At an identical budget the d-STABLE allocation beats a fixed $1/k$ profile by $+1.7$, and this is the entire margin by which

d-STABLE ($71.4 \pm 0.2\%$, Fig. 1) exceeds every stabilizer we test. On `ogbn-arxiv` this margin comes from the allocation's shape, not its training (Sec. 4.2). *The certificate is a separate property with a price.* The a-priori polynomial bound holds only when every branch is norm-bounded; our full-branch-normalized GCN pays 1.8 points for it, and every other configuration carries a measured, not guaranteed, amplification. On ImageNet, where residual nets already train well, the budget buys nothing and costs between 0.5 top-1 points (ResNet-50) and 1.8 (DeiT-Tiny, where vanilla, LayerScale, and a DeepNorm-style rule all beat it)—so we present it as a trade-off, not a universal win.

2 Related Work and Our Contributions

Positioning. Prior depth-stabilization methods commit to one of two regimes, and neither separates the budget from the allocation. A *fixed* schedule (Fixup (Zhang et al., 2019), DeepNorm (Wang et al., 2022), Stable ResNet (Hayou et al., 2021), harmonic decay) sets the per-branch scale by rule, which does pin the total—Fixup's $s_k{=}1/\sqrt{d}$ gives $\sum_k |s_k|{=}\sqrt{d}$—but at a sub-exponential rather than critical rate, and it freezes the allocation before training starts. *Unconstrained learnable* gates (ReZero (Bachlechner et al., 2020), LayerScale (Touvron et al., 2021b), SkipInit (De & Smith, 2020)) free the allocation but fix only the *initial* scale, letting the total drift once the branches activate—to 24.9 on ResNet-50 (Table 5). d-STABLE occupies the space between them. The absolute residual budget is pinned to $\beta \log d$ at every optimization step, while the per-layer profile—sign, magnitude, and where depth is actually "spent"—is learned. This is not an extra penalty placed on ReZero-style gates; it is a reparameterization that decouples two axes prior methods tie together, and it is what converts the product bound $\prod_k (1 + \rho_k |s_k|)$ into the $O(d^{\beta\rho})$ ceiling a drifting budget cannot claim (Theorem 1). A fixed schedule freezes the allocation before training; unconstrained gates let the budget drift. We make the budget an invariant of optimization and ask a different question: **under a stability-preserving budget, where should the network spend its depth?**

Why the total budget is the distinct object. The adjacent lines of work each constrain something else. Spectral normalization and per-layer Lipschitz control (Miyato et al., 2018; Park & Kim, 2024) bound *each factor* of the product rather than the total, which cannot stop $\prod_k (1 + \rho |s_k|)$ from growing exponentially even when every $\|J_g^{(k)}\| \leq 1$. The provable-oversmoothing GNN analyses (Scholkemper et al., 2025; Chen et al., 2025) bound an *asymptotic rate* for one architecture and one failure mode—and the $d{=}64$ collapse we measure is *not* oversmoothing (the residual arms' Dirichlet energy *rises* with depth), so slowing that rate does not by itself address it. Contemporary residual-mixing designs constrain a linear mixing matrix (mHC (Xie et al., 2025)) or aggregation weights (AttnRes (Chen et al., 2026)). d-STABLE is the only one of these with a closed-form depth-dependent amplification bound, and it is architecture-agnostic where the GNN results are not; the budget–branch-norm decomposition (App. E.1) shows the *budget* factor, not the per-branch norm ρ, governs the measured blow-up. **App. A gives the full treatment**—Lipschitz control and residual scaling, depth instability in GNNs and Transformers, depth-dependent residual mixing, the dynamical-systems view, and a point-by-point comparison against the closest designs.

Contributions. This paper makes four main contributions:

- **Method.** We introduce a signed learnable residual-scale parameterization, plus a non-negative softplus variant, that keeps $\sum_k |s_k| = \beta \log d$ exact throughout training while allowing the optimizer to redistribute mass across depth, with β an explicit certificate/accuracy knob.
- **Analysis.** We prove the joint budget law $\sum_k \rho_k |s_k| \leq C \log d \Rightarrow \|J_{\text{eff}}^{(d)}\| \leq d^C$, separating the stabilizing role of the total weighted budget from the expressive role of the allocation profile, and we are explicit about when the certificate is a priori (full-branch normalization) versus a posteriori (budget pinned, ρ measured).
- **Mechanism experiments.** Synthetic residual chains isolate the predicted transition from exponential to polynomial amplification and show that, once the total absolute budget is fixed, allocation shape mainly changes constants rather than the depth-scaling bound.
- **Architecture-scale evidence.** We test the budget across five GNN operators on `ogbn-arxiv`, on heterophilic graphs and a long-range depth stress test where β is tuned, and on ImageNet ResNet-50/152 and DeiT-Tiny—with matched-budget and frozen-allocation controls isolating what the allocation contributes, and the clean-accuracy cost reported against a schedule bound tightened by 5–20 orders of magnitude.

Table 1: **The residual budget's growth rate selects the stability class.** From $\|J_{\text{eff}}^{(d)}\| \le e^{\rho B_d}$, the rate of $B_d = \sum_k |s_k|$ in depth determines the class of the worst-case bound. A logarithmic budget is the unique rate giving a polynomial ceiling; d-STABLE pins it exactly, with β the exponent.

residual budget $B_d = \sum_k \lvert s_k\rvert$	bound $\|J_{\text{eff}}^{(d)}\| \le$	stability class	representative
$\Theta(d)$	$e^{\Theta(d)}$	exponential (collapse)	residual-1 (unscaled)
$\Theta(\sqrt{d})$	$e^{\Theta(\sqrt{d})}$	sub-exponential	Fixup, Stable ResNet ($1/\sqrt{d}$)
$\boldsymbol{\beta \log d}$	$\boldsymbol{d^{\rho\beta}}$	**polynomial**	**d-STABLE**
$\Theta(1)$	$O(1)$	bounded (under-expressive)	—

3 Method and Stability Analysis

We make the budgeted-allocation view precise in three steps: the product structure of the Jacobian reduces stability to a single scalar functional of the schedule; the *growth rate* of that scalar in depth selects the stability class; and d-STABLE pins the rate exactly while leaving the allocation free.

3.1 Residual amplification reduces to one scalar

Throughout, $\|\cdot\|$ is the spectral (induced ℓ_2) operator norm. For a depth-d residual network with explicit per-layer scales,

$$X^{(k+1)} = X^{(k)} + s_k\, g^{(k)}(X^{(k)}), \qquad k = 0, \dots, d-1, \quad s_k \in \mathbb{R}, \tag{2}$$

the end-to-end Jacobian is a product $J_{\text{eff}}^{(d)} = \prod_{k=0}^{d-1}\big(I + s_k J_g^{(k)}\big)$, and the sensitivity to an infinitesimal perturbation δx (finite δx: App. B) obeys

$$\kappa \;=\; \frac{\|\delta X^{(d)}\|}{\|\delta x\|} \;\le\; \|J_{\text{eff}}^{(d)}\|. \tag{3}$$

Submultiplicativity together with $\|J_g^{(k)}\| \le \rho_k$ gives

$$\|J_{\text{eff}}^{(d)}\| \;\le\; \prod_{k=0}^{d-1}\big(1 + \rho_k|s_k|\big) \;\le\; \exp\Big(\underbrace{\textstyle\sum_{k=0}^{d-1} \rho_k|s_k|}_{\text{weighted budget } S_d}\Big). \tag{4}$$

The entire product thus collapses onto a single scalar—the weighted residual budget $S_d = \sum_k \rho_k|s_k|$, combining the per-layer operator strength ρ_k and the scales $|s_k|$. Every statement below is about S_d or its unweighted part B_d (Sec. 3.2).

3.2 The budget's growth rate is the stability class

Theorem 1 (Logarithmic budget law)**.** *If $\|J_g^{(k)}\| \le \rho_k$ for all k and $\sum_{k=0}^{d-1} \rho_k|s_k| \le C \log d$, then $\|J_{\text{eff}}^{(d)}\| \le d^C$.*

Proof. By (4), $\log \|J_{\text{eff}}^{(d)}\| \le \sum_k \log(1 + \rho_k|s_k|) \le \sum_k \rho_k|s_k| \le C \log d$; exponentiating gives $\|J_{\text{eff}}^{(d)}\| \le d^C$. □

The theorem is one point of a hierarchy. Write $\rho = \max_k \rho_k$ and let $B_d = \sum_k |s_k|$ be the (unweighted) residual budget; then (4) reads $\|J_{\text{eff}}^{(d)}\| \le e^{\rho B_d}$, so the *growth rate of B_d in d* fixes the sensitivity class outright (Table 1).

Corollary 2 (Critical budget)**.** *Assume the per-branch norm $\rho = \Theta(1)$ in d. Then $B_d = \Theta(\log d)$ is the* critical budget*: the fastest-growing residual budget whose worst-case sensitivity is still polynomial in depth. Any $B_d = \omega(\log d)$ makes the bound super-polynomial, $e^{\rho B_d} = d^{\omega(1)}$; any $B_d = o(\log d)$ stays polynomially bounded but spends vanishing per-layer capacity (mean $|s_k| \to 0$ faster than $\frac{\log d}{d}$), under-using depth. d-STABLE pins $B_d = \beta \log d$ exactly, placing the network at this critical rate with polynomial exponent $\rho\beta$.*

The budget's rate is the stability class, and $\log d$ is the knee: Fixup and Stable ResNet's $1/\sqrt{d}$ rule sits one class up and an unscaled residual two. The bound is a worst-case envelope; that its exponential and polynomial ends are *attained*, not merely permitted, is verified in Sec. 4.1.

3.3 d-STABLE: PINNING THE CRITICAL BUDGET, FREEING THE ALLOCATION

d-STABLE realizes the critical budget as an exact invariant of optimization while leaving *where* the mass goes entirely to the optimizer. We write

$$s_k = B_d \cdot \frac{r_k}{\sum_j |r_j| + \epsilon}, \qquad B_d = \beta \log d, \qquad r_k = \tanh(u_k), \tag{5}$$

so the normalizer pins the total absolute mass $\sum_k |s_k| = \beta \log d$ at every step, for any value of the parameters $\{u_k\}$. The stabilizer $\epsilon = 10^{-8}$ can only lower the sum, so every upper bound holds (values: App. C.4). What the optimizer learns is the numerator r_k: it can be positive or negative, can drive a layer toward zero (effectively skipping it), or can flip sign to apply a corrective, over-relaxation-style update. The multiplier β is a *fixed* hyperparameter, not learned: we do attempt to learn it (Table 18), but gradient descent on β is essentially flat—the optimizer leaves it near initialization on every dataset—so learning β offers no advantage over setting it. We therefore present d-STABLE as it is, a *fixed budget with a learned allocation*: the budget is controlled by construction, the allocation u_k is learned. Stability is governed by B_d alone; the shape of the allocation is free. We also evaluate the non-negative variant $r_k = \mathrm{softplus}(u_k)$.

β is a knob, not a fixed constant. While β is fixed *within* a run, it is an explicit stability/accuracy dial *across* runs: by Corollary 2 it directly sets the polynomial ceiling $d^{\rho\beta}$, and it *is* the exponent itself under full-branch normalization ($\rho \approx 1$). Raising β from 1 enlarges the budget (and loosens the ceiling) in exchange for capacity; on tasks where the default β=1 is too tight this recovers and then exceeds the unbudgeted gates, with a task-dependent interior optimum (Sec. 4.2). Larger β never leaves the polynomial class.

What guarantees the bound: a priori vs. a posteriori. The ceiling $\|J_{\text{eff}}^{(d)}\| \le d^{\rho\beta}$ needs *both* the pinned budget $\sum_k |s_k| = \beta \log d$ *and* a per-branch bound $\|J_g^{(k)}\| \le \rho$. d-STABLE pins the budget a priori; the per-branch bound is separate. Because a branch is a *composition* of layers, $J_g^{(k)} = \prod_i J_{L_i}$, an a-priori ρ=1 (hence d^{β} by construction) requires spectral-normalizing *every* layer of the branch.

When the residual branch is a single graph convolution, spectral-normalizing that one layer *is* full-branch normalization, and Theorem 1 then holds a priori with $\rho \approx 1$ (realized ≤ 1.15 under one-step power iteration, App. E.1); we report the accuracy of this certified configuration alongside the no-normalization headline (Sec. 4.2).

Otherwise—one normalized layer per ImageNet block, or no normalization on the GNN headline rows—Theorem 1 still holds, but with $\rho = \max_k \|J_g^{(k)}\|$ *read off the trained network* rather than designed: the certificate is *a posteriori*, $\|J_{\text{eff}}^{(d)}\| \le d^{\beta\rho}$. Rather than lean on this conservative bound, we report the empirically measured amplification J_{p95} directly—a p_{95} estimate of $\|J_{\text{eff}}^{(d)}\|$ by JVP power iteration over sampled inputs—which stays of order 10–10^2 across depth for d-STABLE while an unbudgeted residual explodes (Fig. 7, App. E.1).

The quantity d-STABLE controls a priori in every setting is the budget $B_d = \beta \log d$—which is what prevents the residual-1 blow-up ($\rho_k |s_k|$ accumulating to $\Theta(d)$)—while the per-branch ρ is measured, not enforced.

