REGULARIZATION SHAPES THE LIMITS OF DEEP GNNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Depth degradation in graph neural networks (GNNs) can reflect optimization failure, poor generalization, or representation collapse. Distinguishing which mechanisms are active is necessary to determine how they should be addressed. We study this on Minesweeper, one of the few node-classification benchmarks requiring deep GNNs for high performance. We derive a simple mean-field iterative scheme that outperforms GNNs trained from scratch, establishes what is achievable, and highlights low-dimensional communication and recurrence across depth as useful inductive biases for generalization. Explicitly imposing these biases can improve deeper models, but can make training unstable and is motivated by task knowledge. Combining sufficient width with weight decay further allows an unrestricted GCN to match the gain at moderate depths, with weight decay associated with lower-rank message maps. Yet as depth increases, non-regularized models generalize poorly, while strongly regularized ones may fit poorly or train unstably. Improved performance can accompany either increases or decreases in hidden rank and Dirichlet energy, while last-layer measurements need not characterize the whole network. The limits of deep GNNs must instead be understood through the interaction of optimization, regularization, and solution selection.

1 INTRODUCTION

Deeper graph neural networks (GNNs) can propagate information farther and perform more rounds of computation, yet increasing depth often hurts predictive performance. This phenomenon is frequently attributed to *over-smoothing*, where repeated propagation makes node representations increasingly similar (Li et al., 2018; Oono & Suzuki, 2020), and is consequently diagnosed using feature similarity, Dirichlet energy, or representational rank (Chen et al., 2021; Zhou et al., 2021; Roth & Liebig, 2024; Zhang et al., 2026a). But depth degradation—a decrease in performance with depth—is an observation, not a mechanism. Deep GNNs can instead fail through optimization or generalization (Cong et al., 2021; Epping et al., 2024; Scholkemper et al., 2025; Arroyo et al., 2025; Deng et al., 2025; Arnaiz-Rodriguez & Errica, 2026). This tension leads to the basic question: *if a task admits a simple deep solution, when and why does a learned GNN fail to find it?*

We study this question on Minesweeper (Platonov et al., 2023). Minesweeper is an unusual benchmark as found by Rubio-Madrigal & Burkholz (2026): simple, task-agnostic aggregated features, followed by a graph-agnostic learned MLP, match learned GNNs on most standard node-classification datasets—but fall substantially behind on Minesweeper, most likely due to depth-related task dependencies. To resolve this gap, we construct a fixed mean-field iterative rule over label probabilities, which remains accurate at substantial depth, by exploiting the dataset's particular local structure (§ 3.1).

This construction gives a concrete notion of a simple deep generalizing solution. General neural networks are often argued to exhibit a *simplicity bias* to explain their good generalization, favoring simpler functions or features among many solutions that fit the data (Valle-Perez et al., 2019; Gatmiry et al., 2024; Jacobs et al., 2026; Zhang et al., 2026b). Which notion of simplicity is relevant, and when training favors it, remain open questions in learning mechanics (Simon et al., 2026). Our reference makes this task-specific: its communication is low-dimensional, and its computation is recurrent, with the same two-round update repeated across depth. The first question is whether learned GNNs benefit from these two forms of simplicity. We test parameter tying across depth and

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.

impose low-rank message maps (§ 3.2), and find regimes where both improve deeper models (§ 4.1). These variants are diagnostic rather than general remedies, as they restrict trainability and exploit properties of the known solution.

The next question is whether an unrestricted GNN can discover a comparably simple deep solution without prescribing its structure. The Minesweeper GCN selected by the tuning benchmark Luo et al. (2024) is relatively narrow and unregularized, effectively relying on restricted width as an implicit form of regularization. Because this can impede optimization, we instead test wider standard residual GCNs with stronger weight decay, drawing on evidence that weight decay shapes solution selection in overparameterized neural networks (D'Angelo et al., 2023; Kobayashi et al., 2024; Jacobs et al., 2025).

Jointly increasing width and weight decay produces a deeper GCN that substantially improves over the previously selected baseline (§ 4.2). Other configurations either fit poorly or achieve near-perfect training performance but generalize poorly. In the best-performing unrestricted GCNs, weight decay is associated with lower-rank message maps, although earlier layers retain higher ranks than those near the output. The restricted tied rank-one models can reach accurate checkpoints at greater depths, but their performance can fluctuate substantially during training.

These outcomes also change how we interpret common representational diagnostics of deep GNNs (§ 4.3). Matched comparisons show that improved performance can accompany either increases or decreases in last-layer hidden feature rank and normalized Dirichlet energy. These geometric measurements must therefore be interpreted alongside their variation across layers and the model's training and held-out performance.

Contributions.

1. We derive a scalar nonlinear mean-field procedure for Minesweeper that remains accurate at substantial depth, and characterize its convergence and local stability under stated conditions. This explicit computation provides a reference for identifying useful structural biases and testing them in learned GNNs.
2. We show that rank-one message maps combined with parameter tying can yield accurate deep GCN solutions with substantially higher-rank hidden representations. Increasing width and weight decay also improves unrestricted GCNs, which match or exceed the restricted models only at moderate depths.
3. Reference dynamics and complementary experiments show that propagation radius alone does not determine useful computational depth. Matched GCN comparisons further show that improved performance can accompany either increases or decreases in hidden rank and normalized Dirichlet energy, limiting their use as standalone explanations of depth degradation.

2 Related work

Depth in GNNs. Existing deep GNN analyses distinguish among expressivity, optimization, generalization, propagation scope, and representation collapse. Cong et al. (2021) separate expressivity, optimization, and generalization; Deng et al. (2025) model variation in propagation scope across nodes; Zhang et al. (2026a) compare energy- and rank-based measurements of over-smoothing; and Arnaiz-Rodriguez & Errica (2026) separate representation collapse from predictive performance. Our experiments connect these perspectives in a controlled setting: an explicit reference establishes that accurate deep computation is possible, while a depth–width–regularization grid reveals when a learned GCN finds a generalizing implementation and how different failures appear in common diagnostics. A longer discussion of the relation to existing depth analyses is included in § C.

Variational mean-field inference. Variational mean-field inference approximates an intractable graphical-model distribution within a tractable, typically factorized family, yielding local coordinate updates of a variational free energy (Jordan et al., 1999; Wainwright & Jordan, 2008; Blei et al., 2017). These iterations have also been unrolled as tied neural layers (Li & Zemel, 2014), incorporated into recurrent architectures (Zheng et al., 2015), and used to motivate learned representations of structured data (Dai et al., 2016). Whereas standard coordinate-ascent schemes update factors sequentially, synchronous updates can exhibit qualitatively different stability behavior, including oscillations (Plummer et al., 2020; Pati, 2026). We specialize mean-field inference to Minesweeper's

count constraints, deriving a fixed scalar update rather than learning the inference operator. We analyze its sequential and synchronous forms and use it as an explicit reference that separates the computational value of depth from a GNN's ability to learn a generalizing implementation.

Parameter tying across depth. Sharing parameters across propagation steps gives GNNs a recurrent structure, as in early fixed-point models, gated graph networks, and recurrent relational networks (Scarselli et al., 2009; Li et al., 2016; Palm et al., 2018). Subsequent work studies recurrence for equilibrium inference, generalization across graph scales, and parameter efficiency (Gu et al., 2020; Tang et al., 2020; Li et al., 2021), while recent theory establishes its expressivity relative to color refinement (Rosenbluth & Grohe, 2026). A similar principle underlies looped transformers, which reuse a shared block to implement iterative computations without increasing parameter count with depth (Giannou et al., 2023; Saunshi et al., 2025). We study tying as a task-derived bias: the reference's two-round update motivates repeating a two-layer block in a residual GCN. This restriction can improve generalization but also ultimately hinder training.

Late-training generalization and grokking. Grokking denotes generalization that emerges only after a network has learned its training data (Power et al., 2021). It has since been demonstrated beyond modular arithmetic and in conventional deep-learning settings (Liu et al., 2023; Humayun et al., 2024). Recent explanations of grokking emphasize a shift in implicit bias, from kernel-like fitting toward feature-learning or min-norm/max-margin solutions (Lyu et al., 2023; Kumar et al., 2024), consistent with mechanistic evidence that simpler computations eventually displace memorizing ones (Nanda et al., 2023; Merrill et al., 2023; Varma et al., 2024). Depth and decreasing feature rank have also been connected to the onset of late-training generalization (Fan et al., 2024). On Minesweeper, held-out performance continues to improve after training becomes nearly perfect. Our task-derived reference motivates concrete inductive biases as diagnostics for generalization, which give an account of how increasingly better generalizing solutions may be selected.

Explicit regularization. Optimization continues to select among interpolating solutions after training error reaches zero: for separable classifiers, gradient descent converges toward max-margin predictors (Soudry et al., 2018; Lyu & Li, 2020). Explicit regularization—particularly weight decay—can reshape both these dynamics and their limiting bias (D'Angelo et al., 2023; Jacobs et al., 2025). In matrix models, factorized gradient dynamics have been linked to nuclear-norm bias (Gunasekar et al., 2017), while ℓ_2 penalties on factors explicitly favor low-rank products (Kobayashi et al., 2024). We demonstrate this interaction in GNNs: among configurations with near-perfect training performance, weight decay can improve generalization and is associated with lower stable ranks in the later learned message maps.

3 A TASK-DERIVED REFERENCE FOR DEEP MESSAGE PASSING

The Minesweeper node-classification benchmark inherits the local count structure of the original game, which has supported probabilistic formulations (Vomlel & Tichavský, 2014). In Minesweeper, node features encode neighboring mine counts, so message passing can be useful even when neighboring labels are nearly independent. Inspired by this observation, we derive an explicit iterative inference rule over mine probabilities (see § A for mathematical details and convergence proofs). Our solution supports iterative local inference at substantial depth and enables us to separate the difficulty of the task from the difficulty of learning the computation. It also gives us a concrete notion of what simplicity can look like in this task.

3.1 SCALAR ITERATIVE INFERENCE

Definition. Let $x_i \in \{0, 1\}$ indicate whether cell i contains a mine, and let p_i^t denote its mine probability at iteration t. Thus, each node maintains a scalar. Since 20% of cells are mines, we use the prior $p_{\text{mine}} = 0.2$.

The dataset reveals neighboring mine counts for $5{,}000$ cells, with counts of five or more grouped into a single category. Let $\mathcal{O}$ denote these observed cells. For an exact clue c_j, define $E_j(s) = (s - c_j)^2$; for a "≥ 5" clue, use $E_j(s) = (5 - s)_+^2$, where $(u)_+ = \max(u, 0)$. For each cell, we compare the expected count-constraint penalty when it contains a mine with the penalty when it does not.

Table 1: Mean test ROC–AUC (%) over three splits for validation-selected non-backtracking and backtracking configurations. "Locally stable" are configurations passing the numerical local-stability check on every split. Bold iteration counts correspond to 12, 16, 24, and 32 GNN layers.

Reference	Selection	4	**6**	**8**	**12**	**16**	32	64	128	256
Non-backtracking	Stable & best	99.34	99.79	99.88	99.90	99.87	99.83	99.83	99.83	99.83
Backtracking	Best valid.	98.21	98.81	99.06	99.26	99.42	99.73	99.82	99.85	99.87
	Locally stable	96.63	97.47	97.76	98.42	98.96	99.45	99.50	99.50	99.50

Summing these differences over its observed neighboring clues gives

$$r_i(p) = \sum_{j \in \mathcal{O} \cap N(i)} \mathbb{E}_{q_p}\left[E_j(S_{j\setminus i}+1) - E_j(S_{j\setminus i})\right], \qquad S_{j\setminus i} = \sum_{k \in N(j)\setminus\{i\}} X_k,$$

where q_p treats free nodes as independent $X_k \sim \mathrm{Ber}(p_k)$ and fixes training nodes to their labels. Writing $\ell_{p_{\mathrm{mine}}} = \log[p_{\mathrm{mine}}/(1-p_{\mathrm{mine}})]$ and σ as the logistic sigmoid, the damped update is

$$p_i^{t+1} = (1-\alpha)p_i^t + \alpha\,\sigma\big(\ell_{p_{\mathrm{mine}}} - \beta r_i(p^t)\big)\,.$$

We initialize $p_i^0 = y_i$ on training nodes and $p_i^0 = p_{\mathrm{mine}}$ otherwise, and clamp training nodes to y_i after every iteration. The reference therefore uses training labels as persistent boundary conditions. A standard transductive GCN uses these labels only through its training loss and thus needs to fit them. Hard clamping is not essential for accurate deep inference: Appendix A.4 evaluates both soft supervision and a reference without training-label input.

The update excludes the receiving variable X_i from $S_{j\setminus i}$, preventing the immediate return of information along $i \to j \to i$. This defines the non-backtracking reference. Such destination-specific routing is prescribed in belief-propagation and directed-edge architectures (Zhang et al., 2020; Park et al., 2024). The corresponding backtracking version replaces $S_{j\setminus i}$ by $S_j = \sum_{k\in N(j)} X_k$ in the expected energy difference. It therefore retains the receiver's contribution, making its information flow more comparable to standard GCN updates.

In the reference, α and β control inference dynamics: α controls damping, while β controls the strength of the count constraints. Each iteration contains two successive neighborhood aggregations, so we compare T reference iterations with $2T$ GNN layers in terms of message-passing rounds.

Theoretical properties. Section A.1 derives the non-backtracking update from variational mean-field inference. Section A.2 analyzes its sequential or Gauss–Seidel version, in which each node can use updated neighboring states within the same round. Proposition 1 establishes convergence from any initialization, but the limit can be suboptimal. We find that, at $\beta = 0.7$, different initializations reach different stationary points on every split, and that another initialization achieves a lower mean-field free energy than initialization at the prior.

Section A.3 studies the parallel version, which updates all nodes synchronously from the previous state. Equation 9 gives a sufficient condition for local attraction at a stationary point. We evaluate this condition at the point obtained by sequential updates initialized at $p^0 = \mathbf{1}p_{\mathrm{mine}}$. This certificate establishes attraction near the selected stationary point, rather than convergence to a global minimum. Moreover, sequential and parallel updates can approach different stationary points from the same prior initialization.

Finally, Section A.4 studies the backtracking ablation. Its local stability is governed by Eq. 10; for sufficiently small β, the update converges to a unique fixed point from any initialization. Outside this regime, uniqueness and global convergence are not guaranteed. Smaller α takes a smaller step toward the updated probabilities, which can suppress oscillations but may require more iterations to approach a fixed point. Figure 1 illustrates the interaction between constraint strength and damping.

