Understanding Oversmoothing in Deep Graph Neural Networks: Insights from Sparsity, Weights, and Lottery Tickets

Ana Carolina Ramos Cunha

February 22, 2026

1 Introduction

Graph Neural Networks (GNNs) have become a fundamental tool for learning on graph-structured data, achieving strong performance in node classification, link prediction, and graph-level tasks. Architectures such as Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and Graph Isomorphism Networks (GINs) rely on iterative message passing to aggregate information from local neighborhoods. While effective in shallow settings, deep GNNs suffer from a critical limitation known as *oversmoothing*.

Oversmoothing refers to the phenomenon in which node representations become increasingly indistinguishable as the number of layers grows. Repeated neighborhood aggregation causes embeddings to converge toward a low-dimensional subspace dominated by low-frequency graph components, ultimately leading to representation collapse and severe performance degradation. As a result, standard GNN architectures typically fail beyond 5–8 layers.

Recent work has suggested that oversmoothing may not be an inherent structural limitation of GNNs, but rather a consequence of weight optimization dynamics. In particular, the Untrained Graph Neural Networks Tickets (UGTs) framework demonstrates that sparse subnetworks with randomly initialized and frozen weights can outperform fully trained dense networks in deep regimes. This challenges the conventional assumption that training dense weights is necessary for strong performance.

In this report, we reproduce the main experimental findings of the UGTs paper and investigate whether sparsity, initialization, or training dynamics are the primary drivers of oversmoothing. We then extend the analysis by studying additional sparsification strategies and weight reparameterization techniques to better understand the mechanisms that enable deep GNNs to remain stable.

Roadmap. Section 2 situates our work within the broader literature on oversmoothing and lottery tickets in GNNs. Section 3 details our reproduction of the original UGT results. Section 4 presents our extensions, including Unified Graph Sparsification (UGS), Weight Reparameterization (WeightRep), and initialization studies. We conclude in Section 6 with a synthesis of findings and directions for future work.

2 Related Work

2.1 Oversmoothing in Graph Neural Networks

Oversmoothing has been extensively studied as a fundamental limitation of deep GNNs. Early analyses linked the phenomenon to repeated multiplication by graph propagation operators, which act as low-pass filters that suppress high-frequency components of node signals. Chen et al. (3) introduced the Mean Average Distance (MAD) metric to quantify representation smoothness and demonstrated that deep GNNs rapidly collapse node embeddings.

Subsequent theoretical work connected oversmoothing to spectral contraction and rank deficiency of learned representations. These analyses suggest that depth amplifies smoothing effects inherent in message passing, raising the question of whether architectural modifications or training strategies can counteract this collapse.

2.2 Lottery Tickets and Sparsity in GNNs

The Lottery Ticket Hypothesis, originally proposed for dense neural networks, suggests that sparse subnetworks exist within randomly initialized models that can match full-network performance. Ramanujan et al. (2) showed that subnetworks selected without weight training can perform competitively when appropriate masks are learned.

Huang et al. (1) extended this idea to graph neural networks through Untrained GNN Tickets (UGTs). In UGTs, weights remain randomly initialized and frozen, while binary masks over edges are learned via task loss. Surprisingly, such untrained sparse networks can outperform fully trained dense GNNs at greater depths, suggesting that training dynamics may exacerbate oversmoothing.

Chen et al. (4) proposed Unified Graph Sparsification (UGS), which jointly sparsifies model parameters and graph structure through gradient-based training. Unlike UGTs, UGS retains weight optimization, providing a bridge between dense training and untrained sparsity.

More recently, Zhuo et al. (5) introduced Weight Reparameterization (WeightRep), constructing input-dependent weights during training to preserve representation diversity. This approach demonstrates that oversmoothing can be mitigated even under training, provided weights evolve in a structurally appropriate manner.

2.3 Positioning Our Work

Our study builds on these lines of research by systematically comparing: Trained dense GNNs, Untrained sparse GNNs (UGTs), Edge-Popup sparsification, Unified Graph Sparsification (UGS), Weight reparameterization and initialization strategies.