The a-posteriori regime is an inductive bias, not a bound. When ρ is not enforced, d-STABLE does not *bound* the amplification—the ceiling is measured, not guaranteed—so its benefit there comes from the *inductive bias* the fixed-budget parameterization imposes, much as ReZero's comes from its zero-initialization. The two biases differ in kind: ReZero starts each scale at zero and lets the cumulative mass grow freely (it drifts to $\sum_k |s_k|$=24.9 on ResNet-50, Table 5), whereas d-STABLE imposes a *conservation law*—the total mass is held at $\beta \log d$ and only redistributed across depth. The matched-budget control isolates this bias directly: at an identical budget the allocation still wins by $\sim$+1.7—on `ogbn-arxiv` through its shape rather than its training (Sec. 4.2). Accordingly we reserve the words *certificate* and *bound* for the a-priori regime; in the a-posteriori regime the pinned budget is best read as this inductive bias, and a formal analysis of the induced regularization is a natural direction for future work.

Optional components. The analysis also suggests a budget-aware initialization and a depth-aware learning rate that rebalance per-layer signal under a non-uniform allocation. Neither enters the bound, and every reported experiment disables both and uses the parameterization (5) alone (App. C).

4 EXPERIMENTS

We test three regimes. *Synthetic chains* (Sec. 4.1) verify that the budget's growth rate realizes the stability classes of Table 1. *Deep GNNs* (Sec. 4.2) are the regime where depth breaks: an unbudgeted residual collapses, the budget prevents it, and controlled comparisons isolate what the allocation and the certificate each contribute. *ImageNet* (Sec. 4.3) is the regime where residual nets already train well, so the budget buys no accuracy and we account for what it costs. **PyTorch source code is provided in the supplementary for reproducibility.**

4.1 SYNTHETIC VERIFICATION: THE BUDGET'S RATE REALIZES THE STABILITY CLASSES

We study residual chains $X^{(k+1)} = X^{(k)} + s_k\,\sigma(\hat{A}X^{(k)}W^{(k)})$ with $\hat{A} \in \mathbb{R}^{64\times 64}$ random symmetric, $\|\hat{A}\| = 1$, branch operators normalized, over five seeds, reporting the end-to-end amplification κ (3) and $\|J_{\text{eff}}\|$ by power iteration. Five constructions (full detail in App. D.6) isolate the mechanism: an aligned-linear chain shows the d^C bound is *approachable* when its only proof slack is removed (fitted log–log slopes $0.51/1.04/2.14$ vs. predicted $0.5/1/2$, $R^2 \geq 0.999$); a power-law sweep $s_k = c/(k+\tau)^\alpha$ locates the class boundary exactly at $\alpha{=}1$ (the harmonic, $\log d$ budget), with the fitted depth-slope falling from 6.57 ($\alpha{=}0$, linear budget, exponential) through 0.28 ($\alpha{=}0.5$, $\sqrt{d}$, sub-exponential) to 0.009 ($\alpha{=}1$, $\log d$, polynomial); and budget-preserving reshapings (uniform, harmonic, reversed, sparse, signed) all collapse onto one polynomial band, confirming the shape moves only constants, not the class.

Extreme-depth extrapolation (d up to 1024; final training loss, 2 seeds). The class asymmetry holds: *Unconstrained* diverges already at $d{=}128$; *Fixup*'s loss grows $\sim 9\times$ ($0.010 \to 0.018 \to 0.054 \to 0.093$ across $d \in \{128, 256, 512, 1024\}$) as its $\sqrt{d}$ budget keeps growing; fixed-harmonic d-STABLE stays flat (0.006–0.007) at $\Sigma|s_k| = \log d$. Learned budgets fit the task more closely (loss $\sim 10^{-5}$), but ReZero gets there by drifting to $\Sigma|s_k|{=}39$ ($\beta_{\text{eff}} \approx 5.6$ at $d{=}1024$), whereas learnable d-STABLE stays at $\beta_{\text{eff}} \approx 2.1$–$2.4$ at every depth, a $d^{3.3}$ tighter bound. This is Table 1 on a trainable chain.

4.2 DEEP GNNS: WHERE DEPTH BREAKS

Deep message-passing networks are the sharpest depth stress-test. We use a residual GCN stack (Kipf & Welling, 2017) $H^{(k+1)} = H^{(k)} + s_k\,\sigma(\text{GNN}_k(H^{(k)}))$ (hidden 256 on `ogbn-arxiv` (Hu et al., 2020), 64 on the small/heterophilic sets), depths $d \in \{2, \ldots, 64\}$, comparing plain GCN, residual-1 ($s_k \equiv 1$), ReZero, Fixup ($1/\sqrt{d}$), and d-STABLE-tanh/softplus. Unless stated, the GNN rows apply *no* spectral normalization, so they isolate the effect of the budget $\Sigma|s|$ alone; we report accuracy, the residual mass $\Sigma|s|$, and the measured amplification J_{p95} (JVP power iteration over sampled nodes). Unless a control states its own seed set, headline `ogbn-arxiv` numbers are 10 seeds (ReZero: 7).

Table 2: **`ogbn-arxiv` GCN depth scaling** (3 seeds; canonical OGB split). Each cell is test accuracy / $\Sigma|s|$; for d-STABLE, $\Sigma|s| = \log d$ by construction. **Bold** marks the column-best accuracy. This sweep shows the monotone *trend*; the canonical $d{=}64$ headline is the 10-seed 71.4 ± 0.2 (Table 13).

Method	$d{=}2$	4	8	16	32	64
Plain GCN	**69.3**/0	69.8/0	66.6/0	27.8/0	35.2/0	27.5/0
Residual-1	69.3/2.00	**70.6**/4.00	**70.7**/8.00	68.1/16.00	21.6/32.00	16.4/64.00
ReZero	63.0/0.03	65.3/0.05	66.6/0.09	68.2/0.16	69.1/0.27	69.8/0.45
Fixup	69.0/1.41	69.9/2.00	70.0/2.83	70.1/4.00	70.0/5.66	69.5/8.00
d-STABLE (softplus)	67.8/0.69	69.1/1.39	69.5/2.08	69.6/2.77	69.7/3.47	69.5/4.16
d-STABLE (tanh)	67.7/0.69	69.5/1.39	70.6/2.08	**71.3**/2.77	**71.5**/3.47	**71.6**/4.16

Finding 1: the budget turns depth into a usable axis, and the collapse it prevents is gradient expansion. Table 2 gives the depth sweep. Plain GCN collapses by $d{=}16$ (oversmoothing—energy *decays*); residual-1 collapses by $d{=}32$; Fixup and ReZero avoid collapse but saturate at 69–70%; only d-STABLE-tanh improves monotonically with depth. At $d{=}64$ it reaches $\mathbf{71.4 \pm 0.2}\%$ (10 seeds), ahead of ReZero (69.8 ± 0.1) and Fixup (69.5 ± 0.3), well separated across seeds—the gap is several times the per-method standard deviation (Table 13). Crucially, the residual-1 collapse is *not* oversmoothing but *gradient expansion*: its measured amplification J_{p95} explodes from 157 at $d{=}32$ to $7.8{\times}10^6$ at $d{=}64$ while d-STABLE stays of order 10, and its Dirichlet energy *rises* with depth

Table 3: β **is a tunable budget/accuracy knob (controlled long-range depth stress-test, Peptides-func, 3 seeds; 2 for** β**=8 at** d**=16, 32).** Test AP on an *intentionally minimal* backbone—no edge features, mean-pooled readout—so these are same-backbone, budgeted-vs-unbudgeted comparisons that isolate the residual schedule; they are *not* competitive LRGB numbers and are not comparable to edge-feature architectures (GatedGCN/GPS). Raising β makes the pinned budget overtake the unbudgeted gates, with an interior optimum at β=4. No row uses branch normalization, so none is a priori certified.

variant	d=16	d=32	d=64	budget pinned?
scalar d-STABLE, β=1	0.427	0.434	0.445	yes ($\Sigma\|s\|$=$\beta \log d$)
scalar d-STABLE, β=2	0.456	0.470	0.478	yes
scalar d-STABLE, β=4	0.481	0.495	**0.518**	yes
scalar d-STABLE, β=8	0.464	0.468	0.504	yes (turns over)
ref: ReZero / LayerScale / G^2 (unbudgeted)			*.455 / .456 / .465*	no

rather than decaying (App. E.1). Slowing the asymptotic oversmoothing rate would not address this; bounding the budget does.

Finding 2: controlled comparisons isolate what the budget, the allocation, and the certificate each buy. The budget conflates two things—*being small* and *being learned*—and the certificate is a third, separable property. Four controls at d=64 (matched-recipe; App. E.2) separate them:

- **Matched budget (allocation).** Holding $\Sigma|s| = \log d$ identical and comparing a *fixed* harmonic profile against the d-STABLE allocation, the latter wins by $+\mathbf{1.80} \pm \mathbf{0.35}$ at d=32 and $+\mathbf{1.71} \pm \mathbf{0.27}$ at d=64 (5 seeds). The fixed profile (69.83 ± 0.22) lands on the plateau Fixup and ReZero also reach: the budget alone prevents collapse but does not separate d-STABLE from other stabilizers; the allocation does.
- **Frozen allocation (what training adds).** Freezing the converged profile and retraining from scratch gives 71.44 ± 0.19, tied with the learned 71.59 ± 0.13 (5 seeds; all five frozen profiles 71.4–71.8). Here the profile barely trains (within 2% of its initialization; a $100\times$ larger step changes nothing), so the $+1.6$ over $1/k$ comes from the allocation's shape, not learning; a fixed uniform positive profile falls back to 69.6 (App. E.2).
- **Certified configuration (the a-priori price).** With full-branch spectral normalization—exact for a single-conv GCN branch, so Theorem 1 holds a priori with $\rho \approx 1$—d-STABLE reaches 69.65 ± 0.07 at d=64 (69.3 at d=32), about 1.8 points below the no-normalization headline but still far from residual-1's 14.6% collapse. This is the only certified configuration in the paper, and the certificate is a property no competing stabilizer offers; every other GNN, long-range, and ImageNet result pins the budget but reports a measured amplification.
- **Positioning vs. Stable ResNet.** The $1/\sqrt{d}$ per-branch rule ($\Sigma|s| = \sqrt{d}$, sub-exponential class) prevents collapse and reaches 69.47 ± 0.28, level with the fixed $\log d$ profile: at d=64 the class difference does not yet show in accuracy (it does at extreme depth, Sec. 4.1), so the 2.1-point gap to d-STABLE is the allocation's.

Under the same recipe, the learned-gate and deep-GNN baselines requested in review all trail the budgeted allocation at d=64—LayerScale 68.72, Gradient Gating 68.78, GCNII 68.19—and accuracy broadly tracks the budget regimes across baselines (App. E.3); DeepNorm's $(2d)^{1/4}$ rule spends $\Sigma|s| \approx 19$ and lands in the collapse regime (34.1), so we report it as a non-faithful port, not as DeepNet.

Finding 3: β **is a budget/accuracy knob (heterophilic and long-range).** On heterophilic graphs (Table 19) neighbors often carry different labels, so the best residual map is near-identity and the default β=1 budget ($\log 64 \approx 4.2$) is stable but too large; a small-β sweep at d=64 matches or exceeds ReZero on four of six datasets while spending a still-smaller explicit budget. The same knob has a genuine *interior* optimum on a long-range task. On a controlled long-range depth stress-test (Table 3), raising β turns the budget from too-tight into the best config: scalar d-STABLE climbs $0.445 \to 0.478 \to 0.518$ at β=1, 2, 4, overtaking the unbudgeted gates (ReZero 0.455, LayerScale 0.456, Gradient Gating 0.465) with a pinned budget and a measured $J_{p95} \approx 79$ (vs. an unbudgeted residual's 309); β=8 turns over (0.504), so the optimum is interior. Per-channel (diagonal) budgets, with the ∞-norm budget, were also tested and are dominated by scalar-β4 (App. E.1): capacity, set by β, is the lever, not allocation granularity.

Finding 4: allocation family and cross-operator generality. At a fixed $\Sigma|s| = \log d$, the signed (tanh) and non-negative (softplus) allocations are nearly tied on heterophilic accuracy at d=8 (softplus often smaller Jacobian), while signed allocation helps on the homophilic `ogbn-arxiv` (tanh leads softplus by 1.8 at d=16; Table 20)—the budget controls stability, the allocation family controls capacity. The collapse-and-rescue pattern is not GCN-specific: across GraphSAGE, GCN2, GAT, and TransformerConv (Table 4) residual-1 collapses or stalls at the deepest feasible depth while d-STABLE stays competitive with the strongest stabilizer baselines at $\Sigma|s| = \log d$.

Table 4: **Cross-architecture `ogbn-arxiv` depth-collapse test at the deepest full-batch depth per operator** (3 seeds, canonical OGB split). Res1+DE denotes DropEdge (p=0.3); Res1+PN denotes PairNorm. Residual-1 has $\Sigma|s| = d$ in every row, d-STABLE has $\Sigma|s| = \log d$. **Bold** marks the column-best accuracy per row.

