EFFECTIVE DEPTH: A SCALAR CORRECTION CLOSES THE OVERSMOOTHING GAP IN TRAINED GNNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Asymptotic analyses of oversmoothing in graph neural networks assume linear dynamics, i.i.d. layer weights, and constant aggregation across depth. Trained networks violate all three, and these violations are widely cited as the reason the theory fails in practice. Across 324 training runs spanning three architectures, six depths (2–64), and homophilous and heterophilous benchmarks, we ask which violation matters. The i.i.d. assumption is violated severely – the top singular value of trained weights rises from $0.97\times$ to $2.15\times$ the Marchenko–Pastur bulk edge, pooled across architectures – yet trained networks sit no further from the linear i.i.d. prediction than randomly initialized ones (mean absolute slope error 0.071 against 0.188). What explains the gap is one scalar: the total branch gain $\sum_\ell \|h_\ell\|/\|x_\ell\|$ in the residual stream, the network's effective depth. Training collapses it from 49% of nominal depth at $L = 8$ to 14% at $L = 64$, following $L_{\text{eff}} \approx cL^k$ with $k = 0.39\pm0.04$. Supplying that scalar to an otherwise unchanged linear i.i.d. simulation cuts prediction error by 83% at depth 32 and 89% at depth 64. Its layerwise distribution carries no additional information: no profile of equal total predicts worse than the measured one – uniform, reversed, sorted, and three independent shuffles all fall within 0.80–$1.08\times$ ($p > 0.18$) – while halving the total more than doubles the error ($2.16\times$, $p < 10^{-5}$). Fitting the exponent on three depths predicts a held-out depth for the same dataset and architecture at about 1.1 times the oracle's error. Finally we intervene: regularizing total gain in 32-layer GCNs sets it precisely ($r = 0.992$) and moves decay as predicted ($r = -0.943$), with unregularized networks on the same curve. Where the constraint binds, decay varies several-fold at no measurable accuracy cost on Cora and Chameleon; Texas, the smallest graph, pays a real cost, and at extreme targets accuracy tracks distance from a network's own operating point at least as well as it tracks decay. Primary claims are established for GCN, the one architecture where propagation operator and energy metric coincide.

1 INTRODUCTION

Asymptotic oversmoothing theory, pioneered by Oono & Suzuki (2020), predicts exponential decay in the expressive power of graph neural networks as depth increases, assuming linear dynamics (A1), i.i.d. layer weights (A2) and constant aggregation (A3). Trained networks violate all three, and the standard inference is that the theory therefore fails to describe them. That reasoning treats the violations as interchangeable, as though any one alone would explain the gap, and is seldom examined.

Which violation actually carries the discrepancy for trained, finite-depth, residual networks? We compare against depth-matched simulation rather than asymptotic rates, the gap being dataset-specific and ranging from $1.8\times$ to $19.5\times$.

Our primary finding is a negative result: the most visible violation—the i.i.d. assumption (A2)—is not responsible for the gap. Trained GCN weights depart significantly from the Marchenko–Pastur bulk (singular value ratio 1.68 ± 0.48 for GCN, 2.15 pooled across architectures), yet the prediction error does not grow with the departure. The discrepancy is carried instead by the network using far less depth than its nominal architecture: we measure this effective depth as L_{eff}, the sum of per-

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.

layer branch-to-stream norm ratios, and show that what matters is its total, not its distribution across layers.

That residual networks use less depth than they have is not new: Veit et al. (2016) showed residual CNNs behave as ensembles of shorter paths, with depth set by those that actually propagate gradients. We apply this to GNNs, adding a GNN-specific measure of effective depth, the finding that only its total matters, and causal evidence via intervention.

Our primary claims concern GCN with fixed aggregation, and L_{eff} is specific to residual networks; the intervention is at depth 32, on three datasets, for GCN only. Learned aggregation is a separate problem, the baseline simulation being ill-posed for architectures departing from the fixed-operator setting, so we report GAT results without claiming the same mechanism governs its oversmoothing.

We make the following contributions. (1) We measure which of Oono and Suzuki's three assumptions fails to explain the theory-practice gap in trained, residual, finite-depth networks (Sec. 3), showing the i.i.d. violation is robust against the prediction. (2) We introduce effective depth for GNNs and show that depth usage collapses and follows a power law (Sec. 3): a depth-64 residual GCN achieves $L_{\text{eff}} = 8.69$, only 14% of its nominal depth. (3) We show that only the total of the per-layer gain matters, not its distribution (Sec. 4): no total-preserving surrogate profile degrades the prediction, which rules out distributed mechanisms. (4) We inject this scalar into the asymptotic theory, reducing error from 0.075 to 0.025 on GCN and tying the full measured profile overall; it reduces the prediction error by 83% at depth 32 and 89% at depth 64, where the correction beats the full profile from depth 16 upward (Sec. 4), and it extrapolates across depth, with three trained depths predicting a fourth within about $1.1\times$ of the oracle (Sec. 5). (5) We provide causal evidence via a regularisation intervention that directly controls effective depth and reproduces the decay-L_{eff} relationship ($r = -0.943$), confined to the clean band where the constraint binds ($T \leq 12$), settling the sign of the correction and establishing its necessary role (Sec. 6). (6) We clarify the limits of this mechanism: the prediction fails for learned aggregation, the intervention binds only up to $T = 12$, and accuracy costs are dataset-dependent, modest on two datasets while Texas, with 183 nodes and a 65% majority class, pays a measurable cost (Secs. 7 and 9).

These findings do not overturn Oono and Suzuki's theory: they show that once the right scalar is supplied its prediction is recoverable from trained networks, and identify which assumption is not responsible for the remaining gap.

2 Setup

2.1 The theory under audit

Let $G = (V, E)$ be undirected on $N = |V|$ nodes with features $X \in \mathbb{R}^{N\times d}$, and let $\tilde{A} = \tilde{D}^{-1/2}(A + I)\tilde{D}^{-1/2}$ be the augmented symmetrically normalised adjacency. A message-passing layer is

$$X_{\ell+1} = \sigma\Big(\tilde{A}X_\ell W_\ell\Big), \tag{1}$$

and a residual layer $X_{\ell+1} = X_\ell + h_\ell(X_\ell)$, with h_ℓ the branch. Asymptotic theory analyses Eq. (1) as $\ell \to \infty$ and obtains an exponential rate governed by the product of a weight-scale term and a spectral term of $\tilde{A}$ (Oono & Suzuki, 2020). That rate requires the three assumptions introduced in Sec. 1—linear dynamics (A1), i.i.d. layer weights (A2) and constant aggregation (A3)—stated in full in Appendix A. A trained network violates all three, and the standard inference, that the theory therefore fails to describe trained networks, treats the violations as interchangeable. They are not. We measure which one carries the discrepancy.

2.2 Measuring decay

We use the Dirichlet energy induced by the augmented symmetrically normalised Laplacian $\tilde{\Delta} = I - \tilde{A}$, and report the *normalised* quantity

$$E(X) = \operatorname{tr}\Big(X^\top \tilde{\Delta} X\Big), \qquad \hat{E}(X) = \frac{E(X)}{\|X\|_F^2}. \tag{2}$$

Its expansion as a sum over edges, with degrees taken in the self-loop-free graph, is given in Appendix A. The normalisation is not cosmetic: raw energy decays to zero through scale collapse

alone, so a network that uniformly shrinks its features registers as oversmoothing under E while leaving node distinguishability untouched. We log both and report $\hat{E}$. Following Rusch et al. (2023) we take oversmoothing to be the layerwise *exponential* convergence of such a measure. We also log the mean average distance (MAD) over edges but do not report it: that work rules it out as an oversmoothing measure, and we retain it only as an internal consistency check.

For each run we fit $\log \hat{E}(X_\ell)$ linearly in ℓ over $\ell \geq 1$, discarding the input layer and any non-finite or vanishing observation, and report the ordinary least-squares slope with its R^2. The identical routine is applied to trained networks and to every simulation in Sec. 2.5, so protocol comparisons never differ in how the slope is extracted. The exponential form is a genuine fit rather than an early transient, with high R^2 across architecture and depth and few runs fitting poorly (Appendix D).

We compare every trained network against a *depth-matched* simulation, never against the asymptotic limit: the discrepancy between the two is dataset-specific rather than a universal constant, and tracks the spectral gap (Appendix A).