By reproducing the core UGT findings and extending the analysis to additional sparsification and weight-control mechanisms, we aim to disentangle the roles of sparsity, initialization, and training in the emergence of oversmoothing.

3 Reproduction

We reproduce the central experimental results of Huang et al. (1), focusing on three key analyses: 1. Accuracy as a function of network depth, 2. Mean Average Distance (MAD) as a function of depth and training epoch, 3. Accuracy as a function of sparsity level.

Our reproduction covers three architectures (GCN, GAT, GIN) and three citation datasets (Cora, Citeseer, Pubmed), following the original evaluation protocol.

3.1 Formal Setting

Let $G = (V, E)$ denote a graph with adjacency matrix $A \in \mathbb{R}^{n \times n}$ and node feature matrix $X \in \mathbb{R}^{n \times d}$. A L-layer GNN computes node representations iteratively as

$$H^{(0)} = X, \tag{1}$$

$$H^{(\ell+1)} = \sigma\left(\mathcal{P}(A)H^{(\ell)}W^{(\ell)}\right), \tag{2}$$

where: $\mathcal{P}(A)$ is a graph propagation operator (e.g., normalized adjacency for GCN), $W^{(\ell)}$ are learnable weights, σ is a nonlinear activation function.

The model is trained by minimizing cross-entropy loss:

$$\mathcal{L}(\theta) = \sum_{i \in \mathcal{V}_{\text{train}}} \text{CE}\big(f_\theta(X, A)_i, y_i\big), \tag{3}$$

where $\theta = \{W^{(\ell)}\}_{\ell=1}^{L}$.

Here, $f_\theta : \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times C}$ denotes the full L-layer GNN followed by a linear classification layer, mapping node features and the adjacency matrix to class logits for each node, where C is the number of classes.

3.2 Untrained GNN Tickets (UGTs)

In UGTs, weights are randomly initialized and *never updated*. Instead, each weight matrix is paired with a trainable score matrix $S^{(\ell)}$.

A binary mask $m^{(\ell)}$ is constructed via global thresholding:

$$m_{ij}^{(\ell)} = \mathbf{1}\left[S_{ij}^{(\ell)} \geq \tau\right], \tag{4}$$

where τ enforces a desired sparsity level.

The threshold τ is chosen such that a fixed global sparsity level $s \in [0, 1]$ is enforced across all layers, meaning that only the top $(1 - s)$ fraction of scores are retained.

The effective weights become:

$$\widetilde{W}^{(\ell)} = W^{(\ell)} \odot m^{(\ell)}. \tag{5}$$

Only the scores are updated:

$$S \leftarrow S - \eta \nabla_S \mathcal{L}\big(f(X, A; W \odot m), y\big), \tag{6}$$

using a straight-through estimator for the binary mask.

We evaluate sparsity levels of 0.1 and 0.5, and additionally sweep sparsity from 0 to 0.99 for robustness analysis.

3.3 Edge-Popup

Edge-Popup also freezes weights and learns scores, but differs in three aspects: layer-wise top-k thresholding instead of global thresholding, one-shot sparsification instead of gradual pruning, mask construction using $|S|$ instead of raw scores.

We additionally implement a variation using raw scores (without absolute value), observing improved performance at lower sparsities.

3.4 Mean Average Distance (MAD)

To quantify representation smoothness across layers, we adopt the Mean Average Distance (MAD) metric defined in Chen et al. (AAAI 2020).

Let $H^{(\ell)} \in \mathbb{R}^{n \times d}$ denote the matrix of node embeddings at layer ℓ, where $H_i^{(\ell)}$ is the embedding of node i.

