STOP BLAMING AGGREGATION IN DEEP GNNS BASED ON INCOMPLETE EVIDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper argues that aggregation mechanisms in Graph Neural Networks (GNNs) have been unfairly blamed for limiting network depth based on incomplete evidence. The prevailing narrative has led researchers to view aggregation as fundamentally problematic for deep GNNs. However, through reexamining the experimental evidence and theoretical explanations by proper ablations, we show that aggregation's negative effects have been exaggerated while its benefits have been overlooked. Based on our findings, we advocate two actions: First, practitioners should actively deploy deep GNNs on tasks that require large capacity or long-range dependencies, whose potential has already been demonstrated yet remains overlooked. Second, researchers should rigorously investigate the true effects of aggregation on deep GNNs rather than accepting conventional wisdom.

1 INTRODUCTION

Graph Neural Networks (GNNs) have become a central modeling paradigm for structured data, achieving strong empirical performance across a wide range of domains, including social networks (Wang et al., 2019a; Li & Goldwasser, 2019; Huang et al., 2019; Fan et al., 2019), molecules (Gilmer et al., 2017; Zitnik & Leskovec, 2017; Dai et al., 2019), knowledge graphs (Wang et al., 2019b;c; Park et al., 2019), and recommendation systems (Monti et al., 2017; Ying et al., 2018; Jiang et al., 2022; Wu et al., 2022b). A key source of this success lies in their message passing mechanism, which incorporates neighboring information from the graph structure. The message passing mechanism is primarily realized through the aggregation operation, which is one of the three fundamental components of GNN architectures: aggregation, transformation, and activation. Since the transformation and activation operations are structurally identical to those used in multilayer perceptrons (MLPs), aggregation is the defining component that makes GNNs a extension of standard neural networks.

A large body of research has focused on the role of aggregation and its effects on representation learning in GNNs. Early work (Li et al., 2018) has shown that aggregation schemes can be interpreted as a special form of Laplacian smoothing, encouraging neighboring nodes to learn similar representations, which can be beneficial when neighboring nodes share the same label. They also show that, however, the infinitely repeated application of such smoothing results in oversmoothing, a regime in which node representations become indistinguishable. Related phenomena, such as oversquashing (Topping et al., 2022; Deac et al., 2022), have also been introduced to describe limitations arising from information compression during message passing. Together, these studies have provided valuable insights into the mathematical and empirical properties of aggregation in GNNs.

These aggregation-centered explanations, combined with the observation reported in the celebrated GCN paper (Kipf & Welling, 2017) that the best results are obtained with a 2- or 3-layer model, fostered the view that GNNs inherently prefer shallow depth, and performance degrades as depth increases. The subsequent introduction of a demonstration of oversmoothing's exponential occurrence (Oono & Suzuki, 2020; Cai & Wang, 2020; Wu et al., 2023) even reinforces the view by explaining how oversmoothing can affect performance in relatively shallow depth. As a result, the perspective has become widespread across the field, leading researchers to primarily consider shallow architectures when designing GNNs. Indeed, our examination of GNN models reveals that hyperparameter search spaces for network depth are predominantly restricted to fewer than ten lay-

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.

ers (Bo et al., 2021; He et al., 2021; Luo et al., 2024; Pei et al., 2020; Dwivedi et al., 2022; Platonov et al., 2023).

It is important to note, however, that several works have demonstrated the successful construction of deep GNNs. Li et al. (2019) achieved state-of-the-art performance on point cloud semantic segmentation (Armeni et al., 2017) using a 56-layer GNN. Li et al. (2021) trained a 448-layer GNN that achieved the best performance on the large-scale ogbn-proteins dataset (Hu et al., 2020). These results clearly indicate that depth itself is not an insurmountable obstacle for GNNs when properly addressed. Notably, neither work targeted aggregation-induced problems, focusing instead on vanishing gradients and memory efficiency, respectively.

Despite these advances, the narrative that building deep GNNs is fundamentally difficult has persisted. Based on the aggregation-as-obstacle belief, many studies aimed at enabling deeper architectures focus on mitigating oversmoothing (Zhao & Akoglu, 2020; Rong et al., 2020; Chen et al., 2020; Zhou et al., 2021a; Eliasof et al., 2021; Chamberlain et al., 2021; Thorpe et al., 2022; Rusch et al., 2022; Fang et al., 2023; Song et al., 2023; Rusch et al., 2023b; Park et al., 2024a; Su et al., 2024; Pei et al., 2024; Lu et al., 2024). The persistence of this narrative, even in the face of successful deep GNN constructions, reflects the compelling nature of the foundational evidence: empirical observations of shallow-depth optimality coupled with theoretically rigorous aggregation-centered explanations.

However, we argue that a fundamental oversight exists in this evidence: GNNs are, at their core, structured around MLP components. Overfitting in low-capacity regimes and gradient instability, patterns identical to those observed in deep GNNs, are well documented in the broader deep learning literature as inherent to deep MLPs. Since GNNs share the same transformation and activation layers as MLPs, they likely inherit these identical optimization hurdles. Nevertheless, prior work has implicitly blamed aggregation for these challenges without conducting the necessary ablations to isolate message passing from standard MLP dynamics.

Motivated by this observation, we revisit the empirical foundations of deep GNN optimization by disentangling the individual contributions of transformation and activation steps. Through a systematic re-examination of experimental evidence and theoretical frameworks, we demonstrate that the purported negative effects of aggregation have been substantially overstated. Our findings reveal that GNNs actually exhibit greater robustness to increased depth than their MLP counterparts, directly contradicting the prevailing view that aggregation is the primary driver of performance degradation. Theoretically, we show that existing oversmoothing analyses entail a critical yet overlooked implication: learned transformations can actively counteract convergence. Furthermore, we provide evidence that observed exponential oversmoothing is primarily driven by transformation and activation steps, with aggregation's contribution being relatively marginal.

Based on these findings, we argue that aggregation has been unfairly maligned for deep GNN challenges based on incomplete evidence. We advocate two actions. First, practitioners should actively deploy deep GNNs for tasks that require high capacity or long-range dependencies. Second, researchers should rigorously investigate the actual effects of aggregation on optimization and generalization. We present our evidence in Section 3, outline these directions in Section 4, and address credible counterarguments in Section 5.

2 Background

The narrative that aggregation in GNNs inherently limits network depth has been constructed through a series of influential yet incomplete observations. This narrative began with early empirical findings from two highly influential works: Graph Convolutional Networks (GCN) (Kipf & Welling, 2017) and Graph Attention Networks (GAT) (Veličković et al., 2018), both of which reported best performance at shallow depths of 2-3 layers on the Cora, CiteSeer, and PubMed datasets (Sen et al., 2008).

Both architectures share a common structure. At each layer ℓ, node features are updated as: $\mathbf{X}^{(\ell+1)} = \sigma(\hat{\mathbf{A}}\mathbf{X}^{(\ell)}\mathbf{W}^{(\ell)})$, where $\mathbf{X}^{(0)}$ are input features, $\sigma(\cdot)$ is typically ReLU, $\mathbf{W}^{(\ell)}$ are learnable weights, and $\hat{\mathbf{A}}$ is the aggregation operator (a degree-normalized adjacency matrix $\hat{\mathbf{A}}_{\text{GCN}}$ in GCN; a learnable right-stochastic attention matrix $\hat{\mathbf{A}}_{\text{GAT}}$ in GAT). The single layer comprises three

sequential steps: *aggregation* ($\hat{\mathbf{X}}^{(\ell+1)} = \hat{\mathbf{A}}\mathbf{X}^{(\ell)}$), *transformation* ($\tilde{\mathbf{X}}^{(\ell+1)} = \hat{\mathbf{X}}^{(\ell+1)}\mathbf{W}^{(\ell)}$), and *activation* ($\mathbf{X}^{(\ell+1)} = \sigma(\tilde{\mathbf{X}}^{(\ell+1)})$). We note that removing aggregation yields a standard multi-layer perceptron (MLP) ($\mathbf{X}^{(\ell+1)} = \sigma(\mathbf{X}^{(\ell)}\mathbf{W}^{(\ell)})$), while Simple Graph Convolution (SGC) (Wu et al., 2019) removes activation, applying multiple aggregations followed by a single transformation: $\mathbf{X}^{(L)} = \hat{\mathbf{A}}^L\mathbf{X}^{(0)}\mathbf{W}$.

Among these three steps, Li et al. (2018) identified aggregation as the source of depth limitations. They introduced the concept of *oversmoothing*, showing that GCN's aggregation is a special form of Laplacian smoothing. When repeated infinitely, aggregation causes node features resulting in the degree-scaled embedding convergence. Several studies reinforce the claim by proving an asymptotic oversmoothing regime in various forms of aggregation from different perspectives. Keriven (2022) demonstrate that the repeated aggregation process with the right stochastic matrix results in the uniform embedding convergence, using the ergodic theorem. Chamberlain et al. (2021) interpret GATs as a discretization of the heat diffusion equation, implying that the embedding reaches equilibrium. Based on this intuition, Thorpe et al. (2022) provide a theoretical analysis showing that the outputs of GATs asymptotically converge to a fixed value independent of the initial inputs. However, such asymptotic analyses do not explain why performance degradation occurs in GNNs within the non-asymptotic regime, that is, when GNNs have only a small number of layers.

Subsequent work (Oono & Suzuki, 2020; Cai & Wang, 2020; Wu et al., 2023) seem to answer the question, providing a theoretical analysis that the oversmoothing occurs exponentially. To quantify the speed of oversmoothing, these studies first define a node similarity measure $\mu : \mathbb{R}^{N\times d} \to \mathbb{R}_{\geq 0}$ that approaches zero when node embedding converges. With the proposed measure, the exponential oversmoothing can be shown as an exponential decay of the measure, such that

$$\mu(\mathbf{X}^{(\ell)}) \leq Cq^{\ell} \text{ for } \ell = 1, \cdots, L-1, \tag{1}$$

for some constant $C > 0$, and $0 < q < 1$. In Appendix B, we summarize four representative node similarity measures, $\mu_{\texttt{oono}}$, $\mu_{\texttt{cai}}$, $\mu_{\texttt{wu}}$, and $\mu_{\texttt{rusch}}$ from Oono & Suzuki (2020); Cai & Wang (2020); Wu et al. (2023); Rusch et al. (2023a), respectively.By addressing a question left by previous asymptotic analyses, this line of research has further strengthened the theoretical foundation for the belief that aggregation introduces inherent limitations in deep GNNs.

These early observations and persuasive explanations led to the widespread belief that GNNs inherently suffer significant performance degradation as depth increases (Oono & Suzuki, 2020; Zhao & Akoglu, 2020; Zhou et al., 2021a;b; Zhang et al., 2022; Thorpe et al., 2022; Song et al., 2023). This belief was further solidified by the identification of the over-squashing problem, which suggests that the aggregation structure inherently limits the model's ability to learn long-range dependencies. Since then, many studies have either examined the intrinsic problems of aggregation in deep GNN construction more rigorously or attempted to construct deep GNNs and improve their performance by addressing these issues. We summarize these lines of work in Appendix A.

However, some studies challenge such established views. For example, Li et al. (2019) successfully trained a 56-layer GCN by adapting skip connections. Similarly, Li et al. (2021) achieve the best performance using a 448-layer GNN, by addressing memory consumption issues. Zhou et al. (2021b) and Zhang et al. (2022) show no significant performance degradation when repeatedly applying the aggregation process without the transformation step. Some studies (Yang et al., 2020; Cong et al., 2021) show that oversmoothing does not occur after training. Moreover, Rusch et al. (2023a) and Heo et al. (2025) show that node similarity measures show empirically weak correlation with performance.