2.3 Operator matching, and why GCN is primary

The theory propagates with $\tilde{A}$, but measuring a trained network requires choosing an energy functional, and for learned aggregation the two do not agree. We measure GCN with Eq. (2), whose operator is exactly the one the layer propagates with, and GAT and Graph Transformer with the unnormalised $E(X) = \frac{1}{2}\sum_{(i,j)\in E} \|x_i - x_j\|^2$, since their propagation operator is not a fixed normalised adjacency. Neither convention is well posed for learned aggregation: matching the metric leaves the simulation propagating with an operator the network does not use, while keeping the sym-normalised metric makes the simulation internally coherent but compares it against a measurement in a different functional, and under the matched metric a substantial minority of GAT and Graph Transformer baseline simulations predict energy *increase*. We therefore treat **GCN as the primary architecture** for all headline claims and report GAT and Graph Transformer as a secondary analysis under both conventions. GAT's small baseline error is an artefact of the mismatch, not evidence that the theory succeeds there. The consistency check that fixes this, and the per-architecture figures, are in Appendices A and D.

2.4 Residual decomposition and effective depth

For a residual layer $X_{\ell+1} = X_\ell + h_\ell$, energy is not additive:

$$E(x + h) = E(x) + E(h) + 2B(x, h), \tag{3}$$

with B the bilinear form induced by $\tilde{\Delta}$. A systematically negative B would mean the branch actively opposes smoothing. It does not: across 1512 layer-observations (residual, non-collapsed, $\ell \geq 4$) the cross-term is non-negative in 94.9% of cases, with mean fractional contribution $+0.083$. One cell resists this and we name it rather than average it away: trained Graph Transformers are negative in 20.6% of layers, against 0.4% at initialisation, so the aggregate conclusion—no active anti-smoothing mechanism—holds, but not uniformly across architectures (Appendix D).

Di Giovanni et al. (2023) show that graph convolutions can increase Dirichlet energy, depending on the spectrum of the channel-mixing weights, so sharpening is possible in principle. Our measurement is narrower and empirical: in these trained residual networks the residual branch does not systematically oppose smoothing. We claim no impossibility result.

With no cross-term mechanism to appeal to, the residual stream can influence decay only through the magnitude of its branch contributions. We define the **total branch gain**, or effective depth,

$$L_{\text{eff}} = \sum_{\ell} \frac{\|h_\ell\|}{\|x_\ell\|}, \tag{4}$$

a single scalar per network, and a property of the configuration rather than of the seed: within-configuration spread is 0.437 against 2.319 between configurations (Appendix D).

2.5 Simulation protocols

All comparisons are against a linear i.i.d. simulation of the dynamics the theory assumes: no activation, fresh weights W_ℓ with i.i.d. Gaussian entries of fixed scale σ at every layer, and the same

propagation operator,

$$X_{\ell+1} = X_\ell + \tilde{A} X_\ell W_\ell. \tag{5}$$

Crucially σ is a fixed constant, not fitted to any trained or initialised weight scale, so the simulation receives *no* information from the network beyond what a protocol injects explicitly. The remaining construction details are in Appendix A.

Protocols differ only in what gain information the simulation receives. P1, the theory baseline, simulates at nominal depth L with gains free, taking nothing from the trained network. P2 simulates at $\mathrm{round}(L_{\mathrm{eff}})$ layers. P3 forces the per-layer gain to the measured $\|h_\ell\|/\|x_\ell\|$ at every layer. P4 forces a uniform gain whose total matches the measured L_{eff}: the same total as P3 with no layerwise structure. P5 is P4 with L_{eff} predicted from a power-law fit rather than measured. Gain is forced multiplicatively on the branch, so $\|h_\ell\|/\|x_\ell\|$ attains its target exactly. P2 and P4 supply the same scalar and differ only in how it enters—the slope is fitted over the layer range actually simulated, so truncation changes the fitting window while uniform injection does not—which isolates injection from magnitude; P3 against P4 holds the total fixed and removes only the layerwise distribution, isolating distribution from total (Sec. 4). Error is mean absolute slope error against the corresponding trained network.

2.6 Experimental setup

We train GCN, GAT and Graph Transformer at hidden width 64, in residual and non-residual variants, at depths $2, 4, 8, 16, 32, 64$, on Cora (homophilous) and Chameleon and Texas (heterophilous), with three seeds per configuration: 324 runs, the depth-generalisation study (108 runs) and the intervention (72 runs) being trained separately (Appendix A). Three seeds are not three independent observations, so we report configuration-level statistics alongside run-level ones throughout; depths 2 and 4 yield too few points for a stable slope and enter only the spectral analysis of Sec. 3. A run is recorded as collapsed when it predicts at most two classes or barely exceeds the majority-class rate (Appendix A.7), and collapsed runs are excluded, their exclusion reported wherever it changes a population: collapse is overwhelmingly a property of non-residual architectures, and residual GCN never collapses at any depth. We use the canonical public split for each benchmark and never construct our own. Splits, the verification harness run before analysis and the collapse counts are described in Appendix A.

3 What the discrepancy is not, and what it is

Of the three assumptions, (A2) is the one training most visibly breaks and the natural first suspect. The asymptotic rate is derived for weights whose spectrum is a Marchenko–Pastur bulk (Marchenko & Pastur, 1967), and trained networks develop singular values well outside it (Martin & Mahoney, 2021). If that violation separates trained networks from the theory, it should be large and breaking it should make the prediction worse. We find the first and not the second.

The violation is large and universal. For every layer we take the ratio of the top singular value of W_ℓ to the Marchenko–Pastur bulk edge for a matrix of its shape and entry variance, and average over layers; a ratio of 1 means the spectrum sits inside the bulk, as (A2) requires. This depends only on the weights, so it uses all 214 non-collapsed runs, including depths 2 and 4 and the non-residual variants. For GCN ($n = 91$) the ratio is 0.965 ± 0.011 at initialisation, with no run above 1; after training it is 1.68 ± 0.48, with 96.7% of runs above the edge, and the shift is present at every depth. Pooled over architectures it moves from 0.97 to 2.15, and is larger still for the attention architectures (Appendix D). By this measure (A2) is not approximately satisfied by trained networks; it is violated in essentially all of them.

The violation does not degrade the prediction. Each trained network gives a paired control: the same network at initialisation, where (A2) holds by construction. We score both against the same depth-matched residual simulation (P1). Following Sec. 2.6, the comparison uses residual, non-collapsed networks at depths 8–64: for GCN, 36 runs in 12 configurations, every one with a trained spectral ratio above the bulk edge (range 1.04–2.20). GCN also removes the operator ambiguity of Sec. 2.3, its P1 prediction being identical under both conventions. Breaking (A2) does not move trained GCNs away from the theory: their mean absolute slope error is 0.071 ± 0.034

against 0.188 ± 0.136 for the same networks with i.i.d. weights, with 26 of 36 runs closer after training ($p < 10^{-4}$ on runs, 0.027 on configuration means). Pooled over architectures the pattern is the same (0.163 at initialisation, 0.054 trained; 97 runs in 33 configurations). Per-depth rows and the per-architecture breakdown are in Appendix B.

We draw only the one-sided conclusion. It is tempting to read this as training bringing networks *closer* to the theory, but the initialisation error depends on how the initialisation scale compares with the simulation's fixed σ, which has nothing to do with (A2). The direction of the gap is therefore not evidence about (A2); its absence in the other direction is. The same caution explains the one depth that reverses: at depth 64 trained networks are slightly further from the theory than at initialisation (0.049 against 0.036, $p = 0.039$), which is not a growing trained error, the smallest of any depth, but a shrinking initialisation error. Nor does the error follow the violation in dose: within a depth, trained GCNs whose weights depart further from the bulk are no further from the theory, and the within-configuration association, though positive, is an order of magnitude too small to account for the error that remains (Appendix D).