First, define the cosine distance between two node embeddings:

$$D_{ij}^{(\ell)} = 1 - \frac{H_i^{(\ell)} \cdot H_j^{(\ell)}}{\|H_i^{(\ell)}\|_2 \, \|H_j^{(\ell)}\|_2}. \tag{7}$$

Let $M_{\text{tgt}} \in \{0,1\}^{n \times n}$ be a target mask matrix (e.g., excluding self-pairs or restricting to specific node sets). The masked distance matrix is:

$$\widetilde{D}^{(\ell)} = M_{\text{tgt}} \odot D^{(\ell)}. \tag{8}$$

For each node i, define the row-wise mean over non-zero entries:

$$\text{MAD}_i^{(\ell)} = \frac{\sum_{j=1}^{n} \widetilde{D}_{ij}^{(\ell)}}{\sum_{j=1}^{n} M_{\text{tgt},ij}}. \tag{9}$$

The overall Mean Average Distance at layer ℓ is:

$$\text{MAD}^{(\ell)} = \frac{1}{n} \sum_{i=1}^{n} \text{MAD}_i^{(\ell)}. \tag{10}$$

Lower MAD indicates stronger representation collapse.

3.5 Experimental Protocol

We follow the original setup: depth varied from 2 to 20 layers, MAD computed for a 32-layer GCN on Cora, accuracy averaged across multiple runs, Kaiming uniform initialization for frozen weights, identical optimizer and learning rates as the original implementation.

3.6 Reproduced Results

3.6.1 Accuracy vs Depth

Trained dense GNNs exhibit severe degradation beyond 5–8 layers, with accuracy dropping sharply at greater depths. In contrast, UGTs maintains stable performance up to 20 layers across architectures and datasets. Sparsity 0.5 often yields the strongest deep performance.

Edge-Popup improves over dense training but consistently underperforms UGTs in deep regimes.

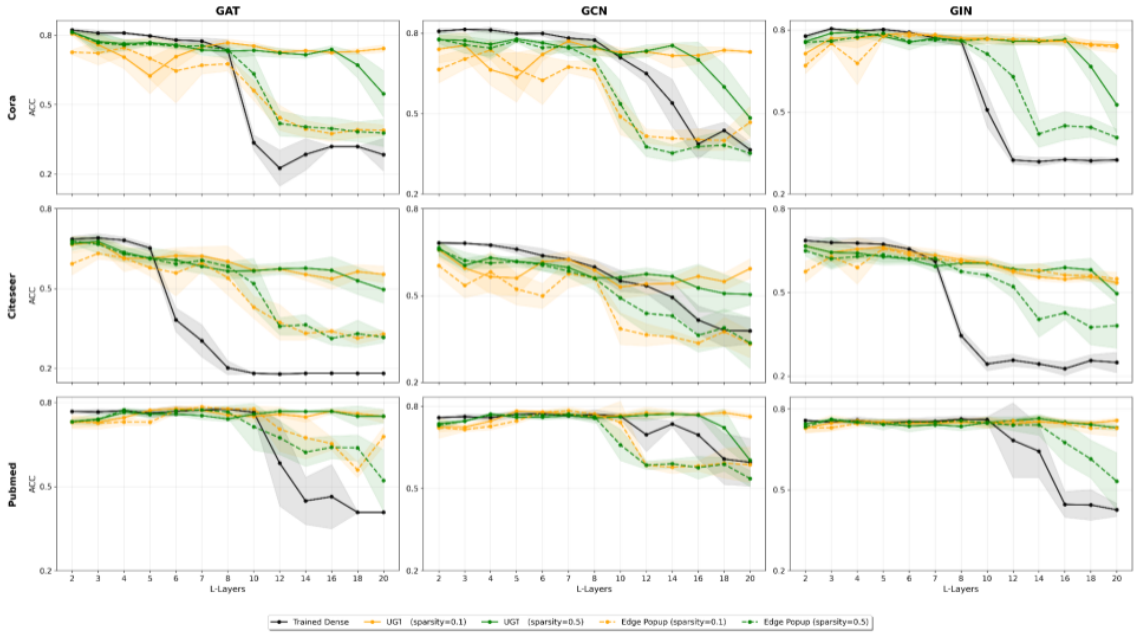

Figure 1: Accuracy versus number of layers for GCN, GAT, and GIN across datasets. Reproduced from our implementation (corresponding to Figure 2 in (1)).

3.6.2 MAD Analysis

Dense GCNs rapidly exhibit representation collapse, with MAD approaching zero within the first few layers. UGTs maintains significantly higher MAD across depth and training epochs, indicating preserved feature diversity.