3 Revisiting the Evidence

The narrative that aggregation limits depth rests mainly on two sources: depth-performance experiments (Kipf & Welling, 2017; Veličković et al., 2018) showing optimal results at shallow layers, and theoretical analysis explaining performance degradation via oversmoothing (Li et al., 2018; Oono & Suzuki, 2020; Chamberlain et al., 2021; Thorpe et al., 2022; Wu et al., 2023). In this section, we revisit these evidence through systematic ablations and careful review of theoretical claims.

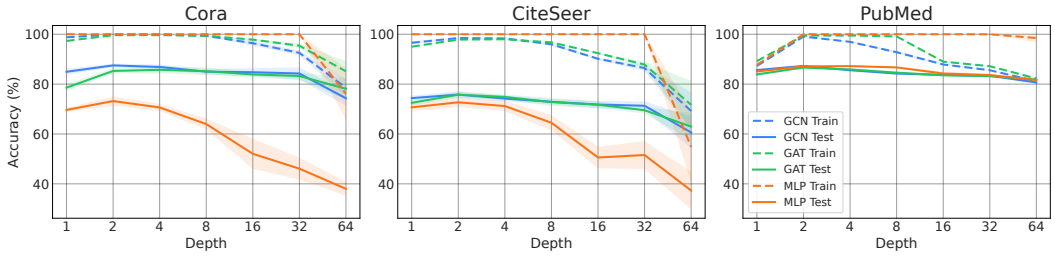

Figure 1: Training and test accuracy of GCN, GAT, and MLP across varying depths (1-64 layers) on three citation network datasets: Cora, CiteSeer, and PubMed. Dashed lines represent average training accuracy and solid lines represent average test accuracy over 5 repeated runs. Shaded regions indicate the standard deviation around the average.

3.1 Depth-Performance Patterns

We evaluate the training and test accuracy of vanilla GCN, GAT, and MLP while varying the depth from 1 to 64 on Cora, CiteSeer, and PubMed datasets used in Kipf & Welling (2017) and Veličković et al. (2018). All models use He initialization (He et al., 2015) for weight parameters. Detailed experimental setups are provided in Appendix F. The results are illustrated in Figure 1.

The experimental results directly refute the prevailing narrative that aggregation-induced issues, such as oversmoothing, cause severe performance degradation as depth increases. The test accuracy of GCN and GAT remains stable up to depth 32 and decreases only moderately at depth 64, whereas MLPs suffer substantially larger declines on Cora and CiteSeer. The training accuracy of GNNs decreases gradually as depth increases, while MLPs fit the training data almost perfectly from depth 2 to 32. Nevertheless, GNNs maintain training accuracy above 80% in almost all cases. The only exceptions are GCN on Cora and CiteSeer and GAT on CiteSeer at depth 64, where the training accuracy of MLPs also falls below 80% and GNNs retain higher training accuracy than MLPs. Taken together, the generalization gap of GNNs is much smaller than that of MLPs: their test accuracy remains high even as their training accuracy decreases. This comparison with MLPs suggests a property that has received little attention: *aggregation contributes to the robustness of test performance as depth increases.*

We note that the results of GCN reported here are inconsistent with those of Kipf & Welling (2017), where a severe performance drop is observed beyond eight layers. We find that the reported drop appears rarely, only under particular random seeds, and that on those seeds MLPs degrade even earlier and more severely than GCNs. We provide the detailed analysis in Appendix C.

We report results on six additional datasets in Appendix D.1: Photo (McAuley et al., 2015), Computers (McAuley et al., 2015), Squirrel (Platonov et al., 2023; Rozemberczki et al., 2021), Chameleon (Platonov et al., 2023; Rozemberczki et al., 2021), Amazon-ratings (Platonov et al., 2023), and Roman-empire (Platonov et al., 2023). The comparative pattern holds for most of them. We further discuss future research directions on how the homophily ratio relates to the performance behavior of deep GNNs in Section 4.

Despite the robust test accuracy, both GCN and GAT achieve their best test performance at shallow depths of two or four layers on these datasets, consistent with prior observations. However, MLPs also achieve their best performance at two or four layers. These results are therefore insufficient to conclude that achieving the best performance at shallow depths is an inherent property of GNNs. Rather, this may reflect the nature of these tasks, which do not require the additional capacity that depth provides. Therefore, to examine whether GNNs inherently prefer shallow depths, evaluation on tasks that demand greater capacity would be necessary. Indeed, GNNs have been shown to benefit from increased depth on large-scale datasets (Li et al., 2021) and tasks that require capturing long-range dependencies (Park et al., 2024a;b), which we discuss further in Section 4.

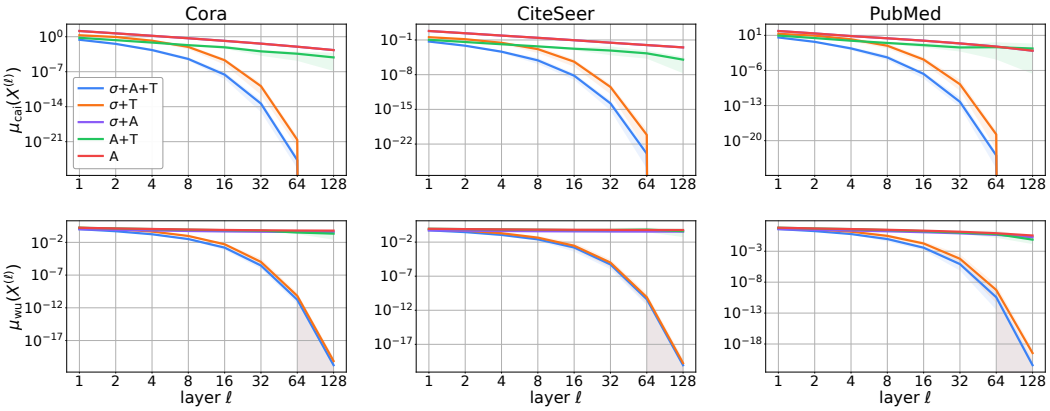

Figure 2: Oversmoothing levels measured by three metrics across 128-layer GCN and GAT evaluated on three datasets: Cora, CiteSeer, and PubMed. For each model, we compare five variations based on the combination of aggregation (A), transformation (T), and activation (σ) steps. For example, A+T refers to a model with aggregation and transformation steps. The solid curves indicate the average over 10 repeated runs, and shaded regions indicate the min-max range.

3.2 Theoretical Analysis

Oversmoothing has been widely recognized as a fundamental obstacle to deep GNN construction. This belief has been solidified by rigorous theoretical analyses and their empirical validations demonstrating oversmoothing's occurrence in GNNs. However, considering the transformation process in GNNs raises an overlooked question: *Can learned transformations prevent oversmoothing?* Intuitively, the answer should be yes. Unless aggregation makes representations perfectly identical, transformations can amplify differences and offset smoothing, even through simple scalar multiplication. Aligned with this intuition, some studies (Yang et al., 2020; Cong et al., 2021; Arnaiz-Rodriguez & Errica, 2026) observe that oversmoothing does not occur after training, suggesting weight parameters learn to offset smoothing effects. This observation motivates closer examination of what conditions the theory actually requires for oversmoothing to occur.

We found that this phenomenon aligns with the theoretical analysis provided by Oono & Suzuki (2020) and Cai & Wang (2020). For example, in GCN, Oono & Suzuki (2020) demonstrate

$$\mu_{\texttt{oono}}(\mathbf{X}^{(\ell)}) \leq (s\lambda)^{\ell}\mu_{\texttt{oono}}(\mathbf{X}^{(0)}) \,, \tag{2}$$

where s is the largest operator norm of $\mathbf{W}^{(\ell)}, \forall \ell$, $\lambda < 1$ is the second largest eigenvalue of $\hat{\mathbf{A}}_{\texttt{GCN}}$. Based on Equation (2), the exponential oversmoothing occurs only when the combined effects of aggregation and transformation drive the node similarity measure toward zero, i.e., $s\lambda < 1$. The theory does not imply that oversmoothing is inevitable; rather, occurrence depends on weight parameters. Since cross-entropy loss encourages separating representations of different classes, we can expect transformations to learn in directions that counteract aggregation's smoothing effect. Similarly, Cai & Wang (2020) demonstrate $\mu_{\texttt{cai}}(\mathbf{X}^{(\ell)}) \leq (s(1-\bar{\lambda})^2)^{\ell}\mu_{\texttt{cai}}(\mathbf{X}^{(0)})$ where $\bar{\lambda}$ indicates smallest non-zero eigenvalue of augmented normalized Laplacian $\mathbf{I}_N - \hat{\mathbf{A}}_{\texttt{GCN}}$. Despite using a different node similarity measure and a slightly different exponential base, the presence of the weight-dependent term s leads to the same conclusion: *oversmoothing is conditional on learned parameters, not inevitable*.

Not all theoretical work has carefully accounted for weight parameters. Some analyses (Li et al., 2018; Chamberlain et al., 2021; Keriven, 2022; Thorpe et al., 2022) focus on oversmoothing in the asymptotic regime of SGC structures, where aggregation is repeated infinitely without intermediate transformations or activations ($\lim_{L\to\infty} \hat{\mathbf{A}}^L\mathbf{X}\mathbf{W}$), thereby sidestepping the question of how learned weights interact with aggregation across layers. Wu et al. (2023) demonstrates exponential oversmoothing in GATs, explicitly considering the transformation step in their analysis. However, in deriving upper bounds on node similarity measures, they treat the weight-dependent term as a constant and factor it out of the exponential. This results in bounds that appear independent of weight parameters, overlooking how transformation influences the exponential rate of oversmoothing.

3.3 Empirical Observation on Exponential Oversmoothing

While the theory's conditional nature suggests oversmoothing is not inevitable with proper optimization, a critical concern remains: *what if aggregation causes oversmoothing so rapidly in initialized networks that optimization cannot even begin?* If node representations converge before any training takes place, gradients cannot flow properly and there is no signal to update the weights. This would render the theoretical escape route via learned transformations practically inaccessible. Previous studies have shown that GNNs overcome oversmoothing when they can be trained, but only at moderate depths: depth 30 in Yang et al. (2020), depth 10 in Cong et al. (2021), and depth 64 in Arnaiz-Rodriguez & Errica (2026), the last obtained by doubling the weights rather than by training. It thus remains possible that beyond a certain depth the collapse at initialization prevents training from starting at all.

Experimental validations of exponential oversmoothing theory appear to confirm this concern. Multiple studies (Oono & Suzuki, 2020; Wu et al., 2023; Rusch et al., 2023a) measure oversmoothing and observe exponential decay approaching zero even at moderate depths around 64 layers. These results have been interpreted as strong evidence that aggregation severely limits practical depth. However, none of the previous studies report the influence of individual steps. Our systematic ablation studies show that the rapid decay cannot be mainly attributed to aggregation. Instead, it is dominated by the combination of transformation and activation, not aggregation.