What remains, and what accounts for it. That error is not small. On average it is 75% of the predicted slope's magnitude, with a consistent sign: at depths 16–64 every trained GCN (27 of 27) decays more slowly than predicted. For GCN, (A3) holds by construction, since the layer propagates with the fixed operator $\tilde{A}$, so neither the violated i.i.d. assumption nor adaptive aggregation accounts for the gap. The simplest remaining account is that these networks do not use their nominal depth. We measure this with L_{eff} of Eq. (4) on the 94 non-collapsed networks of the depth study. A layer whose branch is negligible relative to its stream contributes almost nothing to L_{eff} and almost nothing to the propagation; a network in which every branch matched its stream would have $L_{\text{eff}} = L$. Here L_{eff} is a descriptive quantity, a scalar that organises the sweep; whether it is the *right* scalar, carrying the information the theory is missing, is the question of Sec. 4.

Training uses a shrinking fraction of the layers. A residual GCN of depth 64 has an effective depth of 8.69, and the fraction of nominal depth in use falls steadily, from 0.49 at depth 8 to 0.32, 0.21 and 0.14 at depths 16, 32 and 64. Graph Transformers follow the same pattern at slightly higher L_{eff} (4.76 at depth 8 to 9.55 at depth 64) and GAT at lower (2.88 to 4.66 between depths 8 and 32). No architecture comes near $L_{\text{eff}} = L$ at any depth (Fig. 2, Appendix B): doubling depth adds far less than double the effective depth.

The growth is well described by a power law. Fitting $L_{\text{eff}} = cL^k$ per dataset, GCN gives $k = 0.39 \pm 0.04$, each fit with $R^2 \geq 0.995$ over four depths; across all nine (dataset, architecture) configurations k ranges from 0.16 to 0.53 (mean 0.35 ± 0.12) and 8 of 9 fits have $R^2 \geq 0.96$. Two qualifications apply. GAT and the Chameleon Graph Transformer have only three depths after collapse, and the one poor fit is among them; the architecture means differ, but with three configurations per architecture that difference is not established. And over a factor of eight in depth a power law is not the only candidate: a logarithm fits worse on Cora and Chameleon but marginally better on Texas. We use the power law as a compact description of the sweep, not as a law of training. The per-dataset exponents, fit comparisons and the test on k are in Appendix D.

Effective depth is a property of the configuration. Seeds agree closely: the average seed-to-seed standard deviation of L_{eff} within a configuration is 0.44, against 2.32 between configurations. This is what makes L_{eff} usable as an organising quantity rather than a per-run statistic, and what the prediction experiments of Sec. 5 rely on.

Mechanism. If L_{eff} is how the residual stream controls decay, then among networks with the same graph, operator and depth, those with larger L_{eff} should decay faster. We test this within each (dataset, architecture, depth) cell, removing cell means from both L_{eff} and the measured slope; centring within cells also removes the difference in energy operator between architectures (Sec. 2.3), which is what allows the cells to be pooled. The pooled within-cell correlation has the predicted sign and is modest: $r = -0.26$, $p = 0.041$ (94 runs in 32 cells), about 7% of the within-cell variance. Each architecture alone has the same sign and none is individually significant; for GCN, $r = -0.19$ ($p = 0.36$). The naive correlation across all runs is positive, but it compares architectures whose slopes are measured with different energy operators, and we do not interpret it (Appendix D). With the non-negative cross-term of Sec. 2.4, which rules out the branch actively cancelling smoothing,

this is consistent with L_{eff} acting through branch magnitude. It does not establish it; the direct test is interventional (Sec. 6).

4 IT IS THE TOTAL, NOT THE DISTRIBUTION

Section 3 summarised each residual stream by one number. A trained network does not spread its gain evenly: branches are larger in some layers than others, and the shape varies with depth and architecture. A natural refinement is that the shape, not just the sum, is what the theory is missing, that *which* layers propagate should matter and not only how many. This section tests that directly, and it fails. We randomise the profile and nothing changes; we change its total and the prediction breaks.

The test. For each trained network we take its measured per-layer gains, build surrogates, simulate each under Sec. 2.5, and compare the slope error with that of the measured profile (P3). *Reorderings* keep the exact multiset of gains and change only their order: reversed, sorted ascending, sorted descending, and three independent random shuffles. *Reshapings* keep the total and change the shape: a flat profile at the measured total (P4), and a step placing the whole total in the first layers. *Controls* keep the shape and change the total, halving or doubling it, and are there to show the test has power: if the simulation were insensitive to the gains altogether, every surrogate would pass trivially.

The profile carries no information the total does not. For GCN, no surrogate preserving the total is worse than the measured profile: the error ratios run from 0.80 to 1.08, none significantly above 1 (paired Wilcoxon, all $p > 0.18$), and several are significantly *below* it: sorting ascending (0.84) and each of the three shuffles (0.87, 0.85, 0.80; $p \leq 0.004$). The shuffles are independent and all three land on the same side of the measured profile, so the result does not rest on one lucky permutation. Replacing the profile by a flat one at the same total gives 0.93. Pooled over architectures the reorderings fall between 1.02 and 1.09 and the shuffles between 1.02 and 1.04. Of the eight total-preserving surrogates, one differs at $p < 0.05$ under Wilcoxon: the front step, 6% worse ($p = 0.019$), which survives neither a correction for eight comparisons (Bonferroni threshold 0.006) nor the configuration-level test ($p = 0.06$). The reversed profile is 9% worse, significant under a paired t-test ($p = 0.006$) but not under the Wilcoxon test we use throughout ($p = 0.14$); we report both rather than choose. The controls, by contrast, move the error substantially: halving the total more than doubles the GCN error ($2.16\times$, $p < 10^{-5}$; pooled $1.69\times$). A simulation that detects a change in the total this clearly could also detect a change in the profile. It does not, because the profile does not matter. Figure 3, in Appendix D, plots every surrogate with bootstrap intervals.

Not a dilution effect. The invariance is not an artefact of pooling strongly shaped profiles with flat ones: the penalty for flattening a profile is uncorrelated with how structured it is, and the most structured profiles in the study, those of Graph Transformers at depth 32, are penalised no more than the flatter ones. For GCN the relationship runs the other way, the more a network concentrates gain in its early layers the more flattening *helps*, so where the profile is most distinctive it is at best irrelevant to the prediction (Appendix D).

One scalar is the correction. If the distribution carries nothing, the total should carry everything the profile does. It does: for GCN one scalar does as well as the full profile overall (0.025 against 0.027; configuration-level $p = 0.27$), reducing the theory's error by 44% at depth 8, rising to 73%, 83% and 89% at depths 16, 32 and 64, overall from 0.075 to 0.025, with every one of the 12 configurations improved (configuration-level Wilcoxon $p = 0.0005$). It beats the full profile at every depth from 16 upward ($p \leq 0.020$); depth 8 is the exception, where the profile keeps a small, significant edge (0.061 against 0.067, $p = 0.027$). Pooled over architectures the pattern holds but is weaker: P4 reduces error from 0.057 to 0.037, by 15%, 20%, 64% and 87% across the four depths, the gains at depths 8 and 16 not significant (configuration-level $p = 0.50$ and 0.65). Per-depth values are in Appendix B.

The scalar must be injected, not substituted. P2 uses the same scalar as P4 but applies it differently, simulating a network of $\mathrm{round}(L_{\text{eff}})$ layers instead of L layers with reduced gain. It does no better than the uncorrected theory (GCN 0.073 against 0.075) and is significantly worse than P4

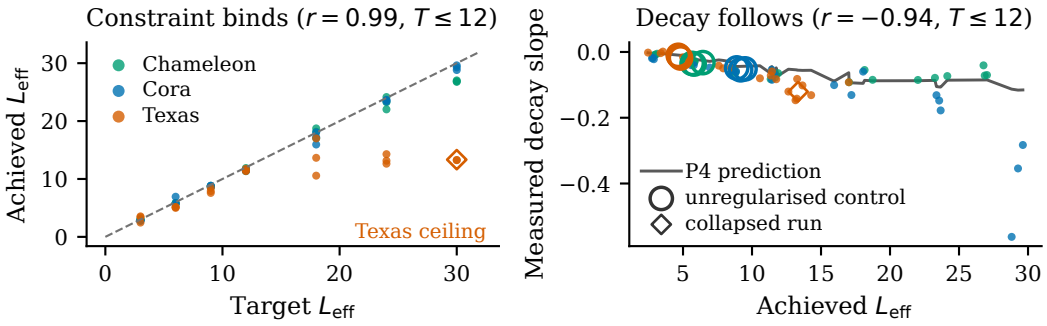

Figure 1: The intervention, plotting all targets $T = 3$–30. Left: achieved against target L_{eff}; the constraint binds for $T \leq 12$ ($r = 0.992$) and ceases to bind on Texas beyond it. Right: measured decay slope against achieved L_{eff}, with the P4 curve overlaid and unregularised controls circled on the same curve. The causal claim is confined to $T \leq 12$, where the constraint binds; Sec. 6 reads the wider range as its boundary.