These findings confirm that oversmoothing is strongly mitigated when weights remain untrained and sparsity is imposed.

3.6.3 Accuracy vs Sparsity

UGTs matches or slightly underperforms trained dense networks across sparsity levels from 0.1 to 0.9. Performance degrades gradually at extreme sparsity (0.95–0.99).

Edge-Popup exhibits earlier degradation, particularly on Citeseer.

Notably, using raw scores instead of $|S|$ improves Edge-Popup at lower sparsities, suggesting that signed score competition enhances expressivity.

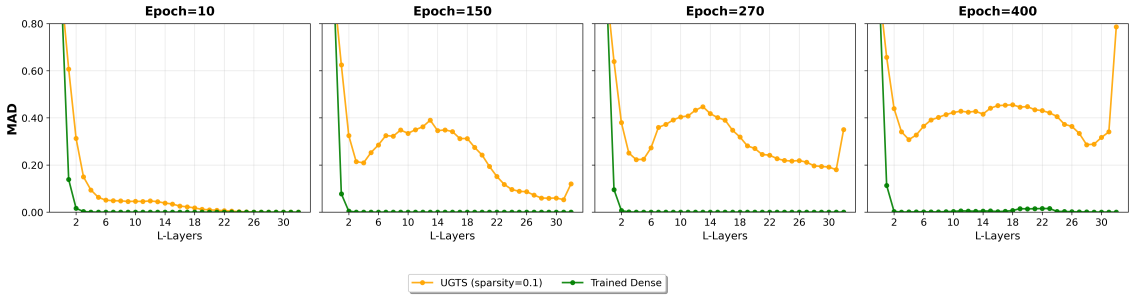

Figure 2: Mean Average Distance (MAD) across layers for a 32-layer GCN on Cora, evaluated at different training epochs. Dense training exhibits rapid representation collapse (MAD $\to 0$), while UGT maintains higher pairwise embedding distances across depth. Reproduction of Figure 4 from (1).

4 Extensions

While UGTs demonstrate that untrained sparse subnetworks can mitigate oversmoothing, several fundamental questions remain:

- Is sparsity alone sufficient to prevent oversmoothing under training?
- Can trained sparse models match the depth robustness of UGTs?
- Is oversmoothing primarily a structural issue or a learning dynamics issue?
- Can weight reparameterization or initialization strategies resolve collapse while retaining training?

To address these questions, we investigate three extensions: (1) Unified Graph Sparsification (UGS), (2) Weight Reparameterization (WeightRep) and Weight Initialization (WeightInit), (3) The interaction between UGT and structured initialization.

4.1 Unified Graph Sparsification (UGS)

4.1.1 Motivation

UGTs suggests that freezing random weights avoids representation collapse. However, this raises a key question: is the benefit due to sparsity itself, or due to the absence of weight training?

Unified Graph Sparsification (UGS) (4) jointly sparsifies both model weights and graph edges while retaining gradient-based weight optimization. UGS therefore separates the effects of sparsity from the absence of training.

4.1.2 Formal Setting

UGS introduces binary masks over both weights and adjacency:

$$\widetilde{W}^{(\ell)} = W^{(\ell)} \odot m_W^{(\ell)}, \tag{11}$$

$$\widetilde{A} = A \odot m_A, \tag{12}$$

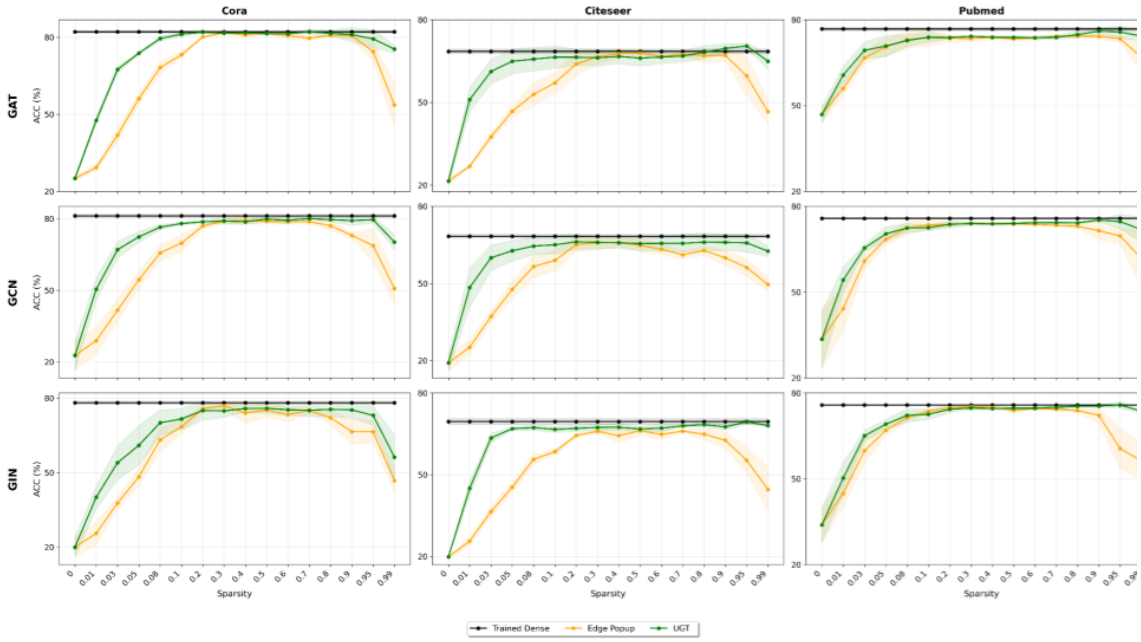

Figure 3: Test accuracy as a function of sparsity across architectures and datasets. UGT maintains competitive performance across a wide sparsity range, while Edge-Popup degrades earlier. Performance declines gradually only at extreme sparsity levels (0.95–0.99). Reproduction of Figure 5 from (1).

and optimizes:

$$\min_{m_W, m_A} L\left(f(X, A \odot m_A; W \odot m_W), y\right) \quad \text{s.t.} \quad \|m_W\|_0 \leq s_W, \ \|m_A\|_0 \leq s_A. \tag{13}$$

Here, $\|\cdot\|_0$ denotes the ℓ_0 pseudo-norm, which counts the number of nonzero entries in the corresponding binary mask.

After training, hard pruning is applied.

4.1.3 Experimental Setup

We compare: trained dense GNN, trained sparse GNN via UGS, untrained sparse GNN (UGTs).

Depth is varied while holding dataset, backbone architecture, and training protocol fixed.

4.1.4 Findings

We observe that UGS collapses rapidly as depth increases, similar to trained dense GNNs. Although sparsity is enforced, representation collapse still occurs.

In contrast, UGTs remains stable across increasing depth.

This demonstrates that sparsity alone does not prevent oversmoothing under training. Instead, weight optimization appears to amplify feature homogenization, even when the model is sparse.