To quantify the independent influence of individual steps, we measure and compare the changes in node similarity using five variants of the GNNs: the original model ($\sigma+\mathsf{A}+\mathsf{T}$), the model with activation and transformation ($\sigma+\mathsf{T}$), the model with activation and aggregation ($\sigma+\mathsf{A}$), the model with aggregation and transformation ($\mathsf{A}+\mathsf{T}$), the model with aggregation-only (A). For example, the model with activation and transformation ($\sigma+\mathsf{T}$) effectively becomes an MLP by excluding the aggregation step. The model with aggregation (A) only can be seen as the SGC. We compute node similarity measures from 128-layer GCN and GAT models at depths $\ell = 1, 2, 4, \cdots, 128$ on Cora, CiteSeer, and PubMed datasets: $\mu_{\texttt{cai}}(\mathbf{X}^{(\ell)})$ for GCN, and $\mu_{\texttt{wu}}(\mathbf{X}^{(\ell)})$ for GAT.

Figure 2 shows the changes in the similarity measure over the layers. Regardless of the dataset and the node similarity measure, rapid exponential decay reaching near-zero values around depth $64 \sim 128$ is observed only when both transformation and activation are present (i.e., $\sigma+\mathsf{A}+\mathsf{T}, \sigma+\mathsf{T}$). All other configurations, models excluding activation (i.e., $\mathsf{A}+\mathsf{T}$), models excluding transformation (i.e., $\sigma+\mathsf{A}$), and aggregation-only models (i.e., A), exhibit significantly lower decay rate. The same patterns are observed on additional datasets with additional node similarity measure $\mu_{\texttt{rusch}}(\mathbf{X}^{(\ell)})$ for GAT (Appendix D.2).

From these results, we conclude that *the influence of the aggregation step is marginal* and the oversmoothing is primarily caused by the combination of transformation and activation steps. The slow decay rate induced by aggregation can be attributed to graph sparsity: sparser graphs exhibit lower decay rates (Rong et al., 2020), and real-world graphs are typically highly sparse (e.g., Cora has sparsity of 10^{-3}). These results do not contradict the theoretical analyses of Cai & Wang (2020) and Oono & Suzuki (2020), which attribute exponential oversmoothing to the combination of aggregation and transformation, nor Wu et al. (2023), which attributes it to aggregation and activation. Indeed, exponential decay does occur in these combinations, albeit at much slower rates than in models including both transformation and activation. However, both the theoretical analyses and empirical validations in these studies overlooked the dominant role of transformation and activation in driving rapid exponential decay.

Then why do transformation and activation steps cause oversmoothing? Note that $\sigma+\mathsf{T}$ is equal to the MLP architecture, and we already have a well-known case where the oversmoothing occurs with MLPs: the *zero-collapsing*, whereby node embeddings converge toward the zero vector (Glorot & Bengio, 2010).[1] This occurs most prominently when Glorot (Xavier) initialization (Glorot & Bengio, 2010), originally designed for linear or tanh activations, is paired with ReLU. In the deep learning community, zero-collapsing has been primarily understood as a source of vanishing gradients rather than as oversmoothing, and researchers have developed effective solutions, most notably He initialization (He et al., 2015), which scales weights appropriately for ReLU.

[1] While neuron saturation describes similar behaviors in sigmoid and hyperbolic tangent activations, we employ the term 'zero-collapsing' to denote the specific scenario where activations collapse toward zero.

Note that we initialize the weight parameters using Glorot (Xavier) initialization (Glorot & Bengio, 2010) to reproduce experimental setup of previous studies (Oono & Suzuki, 2020; Wu et al., 2023; Rusch et al., 2023a).[2] To show how the results change with different initialization choices, we provide the results using He initialization in Appendix D.3, and the results measured after training under both initializations in Appendix E. With He initialization, the decay rates of both the original model ($\sigma+\mathsf{A}+\mathsf{T}$) and the model with activation and transformation ($\sigma+\mathsf{T}$) significantly decrease, and the node similarity measures remain far from zero even at depth 128. After training, the measures of these models no longer decay toward zero and instead grow at deeper layers under He initialization, whereas under Glorot initialization, training barely proceeds and the measures collapse as in the untrained networks.

In summary, the rapid exponential oversmoothing previously attributed to aggregation is, in fact, zero-collapsing, a phenomenon that also occurs in architectures without aggregation, such as MLPs. Returning to our earlier question, the answer is that aggregation induces oversmoothing at a substantially slower rate, slow enough that it poses no concern at the depths typically considered in practice. Also, rapid convergence can be resolved through classical solutions such as proper weight initialization (e.g., He initialization), without requiring GNN-specific architectural innovations.

4 Call to Action: What We Recommend Going Forward

Our analysis reveals that the field has been constrained by an incomplete understanding of aggregation's role in deep GNNs. Moving forward requires two complementary approaches: immediate practical deployment of deep GNNs where they are needed, and rigorous theoretical investigation to understand when and why they work. We address each audience in turn.

4.1 For Practitioners: Deploy Deep GNNs

Deep GNNs already work. Despite the prevailing narrative that aggregation fundamentally limits depth, a parallel line of research has demonstrated successful deep GNN construction. ResGCNs (Li et al., 2019) achieve state-of-the-art performance on point cloud semantic segmentation with 56 layers by employing skip connections to address optimization challenges. RevGNN (Li et al., 2021) solve memory complexity to train 448-layer GNNs, achieving top performance on the large-scale ogbn-proteins dataset. Park et al. (2024a) introduce a reverse GNN structure that makes representations distinguishable, reaching peak performance at 256 layers on several heterophilic datasets. Park et al. (2024b) address the redundant message flow, also known as the backtracking problem, reporting best performance at depth 20 on the Peptides-struct dataset from the long-range graph benchmark (LRGB) (Dwivedi et al., 2022). We note that 20 was the maximum depth explored in their hyperparameter search.

Task-dependent depth requirements. These cases reveal an important pattern: deep GNNs succeed on specific types of tasks. Large-scale datasets like ogbn-proteins benefit from the increased model expressivity that depth provides. Heterophilic datasets and long-range graph benchmarks require the extended receptive fields that only deep architectures can provide (Rusch et al., 2023a). However, this does not mean depth is universally beneficial. Small-scale datasets like Cora and CiteSeer, may continue to perform best with shallow models.

Practical recommendations. For practitioners working with large-scale datasets, long-range dependency tasks, or heterophilic graphs, we recommend actively exploring deep architectures. Adopting techniques from previous success cases, such as skip connections and memory-efficient training methods, can facilitate deep GNN construction. Depth should be treated as an explorable hyperparameter, not an inherent limitation.

[2] One puzzling aspect we notice is that many official GNN implementations (Akiba et al., 2017; Tokui et al., 2015; 2019), including Pytorch Geometric (Fey & Lenssen, 2019), adopt Glorot initialization by default, despite ReLU being the most popular in GNNs. We find that numerous studies (Chen et al., 2020; Oono & Suzuki, 2020; Rong et al., 2020; Zhao & Akoglu, 2020; Zhou et al., 2021a; Rusch et al., 2022; 2023a;b; Park et al., 2024a) also commonly adopt Glorot initialization.

4.2 For Researchers: Advance Deep GNNs

Understanding the real challenges. While deep GNNs can be deployed successfully, our understanding of how aggregation actually affects deep architectures remains incomplete. Accurately characterizing aggregation's properties is essential for advancing deep GNNs.

Our suggestion for the first step is identifying *when smoothing becomes problematic*. Current notions of oversmoothing and node similarity measures (Oono & Suzuki, 2020; Cai & Wang, 2020; Wu et al., 2023; Rusch et al., 2023a) focus on convergence of representations. While they capture complete convergence, they provide no guidance on when representations become problematically similar. Yet even modest smoothing, far from full convergence, can be harmful if it mixes representations of different classes.

For identification, we need explicit connections between smoothing and its consequences for learning, such as optimization and generalization performance. Intuitively, smoothing affects optimization directly: it eases gradient descent when representations of the same class become similar, but impedes it when representations of different classes get mixed. Smoothing also impacts generalization, as recent work reveals that clustered intra-class representations and separable inter-class representations are critical for generalization (Chuang et al., 2021). Rigorously capturing when smoothing helps versus harms optimization and generalization would provide inspiration for deep GNN's development.

Building effective solutions. With accurate understanding of aggregation's properties, the field can develop more powerful deep GNN architectures that practitioners can deploy with confidence. We identify two promising directions for developing such architectures.

First, adapt proven classical techniques to work optimally with GNN structures. While skip connections, batch normalization, and He-initialization enable deep GNN training, these techniques were designed without consideration of GNN architecture. Several studies have adapted such classical deep learning techniques to the graph domain, including skip connections (Zhang et al., 2022; Lu et al., 2024), normalization layers (Cai et al., 2021; Zhou et al., 2021b), and initialization (Li et al., 2023), yielding improvements. Further developing these classical techniques based on an accurate understanding of aggregation's properties, and systematically exploring how to optimally integrate the solutions, would significantly advance practical deep GNN design.

Second, develop solutions that address GNN-specific challenges in layer stacking. Methods proposed to mitigate oversmoothing (Zhao & Akoglu, 2020; Rong et al., 2020; Chen et al., 2020; Zhou et al., 2021a; Eliasof et al., 2021; Chamberlain et al., 2021; Thorpe et al., 2022; Rusch et al., 2022; Fang et al., 2023; Song et al., 2023; Rusch et al., 2023b; Park et al., 2024a; Su et al., 2024; Pei et al., 2024; Lu et al., 2024) are valuable when harmful smoothing dominates and needs to be slowed down. This is particularly relevant for heterophilic graphs, which require long-range propagation but where aggregation is likely to mix representations of different classes. Similarly, addressing phenomena like backtracking, which shows stronger performance correlation (Park et al., 2024b), can yield improvements. The key is building frameworks that guide which solutions to apply for specific graph characteristics and task requirements, enabling practitioners to construct deep GNNs effectively without defaulting to shallow architectures.

5 Alternative Views

While we challenge prevailing beliefs about aggregation in deep GNNs, we acknowledge credible alternative perspectives. We address two main counterarguments.

5.1 "These Issues Are Already Being Addressed"

The counterargument. One might argue that our finding is redundant, that the field has already moved beyond viewing aggregation as fundamentally limiting. After all, numerous recent works have successfully constructed deep GNNs (Li et al., 2019; 2021; Park et al., 2024a;b). In addition, several studies have questioned the severity of oversmoothing or whether aggregation is responsible for the performance degradation of deep GNNs (Roth & Liebig, 2024; Yang et al., 2020; Cong et al., 2021; Zhou et al., 2021b; Zhang et al., 2022; Arnaiz-Rodriguez & Errica, 2026). If researchers are

already building deep GNNs and questioning aggregation-centric explanations, what novel contribution does our finding provide?

Our response. While we acknowledge these important developments, they represent fragmented insights rather than a systematic reframing of the field's understanding. The mainstream discourse continues to treat deep GNN construction as fundamentally difficult due to aggregation-induced issues. Major surveys (Rusch et al., 2023a; Ju et al., 2024; Jin & Zhu, 2025), papers (Li et al., 2018; Alon & Yahav, 2021; Wenkel et al., 2022; Giraldo et al., 2023; Qureshi et al., 2023; Wu et al., 2023; Li et al., 2024; Pei et al., 2024; Errica et al., 2025; Sun et al., 2025), and textbooks (Ma & Tang, 2021; Wu et al., 2022a) still prominently feature oversmoothing and oversquashing as inherent limitations of GNN depth, not as problems that have been adequately addressed or questioned. Even the works that question the severity of oversmoothing or the role of aggregation retain the broader premise that GNNs suffer from depth and focus on identifying its cause within that frame, thereby reinforcing the narrative (see Appendix A for details). This narrative persists in shaping research priorities and practitioner beliefs.