Even a scalar, task-derived inference procedure presents difficulties in selecting and maintaining a solution: convergent updates can reach different objective values, and synchronous updates introduce additional stability constraints. These provide concrete mechanisms to investigate when studying how GNNs learn iterative computation.

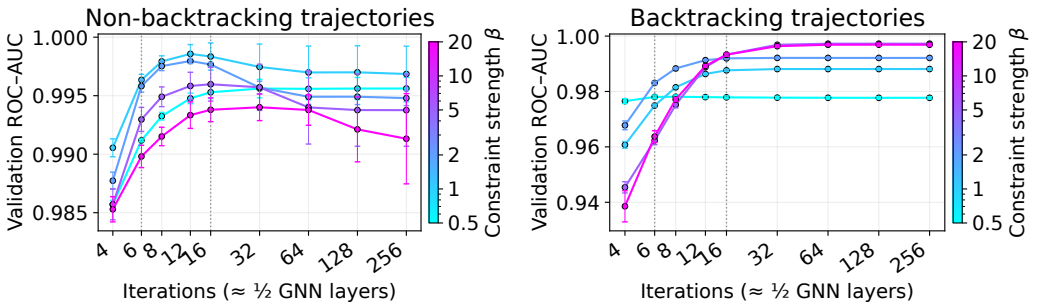

Figure 1: Non-backtracking and backtracking reference trajectories. For each β, α is selected by mean validation ROC–AUC at 16 iterations and held fixed along the trajectory. Higher-β configurations tend to be selected with smaller α.

Results. The reference establishes that local inference can remain accurate at substantial depth, as reported in Table 1. For the non-backtracking budgets, the best-validation configurations pass the local-stability check on every split. For backtracking, unrestricted selection gives higher performance than selection restricted to configurations passing these checks.

To compare the reference with learned predictions, we fit effective $(\hat{\alpha}, \hat{\beta})$ pairs to a GCN's output probabilities at recurring training checkpoints (Fig. 7 in the appendix). The depth-16 GCN is matched to eight reference iterations. The fitted $\hat{\alpha}$ remains near 0.3, whereas $\hat{\beta}$ has an upwards tendency around 1. At the final measured epoch, the reference explains 83% of the variation in the GCN probabilities on the fitting nodes, with probability RMSE of 0.15. Details of the fitting procedure are given in Appendix A.5.

3.2 Task-derived inductive biases

The reference is simple in two structural ways. First, communication is low-dimensional: each node propagates only a single scalar probability. Second, the computation is recurrent: the same two-round update is reused at every iteration rather than learning a different transformation at each depth. These properties give us two task-derived notions of simplicity that learned GNNs should tend towards to potentially overcome overfitting.

Our residual GCN follows Luo et al. (2024). Without dropout or normalization, the update is:

$$H^{(t+1)} = \phi\Big(PH^{(t)}W_a^{(t)} + H^{(t)}W_r^{(t)}\Big) \tag{1}$$

where P is a propagation matrix derived from the graph adjacency matrix, such as the symmetric normalized adjacency matrix, and ϕ is a ReLU.

Low-dimensional communication. The learned channel-mixing matrices $W_a^{(t)}$ control the directions available to transformed messages, so a rank-one map is the closest analogue of the scalar communication used by the reference. To enforce rank one on this operation, we consider the restricted bottlenecked update:

$$H^{(t+1)} = \phi\Big(PH^{(t)}u_{(t)}v_{(t)}{}^{\top} + H^{(t)}W_r^{(t)}\Big) \tag{2}$$

where $u_{(t)}$ and $v_{(t)}$ are learnable vectors whose outer product has rank one. We also consider the corresponding bottleneck with rank 2 for additional flexibility during training. Closeness to the restricted version is quantified by the stable rank of $W_a^{(t)}$, where lower values indicate that fewer message directions dominate (see § B for measurement definitions).

Recurrent structure. The reference reuses the same two-round update at every iteration. A corresponding GCN restriction therefore repeats a two-layer block across depth: $W_q^{(t+2)} = W_q^{(t)}$, for $q \in \{a, r\}$. For unrestricted models, we examine the average and last layers of message-map stable ranks; we report the layerwise profile in Fig. 11 in the Appendix. Period-two parameter tying produces a repeating two-layer rank pattern, providing a reference for comparing how message maps vary across depth.

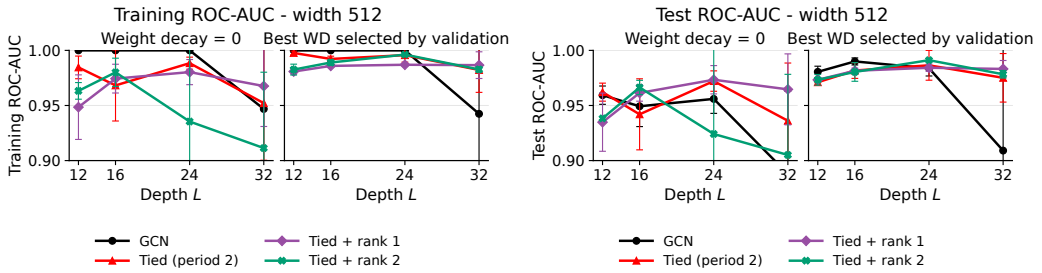

Figure 2: Train (left) and test (right) performance of GCN variants across depth for width 512: an unrestricted GCN, a period-2 parameter-tied GCN, a rank-one GCN, a rank-one tied GCN, and a rank-two tied GCN.

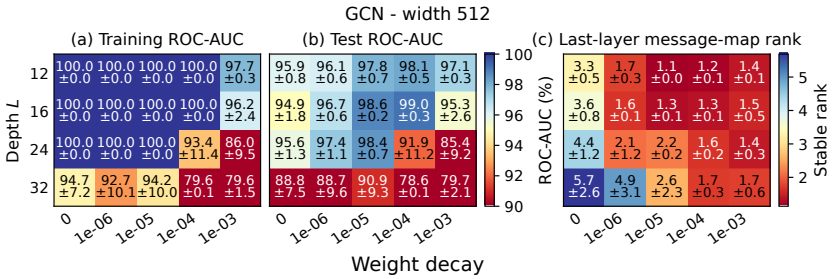

Figure 3: Train ROC-AUC, test ROC-AUC, and last-layer message rank of a GCN with width 512 across depth and weight decay. The best configurations, as seen in (b), combine perfect training performance (a) with lower-rank communication (c).

Accordingly, our experiments on restricted GCNs test the two properties with low-rank message maps and with tied parameters. These restricted models inject knowledge that the reference tells us is sufficient for accurate deep inference. Their role is diagnostic: if they improve deeper GNNs, this provides evidence that biasing the model toward the structure of a simple iterative solution can help, while also revealing the optimization cost of imposing that structure directly.

4 Experimental Results

Our experimental grid evaluates the residual GCN of Luo et al. (2024) on Minesweeper over depth $\in \{12, 16, 24, 32\}$, width $\in \{64, 512\}$, and weight decay $\in \{0, 1e{-}6, 1e{-}5, 1e{-}4, 1e{-}3\}$, for 5000 epochs. Configurations are denoted by $(\text{depth}, \text{width}, \text{wd})$. Complete results and experimental details appear in Appendix D.

Alongside training and validation performance, we measure the normalized Dirichlet Rayleigh quotient and effective rank of the hidden representations following Zhang et al. (2026a). Additionally, we report the stable rank of the GCN message maps, motivated by the low-dimensional communication in our reference. Hidden rank and message-map rank capture different objects: a scalar computation can still use several hidden channels in a ReLU network, for example to approximate the sigmoid update of our reference, while message-map rank measures the channel directions made available by the learned transformation. Definitions are given in Appendix B.

4.1 Simplicity biases can rescue deep GNNs

The task-derived restrictions from § 3.2 test whether the reference's two structural biases can benefit learned GNNs. Figure 2 shows train and test performance for width 512, while Fig. 9 in the appendix additionally reports train, validation, and test performance for widths 64 and 512 for all variants.

Combining rank-one communication with parameter tying yields accurate validation-selected checkpoints across depths even without weight decay, demonstrating that well-generalizing solutions with

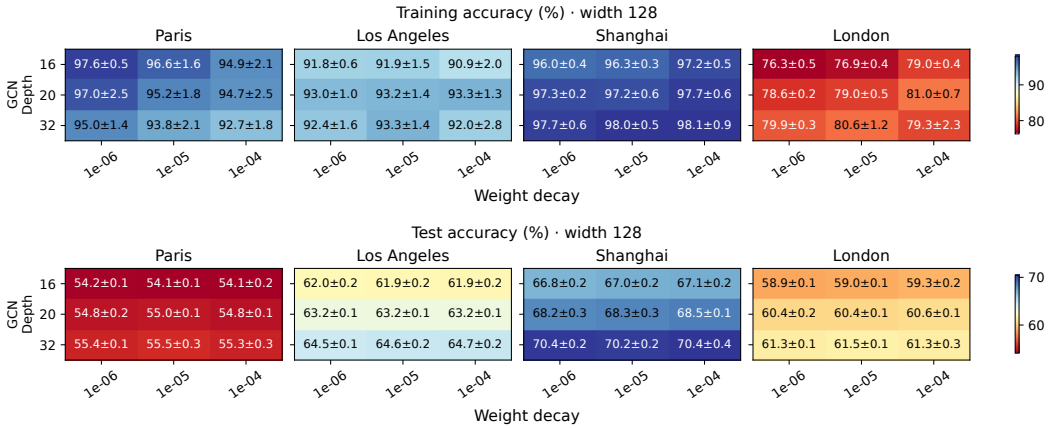

Figure 4: Train and test average accuracy for City Networks (Liang et al., 2026) over 5 seeds.

rank-one message maps exist. This benefit does not imply stable training, as we show for configuration D in Figure 5. Moreover, the untied rank-one model does not consistently recover these gains at greater depths, indicating that low-dimensional communication alone is insufficient on its own.

With weight decay, an unrestricted GCN matches or exceeds the restricted models at depths 12 and 16, falls behind at 24, and degrades substantially at 32. Regularized unrestricted models can therefore recover benefits associated with these structural biases at moderate depth, while restricted models can reach better-performing checkpoints at greater depth. These results motivate the restrictions as diagnostic interventions, while also exposing the difficulty of reliably maintaining the solutions they make accessible.

4.2 Improving performance through depth and regularization

The previously selected Minesweeper GCN of Luo et al. (2024) uses $(12, 64, 0)$ and obtains $97.8\% \pm 0.3$ test ROC-AUC in our setup. We find that jointly increasing width and weight decay produces a better deeper GCN: $(16, 512, 1\mathrm{e}{-4})$ reaches $99.0\% \pm 0.3$ test ROC-AUC.

The neighboring configurations make the interaction between width and regularization more concrete, as we report in Figure 3. Increasing width from 64 to 512 makes better configurations accessible. This benefit of additional capacity is consistent with theoretical results showing that robust interpolation can require substantially more parameters than fitting the training data alone (von Berg et al., 2026). Weakly regularized models interpolate but can still generalize poorly, while stronger weight decay reduces the message-map rank but eventually constrains training performance. The best configuration therefore combines trainability with regularized communication.

To assess whether these observations extend beyond Minesweeper, we also evaluate GCNs on City Networks (Liang et al., 2026), whose labels are constructed using 16-hop neighbourhoods. As shown in Fig. 4, increasing depth continues to improve performance up to 32 layers, and we observe no training failures among the tested configurations, which suggests that further capacity may be necessary. With such large datasets, the question then becomes how to recover benefits of overparameterization in the face of computational constraints.

These results also suggest that additional computational capacity can be beneficial even beyond the propagation depth used to define the task. More broadly, this challenges interpreting problem radius or measured interaction range as the depth a GNN requires (Alon & Yahav, 2021; Bamberger et al., 2025). Reaching the relevant nodes does not determine how much nonlinear computation is useful for processing their information. Together with the Minesweeper results and our reference analysis from Figure 1, this also questions inferring task depth from the best-performing configuration, since this optimum jointly depends on model capacity, regularization, and trainability.

Table 2: Representation geometry on Minesweeper for width 512, averaged over three runs, taken at each run's validation-selected checkpoint.

Configuration	Model	Depth	WD	ROC–AUC (%)		Hidden rank		Message rank		Dir
				Train	Test	Mean	Last	Mean	Last	Last
A	GCN	24	0	100.0	95.6	101.1	21.5	7.2	4.4	0.5
B	GCN	24	10^{-5}	100.0	98.4	57.2	4.2	3.5	2.2	0.3
C	GCN	32	10^{-4}	79.6	78.6	4.4	7.8	1.8	1.7	0.3
D	Tied + rank 1	32	10^{-4}	98.7	98.3	52.1	46.0	1.0	1.0	0.6

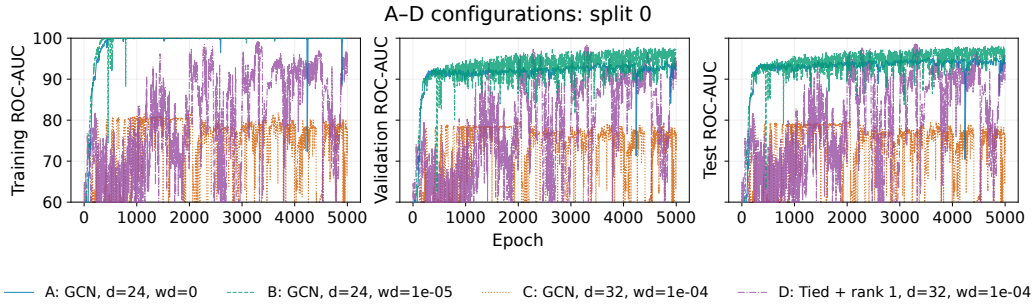

Figure 5: Performance metrics across epochs for split 0 of the A-D configurations.

4.3 Representation geometry and depth degradation

The preceding experiments show that regularization can improve generalization in deep GNNs while reducing message-map rank, but can also impair training at greater depth. We now ask how hidden effective rank and normalized Dirichlet energy describe these outcomes, and whether last-layer measurements are sufficient.