(paired Wilcoxon $p < 10^{-7}$), for the reason given in Sec. 2.5: truncating the network also truncates the window over which the slope is fitted, so P2 answers a different question. Effective depth is the right quantity, but a trained network of depth L is not a shallower network; it is a network of depth L whose layers each do less.

What this does not show. The total is informative but not calibrated. Doubling it does not hurt the GCN prediction as halving does ($1.20\times$, $p = 0.31$; pooled $0.83\times$), because the bias is one-sided: at the measured total P4 predicts slower decay than observed in 89% of GCN runs, including every run at depths 32 and 64, while at double the total it predicts faster decay in 94%. The correct total lies between the measured one and its double, so the measured total is the right organising quantity but not an exact plug-in; Sec. 6 finds the same bias when the total is imposed rather than measured. Finally, every comparison here is between simulations of a trained network: it shows that the profile adds nothing to the linear simulation's prediction, not that the profile is irrelevant to what the trained network computes.

5 Prediction across depth

The power law of Sec. 3 was fitted to every depth of a configuration at once, so we test it on depths it was not fitted on. From three trained depths it predicts a fourth, including a depth deeper than any trained, at close to the error of an oracle that is given the true L_{eff}. From a single trained depth, with the exponent borrowed from other configurations, it still removes 78% of the theory's error for GCN. It fails for GAT, whose exponent does not transfer between datasets: across all architectures 25% of single-anchor predictions are worse than using no correction, concentrated in two GAT configurations, so the prediction result is a GCN and Graph Transformer result and we state it as such. The full analysis, both protocols and both tables are in Appendix E.

6 Intervention: control and its limits

Sections 3 and 4 are observational: L_{eff} is measured, not set. A network that trains to a large L_{eff} might simply be a network for which decay was always going to be slow, for reasons unrelated to L_{eff} itself. To move from association to control we add a penalty during training that pushes L_{eff} toward a chosen target T, $\lambda = 0.03$ applied to GCN at depth 32 across all three datasets and three seeds. If L_{eff} is what governs decay, setting it should set the decay rate; if the earlier sections found only a correlate, forcing L_{eff} should do nothing.

The constraint binds, in a clean band. For $T \in \{3, 6, 9, 12\}$, achieved L_{eff} tracks target almost exactly ($r = 0.992$, $p < 10^{-30}$; mean absolute deviation 0.44), with zero collapsed runs. This is the range Fig. 1 shows.

Decay follows, within each dataset. Across this band, achieved L_{eff} and measured decay slope correlate at $r = -0.943$ overall, and within each dataset separately: -0.939 on Cora, -0.974 on Chameleon, -0.979 on Texas (all $p < 10^{-4}$). P4, evaluated at the achieved L_{eff} rather than the nominal depth, predicts these intervened networks much better than it predicts the observational GCN sweep of Sec. 4 at depth 32: mean absolute error 0.013 against 0.027. Intervened networks are, if anything, easier to predict than the networks L_{eff} was originally measured on.

The tautology objection, and its answer. The obvious worry is that this is circular: the penalty is defined in terms of L_{eff}, so of course L_{eff} predicts decay in penalised networks. The unregularised controls answer this. A control network that happens to reach a given L_{eff} on its own, with no penalty applied, lies on the same curve as a penalised network pushed to that same L_{eff}. Cora's control reaches $L_{\text{eff}} = 9.19$ and decays at -0.0514; the nearest penalised condition, $T = 9$, reaches 8.74 and decays at -0.0577. Chameleon's control reaches 5.97 at -0.0346; $T = 6$ reaches 5.81 at -0.0308. The penalty does not create a relationship between L_{eff} and decay; it moves a network along a relationship that unpenalised networks already sit on.

Limits: the constraint stops binding, and P4 decalibrates. Beyond $T = 12$ the constraint stops binding on Texas, whose achieved L_{eff} reaches a ceiling near 13.75 and does not move however high the target is set; with 183 nodes we read this as a ceiling on how much total branch gain a network this small can carry, though we have not identified the mechanism. P4 decalibrates over the same range and not monotonically: the ratio of measured to predicted decay is U-shaped in the target rather than a constant miscalibration, and on one dataset it reverses sign. The causal claim is therefore confined to the band $T \leq 12$; the per-target deviations, correlations, fit qualities and error growth, including one collapsed run at $T = 30$ and the two error figures that follow from including or excluding it, are in Appendix D.

Reading the intervention as a whole. Within the band where the constraint binds cleanly, L_{eff} is not merely correlated with decay but controls it: pushing it to a target value, on networks the penalty never showed a decay rate, produces the decay rate the earlier sections' curve predicts, and the unpenalised controls rule out circularity. Outside that band the picture is genuinely mixed rather than a clean extension of the same result. On a small graph the network may simply be unable to reach the requested total, and where it can be pushed to extremes, a single scalar total is not enough to calibrate the decay rate precisely, in either direction. We take the causal claim from the clean band and report the wider range as its boundary, not as an extension of it.

7 Accuracy cost of controlling oversmoothing

Decay can be set almost at will, within limits, by controlling one scalar. Does suppressing it cost accuracy? We use the intervention runs of Sec. 6, restricted to the clean band $T \leq 12$ where the constraint binds and P4 holds. On Cora the decay rate varies by $3.5\times$ across the band while test accuracy moves by 0.018, with no significant relationship; on Chameleon decay varies by $9.7\times$ against an accuracy spread of 0.011, and the relationship, if anything, runs the wrong way for an oversmoothing account. At depth 32, on these two datasets, an order-of-magnitude change in decay is not accompanied by a corresponding change in accuracy. Texas is the exception: decay varies by $49.1\times$ and accuracy by 0.126, significantly correlated with L_{eff}. Texas has 183 nodes and a majority class covering 64.9% of test labels, and both the larger decay range and the accuracy cost are consistent with a small graph having less room to absorb an aggressive intervention, matching the ceiling on achievable L_{eff} in Sec. 6. We report it separately rather than averaging it in, since a pooled number would either be driven entirely by Texas or understate what happens there. Correlations and p-values are in Appendix B.

Over the full range of targets the relationship is not monotone, and at extreme targets distance from a network's own unregularised operating point predicts accuracy at least as well as L_{eff} does. We do not read this as ruling out oversmoothing; a network's natural operating point and a well-calibrated decay rate could easily coincide. It does mean the accuracy cost at extreme targets is not uniquely attributable to oversmoothing, and a paper about oversmoothing should say so. The quadratic fits, vertex locations and per-dataset correlations are in Appendix D.

8 Related Work

The asymptotic theory we audit is that of Oono & Suzuki (2020), which predicts exponential decay of expressive power with depth; we do not dispute it, but ask which of its assumptions carries the gap to trained networks. We follow Rusch et al. (2023) in taking oversmoothing to be the layerwise exponential convergence of a similarity measure, and measure it with a normalised Dirichlet energy, in the energy-based framework of Di Giovanni et al. (2023). Martin & Mahoney (2021) establish the departure of trained weight spectra from the Marchenko–Pastur bulk whose explanatory role Sec. 3 tests and rejects. The observation that residual networks use less depth than they have is not new: Veit et al. (2016) show that residual CNNs behave as ensembles of relatively shallow paths, with depth set by the paths that actually propagate gradients. Our L_{eff} is a different quantity from that path-length analysis: it is a scalar total of per-layer branch-to-stream norm ratios in the residual stream, and we show that only the total, not its layerwise distribution, matters for oversmoothing prediction. A fuller discussion, including the mitigation methods we do not study, is in Appendix C.