We argue that these insights failed to become mainstream because they did not directly address the evidentiary foundations, the experimental results and theoretical analyses, that established the aggregation-as-obstacle narrative in the first place. Our contribution is to provide this systematic re-examination of the evidence, unifying fragmented insights into a coherent narrative that can reshape the field's mainstream understanding. By directly addressing the evidentiary gaps that previous work left intact, we aim to shift the field from fragmented corrections to a coherent understanding: aggregation's effects are complex and context-dependent, not uniformly limiting, and research should focus on understanding *when* and *why* depth helps rather than assuming it inherently hurts.

5.2 "Aggregation Is Indeed Problematic"

The counterargument. GNNs attain lower training accuracy than MLPs as depth increases, as shown in Section 3.1. Supporting this observation, several recent theoretical works (Park & Kim, 2024; Arroyo et al., 2025; Keriven, 2026) argue that aggregation makes GNNs harder to train. Moreover, the success of deep GNNs seems to validate aggregation-centric concerns, since their methods address aggregation-related issues. Li et al. (2019) utilize skip connections and normalization, which are known to mitigate oversmoothing (Scholkemper et al., 2024); Park et al. (2024a) modify aggregation's smoothing property; and Non-backtracking GNNs (Park et al., 2024b) improve performance by eliminating message revisitation through the same edges. If fixing aggregation-related issues yields improvements, doesn't this prove aggregation is indeed problematic?

Our response. We do not claim that aggregation presents no challenges. Our claim is that aggregation's actual effects must be accurately understood. Building on such understanding and on the solutions already available, whether modifying aggregation (Park et al., 2024a;b) or adopting classic techniques such as skip connections and normalization, research should explore the new architectural landscape that these solutions enable, rather than continuing to focus on the limitations of GNNs. For example, Park & Kim (2024); Arroyo et al. (2025); Keriven (2026) not only analyze the difficulties of training vanilla GNNs (see Appendix A for details), but also show that skip connections and normalization help to address these difficulties, or explicitly highlight them as an important research direction. Based on these studies, we encourage researchers to advance deep GNNs on top of skip connections and normalization.

6 Conclusion

This paper challenges the narrative that aggregation fundamentally limits depth in GNNs by carefully re-examining experimental evidence and theoretical explanations. Our findings directly call for two actions. Practitioners should explore deep architectures for high-capacity and long-range tasks. Researchers should accurately characterize the effects of aggregation. By moving from incomplete narratives to an accurate understanding, we can unlock the full potential of deep GNNs.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, or interpret results, and formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, create or modify scientific figures, and edit the paper to improve readability. Specifically, based on the code implemented by the authors, the tools were used to organize code and results, write scripts for running experiments efficiently, and check the consistency between our intended experimental setups and their implementation. For writing, the tools suggested alternative phrasings for text drafted by the authors and assisted in translating drafts. We have reviewed all AI-assisted work. All AI-generated code was inspected by the authors, and its outputs were verified by reproducing our previously reported results. All interpretations in this paper were written by the authors, and AI tools were used only to check them against the experimental results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the experimental setups, including data splits, model configurations, initialization, training procedures, and hyperparameter search spaces, in Appendix F, and the dataset statistics in Appendix G. We will release the source code for all experiments upon publication.

REFERENCES

Takuya Akiba, Keisuke Fukuda, and Shuji Suzuki. ChainerMN: Scalable Distributed Deep Learning Framework. In *Proceedings of Workshop on ML Systems in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2021.

Iro Armeni, Sasha Sax, Amir R. Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *CoRR*, abs/1702.01105, 2017.

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.

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.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3950–3957, 2021.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. In *ICML Graph Representation Learning and Beyond (GRL+) Workshop*, 2020.

Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-Yan Liu, and Liwei Wang. Graphnorm: A principled approach to accelerating graph neural network training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1204–1215. PMLR, 18–24 Jul 2021.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1407–1418. PMLR, 18–24 Jul 2021.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1725–1735. PMLR, 13–18 Jul 2020.

Ching-Yao Chuang, Youssef Mroueh, Kristjan Greenewald, Antonio Torralba, and Stefanie Jegelka. Measuring generalization with optimal transport. *Advances in neural information processing systems*, 34:8294–8306, 2021.

Weilin Cong, Morteza Ramezani, and Mehrdad Mahdavi. On provable benefits of depth in training graph convolutional networks. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Hanjun Dai, Chengtao Li, Connor W. Coley, Bo Dai, and Le Song. *Retrosynthesis prediction with conditional graph logic network*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Andreea Deac, Marc Lackenby, and Petar Veličković. Expander graph propagation. In *Learning on Graphs Conference*, pp. 38–1. PMLR, 2022.

Vijay Prakash Dwivedi, Ladislav Rampášek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Moshe Eliasof, Eldad Haber, and Eran Treister. PDE-GCN: Novel architectures for graph neural networks motivated by partial differential equations. In *Advances in Neural Information Processing Systems*, 2021.

Federico Errica, Henrik Christiansen, Viktor Zaverkin, Takashi Maruyama, Mathias Niepert, and Francesco Alesiani. Adaptive message passing: A general framework to mitigate oversmoothing, oversquashing, and underreaching. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15490–15515. PMLR, 13–19 Jul 2025.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*, WWW '19, pp. 417–426, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366748. doi: 10.1145/3308558.3313488.

Taoran Fang, Zhiqing Xiao, Chunping Wang, Jiarong Xu, Xuan Yang, and Yang Yang. Dropmessage: Unifying random dropping for graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):4267–4275, Jun. 2023. doi: 10.1609/aaai.v37i4.25545.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1263–1272. JMLR.org, 2017.

Jhony H Giraldo, Konstantinos Skianis, Thierry Bouwmans, and Fragkiskos D Malliaros. On the trade-off between over-smoothing and over-squashing in deep graph neural networks. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 566–576, 2023.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *Advances in Neural Information Processing Systems*, 2021.

Jaeseung Heo, Kyeongheung Yun, Seokwon Yoon, MoonJeong Park, Jungseul Ok, and Dongwoo Kim. Influence functions for edge edits in non-convex graph neural networks. *arXiv preprint arXiv:2506.04694*, 2025.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

Xiao Huang, Qingquan Song, Yuening Li, and Xia Hu. Graph recurrent networks with attributed random walks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 732–740, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016.

Yuezihan Jiang, Yu Cheng, Hanyu Zhao, Wentao Zhang, Xupeng Miao, Yu He, Liang Wang, Zhi Yang, and Bin Cui. Zoomer: Boosting Retrieval on Web-scale Graphs by Regions of Interest . In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2224–2236, Los Alamitos, CA, USA, May 2022. IEEE Computer Society.

Yufei Jin and Xingquan Zhu. Oversmoothing alleviation in graph neural networks: a survey and unified view: Y. jin, x. zhu. *Knowledge and Information Systems*, 67(12):11259–11285, 2025.

Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, et al. A comprehensive survey on deep graph representation learning. *Neural Networks*, 173:106207, 2024.

Nicolas Keriven. Not too little, not too much: a theoretical analysis of graph (over)smoothing. In *The First Learning on Graphs Conference*, 2022.

Nicolas Keriven. Backward oversmoothing: why is it hard to train deep graph neural networks? In *Forty-third International Conference on Machine Learning*, 2026.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Chang Li and Dan Goldwasser. Encoding social information with graph convolutional networks forPolitical perspective detection in news media. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2594–2604, Florence, Italy, July 2019. Association for Computational Linguistics.

Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9267–9276, 2019.

Guohao Li, Matthias Müller, Bernard Ghanem, and Vladlen Koltun. Training graph neural networks with 1000 layers. In *International conference on machine learning*, pp. 6437–6449. PMLR, 2021.

Jiahang Li, Yakun Song, Xiang Song, and David Paul Wipf. On the initialization of graph neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Jin Li, Qirong Zhang, Wenxi Liu, Antoni B Chan, and Yang-Geng Fu. Another perspective of over-smoothing: Alleviating semantic over-smoothing in deep gnns. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):6897–6910, 2024.

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 Press, 2018.

Weigang Lu, Yibing Zhan, Binbin Lin, Ziyu Guan, Liu Liu, Baosheng Yu, Wei Zhao, Yaming Yang, and Dacheng Tao. Skipnode: On alleviating performance degradation for deep graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):7030–7043, 2024. doi: 10.1109/TKDE.2024.3374701.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. Classic GNNs are strong baselines: Reassessing GNNs for node classification. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

Yao Ma and Jiliang Tang. *Deep learning on graphs*. Cambridge University Press, 2021.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 43–52, 2015.

Federico Monti, Michael M. Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 3700–3710, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

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

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

Moonjeong Park, Jaeseung Heo, and Dongwoo Kim. Mitigating oversmoothing through reverse process of GNNs for heterophilic graphs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39667–39681. PMLR, 21–27 Jul 2024a.

Namyong Park, Andrey Kan, Xin Luna Dong, Tong Zhao, and Christos Faloutsos. Estimating node importance in knowledge graphs using graph neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 596–606, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016.

Seonghyun Park, Narae Ryu, Gahee Kim, Dongyeop Woo, Se-Young Yun, and Sungsoo Ahn. Non-backtracking graph neural networks. *Transactions on Machine Learning Research*, 2024b. ISSN 2835-8856.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Hongbin Pei, Yu Li, Huiqi Deng, Jingxin Hai, Pinghui Wang, Jie Ma, Jing Tao, Yuheng Xiong, and Xiaohong Guan. Multi-track message passing: Tackling oversmoothing and oversquashing in graph learning via preventing heterophily mixing. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 40078–40091. PMLR, 21–27 Jul 2024.

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.

Shaima Qureshi et al. Limits of depth: Over-smoothing and over-squashing in gnns. *Big Data Mining and Analytics*, 7(1):205–216, 2023.

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

Andreas Roth and Thomas Liebig. Rank collapse causes over-smoothing and over-correlation in graph neural networks. In *Learning on Graphs Conference*, pp. 35–1. PMLR, 2024.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

T Konstantin Rusch, Ben Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael Bronstein. Graph-coupled oscillator networks. In *International Conference on Machine Learning*, pp. 18888–18909. PMLR, 2022.

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

T. Konstantin Rusch, Benjamin Paul Chamberlain, Michael W. Mahoney, Michael M. Bronstein, and Siddhartha Mishra. Gradient gating for deep multi-rate learning on graphs. In *The Eleventh International Conference on Learning Representations*, 2023b.

Michael Scholkemper, Xinyi Wu, Ali Jadbabaie, and Michael T Schaub. Residual connections and normalization can provably prevent oversmoothing in gnns. *arXiv preprint arXiv:2406.02997*, 2024.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Yunchong Song, Chenghu Zhou, Xinbing Wang, and Zhouhan Lin. Ordered GNN: Ordering message passing to deal with heterophily and over-smoothing. In *The Eleventh International Conference on Learning Representations*, 2023.

Guangxin Su, Hanchen Wang, Ying Zhang, Wenjie Zhang, and Xuemin Lin. Simple and deep graph attention networks. *Knowledge-Based Systems*, 293:111649, 2024. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2024.111649.