Prior work (Zhang et al., 2026a) acknowledges that rank and energy can correlate poorly with accuracy in residual architectures. We examine how these measurements distinguish improved generalization, poor fitting, and fluctuating training performance through comparisons across configurations, layers, and epochs. Table 2 compares the best-validation checkpoint of four configurations at width 512; we show training curves of split 0 in Figure 5, and their geometry across epochs in Figure 12 in the appendix.

1. **A→B: Reduced overfitting with lower rank and energy.** At fixed depth and width, adding weight decay preserves near-perfect training performance and improves test performance. Both mean and last-layer hidden and message-map ranks decrease, alongside Dirichlet energy. Reduced rank and energy therefore accompany improved generalization.
2. **C→D: Better fitting with higher hidden rank and energy.** At fixed depth, width, and weight decay, parameter tying and rank-one message maps improve both training and test performance. Hidden rank increases both on average and at the last layer, and Dirichlet energy increases, while message-map rank is one throughout. Low-dimensional communication can therefore coexist with richer hidden representations.
3. **B→C: Last-layer rank is not enough.** At the validation-selected checkpoints, C has substantially poorer training and test performance despite a higher last-layer hidden rank (7.8 versus 4.2). However, its layer-average hidden rank is much lower (4.4 versus 57.2), as are both mean and last-layer message-map ranks. Figure 10 in the appendix shows the layerwise distribution, which explains this reversal: B's rank rises sharply in the early layers before declining toward the output, whereas C remains low throughout, consistent with representational collapse.

The trajectories distinguish these checkpoint comparisons from behavior throughout training. On split 0, A and B rapidly reach near-perfect training ROC–AUC, whereas C remains poorly fitting. D reaches high training and held-out ROC–AUC but exhibits repeated large drops, so its final epoch

does not summarize its behavior (Fig. 5). Its hidden ranks and Dirichlet energy also vary substantially across epochs, while its message maps remain rank one throughout (Fig. 12). The selected-checkpoint comparison establishes that the restricted architecture can reach a generalizing solution, while the trajectory reveals that training does not reliably maintain its performance.

In summary, rank and Dirichlet energy can help identify collapse-like behavior, but when this accompanies poor fitting, the failure is already apparent in training performance. Conversely, lower rank and energy can accompany improved generalization, and a low-rank final representation does not imply low-rank representations throughout the network. These measurements must therefore be interpreted across layers and training epochs, alongside fitting and held-out performance, with a clear distinction between the selected checkpoint and the surrounding training trajectory.

5 Conclusion

5.1 Findings and discussion

Our scalar mean-field reference shows how deep message passing can succeed on Minesweeper despite nearly independent neighboring labels and restricted communication channels. The useful information comes from observed count constraints, which repeated nonlinear updates combine across the graph. Homophily and label informativeness alone therefore do not characterize the information available for prediction. The reference also makes concrete the distinction between propagating fixed input features and repeatedly updating the states being propagated, providing a constructive account of the exception identified by Rubio-Madrigal & Burkholz (2026).

Starting from this explicit solution allows us to derive structural biases and test their effects in learned GNNs. Rank-one message maps combined with parameter tying can yield accurate deep solutions with higher-rank hidden representations, while sufficient width and weight decay recover comparable performance in unrestricted GCNs at moderate depths. Successful computation can therefore combine substantial hidden capacity with restricted communication. The instability of some restricted models and the sensitivity to regularization also show how these biases interact with trainability, helping explain the hyperparameter sensitivity of GNN evaluation (Luo et al., 2024; 2025).

These results also change how depth and representation geometry should be interpreted. The reference exposes dependence on damping and initialization, while GCNs on City Networks benefit from additional layers beyond the radius used to construct the labels. Useful computational depth therefore cannot be inferred from propagation radius alone. Likewise, improved performance can accompany either increases or decreases in hidden rank and normalized Dirichlet energy, qualifying their interpretation as evidence of harmful over-smoothing (Li et al., 2018; Oono & Suzuki, 2020; Jamadandi et al., 2024; Zhang et al., 2026a). Together, these findings motivate studying the nonlinear computations that support successful prediction and the conditions under which GNNs learn effective solutions at large depths.

5.2 Open questions and future work

Our findings contribute a controlled GNN case study to the learning-mechanics agenda of Simon et al. (2026), connecting their questions about nonlinear learning, functional complexity, and hyperparameter effects. A central open problem is to theoretically connect the reference's inference dynamics to the training dynamics of learned GNNs. This would help explain to what extent width and regularization favor effective solutions, and how to maintain their performance at greater depths.

The framework can also be extended to other tasks with known iterative solutions, including algorithmic reasoning benchmarks (Veličković et al., 2022), graphical-model inference, and error-correcting decoding (Kuck et al., 2020; Garcia Satorras & Welling, 2021). Reference algorithms provide concrete hypotheses about what information needs to be communicated and how it should be processed across successive steps. Testing these hypotheses in learned models would help establish which forms of computational simplicity transfer across tasks, and when additional depth supports useful computation.

AI USE STATEMENT

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments, implement methods, formulate mathematical claims, assist in the writing of proofs, and interpret results. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, or propose or refine hypotheses; and generating synthetic data sets, assisting with translation, cleaning and reformatting dataset, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, edit a research paper to improve readability, summarize or analyse existing literature, and identify relevant literature. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The code to reproduce results is available with the paper. The proofs of theoretical claims are presented in the appendix.

REFERENCES

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=i80OPhOCVH2.

Adrian Arnaiz-Rodriguez and Federico Errica. Oversmoothing, "oversquashing", heterophily, long-range, and more: Demystifying common beliefs in graph machine learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Q3MisVkuTu.

Alvaro Arroyo, Alessio Gravina, Benjamin Gutteridge, Federico Barbero, Claudio Gallicchio, Xiaowen Dong, Michael M. Bronstein, and Pierre Vandergheynst. On vanishing gradients, over-smoothing, and over-squashing in GNNs: Bridging recurrent and graph learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=N4cyRMuLyl.

Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137(1):91–129, Feb 2013. ISSN 1436-4646. doi: 10.1007/s10107-011-0484-9. URL https://doi.org/10.1007/s10107-011-0484-9.

Jacob Bamberger, Benjamin Gutteridge, Scott le Roux, Michael M. Bronstein, and Xiaowen Dong. On measuring long-range interactions in graph neural networks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=2fBcAOi8lO.

Gleb Bazhenov, Oleg Platonov, and Liudmila Prokhorenkova. Graphland: Evaluating graph machine learning models on diverse industrial data. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2025.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773. URL https://doi.org/10.1080/01621459.2017.1285773.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. In *ICML 2020 Graph Representation Learning workshop*, 2020. URL https://arxiv.org/abs/2006.13318.

Wuyang Chen, Junru Wu, Zhangyang Wang, and Boris Hanin. Principled architecture-aware scaling of hyperparameters. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=HZndRcfyNI.

Yuzhou Chen, Baris Coskunuzer, and Yulia Gel. Topological relational learning on graphs. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=YOc9i6-NrQk.

Weilin Cong, Morteza Ramezani, and Mehrdad Mahdavi. On provable benefits of depth in training graph convolutional networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 9936–9949. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/524265e8b942930fbbe8a5d979d29205-Paper.pdf.

Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2702–2711, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/daib16.html.

Francesco D'Angelo, Aditya Varre, Maksym Andriushchenko, and Nicolas Flammarion. Why do we need weight decay for overparameterized deep networks? In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023. URL https://openreview.net/forum?id=3vBtKsiHEC.

Gangda Deng, Hongkuan Zhou, Rajgopal Kannan, and Viktor Prasanna. Mixture of scope experts at test: Generalizing deeper graph neural networks with shallow variants. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=swwAPbtU07.

Bastian Epping, Alexandre René, Moritz Helias, and Michael T Schaub. Graph neural networks do not always oversmooth. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=nY7fGtsspU.

Simin Fan, Razvan Pascanu, and Martin Jaggi. Deep grokking: Would deep neural networks generalize better?, 2024. URL https://arxiv.org/abs/2405.19454.

Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Benjamin Chamberlain, Michael Bronstein, and Federico Monti. Sign: Scalable inception graph neural networks. In *ICML 2020 Workshop on Graph Representation Learning and Beyond*, 2020.

Víctor Garcia Satorras and Max Welling. Neural enhanced belief propagation on factor graphs. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 685–693. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/garcia-satorras21a.html.

Khashayar Gatmiry, Zhiyuan Li, Sashank J. Reddi, and Stefanie Jegelka. Simplicity bias via global convergence of sharpness minimization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15102–15129. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/gatmiry24a.html.

Angeliki Giannou, Shashank Rajput, Jy-Yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11398–11442. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/giannou23a.html.

Francesco Di Giovanni, James Rowbottom, Benjamin Paul Chamberlain, Thomas Markovich, and Michael M. Bronstein. Understanding convolution on graphs via energies. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=v5ew3FPTgb.

Fangda Gu, Heng Chang, Wenwu Zhu, Somayeh Sojoudi, and Laurent El Ghaoui. Implicit graph neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11984–11995. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8b5c8441a8ff8e151b191c53c1842a38-Paper.pdf.

Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Implicit regularization in matrix factorization. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6152–6160, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Deep networks always grok and here is why. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20722–20745. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/humayun24a.html.

Tom Jacobs, Chao Zhou, and Rebekka Burkholz. Mirror, mirror of the flow: How does regularization shape implicit bias? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=MLiR9LS5PW.

Tom Jacobs, Advait Gadhikar, Celia Rubio-Madrigal, and Rebekka Burkholz. Hyperbolic aware minimization: Implicit bias for sparsity. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XKB5Hu0ACY.

Adarsh Jamadandi, Celia Rubio-Madrigal, and Rebekka Burkholz. Spectral graph pruning against over-squashing and over-smoothing. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 10348–10379. Curran Associates, Inc., 2024. doi: 10.52202/079017-0331. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/140aac600566125915df7e74ff538f66-Paper-Conference.pdf.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, November 1999. ISSN 0885-6125. doi: 10.1023/A:1007665907178. URL https://doi.org/10.1023/A:1007665907178.

Seijin Kobayashi, Yassir Akram, and Johannes Von Oswald. Weight decay induces low-rank attention layers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=oDeqjIM9Sk.

F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001. doi: 10.1109/18.910572.

Jonathan Kuck, Shuvam Chakraborty, Hao Tang, Rachel Luo, Jiaming Song, Ashish Sabharwal, and Stefano Ermon. Belief propagation neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 667–678. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/07217414eb3fbe24d4e5b6cafb91ca18-Paper.pdf.

Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vt5mnLVIVo.

Guohao Li, Matthias Müller, Bernard Ghanem, and Vladlen Koltun. Training graph neural networks with 1000 layers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6437–6449. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/li21o.html.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Yujia Li and Richard S. Zemel. Mean-field networks. In *ICML Workshop on Learning Tractable Probabilistic Models*, 2014. URL https://www.cs.toronto.edu/~zemel/documents/icml14_mfn.pdf.

Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. Gated graph sequence neural networks. In *Proceedings of ICLR'16*, April 2016. URL https://www.microsoft.com/en-us/research/publication/gated-graph-sequence-neural-networks/.

Huidong Liang, Haitz Sáez de Ocáriz Borde, Baskaran Sripathmanathan, Michael M. Bronstein, and Xiaowen Dong. Towards quantifying long-range interactions in graph machine learning: a large graph dataset and a measurement. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fylMiUmg39.

Ziming Liu, Eric J Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zDiHoIWa0q1.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. Classic gnns are strong baselines: Reassessing gnns for node classification. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 97650–97669. Curran Associates, Inc., 2024. doi: 10.52202/079017-3098. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b10ed15ff1aa864f1be3a75f1ffc021b-Paper-Datasets_and_Benchmarks_Track.pdf.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. Can classic GNNs be strong baselines for graph-level tasks? simple architectures meet excellence. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ZH7YgIZ3DF.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJeLIgBKPS.

Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon Shaolei Du, Jason D Lee, and Wei Hu. Dichotomy of early and late phase implicit biases can provably induce grokking. In *The Twelfth International Conference on Learning Representations*, 2023.

William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023. URL https://openreview.net/forum?id=8GZxtu46Kx.

Nimrah Mustafa and Rebekka Burkholz. Dynamic rescaling for training GNNs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=IfZwSRpqHl.

Nimrah Mustafa and Rebekka Burkholz. GATE: How to keep out intrusive neighbors. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=Sjv5RcqfuH.

Nimrah Mustafa, Aleksandar Bojchevski, and Rebekka Burkholz. Are GATs out of balance? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=qY7UqLoora.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1ldO2EFPr.

J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Society for Industrial and Applied Mathematics, 2000. doi: 10.1137/1.9780898719468. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898719468.

Rasmus Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/b9f94c77652c9a76fc8a442748cd54bd-Paper.pdf.

Seonghyun Park, Narae Ryu, Gahee Kim, Dongyeop Woo, Se-Young Yun, and Sungsoo Ahn. Non-backtracking graph neural networks. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=64HdQKnyTc.

Debdeep Pati. Stability of sequential and parallel coordinate ascent variational inference. 2026. URL https://arxiv.org/abs/2603.20929.

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

Sean Plummer, Debdeep Pati, and Anirban Bhattacharya. Dynamics of coordinate ascent variational inference: A case study in 2d ising models. *Entropy*, 22(11), 2020. ISSN 1099-4300. doi: 10.3390/e22111263. URL https://www.mdpi.com/1099-4300/22/11/1263.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. In *ICLR MATH-AI Workshop*, 2021.

Eran Rosenbluth and Martin Grohe. Repetition makes perfect: Recurrent graph neural networks match message passing limit. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40 (30):25168–25175, 2026. doi: 10.1609/aaai.v40i30.39707. URL https://ojs.aaai.org/index.php/AAAI/article/view/39707.

Andreas Roth and Thomas Liebig. Rank collapse causes over-smoothing and over-correlation in graph neural networks. In Soledad Villar and Benjamin Chamberlain (eds.), *Proceedings of the Second Learning on Graphs Conference*, volume 231 of *Proceedings of Machine Learning Research*, pp. 35:1–35:23. PMLR, 27–30 Nov 2024. URL https://proceedings.mlr.press/v231/roth24a.html.

Celia Rubio-Madrigal and Rebekka Burkholz. Fixed aggregation features can rival GNNs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=gSZhNPp103.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=din0lGfZFd.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

Michael Scholkemper, Xinyi Wu, Ali Jadbabaie, and Michael Schaub. Residual connections and normalization can provably prevent oversmoothing in gnns. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 43581–43610, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6c473e69ba261200dd595d07494c1a73-Paper-Conference.pdf.