9 Limitations

Trained-dynamics claims concern residual networks; non-residual ones appear only in Sec. 3's spectrum analysis and the collapse counts. Primary claims rest on GCN and three datasets, with one intervention at depth 32; GCNII, GraphCON and similar oversmoothing-resistant architectures are unstudied. Decay statistics use non-collapsed networks, and collapse is not random: overwhelmingly non-residual, it leaves GAT without depth-64 data and the Graph Transformer's depth-64 results a survivor mean, so the power-law fits and held-out-depth test exclude configurations where training failed; residual GCN never collapses. Three seeds make the depth study 32 configurations, not 94 observations; some results hold only at run level, two depend on pooling (Secs. 3, 4), disagreeing tests are both reported, and the power law is description, not law, a logarithm fitting as well on one dataset. Every prediction comes from a linear simulation at fixed weight scale, so the initialisation comparison licenses only the one-sided conclusion that training does not move networks *away* from the theory; the ablation bears on the *simulation's* prediction, not on what the network computes. L_{eff} is no exact plug-in: P4 under-predicts decay in most GCN runs, its bias U-shaped under intervention, so our P4 accuracy claims are relative, not absolute. For GAT and the Transformer the simulation propagates with $\tilde{A}$ while the slope uses unnormalised energy: neither is coherent, a minority of their baselines predict energy *increase*, GAT's small baseline error is artefact not evidence, and its single-anchor failure is real, a quarter of predictions worse than no correction. The penalty weight came from one calibration run on Cora at $T = 6$, where binding was not monotone in λ and selection used test rather than validation accuracy: a working choice leaving the Cora accuracy at $T = 6$ not strictly held out, though claims compare targets at fixed λ. Chameleon's standard split leaks labels through duplicated nodes (Platonov et al., 2023), bearing most directly on Sec. 7, with its effect on decay unchecked; Texas, at 183 nodes with a 64.9% majority class, is where the intervention stops binding and the only dataset with an accuracy cost, so a small-graph effect cannot be separated from a Texas-specific one (Appendices A, D).

10 Conclusion

The assumption training most visibly breaks is not the one that matters: trained GCNs sit no further from the linear i.i.d. prediction than the same networks at initialisation, and what carries the gap is depth a network declines to use rather than how it distributes the depth it does use. One scalar, the total branch gain, absorbs most of the theory's error; its layerwise distribution adds nothing; and regularising it moves decay as predicted, with unpenalised networks sitting on the same curve. None of this makes L_{eff} the mechanism: the observational link is significant only pooled across architectures, the causal weight rests on the band where the constraint binds cleanly, and the correction is informative rather than calibrated. Non-residual dynamics, oversmoothing-resistant architectures, a coherent operator for learned aggregation, and L_{eff}'s own prediction bias remain open.

REFERENCES

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020a.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020b.

Francesco Di Giovanni, James Rowbottom, Benjamin P. Chamberlain, Thomas Markovich, and Michael M. Bronstein. Understanding convolution on graphs via energies. *Transactions on Machine Learning Research*, 2023.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

Vladimir A. Marchenko and Leonid A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 72(4):507–536, 1967.

Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

Yimeng Min, Frederik Wenkel, and Guy Wolf. Scattering GCN: Overcoming oversmoothness in graph convolutional networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations (ICLR)*, 2020.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations (ICLR)*, 2023.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations (ICLR)*, 2020.

T. Konstantin Rusch, Benjamin P. Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael M. Bronstein. Graph-coupled oscillator networks. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *International Conference on Learning Representations (ICLR)*, 2017.

Andreas Veit, Michael Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Xinyi Wu, Amir Ajorlou, Zihui Wu, and Ali Jadbabaie. Demystifying oversmoothing in attention-based graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in GNNs. In *International Conference on Learning Representations (ICLR)*, 2020.

A REPRODUCIBILITY

A.1 ASYMPTOTIC VERSUS DEPTH-MATCHED COMPARISON

Comparing a finite-depth network against an asymptotic rate is uninformative, and the size of the discrepancy is dataset-specific rather than a universal constant. At $L = 32$ the depth-matched simulated rate exceeds the asymptotic one by $19.5\times$ on Cora, $8.4\times$ on Chameleon and $1.8\times$ on Texas. The gap tracks the spectral gap: Cora's $\lambda^* = 0.9964$ drives its asymptotic rate near zero, and its 78 connected components make eigenvalue-1 deflation load-bearing there. We therefore report the range $1.8\times$–$19.5\times$ rather than a single figure; per-dataset rates are tabulated in Appendix D.

A.2 SPLITS AND VERIFICATION

We use the canonical public split for each benchmark and never construct our own. For Texas and Chameleon we take the first of the ten bundled splits, so reported variability reflects seed variation, not split variation. We retain the standard GEOM-GCN split for Chameleon, despite its known duplicated nodes (Platonov et al., 2023), for comparability with prior work. All reported quantities are produced under a verification harness run before analysis, covering split disjointness, label-shuffle controls scored against the majority-class rate rather than uniform chance, and seed-variation checks.

A.3 SIMULATION CONSTRUCTION

The simulated state starts from Gaussian noise rather than dataset features and is renormalised to unit Frobenius norm after each layer; because $\hat{E}$ is scale-invariant this changes neither the energy trajectory nor the fitted slope, and it prevents numerical overflow at large depth. Slopes are averaged over several independent simulation repetitions per configuration.

A.4 THE THREE ASSUMPTIONS IN FULL

The asymptotic rate of Oono & Suzuki (2020) requires: **(A1) linear dynamics**, nonlinearities removed, bounded, or absorbed into a contraction argument; **(A2) i.i.d. layer weights**, each W_ℓ drawn independently from a fixed distribution, so its spectrum is a bulk with no outliers; and **(A3) constant aggregation**, the propagation operator fixed across depth and not adapting to the data.

A.5 SETUP DETAILS MOVED FROM SECTION 2

Operator consistency check. For GCN the baseline error is identical under both metric conventions (0.0751), because propagation operator and metric coincide; we use this identity as an internal consistency check on the implementation. Under the matched metric 11.1% of GAT and 16.1% of Graph Transformer baseline simulations produce positive slopes, the theory predicting energy increase, where under the sym-normalised metric none do, for any architecture.

Training and exclusions. The depth-generalisation study re-trains the residual, depth-$\{8, 16, 32, 64\}$ cell of the sweep under the analysis harness, a fresh sample at existing design points rather than additional ones, so we report it separately rather than as a single total. The decay slope is fitted over layers $1, \ldots, L$ and requires at least three surviving points: depth 2 provides two and admits no rate by construction, and depth 4 provides four, enough to fit but too few for a stable comparison against simulation.

A.6 EDGE-SUM FORM OF THE ENERGY

Eq. (2) expands over edges as

$$E(X) = \frac{1}{2} \sum_{(i,j)\in E} \left\| \frac{x_i}{\sqrt{1+d_i}} - \frac{x_j}{\sqrt{1+d_j}} \right\|^2 ,$$

where d_i is the degree of node i in the self-loop-free graph. Self-loops are stripped from the edge index at load time, so the edge sum and the trace form agree exactly.

A.7 Collapse and survivorship

A run is recorded as collapsed when it predicts at most two distinct classes on the test set, or when its test accuracy is at most 0.02 above the majority-class rate. Collapse is overwhelmingly a property of non-residual architectures: 99 of 162 non-residual runs collapse, against 11 of 162 residual runs. It is also depth-dependent. At depths $\{2, 8, 32\}$, 44 of 162 runs collapse and *all* of them are non-residual; every residual collapse in the sweep occurs at depth 64 and is confined to learned aggregation—GAT on all three datasets, and Graph Transformer on two. Residual GCN never collapses at any depth; non-residual GCN collapses in 17 of 54 runs.

The trained-dynamics analysis is accordingly an analysis of residual networks, and within those, depth 64 under attention is a survivorship boundary rather than a clean measurement. The same pattern reappears in the depth-generalisation study, where 14 of 108 runs collapse (GAT 9, Graph Transformer 5, GCN 0). Collapsed runs are excluded from analysis and their exclusion is reported wherever it changes a population.

A.8 Reporting convention

Headline quantities are reported for GCN, for the reason given in Sec. 2.3. Where a quantity is instead pooled across architectures we say so explicitly at the point of use, and the per-architecture breakdown appears in Appendix B. We adopt this convention in advance of the results rather than selecting it afterwards; pooled figures are reported alongside the GCN figures throughout, and are in every case the more conservative of the two.

B Per-architecture breakdown

B.1 Accuracy across the clean band

Table 1: Accuracy across the clean band ($T = 3, 6, 9, 12$), GCN, depth 32. "Decay varied" is the ratio of largest to smallest per-target mean $|\mathrm{measured}|$ slope across the four targets; "accuracy spread" the range of per-target mean test accuracy; r, p the Pearson correlation between achieved L_{eff} and test accuracy, config means, $n = 4$ targets.