Arch	d	Plain	Res1	Res1+DE	Res1+PN	ReZero	Fixup	d-STABLE-tanh	d-STABLE-softplus	$\Sigma\|s\|_{\text{d-STABLE}}$
GCN	64	27.5	16.4	16.3	45.6	69.8	69.5	**71.6**	69.5	4.16
SAGE	64	21.6	21.7	21.7	54.7	70.2	69.9	**71.0**	69.3	4.16
GCN2	64	68.1	28.5	27.6	**70.4**	66.6	64.4	68.4	70.3	4.16
GAT	16	21.6	60.7	66.5	71.3	69.0	71.4	**71.8**	71.1	2.77
TransformerConv	8	58.7	64.6	71.8	**71.8**	68.8	71.0	71.6	70.4	2.08

4.3 IMAGENET: WHERE DEPTH ALREADY WORKS, THE BUDGET HAS A PRICE

ImageNet (Deng et al., 2009) tests the budget where vanilla residual networks are already stable, so the question is the *cost* of enforcing it. We train ResNet-50/152 (He et al., 2016) and DeiT-Tiny (Touvron et al., 2021a) with vanilla, ReZero, Fixup, and d-STABLE-tanh/softplus—adding the Transformer-native LayerScale (Touvron et al., 2021b) and a DeepNorm-style rule (Wang et al., 2022) on DeiT-Tiny—spectral-normalizing only the output-facing projection of each branch—one factor of the branch composition, so this is a-posteriori control, verified by the measured J_{p95} (Sec. 3). We use standard recipes—90-epoch SGD for the ResNets and the 300-epoch DeiT-Tiny recipe (Touvron et al., 2021a)—pinned identically across methods. Every reported top-1 is the training-time validation accuracy of the *source* weights rather than a model-EMA, logged the same way for every method, over N independent runs per row (Table 5); the full hyperparameters, the hardware, and the EMA caveat behind that choice are given in the Reproducibility statement.

Finding 1: ResNets preserve accuracy under the fixed budget. d-STABLE-softplus stays within 0.50 points of vanilla on R50 (76.72 vs. 77.22) and 0.59 on R152 (78.85 vs. 79.44) while the schedule bound drops from $10^{6.95}$ to $10^{1.20}$ (R50) and $10^{21.71}$ to $10^{1.70}$ (R152): the fixed budget with learned allocation converts an exponential-in-depth schedule bound into a polynomial one for under a point of clean top-1. The signed tanh variant costs $\sim$3 points on both ResNets. The measured Jacobian does not follow the schedule bound: Fixup has the lowest J_{p95} at both depths—the expected behaviour of one-factor, a-posteriori normalization, which bounds the schedule factor but not the rest of the input-to-logit map. ReZero shows why an initial scale is not the property: starting at zero, its budget *drifts* to 24.9 (R50) and 38.3 (R152), on R50 a bound looser than vanilla's—what matters is an invariant budget, not a small init.

Finding 2: DeiT-Tiny—the budget transfers, the operating point changes. Both d-STABLE rows pin $\Sigma|s_k| = 3.18 = \log 24$ and cut the bound from $10^{10.42}$ to $10^{1.38}$, but here the clean cost is the largest we measure: tanh 70.81 ± 0.17 and softplus 70.84 ± 0.09 against vanilla 72.68 ± 0.18, a gap of 1.8 points, an order of magnitude above the run-to-run spread. The two d-STABLE variants are tied, and every baseline except ReZero is more accurate, including the Transformer-native LayerScale (71.52) and DeepNorm-style rule (72.01). What orders them is the budget: sorted by $\Sigma|s_k|$, accuracy rises monotonically across every schedule whose mass is fixed or allocated (3.18→70.8, 4.72→71.5, 4.90→71.9, 9.12→72.0, 24→72.7), and the lone outlier is ReZero, whose mass drifts rather than being allocated (7.80→70.0). LayerScale lands near d-STABLE's bound ($10^{2.05}$) only after the fact: its mass doubled from 2.40 in training. The *level* of J_{p95} does not separate the methods (all overlap across seeds); its *seed-to-seed spread* does: the budget-controlled schedules are 6–11$\times$ more reproducible (d-STABLE $\pm$4.6/$\pm$4.7, ReZero $\pm$2.8) than the unbudgeted ones (vanilla $\pm$29.9, Fixup $\pm$31.7). Pinning the budget stabilizes the amplification across initializations even where it does not lower it. At d=24 the budget buys nothing back; DeiT-Tiny is the honest boundary of the two-regime story.

Table 5: **ResNet and DeiT-Tiny on ImageNet.** N is the number of independent runs. Top-1 is the training-time validation accuracy of the source (non-EMA) weights, reported as mean ± std over the last five epochs of all N runs, so it mixes seed variance with end-of-training drift. The remaining columns are the post-training evaluation of each run's best retained checkpoint, mean ± std across evaluated runs, which are fewer than N on R152: one for ReZero, Fixup, d-STABLE-tanh; two for vanilla. The schedule bound is $d^{\beta_{\text{eff}}} = e^{\Sigma|s_k|}$. **Bold**: column-best within an architecture; underline: best d-STABLE top-1, both underlined when tied within one std. †ReZero has no budget: its learned mass drifts from zero to a different value per seed and per architecture. ‡Per-channel schedule: the quantity in the bound is $\sum_k \max_c |s_{k,c}|$, each branch's largest channel summed over branches (initialized at $24 \times 0.1 = 2.40$). §Uniform $s_k = (2n)^{-1/4}$ with n=24 branches; DeepNet's encoder rule counts the 12 blocks, giving $s_k = 24^{-1/4}$ and $\Sigma|s_k| = 10.84$.

Arch	Method	N	Top-1 (%)	$\Sigma\|s_k\|$	$d^{\beta_{\text{eff}}}$	Jac p_{95}	$\text{PGD}_{0.01}$	$\text{PGD}_{0.05}$
R50	Vanilla (s_k=1)	3	$\mathbf{77.22} \pm 0.10$	16.00	$10^{6.95}$	53.0 ± 1.2	13.4 ± 0.4	$\mathbf{0.14} \pm 0.03$
	ReZero	3	76.71 ± 0.13	$24.85 \pm 0.17^{\dagger}$	$10^{10.79\dagger}$	59.4 ± 0.9	8.5 ± 0.1	0.01 ± 0.00
	Fixup (s_k=1/$\sqrt{d}$)	3	76.49 ± 0.10	4.00	$10^{1.74}$	$\mathbf{45.0} \pm 1.2$	11.2 ± 0.2	0.02 ± 0.01
	d-STABLE (tanh)	3	74.18 ± 0.69	2.77	$\mathbf{10^{1.20}}$	49.0 ± 2.4	9.1 ± 2.3	0.02 ± 0.01
	d-STABLE (softplus)	3	$\underline{76.72 \pm 0.15}$	2.77	$10^{1.20}$	51.4 ± 0.7	$\mathbf{14.6} \pm 0.1$	0.08 ± 0.01
R152 (d=50)	Vanilla (s_k=1)	3	79.44 ± 0.12	50.00	$10^{21.71}$	50.8 ± 0.4	19.1 ± 1.1	0.86 ± 0.03
	ReZero	2	$\mathbf{79.75} \pm 0.07$	$38.32 \pm 0.46^{\dagger}$	$10^{16.64\dagger}$	51.7	17.9	0.28
	Fixup (s_k=1/$\sqrt{d}$)	2	78.91 ± 0.10	7.07	$10^{3.07}$	**46.4**	**24.4**	**3.34**
	d-STABLE (tanh)	2	76.72 ± 0.28	3.91	$\mathbf{10^{1.70}}$	48.8	17.0	0.33
	d-STABLE (softplus)	2	$\underline{78.85 \pm 0.08}$	3.91	$10^{1.70}$	52.2 ± 6.6	23.5 ± 0.1	3.09 ± 0.24
DeiT-T (d=24)	Vanilla (s_k=1)	3	$\mathbf{72.68} \pm 0.18$	24.00	$10^{10.42}$	130.5 ± 29.9	$\mathbf{10.7} \pm 0.2$	$\mathbf{0.12} \pm 0.02$
	ReZero	3	69.99 ± 0.77	$7.80 \pm 0.48^{\dagger}$	$10^{3.39\dagger}$	$\mathbf{90.1} \pm 2.8$	6.3 ± 0.8	0.01 ± 0.01
	Fixup (s_k=1/$\sqrt{d}$)	3	71.93 ± 0.23	4.90	$10^{2.13}$	137.4 ± 31.7	8.1 ± 0.2	0.05 ± 0.02
	LayerScale	1	71.52 ± 0.01	$4.72^{\ddagger}$	$10^{2.05}$	139.1	5.1	0.00
	DeepNorm-style	1	72.01 ± 0.01	$9.12^{\S}$	$10^{3.96}$	128.9	5.7	0.01
	d-STABLE (tanh)	3	$\underline{70.81 \pm 0.17}$	3.18	$\mathbf{10^{1.38}}$	147.5 ± 4.6	7.2 ± 0.4	0.04 ± 0.01
	d-STABLE (softplus)	3	$\underline{70.84 \pm 0.09}$	3.18	$10^{1.38}$	143.9 ± 4.7	6.7 ± 0.3	0.06 ± 0.03

Finding 3: PGD does not track the schedule bound monotonically. On R50 d-STABLE-softplus has the best $\text{PGD}_{0.01}$ (14.6, above vanilla 13.4), but on R152 Fixup wins and on DeiT-T vanilla wins, because adversarial accuracy depends on the full input-to-logit map, which the residual-stack budget does not constrain; ImageNet supports the controlled-budget claim, not a robustness one.

5 Limitations

The guarantee is a worst-case schedule bound: the ρ_k in Theorem 1 are operator-norm bounds on the *full* residual branch, so the a-priori certificate holds only under full-branch normalization. On ImageNet we normalize only one layer per branch, leaving the formal ρ_k loose; this is why the theoretical schedule bound can move by many orders of magnitude while the measured J_{p95} saturates at an architecture- and recipe-dependent floor (the DeiT-Tiny boundary case). ImageNet rows aggregate only one to three runs, with an interval that mixes seed variance and end-of-training drift, so near-tied orderings carry limited weight (the core depth claims use 7–10 seeds, on GNNs). And DeiT-Tiny is where the trade-off is least favourable: at d=24 vanilla is the most accurate method, 1.8 points above the best d-STABLE variant, and LayerScale and the DeepNorm-style rule also exceed it, so the ImageNet evidence supports the cost accounting rather than an accuracy claim.

6 Conclusion

We studied residual depth as a budgeted-allocation problem: the budget's growth rate sets the stability class, with $\log d$ the critical rate. d-STABLE pins $\sum_k |s_k| = \beta \log d$ as an exact invariant of optimization, so training moves residual mass across layers without changing how much exists. The technical point is not a new inequality but that the stabilizing quantity can be made an invariant rather than a post-hoc property of a learned gate. The split is the message. A small budget prevents the collapse of deep residual GNNs (16–22% at d=64 on `ogbn-arxiv`), bringing them to the 69.5–69.8% plateau that every small-budget stabilizer reaches; the allocation lifts d-STABLE above it, to $71.4 \pm 0.2\%$. Where depth does not break the model, the budget costs 0.5–1.8 top-1 points on ImageNet, and the a-priori certificate costs 1.8 points on GNNs.

REPRODUCIBILITY STATEMENT

Every number in this paper comes from a script in the anonymized supplementary code, run under a recipe stated in the paper. The residual-budget parameterization and the two variants are defined in full in Sec. 3, with the canonical instantiations and the proof of the depth-scaling claim in App. B and the three-component decomposition in App. C; all assumptions are stated where the result is claimed, and we mark the a-priori and a-posteriori regimes explicitly rather than reporting one bound for both. The synthetic verification, including the schedules swept and the exact fitting procedure for the measured amplification exponents, is in App. D. The `ogbn-arxiv` and heterophilic-graph protocols — operators, depths, seed counts, the matched-budget control and the frozen-allocation retrain — are in App. E, and all GNN results are reported over multiple seeds with dispersion. For the ImageNet results we state for each table row how many seeds it aggregates, and we flag the rows that remain single-seed rather than presenting them as matched to the others. Every d-STABLE-specific hyperparameter (ϵ, the initialization of u_k, the schedule learning-rate scale, and the disabled β_{init}, β_{lr}) is listed in App. C.4.

Full ImageNet recipe. ResNet-50/152 use a 90-epoch SGD schedule; DeiT-Tiny uses the 300-epoch recipe of Touvron et al. (2021a) — AdamW at $\text{lr} = 2\times10^{-3}$ (effective batch 2048, no linear-scaling), 10-epoch warmup with cosine decay, weight decay 0.05, stochastic depth 0.1, RandAugment (`m9-mstd0.5-inc1`), Mixup 0.8 / CutMix 1.0, random erasing 0.25, colour jitter 0.4, label smoothing 0.1, gradient clipping 1.0. ResNet-50/152 use SGD (momentum 0.9, weight decay 10^{-4}) with linear learning-rate scaling over 8 GPUs — ResNet-50 at batch 256/GPU and base $\text{lr} = 0.1$ (effective batch 2048, $\text{lr} = 0.8$), ResNet-152 at 128/GPU and 0.05 (effective 1024, 0.4) — a 5-epoch warmup and cosine decay, label smoothing 0.1, gradient clipping 1.0, and no Mixup, CutMix, or RandAugment. Within each architecture one recipe is pinned across all methods, so the comparison and not merely the headline rests on it. All ImageNet runs are full fp32, on Quadro RTX 8000 (which has no native bf16) except the LayerScale and DeepNorm-style rows (A100); DeiT-Tiny uses 8 GPUs $\times$ batch 256. The DeepNorm-style rule scales each branch by $(2n)^{-1/4}$ with n the branch count (24) rather than DeepNet's block count (12), a 16% smaller residual scale, so we report it as a DeepNorm-style rule rather than as DeepNet. A model-EMA (decay 0.99996) is maintained but *not* used for any reported number: the d-STABLE schedule is registered as a shared submodule, which timm's `ModelEmaV2` does not update correctly, so we evaluate the source weights for every method.