P.L. Seidel. *Über ein Verfahren, die Gleichungen, auf welche die Methode der kleinsten Quadrate führt, sowie lineäre Gleichungen überhaupt, durch successive Annäherung aufzulösen*. Abhandlungen der Bayerischen Akademie der Wissenschaften, Mathematisch-Physikalische Klasse. Verlag d. Akad., 1873. URL https://books.google.de/books?id=7j0rSzyqet4C.

Jamie Simon, Daniel Kunin, Alexander Atanasov, Enric Boix-Adserà, Blake Bordelon, Jeremy Cohen, Nikhil Ghosh, Florentin Guth, Arthur Jacot, Mason Kamb, Dhruva Karkada, Eric J. Michaud, Berkan Ottlik, and Joseph Turnbull. There will be a scientific theory of deep learning, 2026. URL https://arxiv.org/abs/2604.21691.

Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, 2018. URL https://arxiv.org/abs/1803.09820.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *J. Mach. Learn. Res.*, 19(1):2822–2878, January 2018. ISSN 1532-4435.

Hao Tang, Zhiao Huang, Jiayuan Gu, Bao-Liang Lu, and Hao Su. Towards scale-invariant graph-related problem solving by iterative homogeneous gnns. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15811–15822. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/b64a70760bb75e3ecfd1ad86d8f10c88-Paper.pdf.

Guillermo Valle-Perez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rye4g3AqFm.

Vikrant Varma, Rohin Shah, Zachary Kenton, Janos Kramar, and Ramana Kumar. Explaining grokking through circuit efficiency, 2024. URL https://openreview.net/forum?id=7Zbg38nA0J.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The CLRS algorithmic reasoning benchmark. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22084–22102. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/velickovic22a.html.

Jiří Vomlel and Petr Tichavský. An approximate tensor-based inference method applied to the game of minesweeper. In Linda C. van der Gaag and Ad J. Feelders (eds.), *Probabilistic Graphical Models*, pp. 535–550, Cham, 2014. Springer International Publishing. ISBN 978-3-319-11433-0.

Jonas von Berg, Adalbert Fono, Massimiliano Datres, Sohir Maskey, and Gitta Kutyniok. The price of robustness: Stable classifiers need overparameterization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=63VXjOFiit.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1–2):1–305, January 2008. ISSN 1935-8237. doi: 10.1561/2200000001. URL https://doi.org/10.1561/2200000001.

Stephen J. Wright. Coordinate descent algorithms. *Math. Program.*, 151(1):3–34, June 2015. ISSN 0025-5610. doi: 10.1007/s10107-015-0892-3. URL https://doi.org/10.1007/s10107-015-0892-3.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 6861–6871. PMLR, 2019.

Kaicheng Zhang, Piero Deidda, Desmond Higham, and Francesco Tudisco. Are we measuring oversmoothing in graph neural networks correctly? In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=r3DOISnvHD.

Yedi Zhang, Andrew M Saxe, and Peter E. Latham. Saddle-to-saddle dynamics explains a simplicity bias across neural network architectures. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=Vit5M0G5Gb.

Zhen Zhang, Fan Wu, and Wee Sun Lee. Factor graph neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8577–8587. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/61c66a2f4e6e10dc9c16ddf9d19745d6-Paper.pdf.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pp. 1529–1537, USA, 2015. IEEE Computer Society. ISBN 9781467383912. doi: 10.1109/ICCV.2015.179. URL https://doi.org/10.1109/ICCV.2015.179.

Kaixiong Zhou, Xiao Huang, Daochen Zha, Rui Chen, Li Li, Soo-Hyun Choi, and Xia Hu. Dirichlet energy constrained learning for deep graph neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=6YL_BntJrz6.

A MATHEMATICAL DETAILS OF THE REFERENCE

We write the scalar reference from Section 3.1 in differential form and give checkable convergence certificates.

A.1 REFERENCE UPDATE AND VARIATIONAL ORIGIN

Let $\mathcal{T}$ be the clamped training nodes and $\mathcal{U} = V \setminus \mathcal{T}$ the free nodes. Clamping conditions the reference inference problem on the observed training labels. This differs from requiring a learned GNN to fit those labels at its output, which does not prescribe its intermediate states.

For $i, k \in \mathcal{U}$, we define the non-backtracking two-hop influence matrix

$$B_{ik} = \sum_{j \in \mathcal{O}} \mathbf{1}\{i \in N(j)\}\mathbf{1}\{k \in N(j) \setminus \{i\}\}.$$

B is symmetric, entrywise nonnegative, and has zero diagonal. The exclusion of the receiving variable mirrors the cavity structure of factor-to-variable sum-product messages, although our update is mean-field rather than strictly belief propagation (Kschischang et al., 2001). The observed clue categories (input features) are $0, 1, 2, 3, 4, \geq 5$. We define

$$E_j(s) = \begin{cases} (s - c_j)^2, & c_j \in \{0, 1, 2, 3, 4\}, \\ (5 - s)_+^2, & c_j = \text{“}\geq 5\text{”}, \end{cases} \qquad (u)_+ = \max\{u, 0\},$$

and $\Delta E_j(s) = E_j(s+1) - E_j(s)$. With $S_{j\setminus i} = \sum_{k \in N(j)\setminus\{i\}} X_k$, we can then write

$$r_i(p) = \sum_{j \in \mathcal{O} \cap N(i)} \mathbb{E}_{q_p}[\Delta E_j(S_{j\setminus i})], \qquad X_k = y_k \quad (k \in \mathcal{T}),$$

where q_p is the factorized distribution defined below. Writing $p^t \in [0,1]^{|\mathcal{U}|}$ for the free probabilities and applying σ componentwise, our reference becomes

$$g(p) = \sigma(\ell_{p_{\text{mine}}}\mathbf{1} - \beta r(p)), \qquad F_\alpha(p) = (1-\alpha)p + \alpha g(p), \qquad p^{t+1} = F_\alpha(p^t). \tag{3}$$

Mean-field representation. Following standard mean-field variational inference (Jordan et al., 1999; Wainwright & Jordan, 2008), consider the Gibbs distribution over the unclamped mine variables

$$P_\beta(x_{\mathcal{U}} \mid c, y_{\mathcal{T}}) = \frac{1}{Z_\beta} \prod_{i \in \mathcal{U}} p_{\text{mine}}^{x_i}(1 - p_{\text{mine}})^{1-x_i} \exp\left(-\beta \sum_{j \in \mathcal{O}} E_j\left(\sum_{k \in N(j)} x_k\right)\right), \tag{4}$$

where $x_i = y_i$ on $\mathcal{T}$ and Z_β normalizes the distribution. Each observed count couples its neighboring cells, so the exact target ranges over $2^{|\mathcal{U}|}$ joint assignments rather than independent marginals. The mean-field approximation replaces this coupled distribution by the fully factorized family

$$q_p(x_{\mathcal{U}}) = \prod_{i \in \mathcal{U}} p_i^{x_i}(1 - p_i)^{1-x_i}.$$

Each factor retains the marginal mean $p_i = \mathbb{E}_{q_p}[X_i]$ but discards correlations: replacing the coupled local field by its expectation is the “mean field.” Up to $\log Z_\beta$, minimizing $\mathrm{KL}(q_p \| P_\beta)$ is equivalent to minimizing

$$\mathcal{F}(p) = \sum_{i \in \mathcal{U}} \mathrm{KL}(\mathrm{Ber}(p_i) \,\|\, \mathrm{Ber}(p_{\text{mine}})) + \beta \sum_{j \in \mathcal{O}} \mathbb{E}_{q_p}\left[E_j\left(\sum_{k \in N(j)} X_k\right)\right]. \tag{5}$$

This is the variational mean-field free energy: its terms are respectively the entropy/prior contribution and expected count-constraint energy.

Probability update. The following update follows coordinate-ascent variational inference, expressed here as coordinate minimization of the free energy (Blei et al., 2017). Changing X_i from zero to one changes the expected energy of factor j by $\mathbb{E}_{q_p}[E_j(S_{j\setminus i}+1) - E_j(S_{j\setminus i})]$. For an exact clue this equals $2(\sum_{k\in N(j)\setminus\{i\}} p_k - c_j) + 1$. For a censored clue the expectation is evaluated over the Bernoulli sum. Equivalently,

$$\frac{\partial \mathcal{F}}{\partial p_i} = \operatorname{logit}(p_i) - \ell_{p_{\text{mine}}} + \beta r_i(p).$$

Thus $g_i(p)$ is the unique coordinate minimizer. Updating every g_i from the same state gives the parallel (Jacobi) map in equation 3; its fixed points are precisely the stationary points of equation 5.

Let us define the following notation for following results. Let $\mathcal{B}(p) = \frac{1}{2}Dr(p)$. Its diagonal is zero and, for $i \neq k$,

$$\mathcal{B}_{ik}(p) = \frac{1}{2} \sum_{j\in\mathcal{O}:\, i,k\in N(j)} \mathbb{E}_{q_p}\left[\Delta^2 E_j(S_{j\setminus\{i,k\}})\right], \qquad \Delta^2 E_j(s) = E_j(s+2) - 2E_j(s+1) + E_j(s).$$

Since $0 \le \Delta^2 E_j(s) \le 2$ for nonnegative integer s, $\mathcal{B}(p)$ is symmetric and $0 \le \mathcal{B}(p) \le B$ entrywise. In particular, $\|\mathcal{B}(p)\|_2 \le \rho(B)$. We write $\underline{\lambda} = \inf_{p\in[0,1]^{|\mathcal{U}|}} \lambda_{\min}(\mathcal{B}(p))$.

A.2 Sequential updates

First consider cyclic coordinate-wise mean field. Let i_t cycle through a fixed ordering of the free nodes, so $i_{t+m} = i_t$ for $m = |\mathcal{U}|$. With t counting individual coordinate updates,

$$p_{i_t}^{t+1} = (1-\alpha)p_{i_t}^t + \alpha g_{i_t}(p^t), \qquad p_k^{t+1} = p_k^t \quad (k \neq i_t), \tag{6}$$

where $0 < \alpha \le 1$. This is a Gauss–Seidel coordinate-descent update (Seidel, 1873; Wright, 2015): $\alpha = 1$ exactly minimizes the active coordinate, while $\alpha < 1$ moves partway to its minimizer.

Proposition 1 (convergence of the sequential update)**.** *For every finite graph, every $\beta > 0$, and every fixed $0 < \alpha \le 1$, the cyclic iterates equation 6 converge to a stationary point of the mean-field free energy equation 5. No contraction condition on B is needed. If*

$$4 + 2\beta\underline{\lambda} > 0, \tag{7}$$

then the free energy is strongly convex, its stationary point is unique, and the sequential update converges to this unique global minimizer from every initialization.

Proof. With every coordinate except i fixed, write $\phi_i(u) = \mathcal{F}(p_{-i}, u)$. The expected count energy is affine in the Bernoulli mean u, whereas the Bernoulli KL term satisfies

$$\phi_i''(u) = \frac{1}{u(1-u)} \ge 4.$$

Thus ϕ_i is 4-strongly convex and its unique minimizer is $g_i(p)$. Strong convexity, applied to the damped step in equation 6, gives

$$\mathcal{F}(p^t) - \mathcal{F}(p^{t+1}) \ge 2\alpha(2-\alpha)\left|g_{i_t}(p^t) - p_{i_t}^t\right|^2 = \frac{2(2-\alpha)}{\alpha}\left|p_{i_t}^{t+1} - p_{i_t}^t\right|^2. \tag{8}$$

Since $\mathcal{F}$ is bounded below, the squared coordinate changes are summable and hence vanish. Bounded logits also place all coordinates in a compact subinterval of $(0,1)$ after one sweep, so every accumulation point satisfies $p_i = g_i(p)$ for all i. Moreover, on this compact box $\mathcal{F}$ is analytic with bounded Hessian, and the gradient after a sweep is controlled by that sweep's coordinate changes. The Kurdyka–Łojasiewicz convergence framework of Attouch et al. (2013) then applies: the sufficient-decrease estimate equation 8, the relative-error bound, and the Kurdyka–Łojasiewicz property of the real-analytic free energy imply that the sweep sequence has finite length and converges to a stationary point. Finally,

$$\nabla^2\mathcal{F}(p) = \operatorname{diag}\left(\frac{1}{p_i(1-p_i)}\right) + 2\beta\mathcal{B}(p) \succeq (4 + 2\beta\underline{\lambda})I.$$

Condition equation 7 therefore makes the free energy strongly convex, proving uniqueness and global optimality of the limit. □

Condition equation 7 only asks the entropy curvature $4I$ to dominate the negative interaction curvature $2\beta\underline{\lambda}$. A conservative sufficient condition is $4 - 2\beta\rho(B) > 0$. If the condition fails, the sequential update still converges, but multiple stationary points can make its limit initialization- and order-dependent.

Dataset check. At $\beta = 0.7$, all eight initializations converge on each of the three splits, yielding eight distinct stationary points per split at an ℓ_∞ clustering tolerance of 10^{-6}. The largest fixed-point residual is 9.37×10^{-11}; the largest pairwise ℓ_∞ distance is 0.976. Thus the stationary point is empirically non-unique in this regime.

The stationary points also differ in free energy. On every split, initialization at the prior converges to a point with higher free energy than another tested initialization. Thus convergence does not ensure selection of the best available variational solution, even for this scalar reference.

A.3 Parallel updates

Our reference, like a GNN layer, updates every node from the same previous state. Although each g_i minimizes one coordinate, simultaneous changes interact through the state-dependent curvature $2\beta\mathcal{B}(p)$, so $\mathcal{F}$ need not decrease. Closely related sequential and parallel mean-field dynamics have been shown to differ even for two variables, with parallel updates exhibiting periodic behavior in regimes where sequential updates do not (Plummer et al., 2020). For two symmetrically coupled variables with exact clues and $B = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, the undamped Jacobian at a symmetric fixed point has eigenvalues $\pm 2\beta p^\star(1 - p^\star)$; beyond unit magnitude, one mode expands and the other alternates.