Li Sun, Zhenhao Huang, Ming Zhang, and Philip S Yu. Deeper with riemannian geometry: Overcoming oversmoothing and oversquashing for graph foundation models. *arXiv preprint arXiv:2510.17457*, 2025.

Matthew Thorpe, Tan Minh Nguyen, Heidi Xia, Thomas Strohmer, Andrea Bertozzi, Stanley Osher, and Bao Wang. Grand++: Graph neural diffusion with a source term. In *International Conference on Learning Representation (ICLR)*, 2022.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

Seiya Tokui, Ryosuke Okuta, Takuya Akiba, Yusuke Niitani, Toru Ogawa, Shunta Saito, Shuji Suzuki, Kota Uenishi, Brian Vogel, and Hiroyuki Yamazaki Vincent. Chainer: A deep learning framework for accelerating the research cycle. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2002–2011. ACM, 2019.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Hao Wang, Tong Xu, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, and Wen Su. Mcne: An end-to-end framework for learning multiple conditional network representations of social network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 1064–1072, New York, NY, USA, 2019a. Association for Computing Machinery. ISBN 9781450362016.

Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 968–977, New York, NY, USA, 2019b. Association for Computing Machinery. ISBN 9781450362016.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 950–958, New York, NY, USA, 2019c. Association for Computing Machinery. ISBN 9781450362016.

Frederik Wenkel, Yimeng Min, Matthew Hirn, Michael Perlmutter, and Guy Wolf. Overcoming oversmoothness in graph convolutional networks via hybrid scattering networks. *arXiv preprint arXiv:2201.08932*, 2022.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871. Pmlr, 2019.

Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Xiaojie Guo. Graph neural networks: foundation, frontiers and applications. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4840–4841, 2022a.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: A survey. *ACM Comput. Surv.*, 55(5), December 2022b. ISSN 0360-0300.

Xinyi Wu, Amir Ajorlou, Zihui Wu, and Ali Jadbabaie. Demystifying oversmoothing in attention-based graph neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Chaoqi Yang, Ruijie Wang, Shuochao Yao, Shengzhong Liu, and Tarek Abdelzaher. Revisiting over-smoothing in deep gcns. *arXiv preprint arXiv:2003.13663*, 2020.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 974–983, New York, NY, USA, 2018. Association for Computing Machinery.

Wentao Zhang, Zeang Sheng, Ziqi Yin, Yuezihan Jiang, Yikuan Xia, Jun Gao, Zhi Yang, and Bin Cui. Model degradation hinders deep graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

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

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 *Advances in Neural Information Processing Systems*, 2021a.

Kuangqi Zhou, Yanfei Dong, Kaixin Wang, Wee Sun Lee, Bryan Hooi, Huan Xu, and Jiashi Feng. Understanding and resolving performance degradation in deep graph convolutional networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2728–2737, 2021b.

Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 07 2017. ISSN 1367-4803.

A Additional Related work

Methods for mitigating oversmoothing. Many studies have attempted to enable deeper stacking of layers without performance degradation by mitigating oversmoothing. PairNorm (Zhao & Akoglu, 2020) maintains total pairwise embedding distances constant through normalization. DropEdge (Rong et al., 2020) randomly removes edges from the graph, thus delaying oversmoothing. Additionally, DropMessage (Fang et al., 2023) unifies and improves random dropping in GNNs by directly masking elements in the message matrix, achieving better generalization, stability, and over-smoothing mitigation. Energetic Graph Neural Networks (Zhou et al., 2021a) incorporate Dirichlet energy into the loss function as a regularizer. Ordered GNN (Song et al., 2023) separately preserves aggregated embeddings within specific hops. Gradient Gating (Rusch et al., 2023b) introduces a mechanism to stop learning individually for each node before local oversmoothing occurs. Park et al. (2024a) propose the framework that can reverse the aggregation process. GCNII (Chen et al., 2020) introduces initial residual connections and identity mapping to solve oversmoothing. LSGAT (Su et al., 2024) adaptively adjusts attention coefficients across layers to avoid oversmoothing. MTGCN (Pei et al., 2024) identifies the root cause of oversmoothing as information loss due to heterophily mixing in aggregation and mitigates it by separating message passing into multiple semantic tracks. SkipNode (Lu et al., 2024) alleviates over-smoothing and gradient vanishing in deep GCNs by selectively skipping message passing for certain nodes. Some studies propose to change the dynamics of GNNs, given that classical GNNs resemble diffusion-like dynamics (Chamberlain et al., 2021). GRAND++ (Thorpe et al., 2022) uses a source term to prevent the convergence of the embedding. PDE-GCN (Eliasof et al., 2021) and Graph-coupled oscillator networks (Rusch et al., 2022) are designed based on the wave function and nonlinear oscillator structures, respectively. However, only a few methods have succeeded in achieving performance improvements with increasing depth.

Works questioning oversmoothing as the cause of degradation. Several studies have questioned the severity of oversmoothing (Yang et al., 2020; Cong et al., 2021; Roth & Liebig, 2024) or whether aggregation is responsible for the performance degradation of deep GNNs (Zhou et al., 2021b; Zhang et al., 2022; Arnaiz-Rodriguez & Errica, 2026). However, these works retain the broader premise that GNNs suffer from depth, and focus on identifying its cause within that frame. Yang et al. (2020), Cong et al. (2021), and Roth & Liebig (2024) attribute the degradation to overfitting, limited generalization ability, and rank collapse, respectively. Zhou et al. (2021b) and Zhang et al. (2022) conduct ablation studies varying the numbers of aggregation and transformation steps, and identify transformation rather than aggregation as the primary source of performance degradation. Arnaiz-Rodriguez & Errica (2026) also challenge common beliefs about oversmoothing and oversquashing, arguing that performance degradation should not be attributed to oversmoothing alone and pointing to other factors such as node separability and vanishing gradients. In contrast, rather than accepting the narrative that GNNs suffer from depth as a premise, we directly address its evidentiary foundations and re-examine the narrative itself.

Works attributing training difficulties to aggregation. Other studies regard optimization as the fundamental cause of oversmoothing and point to adverse effects of aggregation on optimization (Park & Kim, 2024; Arroyo et al., 2025; Keriven, 2026). Park & Kim (2024) and Keriven (2026) show that aggregation also induces oversmoothing in the gradient space, and Arroyo et al. (2025) argue that the contractive nature of aggregation causes representations to collapse toward zero, leading to vanishing gradients. The phenomenon described by Arroyo et al. (2025) is closely related to what we call zero-collapsing, but the two works differ in its cause. Arroyo et al. (2025) attribute the collapse to aggregation, whereas we find that it is driven by the combination of transformation and activation under an improper initialization, and occurs equally in MLPs without aggregation (Section 3.3).

B Node similarity measures

In this section, we provide four node similarity measures used in previous works for completeness and how the exponential oversmoothing is characterized via the similarity measures.

In the literature, we identify two distinct definitions of oversmoothing, each specific to GCNs and GATs, respectively. The following two propositions clarify the distinction between the oversmoothing in GCNs and GATs.

Proposition B.1 (degree-scaled embedding convergence (Cai & Wang, 2020; Li et al., 2018; Oono & Suzuki, 2020)). *In GCNs, there exist* $\mathbf{c} \in \mathbb{R}^d$ *such that* $\lim_{\ell\to\infty} \mathbf{x}_i^{(\ell)} = \sqrt{d_i+1}\mathbf{c}$ *for all nodes* $i \in \mathcal{V}$

Proposition B.2 (uniform embedding convergence (Keriven, 2022; Thorpe et al., 2022; Wu et al., 2023)). *In GATs, there exist* $\mathbf{c} \in \mathbb{R}^d$ *such that* $\lim_{\ell\to\infty} \mathbf{x}_i^{(\ell)} = \mathbf{c}$ *for all nodes* $i \in \mathcal{V}$

These propositions imply that unless all node degrees are equal, i.e., $d_i = d$ for all i, or embeddings converge to the zero vector, i.e., $\lim_{\ell\to\infty} \mathbf{x}_i^{(\ell)} = [0, 0, \cdots, 0]^\top$, GCNs and GATs exhibit distinct behaviors in the asymptotic regime.

Oono & Suzuki (2020) uses distance between embedding $\mathbf{X}$ and a space $\mathcal{M} = \{\tilde{\mathbf{D}}^{\frac{1}{2}}\mathbf{1}_N \otimes \mathbf{w} | \mathbf{w} \in \mathbb{R}^d\} \in \mathbb{R}^{N\times d}$ as a node similarity measure, i.e.:

$$\mu_{\texttt{oono}}(\mathbf{X}) \coloneqq \inf\{\|\mathbf{X} - \mathbf{Y}\|_F \,|\mathbf{Y} \in \mathcal{M}\}\,, \tag{3}$$

where $\mathbf{1}_N = [1, \cdots, 1]^\top \in \mathbb{N}^N$, $\otimes$ denotes Kronecker product, and $\|\cdot\|_F$ indicates Frobenius norm. Embedding $\mathbf{X}$ spans on $\mathcal{M}$, i.e., $\mu_{\texttt{oono}}(\mathbf{X}) = 0$, if and only if $\mathbf{X}$ satisfies Theorem B.1.

For demonstration of the exponential occurrence of oversmoothing in GCN, they show that

$$\mu_{\texttt{oono}}(\mathbf{X}^{(\ell)}) \leq (s\lambda)^\ell \mu_{\texttt{oono}}(\mathbf{X}^{(0)})\,, \tag{4}$$

where s is the largest operator norm of $\mathbf{W}^{(\ell)}, \forall\ell$, $\lambda < 1$ is the second largest eigenvalue of $\hat{\mathbf{A}}_{\texttt{GCN}}$. In the process of demonstration, they show how each aggregation, transformation, and activation step influences the oversmoothing as follows:

$$\mu_{\texttt{oono}}(\hat{\mathbf{A}}_{\texttt{GCN}}\mathbf{X}) \leq \lambda\mu_{\texttt{oono}}(\mathbf{X}),\ \mu_{\texttt{oono}}(\mathbf{X}\mathbf{W}) \leq \|\mathbf{W}\|_2\, \mu_{\texttt{oono}}(\mathbf{X}),\ \mu_{\texttt{oono}}(\sigma(\mathbf{X})) \leq \mu_{\texttt{oono}}(\mathbf{X}), \tag{5}$$

where $\|\cdot\|_2$ denotes operator norm. Equation (5) suggests that exponential oversmoothing occurs due to the aggregation and transformation step.

Empirical results show that $\mu_{\texttt{oono}}(\mathbf{X}^{(\ell)})$ exponentially decays in practice, satisfying the inequality provided by the theory. However, an adequate explanation of why the distance continues to decay even in cases where the upper bound explodes due to the condition $s\lambda > 1$ was not provided.