ETHICS STATEMENT

This work is foundational research on training stability. It uses only public benchmark datasets (`ogbn-arxiv`, the heterophilic node-classification suites, Long-Range Graph Benchmark, and ImageNet-1k), involves no human subjects, no new data collection, and no personally identifying information. We release no pretrained artifact that would itself constitute a misuse risk: the checkpoints accompanying the code are standard backbones at standard scales. A fuller treatment of the societal considerations, including the deployment-layer argument for where risk-gating belongs, is in App. F. We declare no conflicts of interest.

AI USE STATEMENT

We used generative AI tools in three roles, described below. We have reviewed all AI-assisted work.

Aiding and polishing writing, and organizing code. We used these tools to edit, condense and polish prose that we had drafted ourselves, and to help construct LaTeX tables and figure scaffolding. Every resulting sentence was checked by the authors against the underlying results, and no claim in this paper originates in generated text. We also used them to reorganize and document the released code for readability (restructuring scripts, clearer names and comments). Every such change was reviewed by the authors and verified by re-running the unit tests and regenerating the reported tables from the logged runs.

Retrieval and discovery. We used these tools to help surface related work, including the contemporary residual-mixing literature discussed in App. A. Every reference found this way was then read in the original by the authors, and we confirmed that each claim we attribute to a cited work appears in that work; no citation or characterization of prior work rests on an AI-generated summary.

Generating synthetic datasets. We used these tools to help generate the synthetic residual-chain configurations underlying the verification in App. D. These data are checked against theory rather than taken on trust: the purpose of that experiment is to test whether the measured amplification exponent matches the value the budget law predicts, and the generating code was read, re-run and verified by the authors. No benchmark data was generated or modified in this way — the `ogbn-arxiv`, heterophilic, long-range and ImageNet results use the public datasets as distributed.

We did not use generative AI tools to generate research ideas or to derive or prove any theoretical result. AI-assisted scripts were used to aggregate logged runs into the reported tables; every reported number is computed from those logs, and the authors reviewed the aggregation code and checked the resulting numbers against the runs. No experimental result was generated, estimated, or altered by an AI tool. We take full responsibility for the final content of this work, including all text, claims, and artifacts.

REFERENCES

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Haiqiang Mao, Garrison W. Cottrell, and Julian McAuley. Rezero is all you need: Fast convergence at large depth. In *UAI*, 2020.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

Guangyu Chen, Yu Zhang, Jianlin Su, Zhilin Yang, Xinyu Zhou, et al. Attention residuals. *arXiv preprint arXiv:2603.15031*, 2026.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *ICML*, 2020.

Ricky T.Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.

Ziang Chen, Zhengjiang Lin, Shi Chen, Yury Polyanskiy, and Philippe Rigollet. Residual connections provably mitigate oversmoothing in graph neural networks. *arXiv preprint arXiv:2501.00762*, 2025.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

Soham De and Samuel L. Smith. Batch normalization biases residual blocks towards the identity function in deep networks. In *NeurIPS*, 2020.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. Technical report, DeepSeek-AI, 2026. `https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf`.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *ICML*, 2021.

Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of Lipschitz constants for deep neural networks. In *NeurIPS*, 2019.

Soufiane Hayou, Eugenio Clerico, Bobby He, George Deligiannidis, Arnaud Doucet, and Judith Rousseau. Stable ResNet. In *AISTATS*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized PageRank. In *ICLR*, 2019.

Klas Leino, Zifan Wang, and Matt Fredrikson. Globally-robust neural networks. In *ICML*, 2021.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 2018.

Qiyang Li, Saminul Haque, Cem Anil, James Lucas, Roger B Grosse, and Jörn-Henrik Jacobsen. Preventing gradient attenuation in Lipschitz constrained convolutional networks. In *NeurIPS*, 2019.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in Transformers: Theoretical perspectives and the role of rank collapse. In *NeurIPS*, 2022.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *ICLR*, 2020.

MoonJeong Park and Dongwoo Kim. Taming gradient oversmoothing and expansion in graph neural networks. *arXiv preprint arXiv:2410.04824*, 2024.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *NeurIPS*, 2022.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-Depths: Dynamically allocating compute in transformer-based language models. arXiv:2404.02258, 2024.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards deep graph convolutional networks on node classification. In *ICLR*, 2020.

T. Konstantin Rusch, Benjamin P. Chamberlain, Michael W. Mahoney, Michael M. Bronstein, and Siddhartha Mishra. Gradient gating for deep multi-rate learning on graphs. In *ICLR*, 2023.

Michael Scholkemper, Xinyi Wu, Ali Jadbabaie, and Michael T. Schaub. Residual connections and normalization can provably prevent oversmoothing in GNNs. In *ICLR*, 2025. arXiv:2406.02997.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention. In *ICML*, 2021a.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, 2021b.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. DeepNet: Scaling Transformers to 1,000 layers. In *arXiv preprint arXiv:2203.00555*, 2022.

Zhenda Xie, Yixuan Wei, Huanqi Cao, et al. mHC: Manifold-constrained hyper-connections. *arXiv preprint arXiv:2512.24880*, 2025.

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. In *ICLR*, 2019.

Lingxiao Zhao and Leman Akoglu. PairNorm: Tackling oversmoothing in GNNs. In *ICLR*, 2020.

A Related Work — Full Treatment

This appendix expands the positioning given in Sec. 2.

Lipschitz control and residual scaling. Two prior families of stabilization methods constrain different objects. Spectral normalization (SN) (Miyato et al., 2018) and related Lipschitz-control approaches (Fazlyab et al., 2019; Li et al., 2019; Cohen et al., 2019; Leino et al., 2021) apply *depth-uniform* per-layer norm bounds, but cannot prevent the residual product $\prod_k(1+\rho|s_k|)$ from growing exponentially with depth even when each $\|J_g^{(k)}\| \leq 1$. Residual-scale methods do limit branch contributions: ReZero (Bachlechner et al., 2020) initializes learnable per-layer scalars at zero; Fixup (Zhang et al., 2019) fixes $s_k = 1/\sqrt{d}$; DeepNorm (Wang et al., 2022) sets $\alpha = (2d)^{1/4}$ for Transformer scaling. These share the intuition that residual contributions should be modulated with depth but implement it via initialization, fixed schedules, or architecture heuristics, without an explicit constraint on end-to-end sensitivity. Two further initialization/scaling designs share this per-branch character: SkipInit (De & Smith, 2020) zero-initializes a learnable scalar per residual block, and Stable ResNet (Hayou et al., 2021) scales each branch by $1/\sqrt{d}$ so that forward and backward dynamics become sub-exponential rather than exponential in depth. Both fix the *per-branch* scale (at initialization, or by a fixed rule); neither pins the *cumulative* mass $\sum_k |s_k|$, which is free to drift once the branches become active (Sec. 4.3, Table 5: ReZero drifts to $\sum_k |s_k|$=24.9 on R50). Directly matched on our GNN stress-test, the fixed $1/\sqrt{d}$ per-branch rule ($\sum_k |s_k| = \sqrt{d}$) prevents collapse but trails the pinned $\log d$ budget at d=64 (Sec. 4.2). d-STABLE combines the two families by pinning the cumulative sum $\sum_k |s_k|$ at $\beta \log d$ throughout training, so the schedule directly governs the amplification bound (Theorem 1) at every step.

Depth instability in GNNs and Transformers. Depth-induced degradation manifests differently across architectures. We do not claim these failures are literally the same phenomenon; rather, each involves the compounding of small per-layer maps with depth, and we control a bound on that compounding (the end-to-end Jacobian, Sec. 3) that is common to all of them. In GNNs, oversmoothing causes representations to converge with depth (Oono & Suzuki, 2020; Cai & Wang, 2020; Li et al., 2018), motivating interventions such as PairNorm (Zhao & Akoglu, 2020), DropEdge (Rong et al., 2020), propagation-based approaches (Klicpera et al., 2019), and gating schemes such as Gradient Gating (Rusch et al., 2023). A recent line makes the residual remedy precise: residual connections and normalization can *provably* slow the asymptotic oversmoothing rate (Scholkemper et al., 2025; Chen et al., 2025), and Park and Kim (Park & Kim, 2024) show that residuals also induce *gradient* expansion, so per-layer Lipschitz control—not residual mass alone—governs trainability at large depth. These results are asymptotic-rate statements about a specific GNN failure (feature convergence); our budget law instead bounds the end-to-end Jacobian directly and is architecture-agnostic. The distinction matters here: our measurements (App. E.3) show the observed d=64 residual collapse is *not* oversmoothing—the residual arms' Dirichlet energy *rises* with depth rather than decaying, and the collapse is most directly seen in the end-to-end Jacobian, which explodes for an unbudgeted residual while staying $O(10)$ under the budget. Slowing the asymptotic oversmoothing rate (Scholkemper et al., 2025; Chen et al., 2025) therefore does not by itself address it; controlling the Jacobian addresses both the smoothing and the gradient-expansion axes that Park and Kim (Park & Kim, 2024) separate. In Transformers, rank collapse and signal degradation emerge from repeated attention and residual accumulation (Dong et al., 2021; Noci et al., 2022). These are typically addressed through architecture-specific solutions. Our work targets the underlying mechanism—multiplicative accumulation of perturbations—and provides a unified solution applicable across GNNs, CNNs, and Transformers.

Depth-dependent residual mixing. Two contemporary lines of work attack related residual-accumulation failures while constraining different objects. Manifold-Constrained Hyper-Connections (mHC) (Xie et al., 2025), deployed in DeepSeek-V4 (DeepSeek-AI, 2026), expand the residual stream by a factor n_{hc} and project the per-layer linear-mixing matrix B_l onto the Birkhoff polytope of doubly stochastic matrices, ensuring $\|B_l\|_2 \leq 1$ uniformly. Attention Residuals (AttnRes) (Chen et al., 2026), integrated into the Kimi Linear architecture, replace unit-weighted depth recurrence with softmax aggregation $h_l = \sum_{i<l} \alpha_{i\to l} f_i(h_i)$ where $\sum_i \alpha_{i\to l} = 1$ for each target layer l. The three approaches occupy different axes on the residual-mapping design space: d-STABLE constrains cumulative scalar mass $\sum_k |s_k|$, mHC constrains a linear mixing matrix, and AttnRes constrains

aggregation weights. d-STABLE is the only one of the three with a closed-form depth-dependent amplification bound (Theorem 1, polynomial in d).

Discrete depth routing. Mixture-of-Depths (Raposo et al., 2024) also decides where a network spends its depth, but through a different object: a learned router lets each token skip or enter a block under a fixed compute capacity, so the allocation is discrete, per token, and priced in FLOPs. d-STABLE allocates a continuous residual *magnitude* per layer, shared across inputs, under a fixed ℓ_1 budget priced in worst-case sensitivity. The two are compatible—a routed block can still carry a budgeted scale s_k—and neither constrains the quantity the other controls.

Dynamical systems perspective. Deep residual networks can be viewed as discretizations of dynamical systems (E, 2017; Chen et al., 2018), where stability depends on the accumulation of transformations over continuous "time." Our harmonic schedule $s_k = c/(k+\tau)$ is the Euler discretization of the ODE $\dot{x} = (c/t)\, g(x)$, whose solution grows as $O(t^c)$—a polynomial rate. This connection places d-STABLE within the dynamical systems framework while operating on standard discrete architectures. Scaling laws (Kaplan et al., 2020) relate performance to model/data size; our depth scaling law is orthogonal, characterizing how *sensitivity* scales with *depth*.

What the fixed-budget invariant adds over prior depth-scaling designs. Relative to the closest results, the specific contribution is the object being controlled and *when*. Per-branch scaling (Stable ResNet's $1/\sqrt{d}$ (Hayou et al., 2021)) and initial-residual/gate designs (ReZero, LayerScale, SkipInit (Bachlechner et al., 2020; Touvron et al., 2021b; De & Smith, 2020)) fix the *initial* or *per-branch* scale and let the cumulative mass $\sum_k |s_k|$ evolve freely; the provable-oversmoothing GNN analyses (Scholkemper et al., 2025; Chen et al., 2025) bound an *asymptotic rate* for a specific architecture and failure mode; and Park and Kim (Park & Kim, 2024) control the *per-layer* Lipschitz constant. d-STABLE instead pins the *total* $\sum_k |s_k| = \beta \log d$ *exactly, at every optimization step*, which is what converts the product bound $\prod_k (1 + \rho_k |s_k|)$ into an $O(d^{\beta\rho})$ ceiling the drifting-scale designs cannot claim (a learned per-branch gate reaches $\sum_k |s_k|$=24.9 on ResNet-50, Table 5). It is architecture-agnostic where the GNN results are not, and it constrains the cumulative budget where per-layer Lipschitz control constrains each factor—the budget–branch-norm decomposition (App. E.1) shows the *budget* factor, not the per-branch norm ρ, is what governs the measured blow-up. Benchmarking these designs under a single matched recipe bears this out (App. E.3): the learned-gate and deep-GNN baselines (ReZero, LayerScale, GCNII, and Gradient Gating—the last an orthogonal per-node *rate* gate) all trail the budgeted allocation at d=64, and DeepNorm's $(2d)^{1/4}$ rule spends $\sum_k |s_k|$≈19—a large budget that lands in the collapse-prone regime, with or without its post-LayerNorm, so we label it in the tables as a non-faithful port rather than as DeepNet. The total-budget invariant is thus not merely a distinct object but the empirically better-behaved one.