By the contraction-mapping theorem (Ortega & Rheinboldt, 2000), a sufficient condition for the parallel update to converge globally is $\beta\rho(B) < 2$. Indeed, $|Dg(p)| \le (\beta/2)B$ entrywise, so the Euclidean Lipschitz constant of F_α is at most $1 - \alpha + \alpha\beta\rho(B)/2$. This sufficient condition is worst-case; local certificates can also be constructed at particular fixed points.

Local state-dependent certificate. At any fixed point $p^\star$, let $D^\star = \mathrm{diag}(p^\star(1 - p^\star))$. The derivative of the parallel map is $(1 - \alpha)I - 2\alpha\beta D^\star\mathcal{B}(p^\star)$. Because $D^\star\mathcal{B}(p^\star)$ is similar to $(D^\star)^{1/2}\mathcal{B}(p^\star)(D^\star)^{1/2}$, its eigenvalues are real. If each such eigenvalue μ satisfies

$$-\frac{1}{2\beta} < \mu < \frac{2 - \alpha}{2\alpha\beta}, \tag{9}$$

then Ostrowski's local-attraction theorem for fixed-point iterations implies that $p^\star$ is locally attracting (Ortega & Rheinboldt, 2000). The upper bound controls oscillatory $\mu > 0$ modes and relaxes as α decreases. The lower bound controls expanding $\mu < 0$ modes and is independent of α, so damping cannot fix $\mu \le -1/(2\beta)$. Saturation may stabilize either type by shrinking $D^\star$.

The lower bound also has an optimization interpretation: it is equivalent to positive definiteness of the free-energy Hessian at $p^\star$. A negative-curvature direction therefore cannot be stabilized by damping. Even at a strict local minimum with positive-definite Hessian, however, parallel updates need sufficient damping to avoid oscillatory instability.

Dataset check. We compute the stationary point $p^\star$ returned by the sequential solver initialized at $p^0 = p_{\mathrm{mine}}\mathbf{1}$, and evaluate equation 9 using $\mathcal{B}(p^\star)$. We report the stationary residual, the damping allowance, and the spectral radius of the parallel Jacobian. The parallel trajectory from the same initialization is checked separately.

At $\beta = 0.7$, $\rho(B)$ is approximately 23, 27, and 24 on the three splits, so the global contraction bound does not hold. The stationary residuals are below 8.9×10^{-11}; the local damping upper bounds are 0.648, 0.655, and 0.598, respectively.

At $\alpha = 0.4$, the corresponding Jacobian spectral radii are 0.983, 0.986, and 0.985, satisfying the local-attraction condition. For $(\alpha, \beta) = (0.4, 0.7)$, the parallel trajectory has a maximum fixed-point residual of 2.04×10^{-9} after 1024 iterations, while its maximum ℓ_∞ distance from the sequentially obtained stationary point remains 0.936. The two update schedules can therefore approach different stationary points even from the same initialization.

A.4 Backtracking updates

The backtracking ablation retains the receiving variable in each two-hop sum. Let $H \in \{0, 1\}^{|\mathcal{O}| \times |\mathcal{U}|}$ be the clue-to-free-node incidence matrix, $H_{ji} = \mathbf{1}\{i \in N(j)\}$, and define

$$G = H^\top H = B + \mathrm{diag}(H^\top H).$$

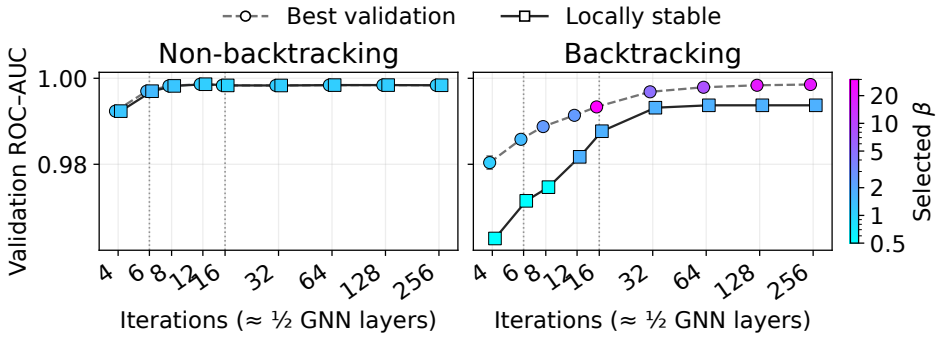

Figure 6: Comparison between the best-validation α-β pairs per iteration and locally certified stable α-β pairs. Iterations 6 to 16 correspond to GNN depths 12 to 32.

The backtracking ablation uses the same expected energy difference, with the receiver included in $S_j = \sum_{k \in N(j)} X_k$:

$$r_{\mathrm{bt},i}(p) = \sum_{j \in \mathcal{O} \cap N(i)} \mathbb{E}_{q_p}[\Delta E_j(S_j)], \quad g_{\mathrm{bt}}(p) = \sigma(\ell_{p_{\mathrm{mine}}} \mathbf{1} - \beta r_{\mathrm{bt}}(p)), \quad F_{\alpha,\mathrm{bt}}(p) = (1-\alpha)p + \alpha g_{\mathrm{bt}}(p).$$

This is a receiver-inclusion ablation, rather than coordinate minimization of equation 5. Let $C(p) = Dr_{\mathrm{bt}}(p)$. For $i, k \in \mathcal{U}$, including $i = k$,

$$C_{ik}(p) = \sum_{j \in \mathcal{O}:\, i,k \in N(j)} \mathbb{E}_{q_p}[\Delta^2 E_j(S_{j \setminus k})].$$

Thus $0 \le C(p) \le 2G$ entrywise. The matrix need not be symmetric. The condition $\beta\rho(G) < 2$ is sufficient for global contraction and a unique fixed point, but uniqueness outside this regime is not asserted.

Stability of the parallel update. At a fixed point, with $D^\star = \mathrm{diag}(p^\star(1-p^\star))$, the Jacobian is $(1-\alpha)I - \alpha\beta D^\star C(p^\star)$. If ν ranges over the possibly complex eigenvalues of $D^\star C(p^\star)$, a sufficient local-attraction condition is

$$|1 - \alpha(1+\beta\nu)| < 1 \qquad \text{for every } \nu. \tag{10}$$

Equivalently, every eigenvalue must satisfy $\mathrm{Re}(1+\beta\nu) > 0$ and $\alpha < 2\,\mathrm{Re}(1+\beta\nu)/|1+\beta\nu|^2$. For exact clues $C = 2G$, giving the bound $\alpha < 2/(1+2\beta\mu_{\max})$, where $\mu_{\max}$ is the largest eigenvalue of $(D^\star)^{1/2} G (D^\star)^{1/2}$.

We evaluate the parallel updates (non-backtracking and backtracking) on the three splits, starting from $p^0 = p_{\mathrm{mine}}\mathbf{1}$, in Figs. 6 and 1. We tune α from 0.001 to 1.0 in 16 values and β from 0.5 to 100 in 22 values. Settings are selected by mean validation ROC–AUC across the three splits, separately at each iteration budget.

Increasing β strengthens the count constraints and changes the Jacobian in equation 10. Smaller α reduces the size of each update and can suppress oscillations, but may require more iterations to reach high accuracy. We compare accuracy at fixed iteration budgets and the additional iterations needed for a locally stable configuration to match the unrestricted configurations' validation scores. At 6, 8, 12, and 16 iterations, the best-validation non-backtracking settings attain mean test ROC–AUCs of 99.8%-99.9%, and satisfy the numerical local-stability check on every split. For backtracking, the corresponding unrestricted scores are 98.8%, 99.1%, 99.3%, and 99.4%, compared with 97.5%, 97.8%, 98.4%, and 99.0% under the local-stability restriction. Instead, the locally stable backtracking setting $(\alpha, \beta) = (0.1, 1.75)$ matches the unrestricted validation scores later, after 15, 18, 23, and 35 iterations, respectively.

Reference without clamping. We replace hard clamping by a finite training-label penalty $\gamma \sum_{i \in T} \mathbb{E}_{q_p}[(X_i - y_i)^2] = \gamma \sum_{i \in T} [y_i(1-p_i) + (1-y_i)p_i]$. All node probabilities are now free

Table 3: Mean test ROC–AUC (%) over three benchmark splits. All settings use the same grids $\alpha \in \{0.03, 0.1, 0.2, 0.4\}$ and $\beta \in \{0.5, 0.7, 2, 10\}$, selected by validation ROC–AUC at each iteration budget. Soft supervision additionally selects γ by validation; the zero-penalty control fixes $\gamma = 0$.

Reference	Label treatment	Iterations			
		4	8	16	256
Non-backtracking	Hard clamping	99.4	99.8	99.8	99.8
	Soft (validation-selected γ)	98.3	99.6	99.8	99.8
	None ($\gamma = 0$)	92.7	94.9	96.2	96.6
Backtracking	Hard clamping	98.1	99.0	99.4	99.9
	Soft (validation-selected γ)	97.2	98.4	99.3	99.9
	None ($\gamma = 0$)	91.5	93.9	95.3	98.1

and initialized at p_{mine}. Defining both fields over all nodes, the parallel update becomes

$$p^{t+1} = (1-\alpha)p^t + \alpha\,\sigma\big(\ell_{p_{\mathrm{mine}}}\mathbf{1} - \beta r_{\mathrm{ref}}(p^t) + \gamma s\big), \qquad s_i = \mathbf{1}_{\{i\in T\}}(2y_i - 1),$$

where r_{ref} is r or r_{bt}, respectively, with no clamped variables. The label penalty is linear in p, so the local stability conditions retain their form, evaluated using the full-node field derivative at the corresponding stationary point.

For soft supervision we tune $\gamma \in \{1, 5, 20\}$ using validation ROC–AUC; the zero-penalty control fixes $\gamma = 0$. Table 3 compares hard clamping, soft supervision, and the zero-penalty control at matched iteration budgets. With $\gamma = 0$, no training labels enter initialization or inference. Validation selects $\gamma = 20$ for both references at all four reported budgets. Hard clamping gives higher mean test ROC–AUC than soft supervision at 4 and 8 iterations; the two are within 0.2 percentage points at 16 and 256 iterations. Without training labels, the 256-iteration scores are 96.6% and 98.1% for non-backtracking and backtracking, respectively.

A.5 Fitting α and β to GCN outputs

For the GCN with depth $L = 16$, width 512, and weight decay 10^{-4}, we fit the non-backtracking reference endpoint to the GCN's output probabilities. We show the fitted values and fitting quality across epochs for split 0 in Figure 7 (all splits present similar behavior).

The analysis covers three benchmark splits and 153 measurements from epoch 0 to 4999. The GCN is matched to $K = L/2 = 8$ reference iterations, each comprising two aggregation steps. The reference starts from $p_{\mathrm{mine}} = 0.2$ on free nodes and clamps the training labels.

Writing z_i for the GCN's output probability and $p_i^K(\alpha, \beta)$ for the reference endpoint, we approximately minimize

$$Q(\alpha, \beta) = \frac{1}{|\mathcal{U}_{\mathrm{fit}}|} \sum_{i\in\mathcal{U}_{\mathrm{fit}}} \big[p_i^K(\alpha, \beta) - z_i\big]^2.$$

The fitting subset $\mathcal{U}_{\mathrm{fit}}$ consists of non-training nodes and is fixed across epochs. We quantify endpoint agreement on these nodes using probability RMSE and

$$R^2_{\mathrm{fit}} = 1 - \frac{\sum_{i\in\mathcal{U}_{\mathrm{fit}}}[p_i^K(\hat\alpha, \hat\beta) - z_i]^2}{\sum_{i\in\mathcal{U}_{\mathrm{fit}}}(z_i - \bar z_{\mathrm{fit}})^2}, \qquad \bar z_{\mathrm{fit}} = \frac{1}{|\mathcal{U}_{\mathrm{fit}}|}\sum_{i\in\mathcal{U}_{\mathrm{fit}}} z_i.$$

When fitting the backtracking reference, it yields a lower $60.5 \pm 1.2\%$ explained variation and higher probability RMSE of 0.227 ± 0.016 at the final epoch.

The fitted parameters describe endpoint agreement and do not establish the GCN's intermediate dynamics. However, elucidating learning mechanics in GNNs is a difficult and open-ended question, and we present one such experimental strategy to address it. We do not claim that GNNs exactly follow the inference dynamics of our reference, but rather suggest it as an explicit tool to improve our understanding of them.

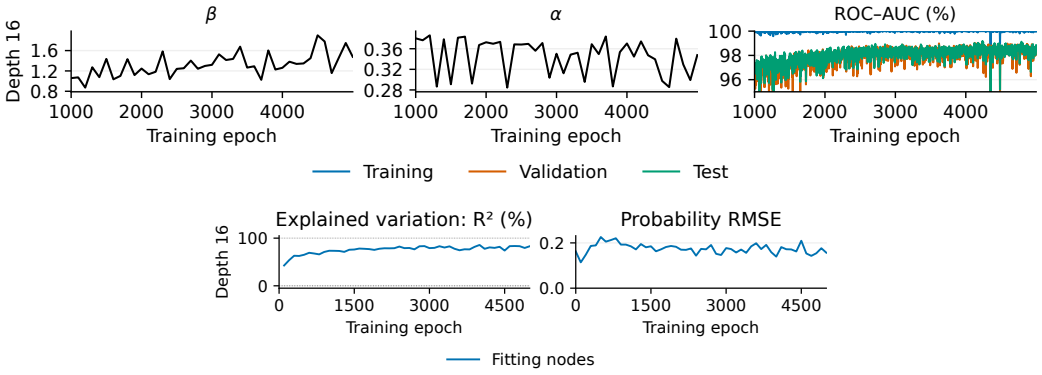

Figure 7: Fitted $(\hat{\alpha}, \hat{\beta})$ through training of a GCN with depth 16, width 512, and weight decay 1e$-$4.

B MEASUREMENT DEFINITIONS

Let $H^\ell \in \mathbb{R}^{n \times d_\ell}$ denote the node-feature matrix after GCN layer ℓ. The representations are measured after the convolution, residual connection and normalization, when present, and the ReLU activation. Unless otherwise specified, all measurements are computed in evaluation mode at the checkpoint with the highest validation ROC-AUC. Last-layer measurements use H^L, immediately before the prediction head. Layer averages exclude the input features and average over the L GCN layers.