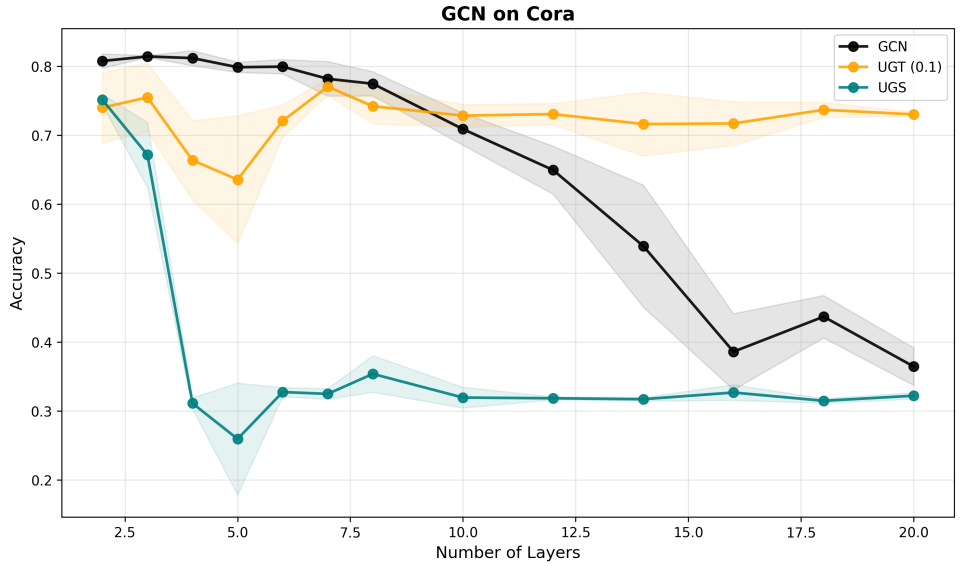

Figure 4: Accuracy versus depth comparing trained dense GNNs, UGS, and UGTs.

4.2 Weight Reparameterization and Initialization

4.2.1 Motivation

Recent work (5) suggests that oversmoothing is not inevitable if weights are constrained to preserve representation diversity. WeightRep dynamically reparameterizes weights to maintain feature correlation structure during training.

We compare: standard dense training, WeightInit (structured initialization only), WeightRep (dynamic reparameterization).

4.2.2 WeightRep Formulation

Let the pre-activation features at layer ℓ be:

$$\hat{H}^{(\ell)} = PH^{(\ell-1)}W^{(\ell)}, \tag{14}$$

where P is the normalized propagation matrix.

Define the covariance (Gram) matrix:

$$\Sigma^{(\ell)} = \hat{H}^{(\ell)\top}\hat{H}^{(\ell)} + \varepsilon I, \tag{15}$$

where $\varepsilon > 0$ ensures numerical stability.

Let $f(\cdot)$ be a spectral function applied to eigenvalues. A common choice is whitening with $f(\lambda) = \lambda^{-1/2}$.

The constructed transformation matrix is:

$$\hat{W}^{(\ell)} = \left(\Sigma^{(\ell)}\right)^{f}. \tag{16}$$

The updated representation is then:

$$H^{(\ell)} = \hat{H}^{(\ell)}\hat{W}^{(\ell)}. \tag{17}$$

Equivalently, this induces an effective weight matrix:

$$\tilde{W}^{(\ell)} = W^{(\ell)}\hat{W}^{(\ell)}, \tag{18}$$

so that:

$$H^{(\ell)} = PH^{(\ell-1)}\tilde{W}^{(\ell)}. \tag{19}$$

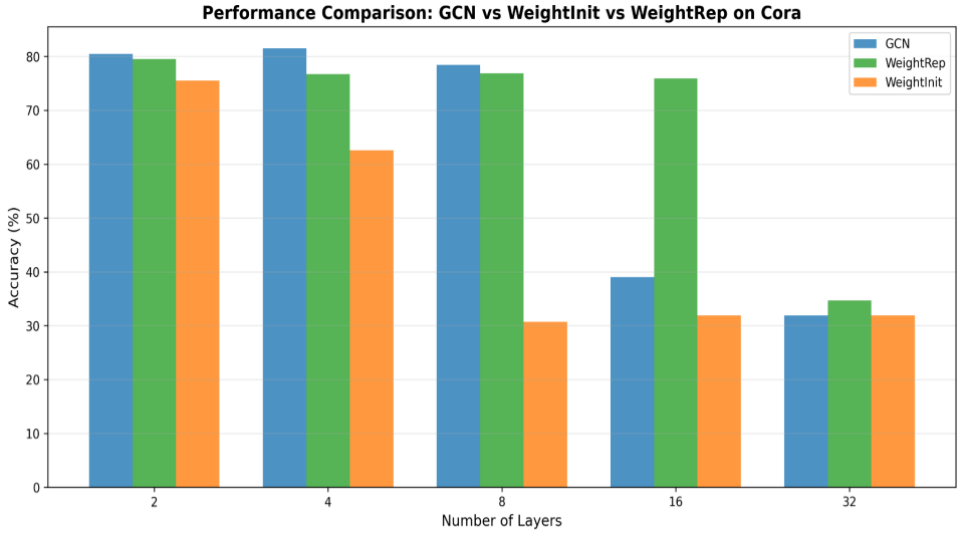

Figure 5: Depth performance comparison between dense training, WeightInit, and WeightRep.