B Canonical Instantiations

Theorem 1 and the class hierarchy of Table 1 yield several special cases used in the main text. Eq. (3) is stated for infinitesimal perturbations. For a finite δx, the mean-value theorem gives $\|\delta X^{(d)}\| \le \sup_{t\in[0,1]} \|J_{\text{eff}}^{(d)}(x + t\,\delta x)\|\, \|\delta x\|$, and every bound below controls this supremum whenever ρ_k bounds $\|J_g^{(k)}\|$ over the whole segment (e.g. globally, as under full-branch normalization).

Corollary 3 (Uniform local bound)**.** *If $\rho_k \le \rho$ for all k and $\sum_k |s_k| \le C \log d$, then $\|J_{\text{eff}}^{(d)}\| \le d^{\rho C}$.*

Corollary 4 (Harmonic scaling)**.** *Let $s_k = \frac{c}{k+\tau}$ with $c, \tau > 0$ and $\rho_k \le \rho$. Then $\sum_k \rho_k |s_k| = O(\rho c \log d)$, and therefore $\|J_{\text{eff}}^{(d)}\| \le d^{O(\rho c)}$.*

Corollary 5 (Budget-preserving scaling)**.** *Let $w_k \ge 0$ and $s_k = B_d \cdot \frac{w_k}{\sum_j w_j}$ with $B_d = C \log d$. If $\rho_k \le \rho$, then $\|J_{\text{eff}}^{(d)}\| \le d^{\rho C}$: the per-layer shape of a budget-preserving allocation does not affect the depth exponent.*

Corollary 6 (Signed and learnable scaling)**.** *Let $r_k \in \mathbb{R}$ and $s_k = B_d \cdot \frac{r_k}{\sum_j |r_j|}$ with $B_d = C \log d$, so that $\sum_k |s_k| = B_d = C \log d$ regardless of the signs of the r_k. If $\rho_k \le \rho$, then $\|J_{\text{eff}}^{(d)}\| \le d^{\rho C}$. This is the d-STABLE parameterization* (5)*: the exponent is fixed for any learned allocation.*

Proposition 7 (Phase transition of the worst-case bound)**.** *Let $S_d = \sum_k \rho_k |s_k|$. The upper bound of Theorem 1 exhibits three regimes:*

- $S_d = O(1)$*: the bound is* $O(1)$ *(bounded);*
- $S_d = \Theta(\log d)$*, i.e.* $c_1 \log d \le S_d \le c_2 \log d$ *for large* d*: the bound* e^{S_d} *lies between* d^{c_1} *and* d^{c_2}*, i.e. it is* $O(d^{c_2})$ *and* $\Omega(d^{c_1})$ *(polynomial);*
- $S_d = \omega(\log d)$*: the bound is* $\omega(d^C)$ *for every fixed* C *(super-polynomial).*

This is the formal statement behind Table 1 and Corollary 2. The statement concerns the upper bound only: individual products can contract through cancellation, so large S_d *permits—but does not force—super-polynomial amplification. A matching lower bound requires a construction in which the per-layer factors align; we verify empirically (Sec. 4.1, App. D.1) that the critical boundary* $S_d = \Theta(\log d)$ *is tight in typical configurations and that* $S_d = \omega(\log d)$ does *produce super-polynomial growth in practice under generic random initializations.*

C Details of the Three Orthogonal Components

The analysis of Sec. 3 separates depth stability into one quantity that enters the bound (the budget) and two that only shape training dynamics. The three-component framework is summarized below.

Component	What it controls	Mechanism	Effect	Guarantee
Forward scaling	stability budget	s_k with $\sum_k \lvert s_k\rvert \le C \log d$	controls accumulation	$O(d^C)$
Initialization	warm-start balance	$W_{\text{init}}^{(k)} \propto \lvert s_k\rvert^{-\beta_{\text{init}}}$	uniform initial signal	no-op under full-branch SN
Optimization	learnability	$\eta_k \propto \lvert s_k\rvert^{-\beta_{\text{lr}}}$	uniform gradient signal	no effect on bound (SGD only)

C.1 Component 1: forward scaling

The first component sets the schedule $\{s_k\}$ via the normalized parameterization (5), with total budget $B_d = \beta \log d$ distributed by coefficients $r_k = \tanh(u_k)$. This is the only component whose form enters the bound: any allocation with $\sum_k |s_k| = O(\log d)$ yields $\left\| J_{\text{eff}}^{(d)} \right\| = O(d^{\rho\beta})$ under bounded local operators.

Local operator control: from hard constraints to flexible regularization. The bound $\left\| J_{\text{eff}}^{(d)} \right\| \le \exp(\sum_k \rho_k |s_k|)$ depends on the local norms $\rho_k = \left\| J_g^{(k)} \right\|$. A sufficient condition for polynomial stability is a uniform $\rho_k \le \rho$, e.g. via spectral normalization (SN) (Miyato et al., 2018), giving $\left\| J_{\text{eff}}^{(d)} \right\| \le d^{\rho C}$. This is not necessary: the analysis requires only that the *aggregate* $\sum_k \rho_k |s_k|$ stay $O(\log d)$, so individual layers may have $\rho_k > 1$ if compensated by smaller $|s_k|$. We therefore treat SN as an optional auxiliary mechanism with three regimes: (i) hard normalization ($\left\| W^{(k)} \right\| \le 1$ every step), strongest guarantee, potentially limiting expressivity; (ii) soft spectral regularization; and (iii) unconstrained operators, relying on the budget alone. Empirically, hard normalization helps most in highly unstable regimes (deep GNNs) while relaxed or absent normalization is better on ImageNet—consistent with stability being governed by the joint $\sum_k \rho_k |s_k|$, the budget providing primary control and the local norms modulating the constant.

C.2 Component 2: budget-aware initialization

A nonuniform schedule attenuates branch outputs unequally at initialization. To equalize effective branch signal from step zero we compensate the base initialization inversely with the scheduled magnitude:

$$W_{\text{init}}^{(k)} = (|s_k^{(0)}| + \epsilon)^{-\beta_{\text{init}}} W_{\text{base}}^{(k)}, \qquad \beta_{\text{init}} \in [0, \tfrac{1}{2}], \tag{6}$$

with $W_{\text{base}}^{(k)}$ a standard (e.g. Kaiming) initialization. The compensated branch output scales as $|s_k^{(0)}|^{1-\beta_{\text{init}}}$, so any $\beta_{\text{init}} < 1$ preserves the ordering of the allocation; the cap $\beta_{\text{init}} \le \frac{1}{2}$ is a conservative heuristic rather than a derived threshold. Component 2 leaves the budget $\sum_k |s_k|$ unchanged but not the weighted budget, and its effect depends on the regime. Under full-branch spectral normalization it is a *no-op* in the forward pass, since $cW/\sigma(cW) = W/\sigma(W)$ for any scalar $c > 0$. Without normalization it raises the initial ρ_k by $(|s_k^{(0)}| + \epsilon)^{-\beta_{\text{init}}}$, so at step zero $S_d \propto \sum_k |s_k^{(0)}|^{1-\beta_{\text{init}}}$, which for a near-uniform allocation grows as $d^{\beta_{\text{init}}}(\log d)^{1-\beta_{\text{init}}}$—outside the logarithmic class. For both

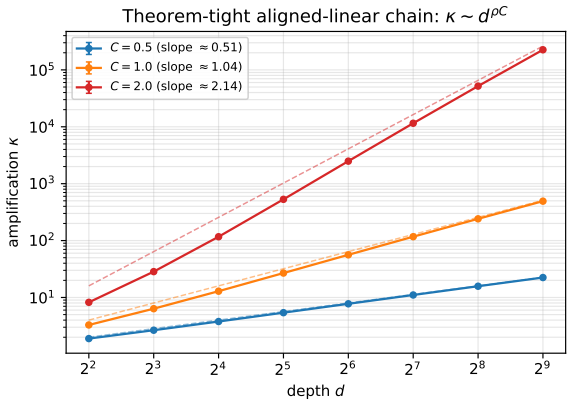


Figure 2: Aligned-linear chain (every branch is the identity, ρ=1): empirical κ vs. depth at three uniform budgets $\sum_k |s_k| = C \log d$. Solid: measured (mean $\pm$ std, 3 seeds); dashed: envelope d^C. The exact product $(1 + C \log d/d)^d$ sits at or below d^C at every depth; empirical curves track the envelope with $R^2 \geq 0.999$.

reasons Component 2 is disabled ($\beta_{\text{init}} = 0$) in every reported experiment. The symbol β_{init} is distinct from the budget scalar β in (5).

C.3 Component 3: depth-aware optimization

Under a uniform learning rate the gradient $\partial \mathcal{L}/\partial W^{(k)}$ inherits the factor s_k, so small-$|s_k|$ layers learn slower. We restore per-layer learnability with

$$\eta_k = \eta_0 \cdot (|s_k^{(0)}| + \epsilon)^{-\beta_{\text{lr}}}, \qquad \beta_{\text{lr}} \in [0.5, 1], \tag{7}$$

applied only to residual-branch parameters. Like Component 2, this affects training dynamics but not the polynomial bound. Its rationale is specific to SGD: Adam-family optimizers normalize each update by $\sqrt{\hat{v}}$, which cancels a constant gradient factor s_k (up to the optimizer's ϵ), so the multiplier would over-compensate low-budget layers there. The multiplier is also static, computed from $s_k^{(0)}$, whereas the learned allocation moves during training; a dynamic multiplier tracking the running s_k would be the consistent choice. Component 3 is disabled ($\beta_{\text{lr}} = 0$) in every reported experiment, including the SGD ResNets.

C.4 Values used in the reported experiments

All reported results use Component 1 alone: $\beta_{\text{init}} = \beta_{\text{lr}} = 0$, so the $\epsilon = 10^{-3}$ of Components 2–3 never enters. The allocation stabilizer in (5) is $\epsilon = 10^{-8}$; it lowers $\sum_k |s_k|$ below $\beta \log d$ by a relative $\epsilon / \sum_j |r_j|$, so every upper bound still holds. The allocation logits are initialized as $u_k \sim \mathcal{N}(0, 0.1^2)$, which keeps $\sum_j |r_j|$ away from zero (the $0/0$ state of an all-zero initialization never arises). On `ogbn-arxiv`, the heterophilic sets, and ImageNet, the schedule parameters u_k are trained at $0.01\times$ the base learning rate with no weight decay; the long-range (LRGB) runs train them at the base learning rate. The budget scalar β is 1 unless a table states otherwise.

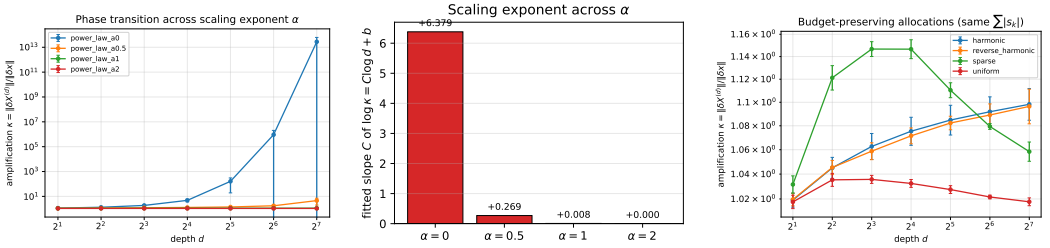
D Results of Synthetic Verification

This appendix gives the full synthetic verification summarized in Sec. 4.1: five untrained constructions that isolate the theorem's mechanism, then a trained comparison against standard residual-scaling baselines with extreme-depth and learning-rate stress probes.

D.1 Theorem-tight construction: aligned linear branch

We first verify that the bound is approachable. With the aligned linear branch $g(X) = X$ (every branch operator is the identity), the chain reduces to the scalar recurrence $y_{k+1} = (1 + s_k)y_k$; for uniform $s_k = C \log d/d$ the exact product satisfies $\prod_k (1 + s_k) = (1 + C \log d/d)^d \leq e^{C \log d} = d^C$

by $\log(1+x) \le x$, with the gap $d^C - \prod_k (1+s_k) \to 0$ as $d \to \infty$. Sweeping $d \in \{4, \dots, 512\}$ at $C \in \{0.5, 1, 2\}$, $\rho = 1$ (Fig. 2), the fitted log–log slopes are $0.51/1.04/2.14$ vs. predictions $\rho C = 0.5/1/2$, $R^2 \ge 0.999$. (The slight excess at C=1, 2 is a finite-depth regression artifact; the product is always at or below d^C.) The bound is near-saturated, confirming the polynomial $d^{\rho\beta}$ ceiling is the right object.



(a) Scaling regimes across schedule families.

(b) Fitted slope C of $\log \kappa = C \log d + b$ across α.

(c) Budget redistribution under fixed $\sum_k |s_k| = \log d$.

Figure 3: Depth scaling law on untrained residual chains. (a) κ vs. depth for $s_k = c/(k+\tau)^\alpha$, $\alpha \in \{0, 0.5, 1, 2\}$. (b) Fitted slope across α: the critical transition sits at α=1 (Prop. 7). (c) Budget-preserving allocations (harmonic, uniform, reverse, sparse) collapse onto one polynomial band once $\sum_k |s_k| = \log d$ is fixed (Cor. 5).