Dataset	Decay varied	Accuracy spread	r	p
Cora	$3.5\times$	0.018	-0.19	0.56
Chameleon	$9.7\times$	0.011	$+0.34$	0.28
Texas	$49.1\times$	0.126	-0.70	0.011

B.2 Effective depth against nominal depth

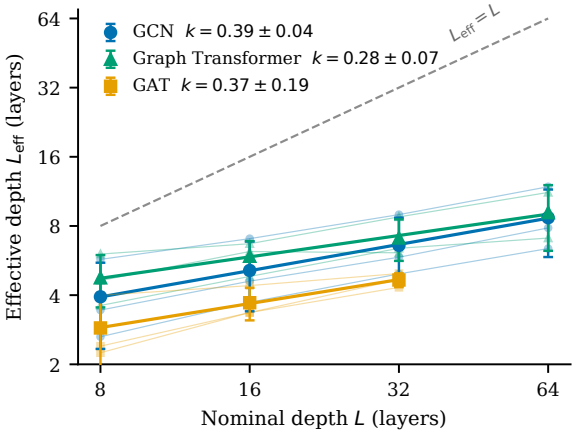

Figure 2: Effective depth grows sub-linearly with nominal depth. Large markers: mean over datasets of the seed-averaged L_{eff}; error bars, standard deviation across the three dataset means. Faint lines: individual datasets. Solid: power-law fit $L_{\text{eff}} = cL^k$ to the architecture means, k in the legend being the mean $\pm$ standard deviation of the per-dataset exponents. Dashed: $L_{\text{eff}} = L$. GAT collapses at depth 64 on every dataset and has no point there; the Graph Transformer point at depth 64 averages two datasets, because all three Chameleon runs collapse, and so is a survivor mean. $n = 94$ non-collapsed runs.

B.3 Prediction error under P1–P4

Table 2: Slope error of residual GCNs under four predictions: the theory at nominal depth (P1), a simulation truncated to $\text{round}(L_{\text{eff}})$ layers (P2), the full measured gain profile (P3), and a uniform gain at the measured total (P4). Mean over 36 runs; $\pm$ is the standard deviation across the three dataset means. Best in bold. P4 against P3, paired Wilcoxon on runs: $p = 0.027$ (P3 better), 0.020, 0.004, 0.004 at depths 8–64. Per-depth rows are in Table 6.

Depth	P1 nominal	P2 truncated	P3 full profile	P4 one scalar
All	0.075 ± 0.034	0.073 ± 0.054	0.027 ± 0.026	$\mathbf{0.025 \pm 0.029}$

B.4 Initialisation versus training

Table 3: Slope error of residual GCNs against the depth-matched simulation (P1), at initialisation, where (A2) holds, and after training, where it does not. Mean absolute differences in $\log \hat{E}$ slope; $\pm$ is the standard deviation across the three dataset-level means. "Closer" counts runs whose trained error is below their initialisation error. p: two-sided paired Wilcoxon signed-rank test on runs ($n = 36$) and on configuration means ($n = 12$). Per-depth rows are in Table 5.

Depth	Error at init	Error trained	Closer	p
All	0.188 ± 0.136	$\mathbf{0.071 \pm 0.034}$	26/36	$< 10^{-4}$ (runs), 0.027 (configs)

B.5 PREDICTION TABLES (P5 AND P6)

Table 4: Prediction of the decay slope without knowing L_{eff}. (a) Leave-one-depth-out: k fitted on the other three depths; P4 is the oracle at the true L_{eff}; rows by depth pool the five testable configurations (GCN on three datasets, Graph Transformer on two), and the last row is GCN alone; p is paired Wilcoxon, P1 against P5, on runs. (b) Single anchor: anchored at $L = 8$ with k borrowed from the other eight configurations, P1 and P6 computed on identical runs; p is paired Wilcoxon, P1 against P6, on runs, and the config-level test (23 configuration–depth cells, seeds averaged) gives $p = 0.015$ overall. Mean absolute slope error throughout; n counts runs.

	n	P1	P4 (oracle)	Predicted	p
(a) Leave one depth out, predicted = P5 from three depths					
Held-out 8	15	0.087	0.053	0.054	0.0001
Held-out 16	15	0.052	**0.016**	0.020	0.004
Held-out 32	15	0.058	0.010	**0.009**	0.0001
Held-out 64	14	0.042	**0.0055**	0.0064	0.0002
All	59	0.060	**0.021**	0.023	—
GCN only	36	0.075	**0.025**	0.027	—
(b) Single anchor, predicted = P6 from $L = 8$ and a borrowed k					
Target 16	27	0.040	—	**0.034**	0.46
Target 32	26	0.045	—	**0.017**	0.0002
Target 64	14	0.042	—	**0.0058**	0.0001
All	67	0.042	—	**0.021**	—
GCN	27	0.060	—	**0.013**	—
Graph Transformer	22	0.039	—	**0.015**	—
GAT	18	**0.019**	—	0.041	—

B.6 PER-DEPTH ROWS OF TABLES 3 AND 2

Table 5: Per-depth rows of Table 3: slope error of residual GCNs against the depth-matched simulation (P1), at initialisation and after training. $n = 9$ runs per depth.

Depth	Error at init	Error trained	Closer	p
8	0.356 ± 0.030	$\mathbf{0.106 \pm 0.052}$	9/9	0.004
16	0.261 ± 0.039	$\mathbf{0.055 \pm 0.024}$	9/9	0.004
32	0.100 ± 0.042	$\mathbf{0.074 \pm 0.006}$	6/9	0.13
64	$\mathbf{0.036 \pm 0.023}$	0.049 ± 0.007	2/9	0.039

Table 6: Per-depth rows of Table 2: slope error of residual GCNs under P1, P2, P3 and P4. $n = 9$ runs per depth.

Depth	P1 nominal	P2 truncated	P3 full profile	P4 one scalar
8	0.120 ± 0.042	0.129 ± 0.059	$\mathbf{0.061 \pm 0.032}$	0.067 ± 0.031
16	0.059 ± 0.016	0.030 ± 0.016	0.023 ± 0.013	$\mathbf{0.016 \pm 0.007}$
32	0.071 ± 0.008	0.051 ± 0.044	0.017 ± 0.004	$\mathbf{0.012 \pm 0.003}$
64	0.051 ± 0.008	0.083 ± 0.048	0.0074 ± 0.0010	$\mathbf{0.0055 \pm 0.0011}$

C RELATED WORK (EXTENDED)

Oversmoothing theory. The observation that node representations become indistinguishable as depth increases is well-established (Li et al., 2018). Oono & Suzuki (2020) developed the asymptotic theory that underpins much of the field: they prove that the expressive power of GNNs decays

exponentially with depth, with the rate governed by the product of weight scales and the spectrum of the propagation operator. Wu et al. (2023) extend this analysis to attention-based architectures, showing that the mechanism behaves differently under learned aggregation. Rusch et al. (2023) provide a unifying framework, defining oversmoothing as layerwise exponential convergence of suitable similarity measures. Our paper does not dispute Oono and Suzuki's asymptotic analysis. Instead, we measure which of its three assumptions—linear dynamics (A1), i.i.d. layer weights (A2), and fixed aggregation (A3)—carries the discrepancy between theory and trained, finite-depth, residual networks. We test (A2) and find that it does not carry the discrepancy, and for GCN (A3) holds by construction.

Measuring oversmoothing. The survey by Rusch, Bronstein and Mishra defines oversmoothing through the exponential convergence of node-similarity measures and reviews a range of candidates. Di Giovanni et al. (2023) ground the definition in energy-based language, showing that graph convolutions act as Laplacian smoothing and control how the Dirichlet energy of features decays. Our Sec. 2.2 normalises the Dirichlet energy to isolate geometric collapse from scale collapse, following their framework. Chen et al. (2020a) introduce the mean average distance (MAD), measuring smoothness through the information-to-noise ratio over edges. We log MAD as an internal consistency check but do not report it, because the survey rules it out as an oversmoothing measure in favour of similarity-based metrics. The normalised energy we use is a methodological choice, not a contribution; similarity measures across the literature play equivalent roles once scale-invariance is enforced.