Cai & Wang (2020) simplify the proof of Oono & Suzuki (2020) by using Dirichlet energy as a measure of oversmoothing, i.e.,

$$\mu_{\texttt{cai}}(\mathbf{X}) \coloneqq \frac{1}{2}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\left\|\frac{\mathbf{x}_i}{\sqrt{1+d_i}} - \frac{\mathbf{x}_j}{\sqrt{1+d_j}}\right\|_F^2. \tag{6}$$

$\mu_{\texttt{cai}}(\cdot) = 0$ also holds if and only if Theorem B.1. They prove that $\mu_{\texttt{cai}}(\mathbf{X}^{(\ell)}) \leq (s(1-\bar{\lambda})^2)^\ell \mu_{\texttt{cai}}(\mathbf{X}^{(0)})$ by showing:

$$\mu_{\texttt{cai}}(\hat{\mathbf{A}}_{\texttt{GCN}}\mathbf{X}) \leq (1-\bar{\lambda})^2\mu_{\texttt{cai}}(\mathbf{X}),\ \mu_{\texttt{cai}}(\mathbf{X}\mathbf{W}) \leq \|\mathbf{W}\|_2\, \mu_{\texttt{cai}}(\mathbf{X}),\ \mu_{\texttt{cai}}(\sigma(\mathbf{X})) \leq \mu_{\texttt{cai}}(\mathbf{X}), \tag{7}$$

where $\bar{\lambda}$ indicates smallest non-zero eigenvalue of augmented normalized Laplacian $\mathbf{I}_N - \hat{\mathbf{A}}_{\texttt{GCN}}$. The demonstration process also implies that the aggregation and transformation step causes the exponential oversmoothing.

Wu et al. (2023) extend the previous theoretical analyses to demonstrate that GATs also lose expressive power exponentially. They define a node similarity measure:

$$\mu_{\texttt{wu}}(\mathbf{X}) \coloneqq \left\|\mathbf{X} - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top\mathbf{X}\right\|_F\,, \tag{8}$$

where $\mu_{\texttt{wu}}(\mathbf{X}) = 0$ if and only if Theorem B.2 holds, instead of Theorem B.1. They show that $\mu_{\texttt{wu}}(\mathbf{X}^{(\ell)})$ approaches zero at an exponential rate as passing the GAT layers, explaining that the

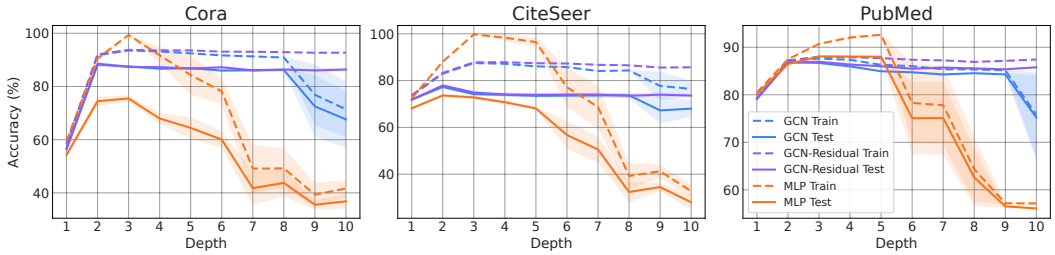

Figure 3: Training (dashed) and test (solid) accuracy of GCN, GCN with residual connections, and MLP across depths 1–10 under the setup of Kipf & Welling (2017), on the data splits that reproduce their reported drop. Shaded regions indicate the standard error over the five folds.

exponential oversmoothing occurred due to the joint spectral radius of a $\hat{\mathbf{A}}_{\texttt{GAT}}$ being less than one. They validate their theory by numerical experiments, showing that $\mu_{\texttt{wu}}(\mathbf{X}^{(\ell)})$ exponentially converges to zero in practice.

Rusch et al. (2023a) introduce one more node similarity measure:

$$\mu_{\texttt{rusch}}(\mathbf{X}) \coloneqq \sqrt{\frac{1}{N}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\|\mathbf{x}_i-\mathbf{x}_j\|_F^2}\,, \tag{9}$$

which is also called Dirichlet energy but has a different form with $\mu_{\texttt{cai}}(\mathbf{X})$. $\mu_{\texttt{cai}}(\cdot)$ satisfies $\mu_{\texttt{wu}}(\mathbf{X}) = 0$ if and only if Theorem B.2 holds. Although the measure was not utilized in theoretical analysis, Rusch et al. (2023a) provide the experimental results that $\mu_{\texttt{rusch}}(\mathbf{X}^{(\ell)})$ converges exponentially to zero when ℓ increases in GCN, GAT, and GraphSAGE (Hamilton et al., 2017).

C Reproducing the Depth Experiments of Kipf & Welling (2017)

Setup. We follow the setup reported by Kipf & Welling (2017): 5-fold cross-validation using all labels on Cora, CiteSeer, and PubMed, 400 training epochs without early stopping, the Adam optimizer with a learning rate of 0.01, dropout of 0.5 on the first and last layers, L2 regularization of 5×10^{-4} on the first layer, 16 hidden units, and Glorot initialization. We vary the depth from 1 to 10 and consider both the standard GCN and GCN with residual connections between hidden layers, as in their work. For comparison, we also train MLPs of the same depths with the same hyperparameters.

The reported drop appears only under particular splits. The official implementation fixes neither the data split nor the random seed. Under most splits, GCN remains fairly robust up to depth 10, and reproducing the reported drop requires searching over the split seed. Among 500 random seeds, only four produce a drop of at least 10 percentage points on all three datasets simultaneously. Figure 3 shows the results on the splits that best reproduce the reported curves for each dataset. Even on these splits, MLPs begin to degrade at shallower depths than GCNs and degrade far more severely.

The drop largely disappears with a validation set. The setup of Kipf & Welling (2017) does not use a validation set, and thus applies neither early stopping nor learning rate tuning. On the same splits, we hold out one of the five folds for validation (resulting in a 60%/20%/20% split), apply early stopping with a patience of 100 epochs, and tune the learning rate within $\{0.0005, 0.001, 0.005, 0.01\}$. As shown in Figure 4, this recovers most of the lost accuracy at deep layers. Note that these results are obtained with the standard GCN, without the residual connections; the drop is thus largely resolved without any architectural change.

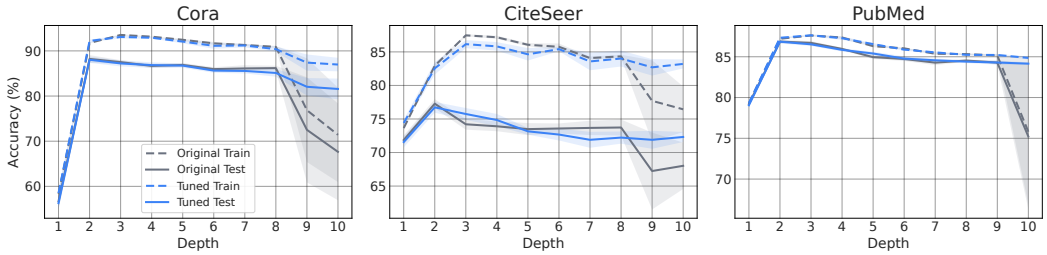

Figure 4: Training (dashed) and test (solid) accuracy of GCN on the same splits as Figure 3. *Original* follows the setup of Kipf & Welling (2017) without a validation set, and *Tuned* uses validation-based early stopping and learning rate tuning. Shaded regions indicate the standard error over the five folds.

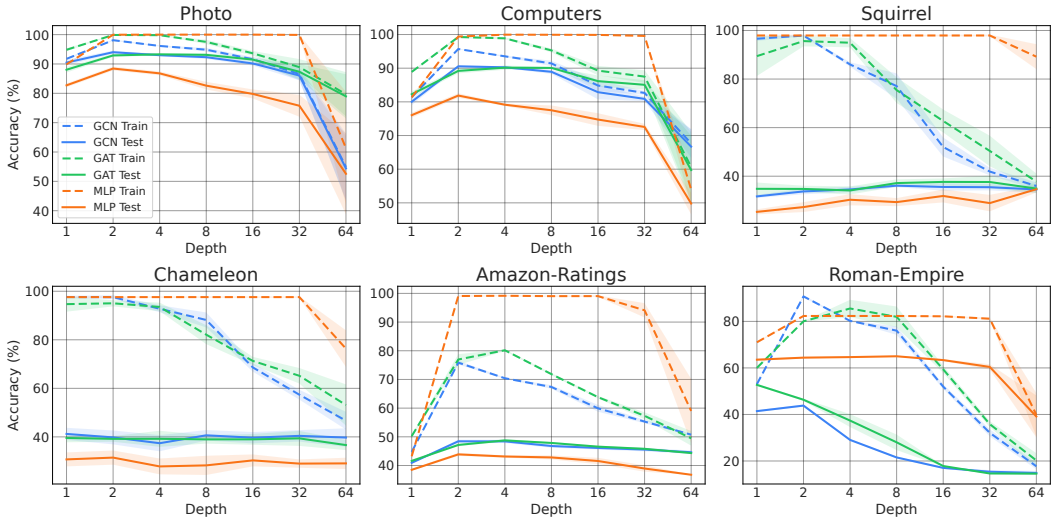

Figure 5: Training and test accuracy of GCN, GAT, and MLP across varying depths (1-64 layers) on two homophilic datasets, Photo and Computers, and four heterophilic datasets, Squirrel, Chameleon, Amazon-ratings, and Roman-empire. Dashed lines represent average training accuracy and solid lines represent average test accuracy over 5 repeated runs. Shaded regions indicate the standard deviation around the average.

D Supplementary Experimental Results

D.1 Supplementary Experimental Results of Depth-Performance

We provide supplementary experimental results extending Figure 1. Following the same setup, we further evaluate on two additional homophilic datasets, Photo and Computers, and four heterophilic datasets, Squirrel, Chameleon, Amazon-ratings, and Roman-empire. All models use He initialization (He et al., 2015) for weight parameters. The results are illustrated in Figure 5.

As summarized in Section 3.1, the comparative pattern holds on most of these datasets: GNNs maintain higher test accuracy than MLPs at deep layers even when their training accuracy decreases. In particular, on heterophilic graphs, GNNs show sharper declines in training accuracy at deeper depths than on homophilic graphs, yet their test accuracy is largely preserved in most cases.

The only exception is Roman-empire, where the test accuracy of GNNs degrades more substantially with depth than that of MLPs. However, it is difficult to regard a property observed on a single dataset as a general characteristic of GNNs. This behavior may be related to the chain-like structure of Roman-empire, whose edges mainly connect consecutive or syntactically dependent words in a single long text. Further study is needed to understand which graph structures make deep GNNs struggle.

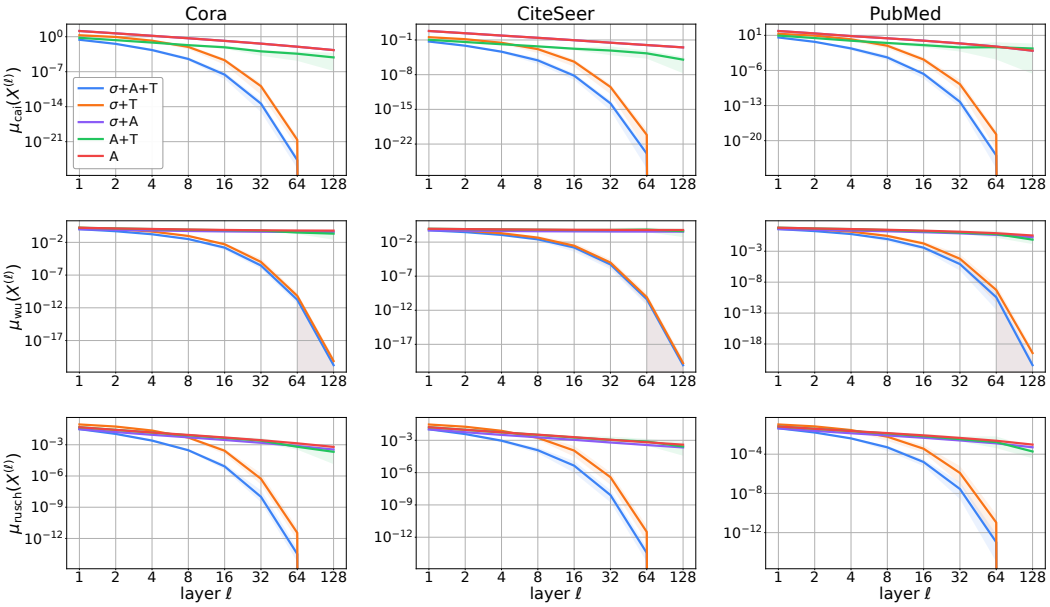

Figure 6: Oversmoothing levels measured by three metrics across 128-layer GCN and GAT evaluated on three datasets: Cora, CiteSeers, and PubMed. For each model, we compare five variations based on the combination of aggregation (A), transformation (T), and activation (σ) steps. For example, A+T refers to a model with aggregation and transformation steps. The solid curves indicate the average over 10 repeated runs, and shaded regions indicate the min-max range.