D.2 Phase transition across power-law schedules

Sweeping $s_k = c/(k+\tau)^\alpha$ across $\alpha \in \{0, 0.5, 1, 2\}$ on the standard ReLU chain $g(X) = \mathrm{ReLU}(\hat{A}XW^{(k)})$ (Fig. 3b), the fitted depth-slope of $\|J_{\text{eff}}\|$ falls from 6.57 at α=0 (linear budget, exponential) to 0.28 at α=0.5 ($\sqrt{d}$, sub-exponential) to 0.009 at α=1 (harmonic, $\log d$, polynomial) to ≈ 0 at α=2 (bounded). The transition is sharp at the predicted α=1, and the harmonic case fits a polynomial with $R^2 = 0.93$—Table 1 on the untrained chain.

D.3 Budget allocation: shape affects constants, not depth scaling

Holding $\sum_k |s_k| = c \log d$ fixed and varying the profile (uniform, harmonic, reversed, sparse; Fig. 3c), all four collapse into a tight band $\kappa \in [1.02, 1.10]$ at d=128: the per-layer arrangement is a constant factor, the total mass governs the depth-scaling (Cor. 5).

D.4 Signed budget: cancellation under the same absolute mass

Corollary 6 predicts the bound depends only on $\sum_k |s_k|$. Testing three signed schedules renormalized to $\sum_k |s_k| = \log d$ (alternating, random, sparse-signed) on the ReLU chain over $d \in \{4, \dots, 512\}$ (Fig. 4), the fitted slope of $\|J_{\text{eff}}\|$ is $+0.010/+0.010/-0.027$, all far below the bound $\rho\beta = 1$. Signed allocations inherit the same ceiling and are typically smaller, as $(I + s_k J_k)$ cancellation—which the $\log(1+x) \le x$ step does not exploit—tightens the result.

D.5 Random nonlinear sanity: the bound is conservative in the typical case

On the random ReLU chain the $\rho\beta = 1$ bound predicts slope 1.0, so the empirical amplification (harmonic slope 0.009) is $\sim$2 orders of magnitude below the ceiling: the bound captures the worst-case capacity of the constraint set (block 1 approaches it), while typical directions traverse a milder portion. The same gap surfaces in Sec. 4.2 (residual-1 GCN J_{p95} vs. the schedule bound) and Sec. 4.3 (DeiT-T). Parametrically (Fig. 5), the empirical slope grows monotonically with ρ ($0.005 \to 0.019$), the harmonic scale c ($0.001 \to 0.315$), and C ($0.000 \to 0.181$), all within the envelope and linear-in-parameters as Corollaries 3–5 predict.

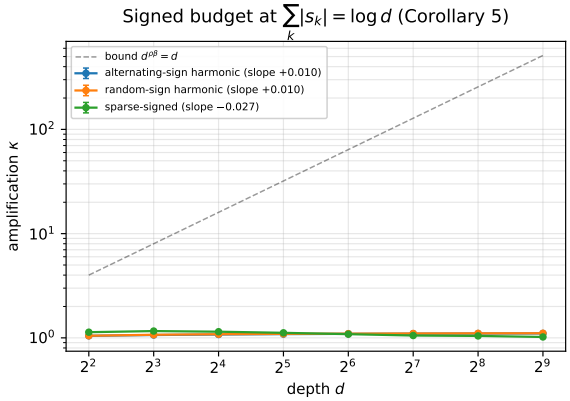


Figure 4: Signed allocations under $\sum_k |s_k| = \log d$. All three sit far below the unsigned bound $d^{\rho\beta} = d$ (dashed). Cancellation tightens the slope by $\sim$2 orders of magnitude.

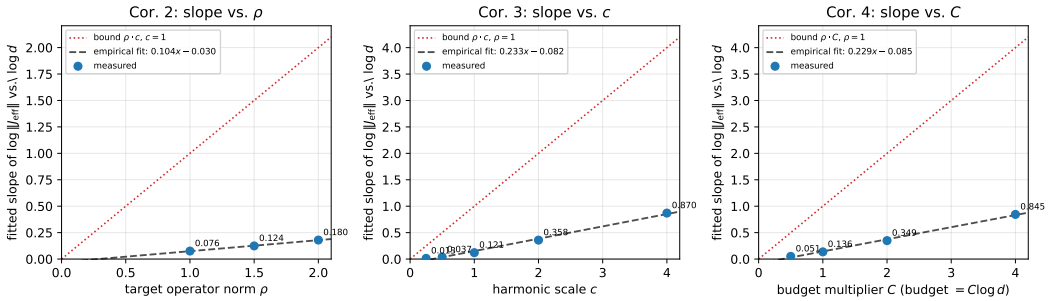


Figure 5: Quantitative verification of Corollaries 3, 4, 5: empirical slope (random ReLU chain) grows monotonically with ρ, c, and C, within the upper-bound envelope; linear in ρ, approximately linear in c, C with a superlinear correction above 2 as the $\log(1 + x) - x$ slack accumulates.

D.6 Design-space comparison: initialization and trained

We evaluate d-STABLE, ReZero, Fixup, and unconstrained scaling on the same chain, at initialization and after training on a teacher–student reconstruction task (random two-layer $\tanh$ MLP teacher; MSE, Adam lr 10^{-2}, B=16, 500 steps, 3 seeds; Fig. 6).

At initialization, unconstrained overflows at d=128 ($\sum_k |s_k| = d$, exponential); Fixup sits super-polynomial ($\sum_k |s_k| = \sqrt{d}$); ReZero is stable only by suppressing the branch ($s_k \equiv 0$, identity map); d-STABLE keeps a nonzero contribution ($\|J_{\text{eff}}\| = 2.97$, $\sum_k |s_k| = \log d$) in the polynomial regime by construction. After training, d-STABLE stays controlled (κ=1.13, $\|J_{\text{eff}}\|$ =4.54 at d=128); unconstrained is catastrophic; ReZero's $\sum_k |s_k|$ drifts to 29.4 (κ=2.92, $\|J_{\text{eff}}\|$ =18.4)—a 4$\times$ gap. The learnable d-STABLE of the synthetic trainer (β learned in log-space, allocation as in Cor. 6) holds $\beta_{\text{eff}} \approx 2.85$ across $d \in \{2, 8, 32, 128\}$ for a $\sim$700$\times$ loss reduction, while ReZero's β_{eff} drifts $1.78 \to 5.65$ (bound looser by $d^{3.87} \approx 10^8$ at d=128). This "ReZero drifts during training" finding recurs in the GNN (Table 2) and ImageNet (Table 5) sections at scale.

Learning-rate robustness. Across lr $\in \{10^{-4}, \ldots, 10^{-1}\}$ (7 values) at d=64, 3 seeds, all budget-preserving methods (d-STABLE fixed/learnable, Fixup) and ReZero train at every rate (0/3 divergences); unconstrained diverges partially at lr=10^{-4} (1/3) and completely at 10^{-1} (3/3)—the $O(d)$ budget admits unstable trajectories the $\log d$ budget forbids.

E GNN Experiments — Full per-dataset tables and controls

E.1 Budget $\times$ branch-norm decomposition: ρ does not predict the blow-up

The a-posteriori certificate (Sec. 3) bounds the amplification by $\prod_k (1 + \rho_k |s_k|)$; in logs this is the weighted budget $\sum_k \rho_k |s_k| \le \rho \, \Sigma|s|$, the per-branch norm $\rho = \max_k \|W^{(k)}\|$ times the residual

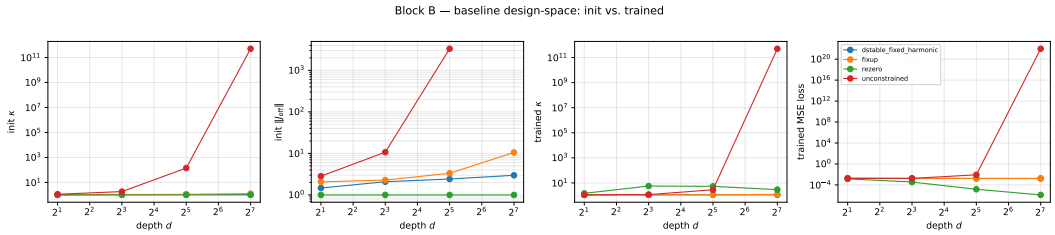


Figure 6: d-STABLE vs. ReZero, Fixup, unconstrained on the synthetic chain. Left: init κ and $\|J_{\text{eff}}\|$ vs. depth. Right: κ and MSE after training. ReZero starts trivially stable (κ=1, branch suppressed) but drifts to κ=2.92 after training ($\sum_k |s_k|$ grows $0 \to 29.4$); d-STABLE (harmonic) and Fixup stay in the polynomial band throughout.

Table 6: **Extreme-depth extrapolation** (synthetic chain; final training loss after 150 steps and $\sum_k |s_k|$ at the end of training, mean over 2 seeds). Unconstrained diverges at every depth $d \geq 128$. Fixup's loss grows with depth as its budget $\sqrt{d}$ grows. Fixed-harmonic d-STABLE holds the loss flat at $\Sigma|s_k| = \log d$. Learnable d-STABLE (β learned) and ReZero both reach $\sim 10^{-5}$, but ReZero's mass drifts to 39 ($\beta_{\text{eff}} \approx 5.6$ at d=1024) while d-STABLE stays at $\beta_{\text{eff}} \approx 2.3$—a $d^{3.3}$ tighter bound.

Method	Final loss by depth				$\sum_k \lvert s_k \rvert$ by depth			
	d=128	256	512	1024	128	256	512	1024
Unconstrained	1×10^{27}	diverged	diverged	diverged	128	256	512	1024
Fixup	0.010	0.018	0.054	0.093	11.31	16.00	22.63	32.00
d-STABLE (harmonic)	0.006	0.007	0.006	0.007	4.85	5.55	6.24	6.93
ReZero	6×10^{-6}	2×10^{-6}	9×10^{-6}	7×10^{-6}	24.84	29.77	32.20	39.03
d-STABLE (learnable)	4×10^{-5}	1×10^{-4}	1×10^{-5}	2×10^{-5}	11.49	13.07	13.23	15.87

budget $\Sigma|s|$. To show these two factors are separable—and that what d-STABLE controls (the budget) is the one that matters—we measure both on the same trained models. On `ogbn-arxiv` GCN at $d \in \{32, 64\}$ (3 seeds) we cross the budget axis (d-STABLE $\beta \in \{1, 2, 4\}$; Fixup, $\Sigma|s| = \sqrt{d}$; residual-1, $\Sigma|s| = d$) with spectral normalization off/on, recording $\Sigma|s|$, ρ, the measured J_{p95}, and accuracy (Table 7, Fig. 7).

The decomposition is decisive. (i) J_{p95} stays $O(10)$ and flat in depth for every d-STABLE budget (28–30 at both depths), while residual-1 explodes from 157 to 7.8×10^6 across d=32 $\to$ 64—the Jacobian blow-up the budget prevents, measured end-to-end. (ii) ρ does not predict J_{p95}: without SN, residual-1 has the *lowest* ρ (2.05–2.57) yet the largest J_{p95}, because the depth-accumulated product is driven by $\Sigma|s|$=d, not the per-layer norm SN controls. (iii) SN clamps J_{p95} but not accuracy: residual-1+SN at d=64 is forced to J_{p95}=7.0 yet still collapses to 11.7%—a small Jacobian alone is not sufficient; only the budget delivers both bounded J_{p95} and 71.5%.

A-priori certificate and its accuracy at full-branch SN. For GCN the residual branch is a *single* conv, so spectral-normalizing it pins $\|J_g^{(k)}\| \leq \rho \approx 1$ a priori, and Theorem 1 gives $\|J_{\text{eff}}^{(d)}\| \leq d^{\rho\beta}$ by construction. The normalization divides by a one-step power-iteration estimate of $\sigma(W)$, which can undershoot, so the realized branch norm is 1.00–1.15 across all 30 SN checkpoints of Table 7, not exactly ≤ 1; converging the power iteration would make $\rho \leq 1$ exact. Table 8 compares against the stricter d^{β}: the measured J_{p95} stays 25–30 across $\beta \in \{1, 2, 4\}$, $d \in \{32, 64\}$ and sits strictly below d^{β} in every cell—tightly at β=1 yet flat in depth. This certified configuration's *accuracy* is 69.3% at d=32 and 69.65 ± 0.07% at d=64 (3 seeds), about 1.8 points below the no-normalization headline (71.4 ± 0.2) but far above residual-1's 14.6% collapse: the a-priori polynomial certificate—the property no competing stabilizer has—at a small, quantified price. (d=64 cells were trained full-batch with `expandable_segments`.)

Energy-decay exponents, and why we demote the energy ranking. We fit the per-layer normalized Dirichlet energy $E_\ell = \sum_{(i,j)\in E} \|h_i - h_j\|^2 / (2 \operatorname{tr} h^\top h)$ through the d=64 network to a log-linear exponent γ (with our sign convention γ<0 means energy *grows* with depth—the gradient-expansion direction—and γ>0 means it *decays*, the classical oversmoothing signature), over 7 seeds (Fig. 8, Table 9). This settles one thing decisively: **the d=64 residual collapse is not oversmoothing**—every residual method has γ<0 (energy grows), and only the non-residual plain GCN decays (γ>0).