4.3 WeightInit Formulation

Instead of modifying weights during training, we apply a one-time representation adjustment at initialization.

Let $W^{(\ell)}$ be initialized using a standard random initialization (e.g., Glorot initialization).

After computing the initial propagated features $\hat{H}^{(\ell)} = PH^{(\ell-1)}W^{(\ell)}$, we construct:

$$\Sigma^{(\ell)} = \hat{H}^{(\ell)\top}\hat{H}^{(\ell)} + \varepsilon I, \tag{20}$$

and define:

$$\hat{W}^{(\ell)} = \left(\Sigma^{(\ell)}\right)^{f}. \tag{21}$$

The initialized representation becomes:

$$H^{(\ell)} = \hat{H}^{(\ell)}\hat{W}^{(\ell)}. \tag{22}$$

4.3.1 Findings

Standard dense training degrades rapidly with depth. WeightInit provides limited improvement, indicating that careful initialization alone cannot prevent collapse.

WeightRep, however, maintains strong performance at larger depths.

This suggests that oversmoothing arises from training dynamics rather than purely architectural limitations. Proper weight evolution during training is critical.

4.4 UGT Initialization Study

Finally, we investigate whether UGT robustness depends on random initialization.

We evaluate UGT under standard Kaiming initialization and WeightInit-style structured initialization.

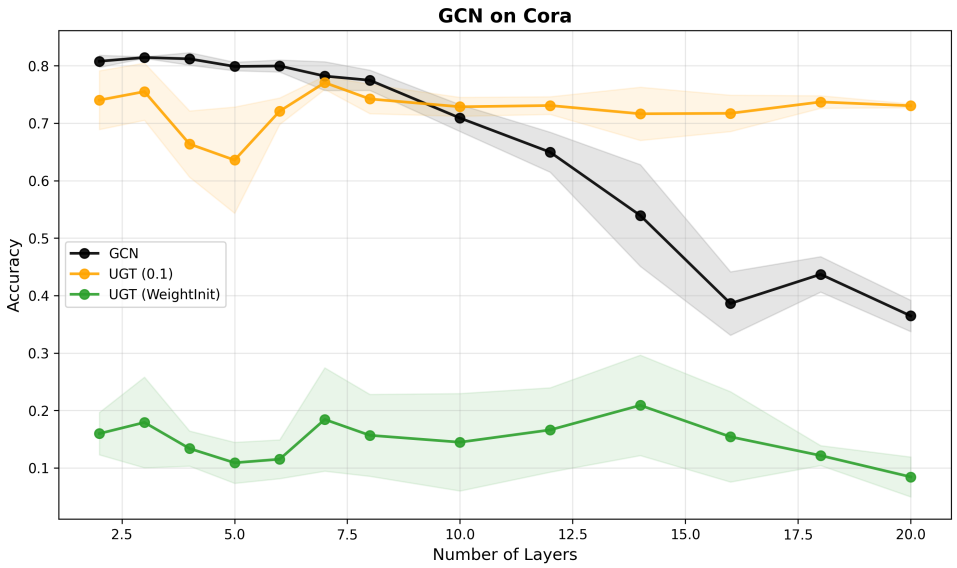

Figure 6: UGT performance under different initialization strategies.

4.4.1 Findings

Standard UGT remains robust across depth. Surprisingly, applying WeightInit severely degrades UGT performance.

This indicates that UGT robustness relies on high-entropy, unstructured random weights. Structured initialization constrains the representational space in a way that conflicts with mask-based selection.

Thus, UGT and WeightRep prevent oversmoothing through fundamentally different mechanisms: UGT avoids destructive training dynamics entirely, while WeightRep modifies them.

4.5 Synthesis of Extension Results

Across all experiments, a consistent pattern emerges:

- Dense training amplifies representation collapse.
- Sparsity without training (UGTs) mitigates collapse.
- Sparsity with training (UGS) does not prevent collapse.
- Structured dynamic weights (WeightRep) can preserve diversity under training.