D.2 Supplementary experimental results for exponential oversmoothing

We provide supplementary experimental results extending Figure 2. Following the same setup, we compute node similarity measures from 128-layer GCN and GAT models at depths $\ell = 1, 2, 4, \cdots, 128$ on Cora, CiteSeer, PubMed, Photo, Computers, Squirrel, Chameleon, Amazon-Ratings, and Roman-Empire datasets: $\mu_{\texttt{cai}}(\mathbf{X}^{(\ell)})$ for GCN, and $\mu_{\texttt{wu}}(\mathbf{X}^{(\ell)})$ and $\mu_{\texttt{rusch}}(\mathbf{X}^{(\ell)})$ for GAT. The results are illustrated in Figure 6, Figure 7 and Figure 8.

D.3 Supplementary experimental results for exponential oversmoothing with He initialization

We demonstrate that He initialization can mitigate exponential oversmoothing in GNNs. We compute node similarity measures from 128-layer GCN and GAT models at depths $\ell = 1, 2, 4, \cdots, 128$ on Cora, CiteSeer, PubMed, Photo, Computers, Squirrel, Chameleon, Amazon-Ratings, and Roman-Empire datasets: $\mu_{\texttt{cai}}(\mathbf{X}^{(\ell)})$ for GCN, and $\mu_{\texttt{wu}}(\mathbf{X}^{(\ell)})$ and $\mu_{\texttt{rusch}}(\mathbf{X}^{(\ell)})$ for GAT. Figure 9, Figure 10, and Figure 11 show the changes of the node similarity measure across layers in GCN and GAT, respectively, with He initialization adopted. We observe that the decay rate significantly decreases compared to results with Glorot initialization shown in Figure 2.

E Node Similarity Measures after Training

We measure $\mu_{\texttt{cai}}(\mathbf{X}^{(\ell)})$ of the five variants in Section 3.3 after training 64-layer GCNs on Cora, CiteSeer, and PubMed, under He and Glorot initialization. The experimental details are provided in Appendix F. The results are illustrated in Figure 12.

With He initialization, the measures of σ+A+T, σ+T, and A+T no longer decay toward zero after training, and instead grow by several orders of magnitude at deeper layers. Only the variants without transformation (σ+A and A) still decay, at the same slow rate as before training, since aggregation in GCN has no learnable parameters. With Glorot initialization, training barely proceeds, yielding

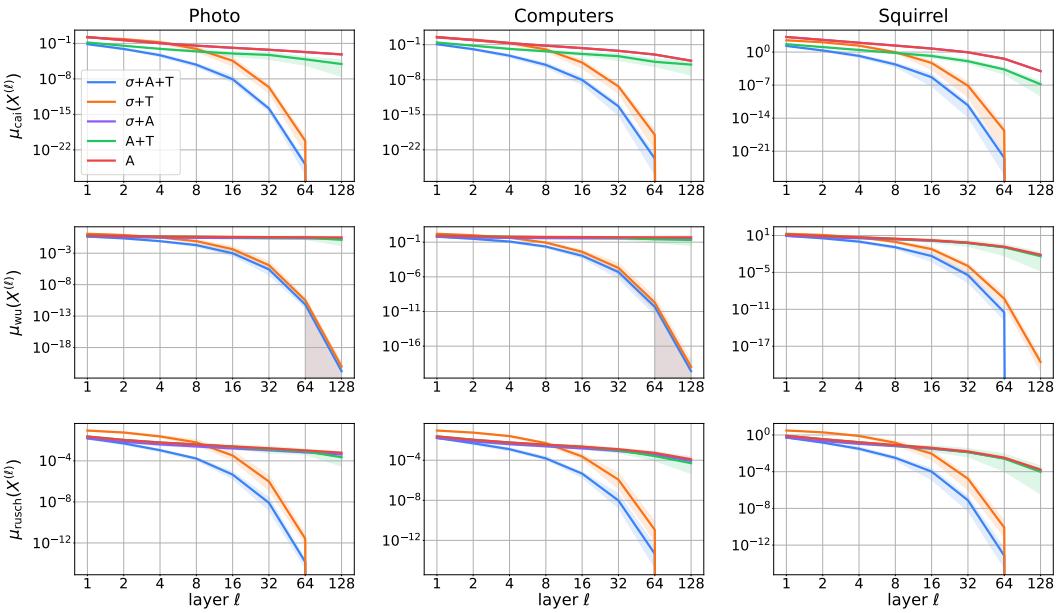

Figure 7: Oversmoothing levels measured by three metrics across 128-layer GCN and GAT evaluated on three datasets: Photo, Computers, and Squirrel. For each model, we compare five variations based on the combination of aggregation (A), transformation (T), and activation (σ) steps. For example, A+T refers to a model with aggregation and transformation steps. The solid curves indicate the average over 10 repeated runs, and shaded regions indicate the min-max range.

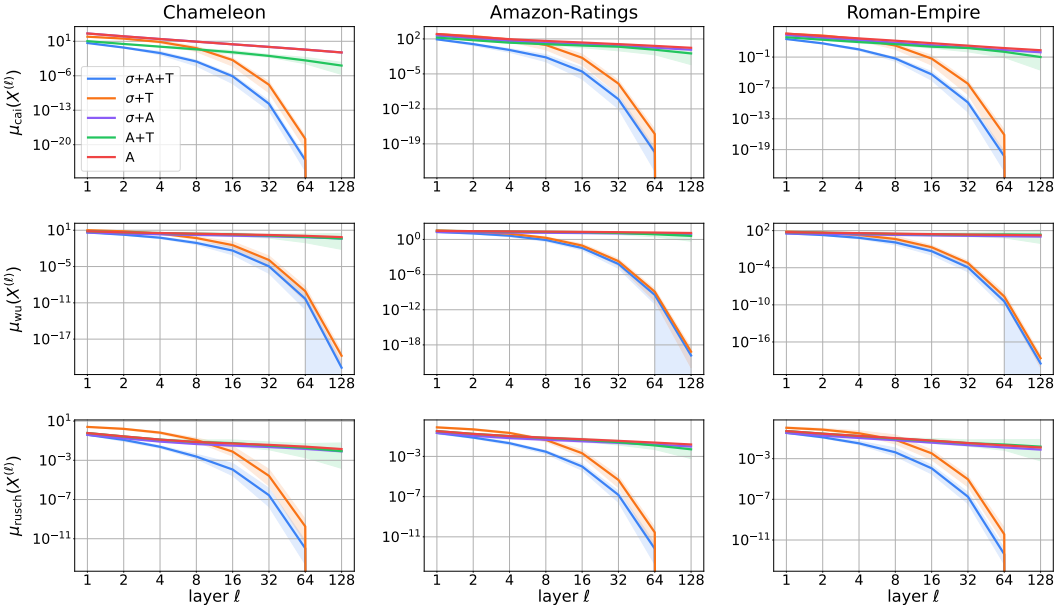

Figure 8: Oversmoothing levels measured by three metrics across 128-layer GCN and GAT evaluated on three datasets: Chameleon, Amazon-Ratings, and Roman-Empire. For each model, we compare five variations based on the combination of aggregation (A), transformation (T), and activation (σ) steps. For example, A+T refers to a model with aggregation and transformation steps. The solid curves indicate the average over 10 repeated runs, and shaded regions indicate the min-max range.

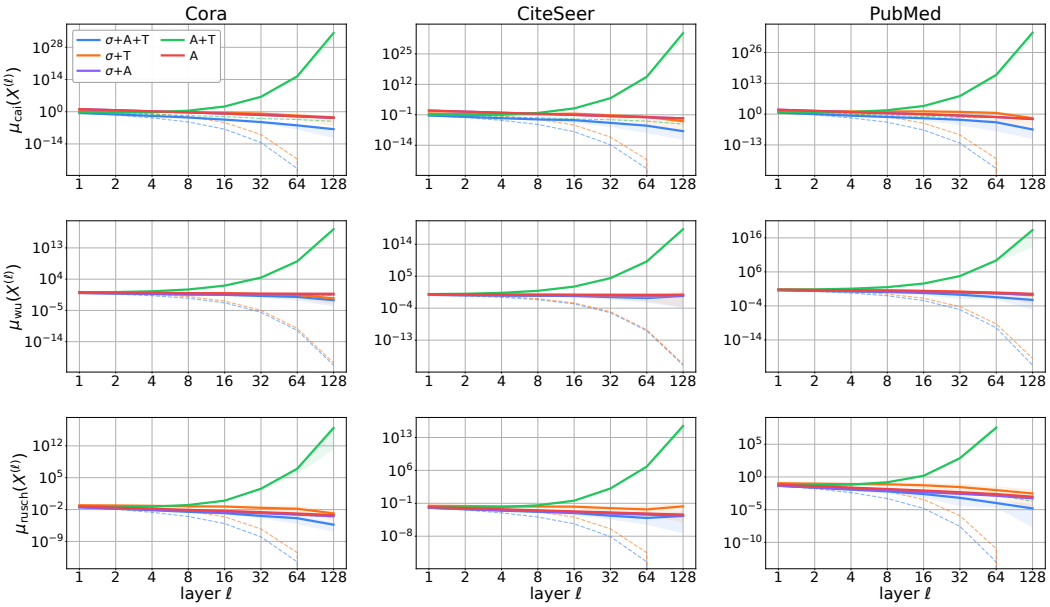

Figure 9: Oversmoothing levels, measured by three metrics across 128-layer GNN model with He initialization, evaluated on three datasets: Cora, CiteSeer, and PubMed. Solid lines denote He initialization and dashed lines denote the corresponding results with Glorot initialization. For each model, we compare five variations based on the combination of aggregation (A), transformation (T), and activation (σ) steps. For example, A+T refers to a model with only aggregation and transformation steps. The solid curves indicate the average over 5 repeated runs, and shaded regions indicate the min-max range.

test accuracy near chance level ($20.0 \pm 10.7\%$, $16.9 \pm 1.3\%$, and $39.4 \pm 0.6\%$ on Cora, CiteSeer, and PubMed, respectively), and the measures collapse to zero as in the untrained networks.

Based on these results, we conjecture that the exponential oversmoothing that Wu et al. (2023) report in trained networks arises because training does not proceed due to zero-collapsing under Glorot initialization, although the initialization is not specified in their work.