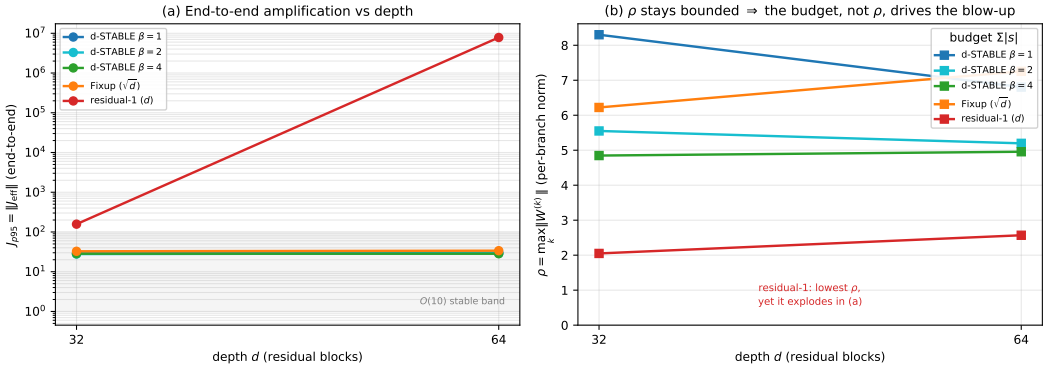


Figure 7: **Budget × branch-norm decomposition on `ogbn-arxiv` GCN** (no SN, 3 seeds). **(a)** Measured J_{p95} vs. depth: every d-STABLE budget and Fixup hold the $O(10)$ band while residual-1 ($\Sigma|s|{=}d$) explodes $157 \to 7.8{\times}10^6$. **(b)** Per-branch norm $\rho = \max_k \|W^{(k)}\|$ stays in $[2, 8]$ for *all* methods—residual-1 is the lowest—so nothing in ρ forecasts the blow-up. The only diverging quantity is the budget $\Sigma|s|$, which d-STABLE pins to $\beta \log d$.

Table 7: **Budget × branch-norm grid** (`ogbn-arxiv` GCN, 3 seeds). ρ is the largest spectral norm over residual-branch weights as used in the forward pass; the input encoder and classifier lie outside the product bound and are excluded. Bold: smallest ρ without SN. Residual-1 owns the smallest unnormalized ρ yet the largest J_{p95}; SN clamps residual-1's J_{p95} to 7.0 at $d{=}64$ but accuracy still collapses (11.7%). $d{=}64$ d-STABLE+SN cells are deferred to Table 8.

d	method (SN)	$\Sigma\|s\|$	ρ	J_{p95}	acc%
32	d-STABLE $\beta{=}1$	3.47	8.30	28.9	71.5
32	d-STABLE $\beta{=}2$	6.93	5.55	27.8	71.6
32	d-STABLE $\beta{=}4$	13.9	4.85	28.4	71.5
32	Fixup	5.66	6.22	32.7	70.0
32	residual-1	32.0	**2.05**	157	21.6
32	residual-1 + SN	32.0	1.01	29.3	63.6
64	d-STABLE $\beta{=}1$	4.16	6.80	30.1	71.5
64	d-STABLE $\beta{=}2$	8.32	5.20	28.9	71.6
64	d-STABLE $\beta{=}4$	16.6	4.96	28.4	71.1
64	Fixup	8.00	7.26	33.8	69.2
64	residual-1	64.0	**2.57**	$\mathbf{7.8{\times}10^6}$	16.3
64	residual-1 + SN	64.0	1.00	7.0	11.7

Residual-1 sits at the highest energy from layer $\approx$5 ($E_1{=}43$, saturating near 172), a high plateau coinciding with its accuracy collapse.

We do *not*, however, claim d-STABLE is "most controlled on energy." The 95% CIs of the mean γ separate d-STABLE's slope from Fixup's and ReZero's, but this yields no clean ranking: the pathological residual-1 has the *smallest* $|\gamma|$ of all (its curve is flat only because it saturates high), so slope alone cannot be the control criterion, and d-STABLE ends at the lowest final energy but does not start lowest. We therefore report the energy fit only as evidence that the collapse is not oversmoothing, and rely on the measured J_{p95} (Table 7) for the mechanism. The cosine feature-MAD (Table 10), which residual-1's magnitude blow-up does not confound, is monotone in the budget and corroborates this.

E.2 Controlled comparisons: matched budget, frozen allocation, positioning, second graph

All `ogbn-arxiv` GCN, SN-free (the headline configuration). Tables 13, 14, 11 and 12 report population std (over seeds); all other tables report sample std.

(i) Matched budget (Table 11): a fixed harmonic schedule ($s_k \propto 1/k$, renormalized to $\Sigma|s| = \log d$) and learned d-STABLE share an *identical* budget, yet the d-STABLE allocation wins by $+1.80$ pp at $d{=}32$ and $+1.71$ pp at $d{=}64$ (5 seeds). With the budget held equal, the entire gap is the allocation's.

Table 8: **A-priori certificate: d-STABLE + full-branch SN on GCN** (`ogbn-arxiv`, 3 seeds). A GCN branch is one conv, so SN gives a-priori $\rho \approx 1$ (realized ≤ 1.15) and $\|J_{\text{eff}}^{(d)}\| \leq d^{\rho\beta}$; we compare against the stricter d^{β}. Measured J_{p95} is below d^{β} in every cell and stays $O(10)$ as depth doubles; the β=1 rows are the tight, informative ones.

d	β	$\Sigma\lvert s\rvert$	J_{p95}	a-priori d^{β}	J_{p95}/d^{β}
32	1	3.47	26.4	32	0.82
32	2	6.93	27.5	1.0×10^{3}	0.027
32	4	13.9	27.8	1.0×10^{6}	2.7×10^{-5}
64	1	4.16	24.9	64	0.39
64	2	8.32	27.8	4.1×10^{3}	6.8×10^{-3}
64	4	16.6	29.7	1.7×10^{7}	1.8×10^{-6}

Table 9: **Energy-decay exponent** γ (`ogbn-arxiv` GCN d=64, 7 seeds, 0–6—a different seed set from Fig. 8; $\pm$ = sample std). $\gamma<0$ = energy grows (gradient expansion); $\gamma>0$ = decays (oversmoothing). Every residual method grows (so none oversmooth); only plain GCN decays. The CIs separate d-STABLE from Fixup/ReZero, but residual-1 has the smallest $|\gamma|$ (flat-at-high-plateau), so slope is not a clean control ranking—we use J_{p95} for the mechanism.

method	γ (7 seeds)	E_1	E_{last}
residual-1	-0.008 ± 0.0002	43	172
d-STABLE	-0.026 ± 0.007	6.4	83
Fixup	-0.039 ± 0.004	5.0	96
ReZero	-0.078 ± 0.004	1.4	144
plain GCN (non-residual)	$+0.008$ (decays)	—	—

(ii) Frozen allocation (Table 12): freezing the converged profile and retraining from scratch at the same budget gives 71.44 ± 0.19, statistically tied with the learned 71.59 ± 0.13 (5 seeds), for *all five* frozen profiles (71.44–71.80). The frozen profile beats the fixed $1/k$ harmonic (69.83) by $+1.6$, but not because it was learned. Recomputing each run's initial allocation from its seed, the trained profile lies within 0.8–1.7% of it with no sign flips (the schedule step is 5×10^{-5} over 500 full-batch steps), and the cross-seed $|s|$ cosine (0.645) equals that of independent random profiles (0.642 ± 0.053). Two controls on seeds 3–7 (sample std) locate the effect: training the allocation at $100\times$ the schedule step leaves accuracy unchanged (71.44 ± 0.27 vs. 71.48 ± 0.20), while a fixed uniform positive profile at the same budget falls to 69.57 ± 0.37 (seeds 3–6), the Fixup/ReZero plateau. The margin is therefore carried by the shape of the random initial allocation—most plausibly its mixed signs, as about half the branches enter with $s_k < 0$—rather than by its training; a uniform-magnitude, random-sign control would isolate the signs. Where the schedule trains at a larger effective step (LRGB, ImageNet), profiles move far from initialization, and all but ResNet-50-tanh agree across seeds ($|s|$ cosine 0.82–1.00).

(iii) Positioning vs. Stable ResNet (Table 12): the $1/\sqrt{d}$ per-branch rule ($\Sigma|s| = \sqrt{d}$, sub-exponential class) reaches only 69.47 ± 0.28 at d=64 (5 seeds), ≈2.1 points below learned d-STABLE on the same seeds but level with the fixed $1/k$ profile at $\Sigma|s| = \log d$ (69.83 ± 0.22). At this depth the $\sqrt{d}$-vs-$\log d$ class difference of Table 1 does not yet separate accuracy—it does in the extreme-depth chains of Sec. 4.1—so the gap is attributable to the allocation.

(iv) Seed variability (Table 13): at d=64 with up to 10 seeds (mean $\pm$ std), d-STABLE reaches 71.41 ± 0.19—corroborating the 71.4 ± 0.2 headline—and leads ReZero by $+1.6$ and Fixup by $+1.9$, a gap several times the per-method standard deviation; residual-1's ± 7.4 std is the collapse signature.

(v) Second homophilic graph (Table 14): on `pubmed` the pattern replicates—residual-1 falls $53.6 \rightarrow 40.9$ across d=32 $\rightarrow$ 64 while d-STABLE holds $77.8 \rightarrow 77.2$ and is best at both depths.

E.3 Added baselines under the matched recipe

We add the learned-gate and deep-init baselines requested in review (LayerScale, DeepNorm/DeepNet, GCNII, Gradient Gating), all under the single matched recipe of Sec. 4.2 (`ogbn-arxiv` GCN, d=64, hidden 256, dropout 0.3, lr $= 5\times10^{-3}$, 500 epochs, no SN unless "+LN"). Gradient Gating (G^2) (Rusch et al., 2023) is implemented from the authors' reference (weight-tied convs, intrinsic to

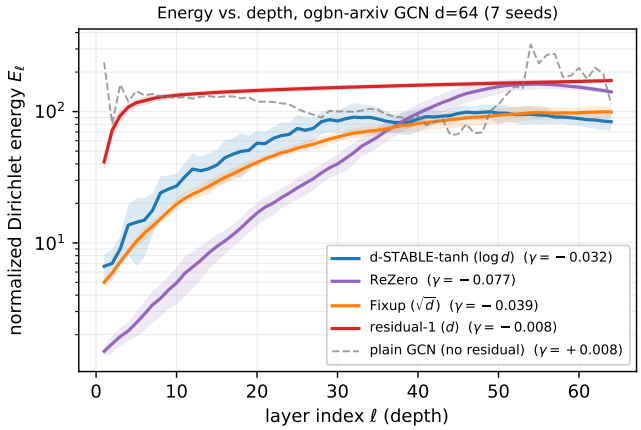


Figure 8: **Normalized Dirichlet energy vs. depth**, `ogbn-arxiv` GCN d=64 (7 seeds, 3–9; plain: 10). residual-1 (red) saturates at the highest energy from layer ≈5—a high plateau, not oversmoothing; only the non-residual plain GCN (dashed) decays. Every residual arm grows (γ<0), so none oversmooth; see Table 9 for the demotion.

Table 10: **Oversmoothing (cosine feature-MAD) vs. budget** (`ogbn-arxiv` GCN, no SN, 3 seeds). Final-layer feature-MAD (0 = directional collapse) is monotone in $\Sigma|s|$ and tracks accuracy—the same budget axis that controls J_{p95}. Reported instead of Dirichlet energy, which residual-1's magnitude blow-up confounds.

d	method	$\Sigma\|s\|$	feature-MAD	acc%
32	d-STABLE β=1	3.47	**0.77**	71.5
32	Fixup ($\sqrt{d}$)	5.66	0.46	70.0
32	residual-1 (d)	32.0	0.00	21.6
64	d-STABLE β=1	4.16	**0.84**	71.5
64	Fixup ($\sqrt{d}$)	8.00	0.44	69.2
64	residual-1 (d)	64.0	0.00	16.3

the method). For DeepNorm (Wang et al., 2022) we port the residual scaling $s_k = (2d)^{-1/4}$, reported without post-LayerNorm ("DeepNorm-style") and with it ("DeepNet"); this ports the scaling rule only and is not a claim about DeepNet in its native 1000-layer post-LN Transformer setting—hence a non-faithful port on this depth-stress GNN. Accuracy broadly tracks the budget regimes (Table 15): the small-budget methods train, $\Sigma|s|$≈19 is unstable, and $\Sigma|s|$=64 collapses; neither LayerNorm nor SN substitutes for the budget.

E.4 Long-range depth stress-test: full β-sweep and the diagonal budget

The main-text Table 3 summarizes the scalar β-sweep on the controlled long-range stress-test (Peptides-func, minimal backbone—no edge features, mean-pooled readout; *not* competitive LRGB numbers, not comparable to GatedGCN/GPS). Table 16 gives the full grid, including the per-channel (diagonal) generalization. Upgrading the scalar gate to $\mathrm{diag}(s_k)$ requires an ∞-norm budget for the bound ($\|\mathrm{diag}(s_k)\|_2 = \max_c |s_{k,c}|$, so the quantity in the bound is $\Sigma_k \max_c |s_{k,c}| = \beta \log d$; budgeting the mean lets it float). We implement both—the ∞-norm variant, which pins that quantity, and a mean-normalized one, which does not. Neither is certified a priori here: this backbone applies no branch normalization, so ρ is measured, not bounded. Both are dominated by scalar-β4 at the optimal budget (best diagonal 0.495 vs. scalar 0.518 at d=64). Per-channel granularity is real but second-order to capacity: β is the lever, and the scalar budget is the right abstraction. The ∞-norm variant is kept as an appendix design-justification, not a headline.