These results suggest that oversmoothing is primarily a learning dynamics failure rather than a structural inevitability of message passing.

Controlling representation evolution — either by eliminating weight training (UGTs) or constraining it dynamically (WeightRep) — is key to enabling deep GNNs.

5 Discussion

Our results collectively suggest that oversmoothing is not an unavoidable consequence of depth in Graph Neural Networks, but rather a byproduct of how representations evolve under training dynamics.

5.1 Is Oversmoothing Structural or Learned?

Traditional explanations attribute oversmoothing to repeated multiplication by graph propagation operators, which act as low-pass filters. While this spectral perspective explains why representations contract with depth, our experiments reveal that structural smoothing alone does not fully account for collapse.

Specifically:

- UGTs — which use the same propagation operators as dense GNNs — do not collapse at comparable depths.
- UGS — which enforces sparsity but retains training — still collapses.

Thus, sparsity of connectivity is insufficient on its own. The critical factor appears to be *how weights are optimized*.

5.2 The Role of Training Dynamics

Training dense weights introduces strong coupling between layers. Gradients encourage homogenization of features across neighborhoods, accelerating the contraction of representation space.

UGTs avoid this phenomenon entirely by freezing weights. Instead of adapting weights, they adapt connectivity via mask learning. This decouples representation propagation from destructive weight updates.

WeightRep provides further evidence that training can be compatible with depth — provided weight evolution is constrained to preserve feature diversity. Unlike UGTs, which eliminate weight training, WeightRep modifies the geometry of weight updates.

5.3 Entropy and Initialization

Our UGT initialization study reveals a subtle but important insight: robustness depends on high-entropy, unstructured weights. When structured initialization (WeightInit) is imposed, UGT performance degrades.

This suggests that mask learning in UGT relies on a rich random feature basis from which useful subnetworks can be selected. Constraining this basis reduces expressivity.

6 Conclusion

In this work, we reproduced the principal findings of Untrained Graph Neural Networks Tickets and systematically investigated the mechanisms underlying oversmoothing in deep GNNs.

Our reproduction confirms three central observations:

1. Standard dense GNNs suffer severe performance degradation beyond moderate depth.
2. Untrained sparse subnetworks (UGTs) maintain stable accuracy even at 20+ layers.
3. UGTs preserves representation diversity, as measured by Mean Average Distance (MAD), whereas dense models rapidly collapse.

Building on this foundation, our extensions clarify the role of sparsity, initialization, and training:

- Sparsity alone does not prevent oversmoothing, as demonstrated by the rapid collapse of UGS under training.
- Proper weight evolution (WeightRep) enables deep training without collapse, indicating that oversmoothing is not structurally inevitable.
- UGT robustness depends on high-entropy random initialization, highlighting the importance of expressive random feature bases.

Collectively, these findings support a central thesis:

> Oversmoothing is primarily a learning dynamics failure rather than a fundamental architectural limitation of message passing.

Deep GNN failure arises when training drives representations toward a low-rank, homogeneous subspace. Preventing collapse requires either eliminating destructive weight updates (UGTs) or actively constraining weight evolution (WeightRep).

Future work may explore hybrid approaches that combine sparse connectivity, dynamic weight control, and theoretical guarantees on representation rank preservation. Understanding how representation geometry evolves during training remains a promising direction for building stable, deep graph models.

References

[1] T. Huang et al. You Can Have Better Graph Neural Networks by Not Training Weights at All: Finding Untrained GNNs Tickets. In Learning on Graphs Conference (LoG), 2022.

[2] V. Ramanujan et al. What's Hidden in a Randomly Weighted Neural Network? In CVPR, 2020.

[3] D. Chen et al. Measuring and Relieving the Over-smoothing Problem for Graph Neural Networks from the Topological View. In AAAI, 2020.

[4] T. Chen et al. A Unified Lottery Ticket Hypothesis for Graph Neural Networks. In ICML, 2021.

[5] Z. Zhuo et al. Graph Neural Networks (with Proper Weights) Can Escape Oversmoothing. In ACML, 2024.