F Experimental Details

In this section, we describe the details of our training setup for experiments in Section 3.1 and Section 3.3. Our experiments were conducted on an AMD EPYC 7513 32-core processor and a single NVIDIA RTX A5000 or RTX 3090 GPU with 24GB of memory.

To obtain the results shown in Figure 1 in Section 3.1 and Figure 5, we train each model on five train/validation/test splits per dataset. For Cora, CiteSeer, PubMed, Computers, and Photo, we randomly split the nodes into $60\%/20\%/20\%$ for training, validation, and test, using five fixed random seeds. This follows the depth experiments of Kipf & Welling (2017), which use all labels with a 20% test set in each fold of 5-fold cross-validation, except that we additionally hold out 20% of the nodes as a validation set for early stopping and learning rate selection. For Squirrel, Chameleon, Amazon-ratings, and Roman-empire, we use the first five of the ten splits provided by Platonov et al. (2023). All models are vanilla architectures without bias terms, dropout, weight decay, normalization layers, or residual connections. We configure GAT with a single attention head, fix the hidden dimension to 64, and vary the depth within $\{1, 2, 4, \cdots, 64\}$. We train with the Adam optimizer for a maximum of $2,000$ epochs, using early stopping on the validation accuracy with a patience of $1,000$ epochs, and report the test accuracy of the checkpoint with the highest validation accuracy. For each model, dataset, and depth, we search the learning rate within $\{0.0005, 0.001, 0.005, 0.01\}$ and select the one that achieves the highest training accuracy.

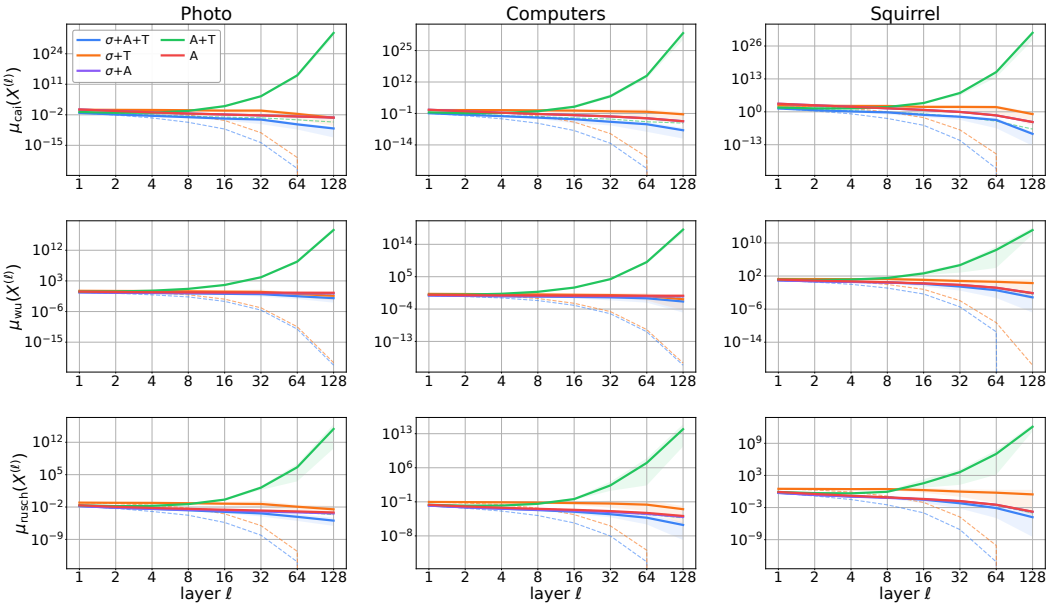

Figure 10: Oversmoothing levels, measured by three metrics across 128-layer GNN model with He initialization, evaluated on three datasets: Photo, Computers, and Squirrel. For each model, we compare five variations based on the combination of aggregation (A), transformation (T), and activation (σ) steps. For example, A+T refers to a model with only aggregation and transformation steps. The solid curves indicate the average over 5 repeated runs, and shaded regions indicate the min-max range.

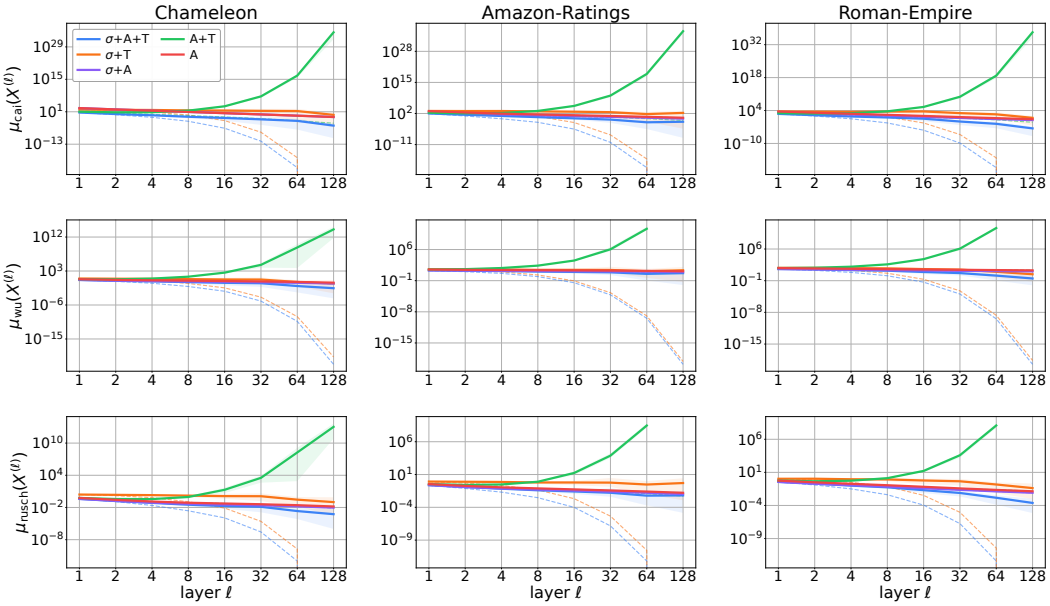

Figure 11: Oversmoothing levels, measured by three metrics across 128-layer GNN model with He initialization, evaluated on three datasets: Chameleon, Amazon-Ratings, and Roman-Empire. For each model, we compare five variations based on the combination of aggregation (A), transformation (T), and activation (σ) steps. For example, A+T refers to a model with only aggregation and transformation steps. The solid curves indicate the average over 5 repeated runs, and shaded regions indicate the min-max range.

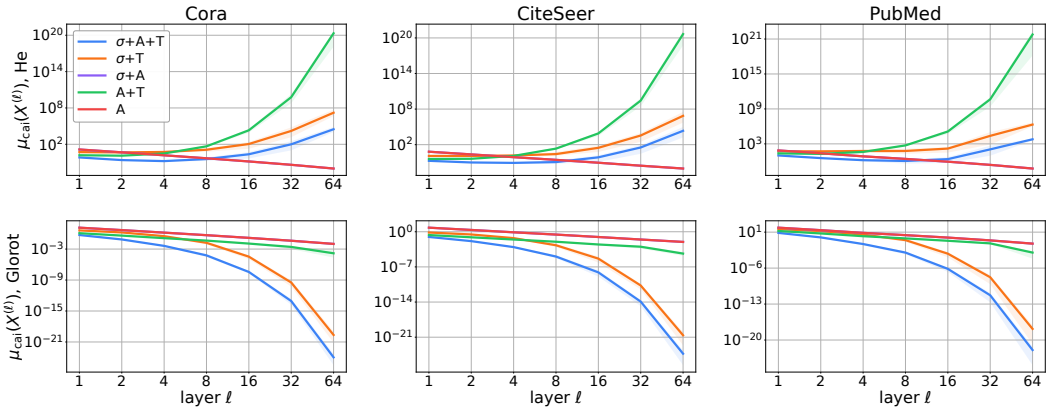

Figure 12: Oversmoothing levels measured by $\mu_{\texttt{cai}}$ in 64-layer GCNs after training, evaluated on Cora, CiteSeer, and PubMed. The top and bottom rows show the results with He and Glorot initialization, respectively. For each model, we compare five variations based on the combination of aggregation (A), transformation (T), and activation (σ) steps. The solid curves indicate the average over 5 repeated runs, and shaded regions indicate the min-max range.

We initialize the weights of all models with He initialization (He et al., 2015), i.e., Kaiming uniform initialization with fan-in mode and the gain for ReLU. Note that the default initializations of the libraries we use differ from this. PyTorch Geometric (Fey & Lenssen, 2019) initializes the weights of GCN and GAT with Glorot initialization (Glorot & Bengio, 2010), while PyTorch (Paszke et al., 2019) initializes linear layers, which constitute the MLP, with a variant of Kaiming uniform initialization whose variance is one sixth of that of He initialization for ReLU, so that does not correspond to He initialization for ReLU networks. We therefore explicitly re-initialize the weights of all three models with He initialization, so that GCN, GAT, and MLP share the same initialization scheme.

To obtain the results presented in Figure 2 in Section 3.3, we measure node similarity without training, since our goal is to answer the main question of Section 3.3: *what if aggregation causes oversmoothing so rapidly in initialized networks that optimization cannot even begin?* Following Wu et al. (2023), we set the hidden dimension to 32 and use a single-head configuration for GAT.

To obtain the results presented in Figure 12 in Appendix E, we train 64-layer GCNs with a hidden dimension of 64, following the training setup of Figure 1. With He initialization, we use the configuration with the highest validation accuracy from a grid search over the learning rate $\{0.0005, 0.001, 0.005, 0.01\}$, dropout $\{0, 0.5\}$, weight decay $\{0, 5 \times 10^{-4}\}$, and hidden dimension $\{16, 64\}$. With Glorot initialization, we use no dropout or weight decay and select the learning rate within $\{0.0005, 0.001, 0.005, 0.01\}$ by validation accuracy.

G Dataset Statistics

We provide detailed statistics and explanations about the dataset used for the experiments in Table 1 and the paragraphs below.

Cora, CiteSeer, and PubMed Each node represents a paper, and an edge indicates a reference relationship between two papers. The task is to predict the research subjects of the papers.

Computers and Photo Each node represents a product, and an edge indicates a high frequency of concurrent purchases of the two products. The task is to predict the product category.

Squirrel and Chameleon Each node represents a Wikipedia page, and an edge indicates a link between two pages. The task is to predict the monthly traffic for each page. We use the classification version of the dataset, where labels are converted by dividing monthly traffic into five bins. We adopted the filtering process to prevent train-test data leakage as recommended by (Platonov et al., 2023).

Table 1: Statistics of the datasets utilized in the experiments.

Dataset	# nodes	# edges	# features	# classes
Cora	2,708	5,278	1,433	7
CiteSeer	3,327	4,552	3,703	6
PubMed	19,717	44,324	500	3
Computers	13,752	245,861	767	10
Photo	7,650	119,081	745	8
Squirrel	2,223	46,998	2,089	5
Chameleon	890	8,854	2,325	5
Roman-empire	22,662	32,927	300	18
Amazon-ratings	24,492	93,050	300	5

Roman-empire Each node represents a word extracted from the English Wikipedia article on the Roman Empire, and an edge indicates a grammatical or sequential relationship between words. The task is to predict the part-of-speech tag of each word.

Amazon-ratings Each node represents a product from the Amazon co-purchasing network, and an edge indicates a frequent co-purchase between two products. The task is to predict the product category based on user co-purchasing patterns.