E.5 Full heterophilic stability and β-sweep

Table 17 expands the main-text heterophilic table with $\Sigma|s|$ and the schedule-only proxy alongside the empirical Jacobian; the proxy is a Theorem 1 bound only for the d-STABLE rows (where SN

Table 11: **Matched-budget control** (5 seeds). Fixed $1/k$ vs. learned at identical $\Sigma|s| = \log d$.

d	schedule	test acc%
32	harmonic-fixed	69.67 ± 0.16
32	d-STABLE learned	$\mathbf{71.47 \pm 0.31}$
64	harmonic-fixed	69.83 ± 0.22
64	d-STABLE learned	$\mathbf{71.54 \pm 0.16}$

Table 12: **Frozen allocation and $1/\sqrt{d}$ positioning** (d=64, 5 seeds).

configuration	test acc%
d-STABLE learned	71.59 ± 0.13
d-STABLE frozen profile	71.44 ± 0.19
fixed $1/k$ harmonic	69.83 ± 0.22
$1/\sqrt{d}$ (Stable ResNet)	69.47 ± 0.28

Table 13: **Seed variability, `ogbn-arxiv` GCN** d=64 (mean $\pm$ std). d-STABLE leads Fixup/ReZero by a gap several times the per-method std; residual-1 collapses with high variance.

method	seeds	test acc%
d-STABLE-tanh	10	$\mathbf{71.41 \pm 0.19}$
ReZero	7	69.77 ± 0.14
Fixup	10	69.52 ± 0.30
residual-1	10	14.60 ± 7.39

Table 14: **Second homophilic graph** (`pubmed` GCN, 3 seeds).

d	method	test acc%
32	d-STABLE	$\mathbf{77.77 \pm 1.09}$
32	residual-1	53.60 ± 10.57
64	d-STABLE	$\mathbf{77.20 \pm 0.57}$
64	residual-1	40.90 ± 0.28

enforces $\rho_k \leq 1$). Table 18 is the full β-sweep behind the "$\beta \leq 0.3$ matches/exceeds ReZero on 4 of 6" headline.

E.6 Why GCN2 and GPS behave differently

The depth-collapse and budget-tightening story generalizes cleanly across backbones without strong internal stabilizers (GCN, GAT, GraphSAGE, TransformerConv). Two architectures behave differently and should be read as diagnostic. **GCN2** (Chen et al., 2020) carries a built-in schedule—an initial-residual $\alpha H^{(0)}$ and an identity-mapping mix $(1-\gamma_k)I$ with $\gamma_k = \log(\lambda/k+1) \to 0$ for layer $k \geq 1$ (written γ_k to avoid a clash with our budget scalar β)—so its effective per-layer amplification is already bounded below 1 on homophilic graphs; the d-STABLE budget is in that sense embedded. But its fixed α cannot down-weight destructive propagation per-layer, so it fails on heterophily, where d-STABLE-wrapped GCN2 recovers +30 points (Texas+Cornell, d=8). **GPS** (Rampášek et al., 2022) adds a content-adaptive global-attention branch (per-pair learned weights independent of the edge set), which a scalar per-layer budget cannot replicate; GPS is thus not a clean test of the scalar-budget mechanism. The appropriate framing: d-STABLE supplies a *stability floor*—an explicit d^{β} schedule bound for architectures whose native mechanism does not already provide comparable depth control.

F Broader Impacts

d-STABLE is foundational research on training stability. It does not propose a deployable system, a generative model, or a domain-specific application; it proposes a parameterization of the residual-scaling schedule that converts an exponential schedule bound into a polynomial one. We see the broader impacts as follows.

Positive societal impacts. Depth instability has been documented as a failure mode in domains where deeper models have been pursued for accuracy reasons: graph-based recommendation and fraud-detection systems (oversmoothing on deep GCN/SAGE stacks), medical imaging (deep ResNet/UNet variants where training brittleness inflates compute), and scientific surrogate models (deep operator networks for PDE/protein/materials prediction). A controlled-budget schedule that turns the worst-case schedule bound from $\exp(\Theta(d))$ into $d^{\rho\beta}$ at the same parameter count addresses a known practical failure mode in those settings, without requiring architecture-specific tuning. To the extent that this reduces the need for repeated retraining, it lowers the energy and compute footprint of training large residual networks — a small but measurable contribution to the efficiency dimension of responsible ML.

Negative societal impacts. The contribution is generic, so direct misuse is limited; the most plausible indirect harm is that more reliable training of deep architectures could accelerate the development of

Table 15: **Expanded d=64 baseline comparison** (`ogbn-arxiv` GCN, matched recipe). $\Sigma|s|$ is the trained residual mass; "—" marks non-residual methods (GCNII, plain, G^2). d-STABLE leads all learned-gate and deep-GNN baselines; DeepNorm-style spends $\Sigma|s|\approx 19$ and lands in the collapse regime, so it is a non-faithful port, not DeepNet.

method (d=64)	$\Sigma\|s\|$	test acc%
d-STABLE-tanh + LN	4.16	$\mathbf{72.30 \pm 0.12}$
d-STABLE-tanh	4.16	71.39 ± 0.22
ReZero	0.45	69.77 ± 0.15
Fixup	8.00	69.64 ± 0.26
LayerScale	0.18	68.72 ± 0.13
Gradient Gating (G^2)	—	68.78 ± 0.22
GCNII	—	68.19 ± 0.19
DeepNorm-style (non-faithful port)	19.03	34.12 ± 21.77
residual-1 + LN	64	29.24 ± 4.12
DeepNet (DeepNorm+LN)	19.03	28.13 ± 0.94
plain GCN	—	25.00 ± 10.40
residual-1	64	13.87 ± 7.84

Table 16: **Full β-sweep on the long-range stress-test** (Peptides-func, minimal backbone, 3 seeds except scalar β=8 at d=16, 32 (2) and diagonal-∞ β=8 at d=64 (1); test AP). Scalar vs. per-channel diagonal (∞-norm budget pinned; mean budget, $\Sigma \max_c |s|$ floating). No row uses branch normalization. Scalar-β4 is the best config; both diagonal variants are dominated by it. **Bold** = best overall. Same-backbone comparison only.

variant	budget	d=16	d=32	d=64
scalar, β=1	$\Sigma\|s\|=\log d$	0.427	0.434	0.445
scalar, β=2	$2\log d$	0.456	0.470	0.478
scalar, β=4	$4\log d$	0.481	0.495	**0.518**
scalar, β=8	$8\log d$	0.464	0.468	0.504
diagonal-∞ (pinned), β=4	$\Sigma \max_c \|s\|=4\log d$	0.461	0.465	0.469
diagonal-∞ (pinned), β=8	$8\log d$	0.477	0.482	0.482
diagonal-mean (floating), β=4	$\Sigma \mathrm{mean}_c\|s\|=4\log d$	0.478	0.483	0.495
ref: ReZero / LayerScale / G^2 (unbudgeted)				*.455 / .456 / .465*

downstream models whose deployment carries its own risks (generative misinformation, surveillance, decision systems with disparate impact). Consistent with the ICLR Code of Ethics for foundational research, we note that risk-gating for such applications should occur at the deployment layer (data curation, evaluation, deployment auditing) rather than at the schedule-stabilization layer; d-STABLE does not change the set of models that are *possible* to train, only the worst-case stability bound at a given depth. We do not release any pretrained model that would itself constitute a misuse risk: all checkpoints accompanying the supplemental code are standard backbones (ResNet, DeiT, GCN/SAGE/GAT/GCN2/TransformerConv) at standard scales on public datasets, with no novel data collection.

Mitigation. Beyond the deployment-layer point above, we adopt the standard mitigations for foundational ML work: anonymized supplemental code, public-only datasets, and explicit disclosure of compute scope (App. D.6, the GNN sweep description in §4.2, and the ImageNet recipe disclosure in §4.3) so that reviewers and downstream users can assess the cost–benefit profile of the technique.

Table 17: **Full heterophilic stability at** d=64. Schedule-only proxy $d^{\beta_{\text{eff}}}$ is a Theorem 1 bound only for the d-STABLE rows; for the others it is the schedule contribution. **Bold** J_{p95} marks the smallest per row among the four non-residual-1 methods.

	Residual-1	ReZero	Fixup	d-STABLE-tanh	d-STABLE-softplus
$\Sigma\|s_k\|$ *at* d=64					
Chameleon	64.0	0.39	8.00	4.16	4.16
Texas	64.0	0.04	8.00	4.16	4.16
Wisconsin	64.0	0.07	8.00	4.16	4.16
empirical J_{p95}					
Chameleon	5.5×10^{8}	39.1	55.6	**25.1**	41.1
Texas	6.0×10^{7}	**11.1**	44.5	12.1	11.3
Wisconsin	1.3×10^{6}	13.8	19.9	11.2	**9.6**

Table 18: β**-sweep for d-STABLE-tanh on heterophilic GCN at** d=64 (10 splits). **Bold** = best d-STABLE column per row; right column is ReZero. Gradient descent on β is essentially flat, so the fixed small-β columns are the principled choice; the bound $d^{\rho\beta}$ tightens with β.

Dataset	β=0.01	0.10	0.30	1.00	3.00	learn-β	ReZero
Chameleon	51.2	**51.9**	50.6	47.3	40.7	50.7	51.3
Squirrel	35.2	**36.5**	35.3	34.0	30.4	35.1	36.1
Actor	**35.6**	35.3	34.8	34.7	33.8	34.9	35.3
Texas	**78.9**	78.9	78.1	75.7	64.9	78.6	80.3
Cornell	73.5	74.3	**74.6**	69.7	54.1	73.8	75.7
Wisconsin	84.3	**85.5**	84.5	80.0	69.6	84.3	85.3

Table 19: **Heterophilic GCN at** d=8 **(peak heterophilic depth) and** d=64 **(depth-collapse stress test)**. Each non-plain main cell is test accuracy / J_{p95}; accuracy is the 10-split mean, J_{p95} averaged over 3 stability splits. The final column reports the best fixed small-β d-STABLE-tanh accuracy from the d=64 sweep $\beta \in \{0.01, 0.03, 0.1, 0.3\}$, as accuracy / $\Sigma|s|$. **Bold** marks the highest accuracy in each row.

Depth	Dataset	Plain acc	Res1 acc/J	ReZero acc/J	Fixup acc/J	d-STABLE-tanh acc/J	d-STABLE-softplus acc/J	small-β d-STABLE acc/$\Sigma\|s\|$
8	chameleon	31.0	42.3/67.3	**52.1**/39.1	47.3/50.3	48.1/54.7	48.2/49.7	–
	squirrel	26.9	30.9/65.5	**35.9**/24.3	33.8/23.6	34.5/13.3	35.3/28.4	–
	actor	25.2	34.4/18.8	**35.2**/25.4	34.9/16.1	34.9/25.8	34.8/22.7	–
	texas	58.4	63.5/29.7	77.6/8.8	74.9/12.0	76.2/12.7	**80.0**/8.6	–
	cornell	44.1	52.4/23.8	**75.1**/11.7	67.6/10.9	70.0/17.3	69.7/11.2	–
	wisconsin	53.7	70.6/24.3	**82.7**/12.3	82.2/9.7	79.0/14.4	81.8/8.3	–
64	chameleon	26.2	$22.3/5.5\times10^{8}$	51.3/39.1	40.0/55.6	47.3/25.1	47.9/41.1	**51.9**/0.42
	squirrel	32.1	$19.6/4.5\times10^{8}$	36.1/24.6	27.4/18.0	34.0/27.9	35.0/19.5	**36.5**/0.42
	actor	25.2	$25.2/8.2\times10^{3}$	35.3/25.5	34.8/23.4	34.7/26.7	34.9/21.0	**35.6**/0.04
	texas	57.6	$58.9/6.0\times10^{7}$	**80.3**/11.1	58.6/44.5	75.7/12.1	75.9/11.3	79.7/0.12
	cornell	42.7	$42.7/3.9\times10^{7}$	**75.7**/12.6	52.4/20.5	69.7/10.7	69.5/8.0	74.6/1.25
	wisconsin	48.2	$48.0/1.3\times10^{6}$	85.3/13.8	69.4/19.9	80.0/11.2	81.6/9.6	**85.5**/0.42

Table 20: **Signed (tanh) vs. non-negative (softplus) d-STABLE schedules**, same log-depth budget. Signed wins on the homophilic `ogbn-arxiv` by 0.4–1.7 points; softplus gives slightly smaller Jacobians on most heterophilic graphs at d=8. **Bold** marks the column-best per row (accuracy and J_{p95} separately).

Dataset / depth	tanh acc	softplus acc	Δ	tanh J_{p95}	softplus J_{p95}
Heterophilic (mean over 10 splits, d=8*)*					
chameleon	48.1	**48.2**	+0.2	54.7	**49.7**
squirrel	34.5	**35.3**	+0.8	**13.3**	28.4
actor	**34.9**	34.8	−0.0	25.8	**22.7**
texas	76.2	**80.0**	+3.8	12.7	**8.6**
cornell	**70.0**	69.7	−0.3	17.3	**11.2**
wisconsin	79.0	**81.8**	+2.7	14.4	**8.3**
ogbn-arxiv (canonical split, mean over 3 seeds)					
d=4	**69.5**	69.1	+0.4	**28.6**	28.9
d=8	**70.6**	69.5	+1.1	**28.2**	30.4
d=16	**71.3**	69.6	+1.7	**27.5**	30.4