Mitigation methods. A substantial body of work proposes architectural and training modifications to suppress oversmoothing. PairNorm (Zhao & Akoglu, 2020) constrains pairwise feature distances to remain bounded across layers. DropEdge (Rong et al., 2020) randomly removes edges during training to reduce information flow. Scattering GCN (Min et al., 2020) applies wavelet transforms to preserve information at multiple scales. GraphCON (Rusch et al., 2022) models GNN layers as discretisations of coupled oscillator dynamics. GCNII (Chen et al., 2020b) introduces initial residual connections and identity mapping to enable deep training. We do not study any of these methods and make no claim about their effectiveness. The intervention in Sec. 6 controls a network's effective depth during training; this is neither a mitigation in the sense of these prior methods nor evidence about whether they work.

Weight spectra and implicit regularisation. Martin & Mahoney (2021) establish that trained deep networks develop singular values well outside the Marchenko–Pastur bulk, forming a spike at the top of the spectrum, which they interpret as implicit self-regularisation. This observation motivates our Sec. 3, where we test whether this violation of the i.i.d. assumption (A2) explains why the theory fails on trained networks. We find that trained GCN weights depart significantly from the bulk (ratio 1.68 ± 0.48), yet the prediction error does not grow with the departure—a robustness result only, since the gap's sign depends on the simulation's fixed weight scale. The implication is that spectral outliers, while structurally important, are not the missing piece in the oversmoothing prediction.

Effective depth and signal propagation. Schoenholz et al. (2017) study information propagation in random networks, identifying depth scales that naturally limit the maximum trainable depth. Veit et al. (2016) show that residual CNNs can be understood as ensembles of paths of varying length, with effective depth determined by the paths actually used during training rather than the nominal network depth. The two works share a core insight: residual networks use much less depth than their nominal architecture suggests. Our Sec. 3 applies this to GNNs, measuring effective depth L_{eff} as the sum of per-layer branch-to-stream norm ratios. This is a different quantity from Veit et al.'s path-length analysis. In their framework, effective depth arises from the distribution of path lengths and their contribution to the forward pass; gradients flow primarily through short paths. In ours, L_{eff} is a scalar measuring the total branch gain in the residual stream, and we show that the distribution of layer-wise gains does not matter—only the total does. Whether L_{eff} is the mechanism linking residual depth to oversmoothing cannot be settled from our observational results; the causal evidence comes from the intervention of Sec. 6, which is confined to $T \leq 12$ for GCN on the three datasets we tested. We do not claim priority for the notion of effective depth in GNNs; we measure it, describe it with a power law as a compact summary, and test its predictive role for decay.

D EXTENDED STATISTICS

Material removed from the body for length. Nothing here is new; each block is cited from the section it was removed from.

D.1 ACCURACY OVER THE FULL TARGET RANGE

Extending the correlation between achieved L_{eff} and test accuracy to the full range of targets (T up to 30) gives a negative linear relationship on all three datasets (Cora -0.58, Chameleon -0.59, Texas -0.78), but a linear fit is the wrong shape for at least one of them. On Chameleon a quadratic fits much better than a line ($R^2 = 0.65$ against 0.35), with its vertex at $L_{\text{eff}} \approx 11.7$: accuracy rises through the clean band and falls beyond it, rather than falling throughout. The same fit on Cora and Texas does not improve materially over the line, and both vertices fall outside the tested range (35.7 and -25.4). Because the vertex sits close to where each dataset's own unregularised control lands ($L_{\text{eff}} = 9.19$ on Cora, 5.97 on Chameleon, 4.73 on Texas), we also checked distance from the control directly: it predicts accuracy at least as well as L_{eff} itself on every dataset ($r = -0.54$ Cora, -0.64 Chameleon, -0.80 Texas), and better than L_{eff} on Cora and Chameleon. Pushing a network's decay rate away from where it settles on its own, in either direction, tracks accuracy loss at least as well as oversmoothing does.

D.2 CROSS-TERM AND EFFECTIVE-DEPTH DISPERSION

Per-architecture cross-term sign: trained Graph Transformers are negative in 20.6% of layers, against 0.4% at initialisation; GCN 1.6% and GAT 4.8%. Across 1512 layer-observations the cross-term is non-negative in 94.9% of cases (94.1% strictly positive; 12 observations are exactly zero), with mean fractional contribution $+0.083$. Dispersion of L_{eff}: within-configuration s.d. 0.437 against between-configuration s.d. 2.319.

D.3 PROFILE ABLATION, ALL SURROGATES

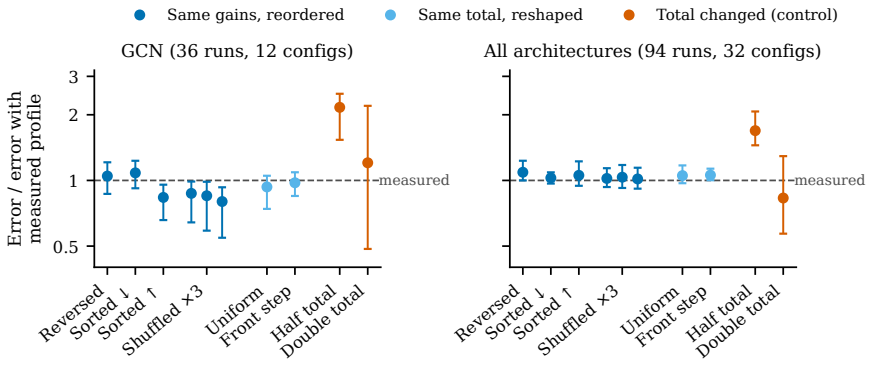

Figure 3: Randomising the gain profile leaves prediction error unchanged; changing its total does not. Each point is the mean slope error under a surrogate profile divided by the mean error under the measured profile, so 1 (dashed) means no change. Error bars are 95% intervals from a bootstrap over configurations (4000 resamples), respecting the three-seeds-per-configuration structure. The three random shuffles are independent draws, shown separately. Left: GCN, $n = 36$ runs in 12 configurations. Right: all architectures, $n = 94$ runs in 32 configurations.

D.4 INTERVENTION LIMITS

From Sec. 6, the intervention limits. Beyond $T = 12$ the picture changes, and differently on Texas. On Cora and Chameleon the constraint keeps tracking through $T = 30$ (mean absolute deviation over $T \geq 18$: 0.73 and 1.56); on Texas achieved L_{eff} reaches 13.75 at $T = 18$ and is flat thereafter at 13.36 and 13.26. At $T = 18$ this is not yet a hard ceiling—two of three seeds are still close to target and one far below (s.d. 3.2)—but by $T = 24$ all three seeds agree and the

network cannot move further however high the target. Texas has 183 nodes; we read this as a ceiling on how much total branch gain a network this small can carry, though we have not identified the mechanism. P4 decalibrates over the same range, and not monotonically: the ratio of measured to predicted decay is 2.15 at $T = 3$, falls to 1.22 at $T = 12$–18, and rises to 2.09 at $T = 30$, U-shaped rather than a constant miscalibration. At $T = 30$ Cora's measured decay (-0.35, -0.56, -0.28) is three to five times steeper than P4's -0.115, with fits still good ($R^2 = 0.80$–0.93), so the discrepancy is not fit noise; Chameleon at $T = 30$ shows the opposite pattern at smaller scale (-0.04 to -0.07 against a P4 prediction of -0.085), with fit quality also declining ($R^2 = 0.59$–0.75). The full-range correlation between achieved L_{eff} and decay, pooling all three datasets, drops to $r = -0.674$; computed separately it is still -0.82 on Cora and -0.90 on Texas, and weakest on Chameleon at -0.66, so the drop is partly a between-dataset scale effect rather than a breakdown within each dataset. Mean absolute error under P4 grows from 0.007 at $T = 3$ to 0.016, 0.026, 0.044 and 0.130 at $T = 12$, 18, 24 and 30; the $T = 30$ figure includes one collapsed run (Texas, seed 2, which the intervention itself destabilised enough to reach a majority-class prediction), and is 0.123 with it excluded under our collapse rule (Sec. 2.6). We report the difference rather than choosing one number, since a recomputation that does not filter collapsed runs will give 0.130.

D.5 Statistics moved from the body

These values were stated in the body of an earlier draft and are retained here unchanged; each is cited from the section it was removed from.