B.1 NORMALIZED DIRICHLET RAYLEIGH QUOTIENT

Let A and D denote the adjacency and degree matrices of the undirected graph without self-loops, and let $L = D - A$, $\widetilde{A} = A + I$, $\widetilde{D} = D + I$. The edge list used by the implementation contains both directions of every undirected edge and one self-loop per node. For node representation h_i^ℓ, we compute

$$\mathrm{Dir}(H^\ell) = \frac{\sum_{(i,j)\in\mathcal{E}^{\leftrightarrow}} \left\| h_i^\ell - h_j^\ell \right\|_2^2}{\sum_{(i,j)\in\mathcal{E}^{\leftrightarrow}} \left(\left\| h_i^\ell \right\|_2^2 + \left\| h_j^\ell \right\|_2^2 \right)}.$$

Because non-self edges occur in both directions, this ratio is equivalently

$$\mathrm{Dir}(H^\ell) = \frac{\mathrm{tr}\left((H^\ell)^\top L H^\ell \right)}{\mathrm{tr}\left((H^\ell)^\top \widetilde{D} H^\ell \right)}.$$

This is a scale-invariant, multi-feature graph Rayleigh quotient (Cai & Wang, 2020; Giovanni et al., 2023). To see its relationship with the conventional normalized Dirichlet energy, define $X^\ell = \widetilde{D}^{1/2} H^\ell$, $\widetilde{\Delta} = I - \widetilde{D}^{-1/2} \widetilde{A} \widetilde{D}^{-1/2}$. Then

$$\mathrm{Dir}(H^\ell) = \frac{\mathrm{tr}\left((X^\ell)^\top \widetilde{\Delta} X^\ell \right)}{\left\| X^\ell \right\|_F^2}.$$

For a regular graph, this reduces directly to the normalized Dirichlet energy of H^ℓ. Note that Minesweeper is a grid and thus almost a regular graph. Smaller values indicate that adjacent representations are more similar relative to their magnitudes. Self-loops contribute zero to the numerator and contribute to the normalization in the denominator.

We primarily report the last-layer value $\mathrm{Dir}_{\mathrm{last}} = \mathrm{Dir}(H^L)$. As a supplementary summary, we also report $\mathrm{Dir}_{\mathrm{avg}} = \frac{1}{L} \sum_{\ell=1}^{L} \mathrm{Dir}(H^\ell)$.

B.2 Effective hidden rank

We use the effective rank of the hidden-feature matrix (Zhang et al., 2026a). We select $4{,}096$ approximately evenly spaced node indices for efficiency. Let H_S^ℓ denote the resulting uncentered feature matrix, and let $\sigma_1, \ldots, \sigma_r$ be its singular values. Defining $p_k = \frac{\sigma_k}{\sum_{j=1}^r \sigma_j}$, the effective rank is

$$\mathrm{Erank}(H_S^\ell) = \exp\left(-\sum_{k=1}^{r} p_k \log p_k\right),$$

where $0 \log 0 = 0$. The effective rank approaches one when a single singular direction dominates and increases as the representation uses more singular directions. Our primary hidden-rank measurement is $\mathrm{Hidd}_{\mathrm{last}} = \mathrm{Erank}(H_S^L)$. We additionally report $\mathrm{Hidd}_{\mathrm{avg}} = \frac{1}{L}\sum_{\ell=1}^{L} \mathrm{Erank}(H_S^\ell)$. Note that these measurements will vary with width and depth, so absolute values have to be compared carefully.

B.3 Stable rank of the message map

Let $W^\ell \in \mathbb{R}^{d_{\ell-1} \times d_\ell}$ denote the learned channel-mixing matrix of GCN layer ℓ. Before aggregation, the message sent from node j to node i has the form $m_{j\to i}^\ell = \widehat{A}_{ij} h_j^{\ell-1} W^\ell$. Because $\widehat{A}_{ij}$ is a scalar, a rank-one W^ℓ forces every transformed message to lie in a one-dimensional channel subspace, independently of the hidden features. This is the GCN analogue of the scalar message map in our constructed solver.

We measure how closely the learned map approaches rank one using its stable rank,

$$\mathrm{Srank}(W^\ell) = \frac{\|W^\ell\|_F^2}{\|W^\ell\|_2^2} = \frac{\sum_k \sigma_k(W^\ell)^2}{\sigma_1(W^\ell)^2}.$$

A value near one indicates that the message map is dominated by one singular direction. A larger value indicates that the map makes additional message directions available, although it does not establish that the hidden features use all of them. For stability purposes, any map with $\|W^\ell\|_F \le 10^{-10}$ receives rank zero as it is considered collapsed.

We report $\mathrm{Msg}_{\mathrm{last}} = \mathrm{Srank}(W^L)$, $\mathrm{Msg}_{\mathrm{avg}} = \frac{1}{L}\sum_{\ell=1}^{L} \mathrm{Srank}(W^\ell)$.

C Relation to existing depth analyses

C.1 Expressivity, receptive field, and sequential computation

Cong et al. (2021) provide a theoretical starting point for our study. They show that depth does not intrinsically reduce the expressivity of a standard GCN, give a convergence guarantee under an overparameterization condition, and analyze the transductive generalization gap through uniform stability. They then propose Decoupled GCN (DGCN), which retains access to large receptive fields while reducing the dependence of their generalization bound on successive learned transformations. We focus on a different question: when can a practical finite-width GCN learn a generalizing computation that interleaves propagation with nonlinear transformations across depth?

Let $P \in \mathbb{R}^{n\times n}$ denote the normalized propagation matrix and $X \in \mathbb{R}^{n\times d}$ the input features. Defining

$$B_\ell = \alpha_\ell \left(\beta_\ell W^{(\ell)} + (1-\beta_\ell) I\right), \qquad B = [B_1^\top, \ldots, B_L^\top]^\top,$$

DGCN can be written as

$$Z_{\mathrm{DGCN}}(X) = \sum_{\ell=1}^{L} \alpha_\ell P^\ell X \left(\beta_\ell W^{(\ell)} + (1-\beta_\ell) I\right). \tag{11}$$

Before its prediction head, DGCN therefore applies a learned linear transformation to a fixed multiscale propagation bank. This places it close to decoupled fixed-propagation methods such as SGC

(Wu et al., 2019) and SIGN (Frasca et al., 2020), and makes it structurally analogous to fixed aggregation features (FAF) (Rubio-Madrigal & Burkholz, 2026). DGCN exposes multiple propagation radii but does not propagate intermediate nonlinear states.

In contrast, omitting dropout, our residual GCN applies

$$H^{(t+1)} = \phi\Big(\mathrm{Norm}^{(t)}\left(PH^{(t)}W^{(t)} + H^{(t)}W_r^{(t)}\right)\Big),$$

so each propagation step acts on the output of the preceding nonlinear residual block. This sequential computation cannot generally be reduced to a linear transformation of $\{P^\ell X\}_{\ell=1}^{L}$. Our reference similarly computes $p^{(T)} = \Phi_c^{\circ T}(p^{(0)})$ by repeatedly applying a nonlinear, task-dependent update.

This distinction is consequential on Minesweeper. Recent FAF variants obtain approximately 90% ROC-AUC under extensive tuning, substantially below carefully tuned nonlinear GNNs. DGCN performs similarly: our best 12-layer, width-64 run, selected with weight decay $1\mathrm{e}{-4}$, obtains $87.98\% \pm 0.37$ test ROC-AUC. Access to several propagation radii is therefore insufficient to reproduce the task-specific iterative computation.

The theoretical results of Cong et al. (2021) distinguish expressivity, optimization, and generalization, but do not resolve the failures studied here. Their WL-based result characterizes which neighborhoods an appropriately parameterized GCN can distinguish in principle, without guaranteeing that training finds such a parameterization. Their convergence guarantee assumes square loss and a final hidden width $d_L \geq N$, whereas our models use $d_L \leq 512$ for $N = 10{,}000$. Their transductive stability analysis bounds the training–test gap by measuring sensitivity to exchanging a labeled training node with a test node, but does not characterize solution selection. Together, these results do not explain why some tested GCN configurations fit poorly, or why configurations with similarly high training ROC–AUC generalize differently on Minesweeper. Our explicit reference establishes task-level achievability, while our model grid identifies these outcomes empirically.

C.2 Node-adaptive scope selection

Deng et al. (2025) propose Moscat, a test-time mixture of GNN experts trained at different depths whose predictions are combined by a node-adaptive gate. Their PAC-Bayesian subgroup analysis predicts that the depth minimizing the generalization bound varies with a subgroup's homophily relative to the training distribution, with larger disparities on heterophilous graphs.

Moscat and our work address different responses to depth-dependent generalization. Moscat combines separately trained shallow and deep experts, showing that predictions obtained at different propagation scopes can be complementary. Its theoretical analysis considers fixed L-hop mean-aggregated features followed by an MLP, so depth changes propagation scope rather than the number of sequential nonlinear computation steps. Our reference and parameter-tied models instead concern repeated nonlinear computation within a single model, as mentioned above.

Empirical comparison. Because Moscat uses held-out labeled nodes to train its gate, we evaluate two label allocations. In the first, Moscat reserves 20% of the training nodes for gate training, while the comparison GCN uses the full training set. In the second, Moscat trains its gate on 90% of the validation nodes and evaluates validation performance on the remaining 10%; the comparison GCN is trained on the original training nodes plus the same 90% of validation nodes. Figure 8 reports validation and test ROC-AUC at depths 12 and 16 for the standard 2500 epochs.

Across both settings, allocating labels to the additional gate does not improve over using the same available labels to train one well-tuned GCN. The mixture also requires training several experts and a gate: at width 512 and depth 16, average runtime on an NVIDIA A100 increases from approximately 5 minutes for the single GCN to 45 minutes for Moscat. This result does not contradict Moscat's node-adaptive objective; but it shows that, on Minesweeper, improving the training regime of one deep model is more effective than post-hoc scope selection.

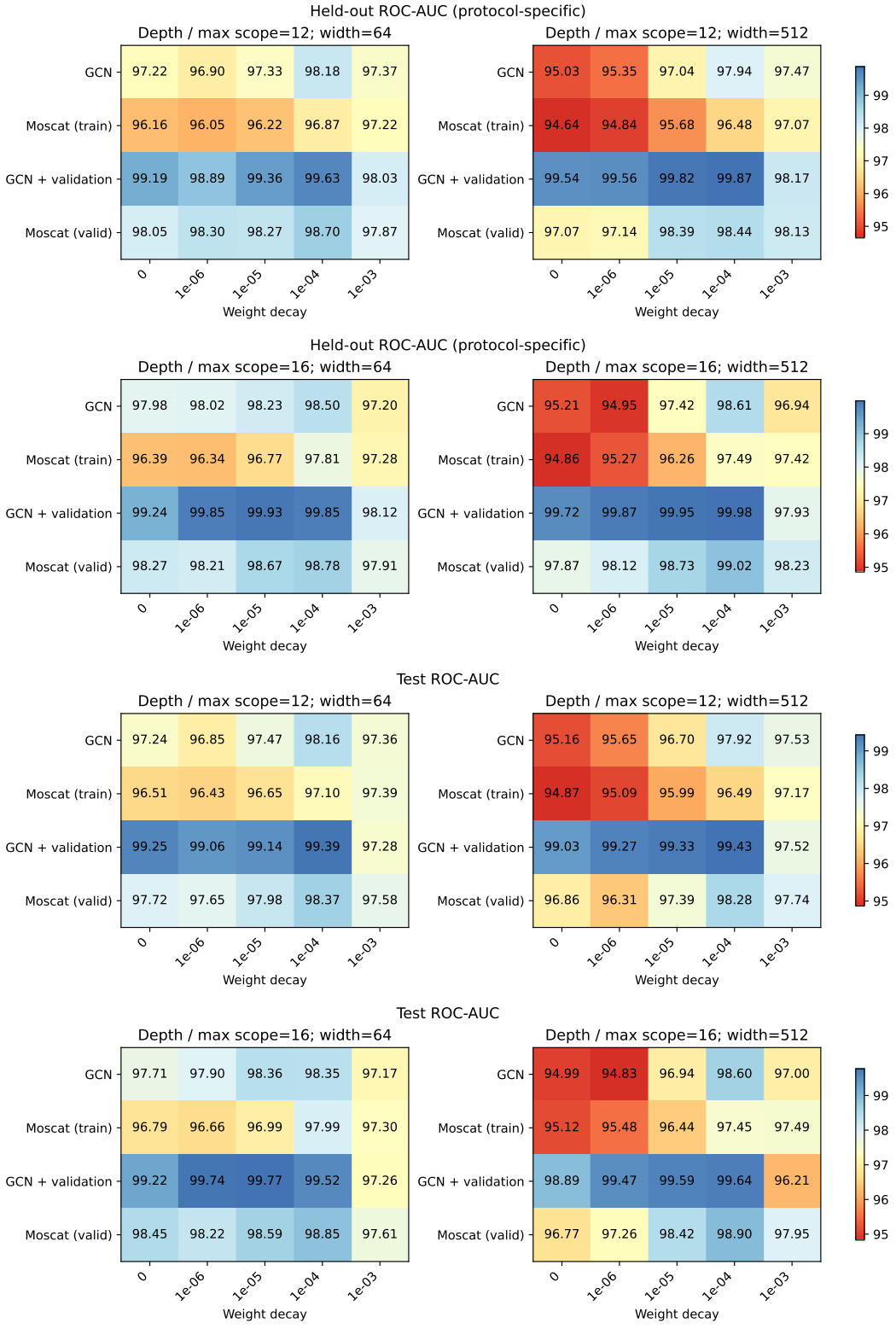

Figure 8: Comparison of Moscat (Deng et al., 2025) and a single GCN at depths 12 and 16 for 2500 epochs. Top reports validation ROC-AUC, and bottom reports test ROC-AUC. We consider two gate-training sets: 20% of the training nodes or 90% of the validation nodes. In each setting, the single-GCN comparison uses all nodes available to the Moscat experts and gate for model training.

C.3 Collapse measurements and predictive performance

Zhang et al. (2026a) study which representation measurements identify over-smoothing in finite, trained GNNs. They show that absolute Dirichlet-like energies depend on feature scale and can remain far from their asymptotic limit after predictive performance has deteriorated. They instead advocate continuous rank relaxations, particularly numerical and effective rank, which are scale invariant and do not require a predetermined dominant eigenspace. Their theory establishes convergence of numerical rank to one for linear GCNs and for a nonnegative nonlinear setting under stated conditions.

Arnaiz-Rodriguez & Errica (2026) similarly challenge universal claims about over-smoothing. Through conceptual counterexamples, they show that collapse depends on architecture, feature scaling, and measurement: small changes can reverse the trend of Dirichlet energy, and raw energy can disagree with its normalized counterpart. Their central methodological point is that representation collapse, loss of class separability, and decreasing predictive performance are distinct claims.

Our measurement design follows these critiques. We report a scale-invariant Dirichlet Rayleigh quotient together with effective hidden rank, and compare independently trained models rather than treating the layers of one untrained network as a depth sweep. Our results support the limitations of Dirichlet-like quantities while also qualifying rank as a practical but not complete diagnostic. Across our residual-GCN grid, validation performance can deteriorate while effective rank either decreases or increases. Weight decay can improve training and validation performance even as hidden rank decreases further. Decreasing rank can therefore indicate harmful collapse, removal of poorly generalizing variation, or a useful low-dimensional bias. Conversely, growing rank can accompany poor generalization even when training performance is unreliable.

Neither Dirichlet energy nor hidden rank therefore identifies a failure mechanism by itself. Their interpretation requires training and validation performance, variability across splits, and the depth–width–regularization regime in which they arise. This also complements a boundary case identified by Zhang et al. (2026a): when residual GCN performance remains stable, all their tested measurements, including rank, correlate poorly with accuracy. We study a residual, normalized architecture in a regime where depth degradation remains and intervene directly on width and weight decay.

C.4 Implications for fixed-aggregation comparisons

Fixed aggregation features are important baselines, particularly because many benchmarks can be addressed without learned message passing (Rubio-Madrigal & Burkholz, 2026). However, a performance gap supports a claim about the model class only when the learned baseline is evaluated under an appropriate training regime. For example, Bazhenov et al. (2025) tune learning rate and dropout for their GNNs, but hold depth and width fixed and do not report a search over weight decay. Their strong neighborhood feature aggregation (NFA) results show that fixed one-hop statistics can be highly competitive, but do not determine whether a remaining gap arises from learned message passing itself or from the depth, width, and regularization settings used to train it.

D Experimental results

Figure 9 compares validation, test, and training performance across the widths and depths of five models: an unrestricted GCN, a parameter-tied GCN, a rank-one, parameter-tied GCN, and a rank-two, parameter-tied GCN. Figures 10 and 11 show layer-wise measurements of the variants for selected configurations. Figure 12 shows epoch-wise evolution of representation measurements for the four selected configurations in the main text.

Appendix D.1 reports training, validation, and test ROC-AUC at the checkpoint selected by validation ROC-AUC, together with last-layer and layer-average representation measurements.

The remaining hyperparameters follow Luo et al. (2024): three splits, Adam with learning rate 0.01, dropout 0.2, BatchNorm, a learnable residual connection, a two-layer MLP predictor, 5000 training epochs, and validation-based checkpoint selection. We add a pre-linear layer that transforms the narrow input features into the width of the shared convolutional matrix; Luo et al. (2024) also tunes

this hyperparameter, but does not find performance differences for Minesweeper. Note that, in the tied architectures, normalization and residual parameters are also shared across layers.

Weight decay and learning rate are often coupled hyperparameters: larger weight decay may benefit from larger learning rates. The strength of this interaction, however, depends on the optimizer and parameterization (Smith, 2018); in particular, Adam can substantially reduce this coupling. Conversely, increasing width or depth can impair optimization stability and may require smaller learning rates (Chen et al., 2024), while parameterization also affects trainability (Mustafa et al., 2023; Mustafa & Burkholz, 2024a;b). To limit confounding effects, we therefore fix the learning rate rather than tuning it jointly with the other hyperparameters. In particular, larger learning rates further destabilize training, whereas smaller learning rates do not attain comparable performance within the already generous training budget, likely because of slower convergence.

The weight-decay coefficient should likewise not be interpreted as a one-to-one measure of regularization strength across architectures. For example, tied and untied parameterizations incur different penalties under reuse, while factorized parameterizations penalize $\|u\|^2 + \|v\|^2$, which has different geometry from a penalty on the effective matrix $\|W\|_F^2$. Thus, the same coefficient can induce different effective regularization in different models. Although we use matched coefficients in one of the selected controlled comparisons to make specific points about the effect of architectural changes, this should not be taken to imply that the corresponding models receive identical effective regularization.

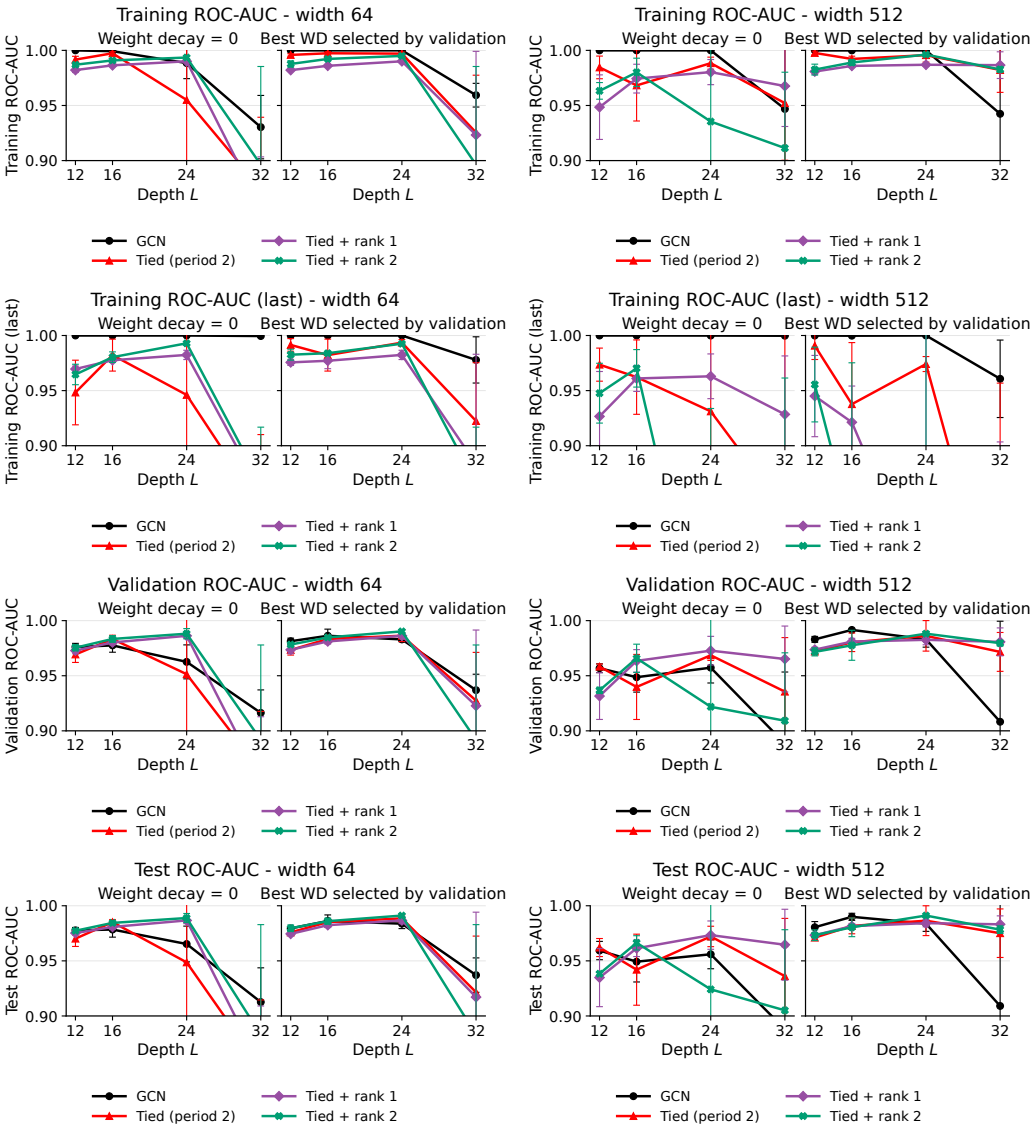

Figure 9: Performance plots for widths 64 (left) and 512 (right) across different depths of a GCN, a parameter-tied GCN, a 1-rank GCN, a 1-rank parameter-tied GCN, and a 2-rank parameter-tied GCN.

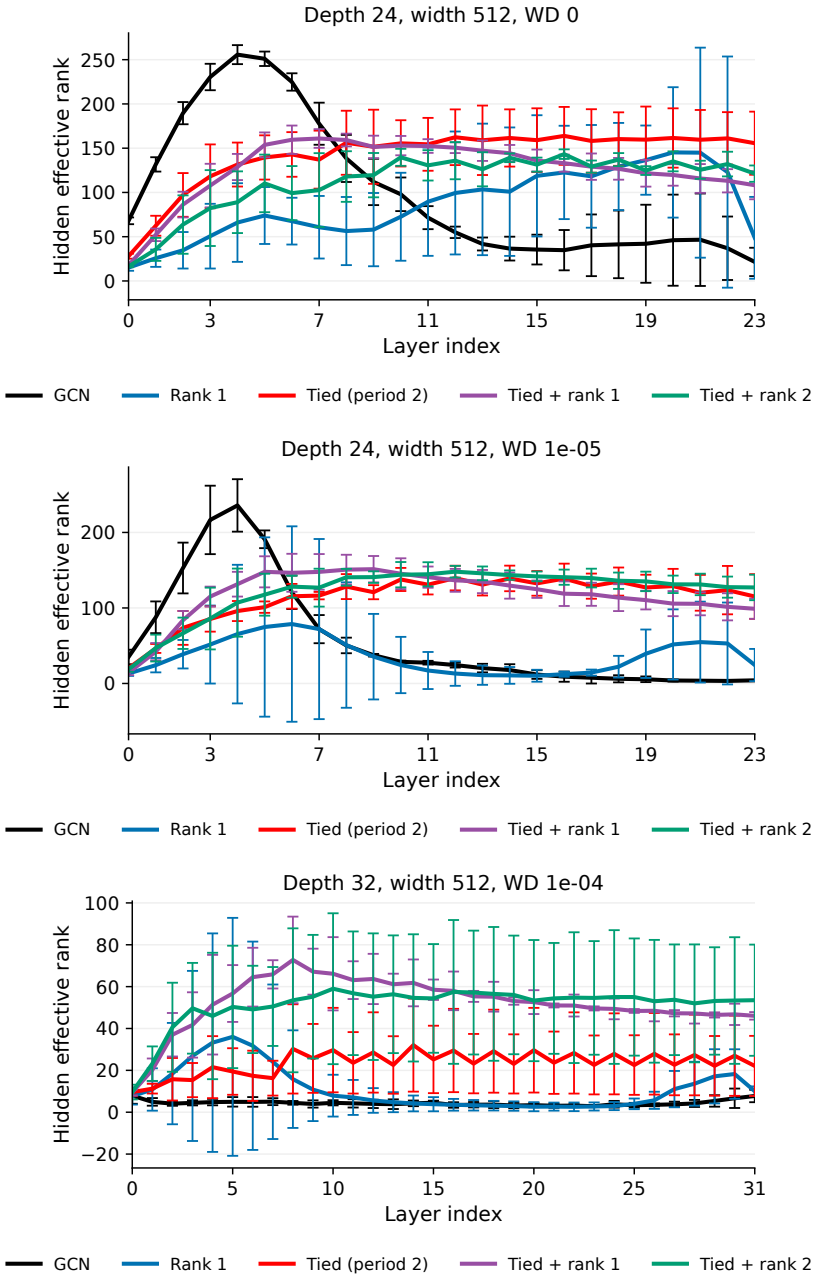

Figure 10: Layer-wise representational geometry of GCN variants: hidden effective rank.

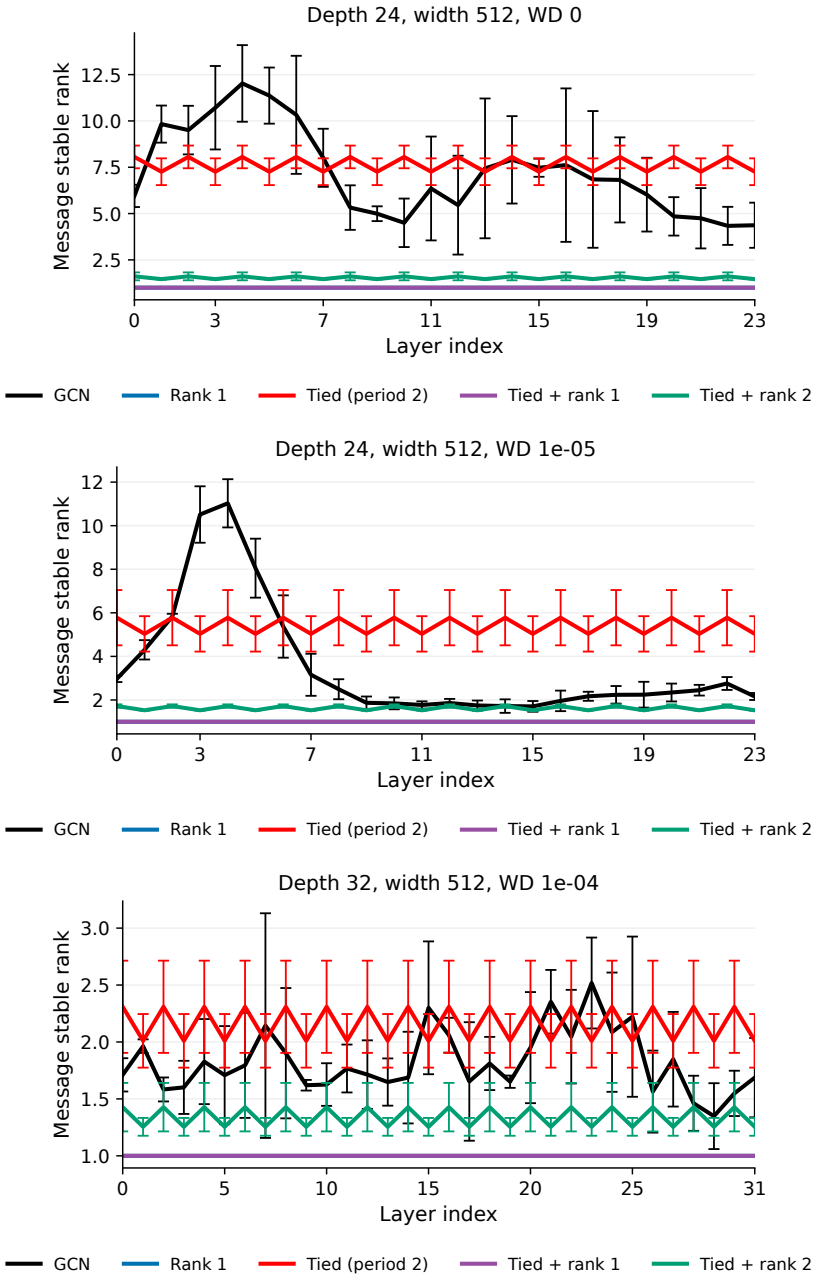

Figure 11: Layer-wise representational geometry of GCN variants: message-map rank.