- Sec. 2.2, fit quality: R^2 ranges 0.66–0.94 across architecture and depth, and only 9.6% of runs fall below 0.5.
- Sec. 2.6: depths 2 and 4 supply 92 of the 214 valid runs of the spectral analysis.
- Sec. 2.6, collapse: 99 of 162 non-residual runs collapse, against 11 of 162 residual runs.
- Sec. 3: the trained spectral ratio rises from 1.54 at depth 2 to 1.77 at depth 16; pooled by architecture it is GAT 2.22 and Graph Transformer 2.68, with the per-layer maximum averaging 2.65.
- Sec. 3: per-dataset exponents $k = 0.35$, 0.39 and 0.42 on Cora, Chameleon and Texas; the single poor fit has $R^2 = 0.72$.
- Sec. 3, mechanism: the pooled within-cell test has $\mathrm{df} = 61$; the GCN-only test has $n = 36$; the naive across-run correlation is $r = +0.21$.
- Sec. 5, P5 per dataset: Cora $0.081 \to 0.038$, Chameleon $0.074 \to 0.025$, Texas $0.071 \to 0.018$. Extrapolated L_{eff} error at depth 64: GCN 2.1%, 5.9%, 7.3%; Graph Transformer 8.3%, 20.4%.
- Sec. 5, P6 on GAT: Chameleon $0.009 \to 0.061$, Cora $0.023 \to 0.056$, Texas $0.025 \to 0.007$.

D.6 Asymptotic versus depth-matched rate

Table 7: Asymptotic versus depth-matched simulated decay rate at $L = 32$. The discrepancy is dataset-specific and tracks the spectral gap λ^*; it is not a universal factor. Rates are slopes of $\log \hat{E}$ per layer.

Dataset	λ^*	Asymptotic	Matched L=32	Ratio	Components
Cora	0.996379	-0.0073	-0.1414	$\mathbf{19.5}\times$	78
Chameleon	0.993833	-0.0124	-0.1037	$8.4\times$	1
Texas	0.959570	-0.0825	-0.1454	$1.8\times$	1

D.7 Dose-response of the spectral violation (Sec. 3)

Nor does the error follow the violation in dose. Within a depth, trained GCNs whose weights depart further from the bulk are no further from the theory (Spearman $\rho = -0.11$, $p = 0.53$). Among seeds

of a single configuration there is a positive association ($r = 0.49$), but its size is small: one within-configuration standard deviation of the spectral ratio corresponds to about 0.005 in slope error, an order of magnitude below the 0.071 that remains to be explained.

D.8 Power-law fit comparisons and the test on k (Sec. 3)

Fitting $L_{\text{eff}} = cL^k$ per dataset, GCN gives $k = 0.35$, 0.39 and 0.42 on Cora, Chameleon and Texas (0.39 ± 0.04), each with $R^2 \geq 0.995$ over four depths. Across all nine (dataset, architecture) configurations k ranges from 0.16 to 0.53 (mean 0.35 ± 0.12), and 8 of 9 fits have $R^2 \geq 0.96$. Two qualifications apply. First, GAT and the Chameleon Graph Transformer have only three depths after collapse, and the one poor fit ($R^2 = 0.72$) is among them. The architecture means in Fig. 2 differ, but with three configurations per architecture the difference is not established (one-way ANOVA on k, $F = 0.70$, $p = 0.53$). Second, the sweep spans a factor of eight in depth, and over that range a power law is not the only candidate. A logarithm, $L_{\text{eff}} = a + b \log L$, fits worse on Cora and Chameleon (RMSE 0.39 and 0.22 against 0.16 and 0.06) but marginally better on Texas (0.09 against 0.10). We use the power law as a compact description of the sweep, not as a law of training.

D.9 Dilution correlations, median split and the Graph Transformer case (Sec. 4)

An invariance pooled over many networks could hide a real effect in the networks whose profiles are strongly shaped, diluted by the many whose profiles are nearly flat anyway. If so, the cost of flattening the profile should grow with how structured it is. It does not. Pooled over architectures, the penalty for replacing the measured profile by a flat one is uncorrelated with the profile's coefficient of variation ($r = +0.07$, $p = 0.52$) and with the fraction of the total in its first quarter of layers ($r = +0.06$, $p = 0.58$). Splitting at the median coefficient of variation, the more structured half is penalised no more than the flatter half ($+0.0017$ against $+0.0020$). The most structured profiles in the study, those of Graph Transformers at depth 32 (mean coefficient of variation 1.57), lose 0.0015 in absolute error when flattened (0.0074 to 0.0089; $p = 1.0$, $n = 8$). For GCN the relationship runs the other way. The more a network concentrates its gain in its early layers, the more flattening *helps* ($r = -0.54$, $p = 0.0006$; within depth $r = -0.35$, $p = 0.038$). Where the profile is most distinctive, it is at best irrelevant to the prediction.

E Prediction across depth: full analysis

Section 3 fitted $L_{\text{eff}} = cL^k$ to every depth at once, so the fit could not fail on a depth it had seen. We now ask whether it predicts depths it has not: holding out one depth and fitting on the other three (P5), and training only the shallowest depth and borrowing k from other configurations (P6). Both extrapolate L_{eff} to the target depth, spread the predicted total uniformly over its L layers as in P4, simulate, and compare with the measured slope. The references are the uncorrected theory (P1) and P4 at the target's *true* L_{eff}; P4 is an oracle, bounding what P5 and P6 can achieve rather than competing with them. The GCN errors reproduced for P1 and P4 match Table 2, a check on the pipeline. All tables are in Appendix B.

Leave one depth out. The test needs three depths after removing the held-out one, so it applies only where all four survive training: GCN on three datasets and the Graph Transformer on Cora and Texas, GAT and the Chameleon Graph Transformer having collapsed at depth 64. Holding out depth 16 or 32 is interpolation; holding out 64, predicted from depths 8–32, predicts a deeper network than any trained. Not knowing L_{eff} costs little: P5's error is about 1.1 times the oracle's, and P5 improves on the theory at every held-out depth, by 38% at depth 8 rising to 85% at depth 64, a 64% overall reduction for GCN. The extrapolated L_{eff} is itself accurate, with mean relative error 3.9% for GCN and 10.0% for the Graph Transformer (Appendix B). The test says nothing about GAT, which has too few surviving depths to be tested.

A single anchor. The harder test trains only depth 8 and takes k from the other eight configurations, so a configuration's own data never sets its own k; the anchor fixes c and P6 predicts depths 16, 32 and 64. For GCN one trained depth is nearly enough: P6 cuts the theory's error by 78%, coming within about 20% of the oracle that knows the true L_{eff}. The price is paid in L_{eff} itself,

the anchored estimate being off by roughly twice as much as the three-depth fit. The Graph Transformer behaves similarly. Pooled over architectures the improvement is significant at depths 32 and 64 but not at depth 16, the target closest to the anchor, and that is where GAT's failure weighs most (Appendix B).

Where it fails: GAT. Pooling hides a failure: on GAT, P6 is worse than the uncorrected theory. The comparison flatters P1, whose GAT error is artificially small, since under the matched metric some of GAT's P1 slopes are positive, which a smoothing operator cannot produce (Sec. 2.3). The failure is real nonetheless: GAT's P6 error is three times GCN's, worsening on Chameleon and Cora while Texas improves, and its anchored L_{eff} is off by 19%, the largest of any architecture. The cause is visible in the exponents: GAT's fitted k ranges from 0.16 on Cora to 0.53 on Texas, the widest spread of any architecture, so a borrowed k is a poor guess for GAT in particular. Across all architectures, 17 of 67 single-anchor predictions (25%) are worse than no correction, the two configurations harmed on average being Chameleon GAT and Cora GAT (Appendix B). After seeing this failure we tried a single alternative: borrowing k only from configurations of the same architecture. The alternative was specified in the evaluation code before it was run, but chosen only after the first approach failed on GAT, so we report it as post hoc. It helps slightly overall ($p = 0.021$) and GCN improves, but it does not rescue GAT: its slope error barely moves, its L_{eff} error gets worse, and the same 17 of 67 predictions remain worse than no correction (Appendix B). With only two same-architecture donors per configuration, this is not a result we can build on. The prediction result is therefore a GCN and Graph Transformer result, and we state it as such.