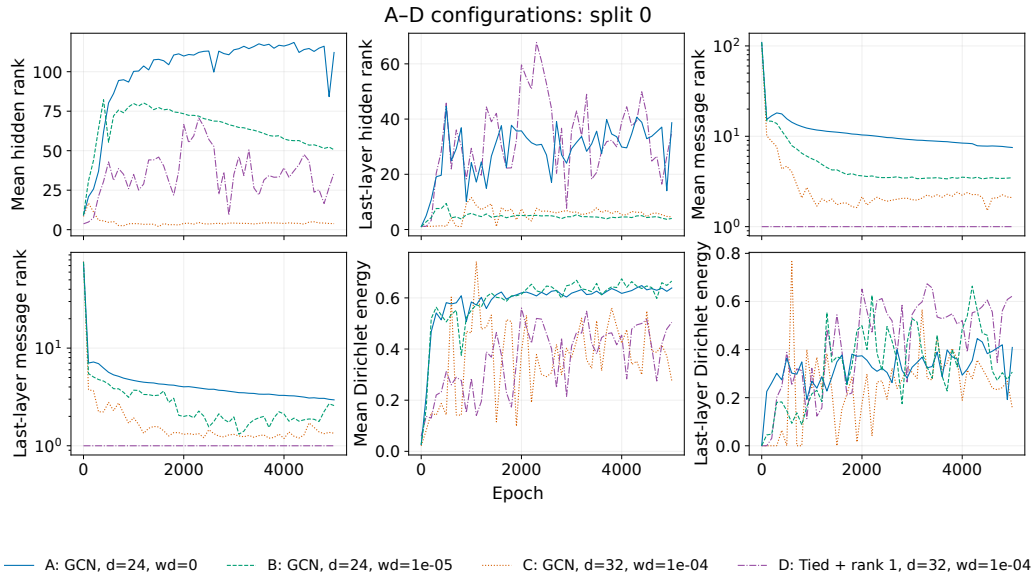

Figure 12: Representation geometry metrics across epochs for split 0 of the A-D configurations in § 4.3.

D.1 GCN HEATMAPS

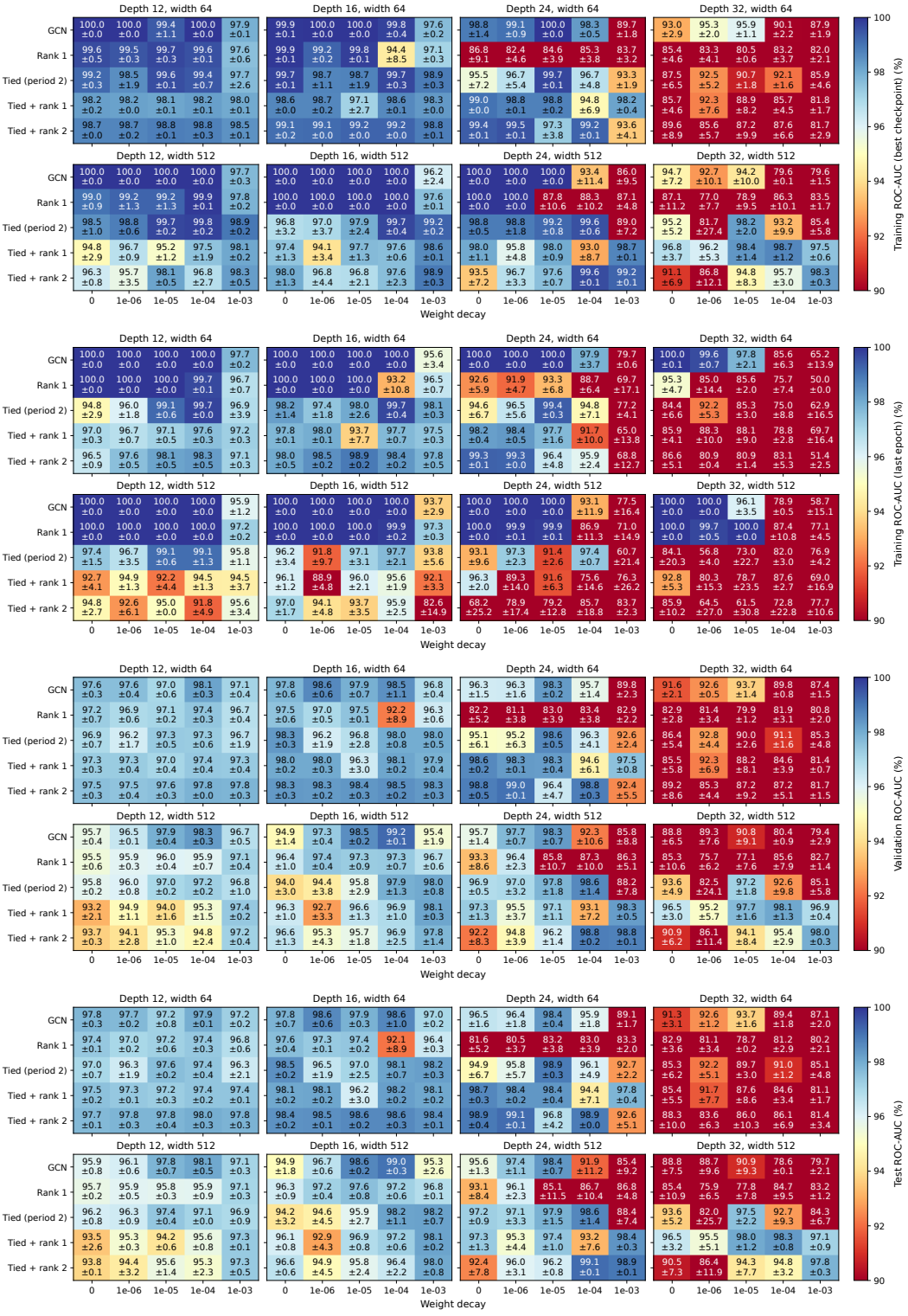

Figure 13: GCN performance on Minesweeper.

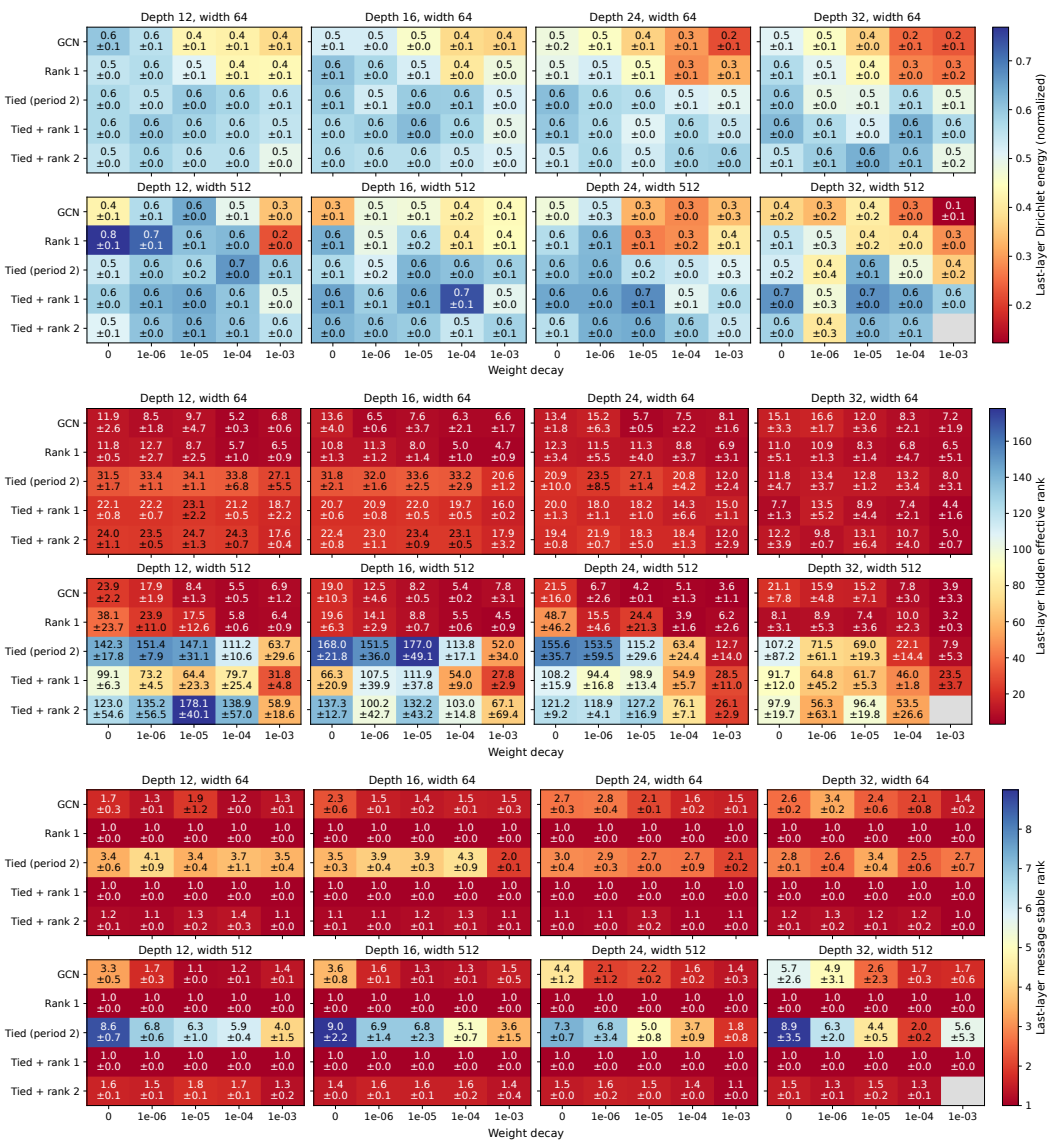

Figure 14: GCN representation metrics on Minesweeper (last layer).

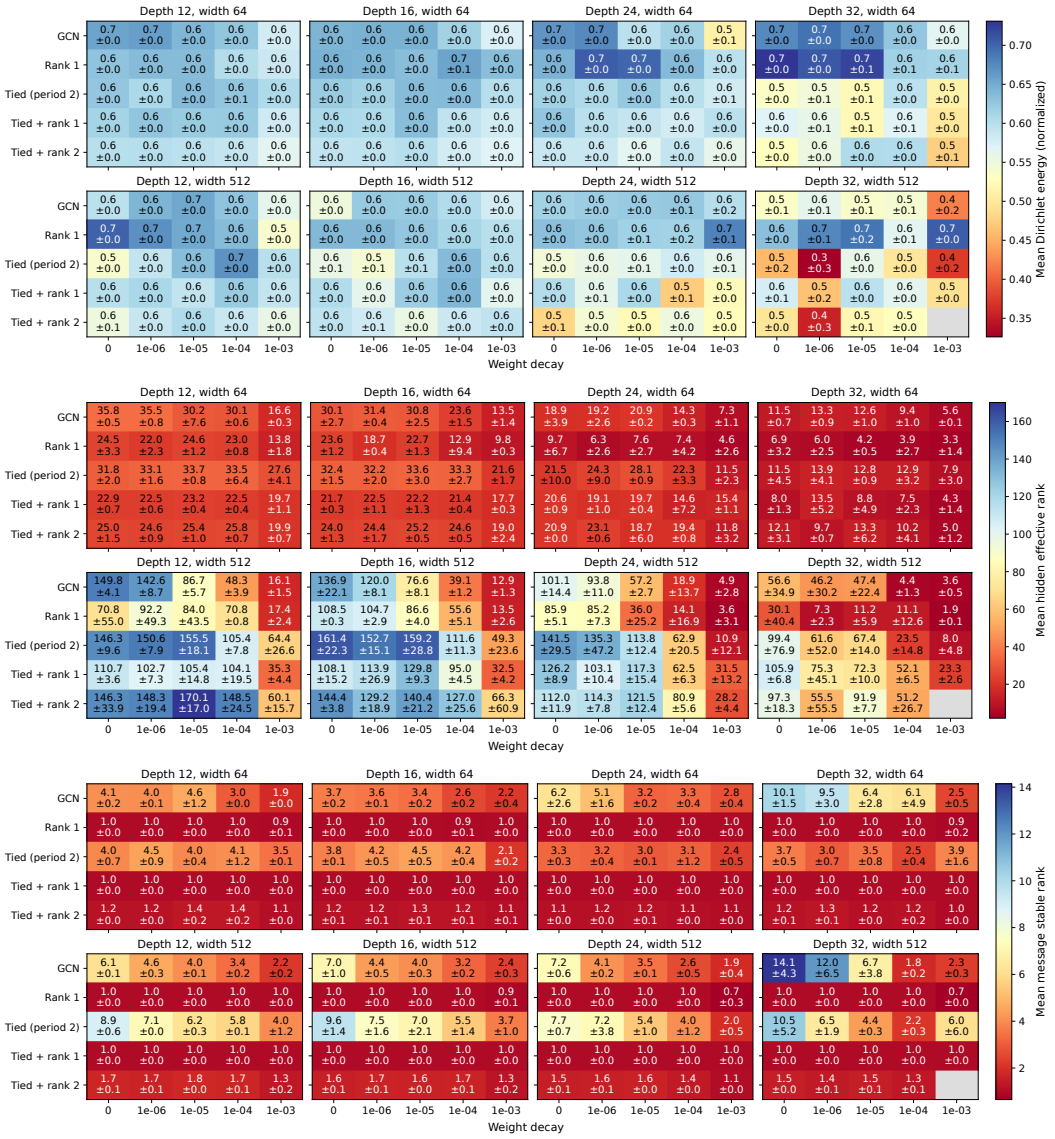

Figure 15: GCN representation metrics on Minesweeper (layer average).