GOING DEEPER, STAYING DISTINCT: GRAPH LEARNING THROUGH MINIMAL SUBSTITUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep graph models need to gather information from distant nodes while preserving the differences between node representations. In conventional message passing, this wider reach is obtained by repeatedly mixing neighboring features, which can gradually erase the distinctions needed for downstream tasks. We introduce **D**eep **M**inimal **S**ubstitution **G**raph **L**earning (DMSGL), which views deep propagation as dynamic succession within a complex network, assuming this succession follows a widely used yet structurally minimalist model of Minimal Substitution. Inspired by *Minimal Substitution*, we design a *Message-substitution* mechanism that treats the layer-wise evolution of node features as dynamic succession. Each layer incorporates nonlocal features while retaining the graph structure learned through local propagation. This broadens the information available to a node without tying long-range access entirely to repeated local smoothing. We analyze how representations vary under DMSGL and establish conditions for retaining discriminative information through deep propagation. Extensive experiments demonstrate that DMSGL maintains stable performance. Furthermore, as the model depth increases, the quality of the learned representations improves without succumbing to the over-smoothing problem typically associated with traditional message-passing mechanisms. Together, these results show that DMSGL supports going deeper while keeping node representations distinct.

1 INTRODUCTION

The depth of a graph model determines both how far information can travel and how node representations evolve. In message-passing GNNs, each layer updates a node from its neighbors, so stacking layers turns local exchanges into a graph-wide diffusion process (Gilmer et al., 2017). This process enables GNNs to encode relational structure for downstream tasks (Ju et al., 2024; Chen et al., 2024). It also creates a structural tension: repeated mixing reduces variation between node representations (known as over-smoothing) and weakens class separation (Oono & Suzuki, 2020; Wu et al., 2023). The central problem is to make depth expand information access while preserving the distinctions carried by node features.

Existing approaches control this tension by modifying the propagation process. Residual connections Yang et al. (2022) retain earlier states, random dropping changes message paths Feng et al. (2020); Do et al. (2021); Rong et al. (2020); Lu et al. (2024); Fang et al. (2023), normalization regulates feature variation Zhao & Akoglu (2020); Zhou et al. (2020); Oono & Suzuki (2020), and multi-hop or alternative dynamics alter the propagation operator Lu et al. (2024); Maskey et al. (2023); Tang et al. (2024); Choi et al. (2023). These methods preserve information as propagation deepens. The layerwise evolution of node features, however, remains expressed mainly as repeated aggregation, leaving the succession of feature states across the network implicit. Modeling this succession offers a direct way to rethink how deep graph propagation unfolds.

We make this succession explicit through the *Minimal Substitution* model. In a complex network, Minimal Substitution describes dynamic succession through relative attractiveness, popularity, and age (Jin et al., 2019); it has also been used for information diffusion prediction and label propagation (Wang et al., 2025; 2024). Building on this model, we propose *Message-substitution*, a feature

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

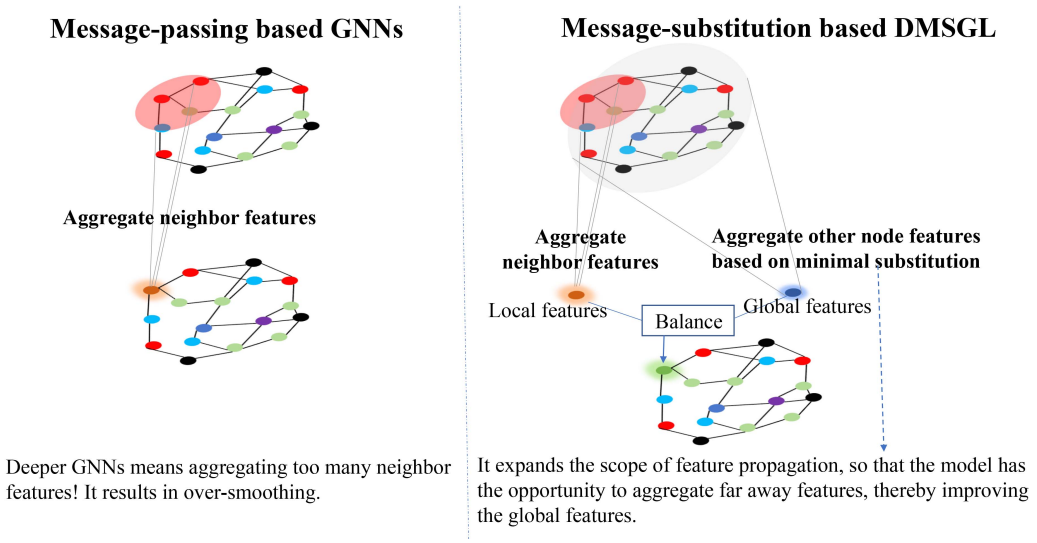


Figure 1: Message-passing and Message-substitution.

propagation mechanism that describes the layer-wise evolution of node representations as dynamic succession. We map relative attractiveness, popularity, and age to feature similarity, receiving-node degree, and propagation depth, respectively. These factors jointly determine how features from nodes across the graph contribute to each receiving node. Message-substitution enables direct feature interactions beyond observed neighborhoods, allowing nonlocal features to participate in representation learning at every layer. We develop Deep Minimal Substitution Graph Learning (DMSGL) by combining Message-substitution with local graph propagation. At each layer, a small set of stochastic macroscopic hubs supplies global features to all nodes. Resampling the hubs allows successive layers to learn from different parts of the graph. A learned coefficient combines the global and local features, while residual updates carry previous representations forward. The training objective further encourages variation in high-frequency representation components. Together, these components allow deeper propagation to incorporate additional nonlocal features while preserving local structure and useful node distinctions. Figure 1 illustrates the feature propagation process. Our contributions are as follows:

1. We introduce a network-succession perspective for deep graph learning, treating the layer-wise propagation of node features as dynamic succession among nodes. Built on this, we propose Message-substitution, which uses feature similarity, receiving-node degree, and propagation depth to determine the contributions of nonlocal features, enabling direct interactions with nodes sampled throughout the graph.
2. We develop DMSGL by combining global substitution with adaptive local propagation and residual updates. We analyze the change in representation variation under this update and derive a finite-depth bound that characterizes the retention of nontrivial representation variation.
3. We evaluate DMSGL on six node classification benchmarks across model depths. At 64 layers, DMSGL achieves the highest mean accuracy among the evaluated baselines on all six datasets, with its largest gain on ogbn-arxiv. These results show that succession-based propagation can support deeper models while keeping node representations distinct.

2 Related Work

Graph representation learning maps high-dimensional, sparse graph-structured data into low-dimensional real-valued representations Ju et al. (2024); Chen et al. (2024). Deep GNNs achieve this through repeated feature propagation, but increasing depth can make node representations increasingly similar, leading to over-smoothing. Existing approaches address this issue by modifying how information is preserved, propagated, or normalized.

Residual connections preserve earlier representations and improve gradient flow across layers. Difference-based residuals Yang et al. (2022), curriculum residuals with soft graph normalization Li et al. (2024), and hyperbolic residual functions Xue et al. (2024) all retain information from previous states. Their propagation at each new layer, however, remains based on local message passing. Residual paths preserve earlier features but do not provide a direct source of graph-wide information.

Graph rewiring and stochastic perturbation alter message-passing paths by adding or removing edges, nodes, or messages Fesser & Weber (2024); Giraldo et al. (2023). GRAND and DropNode randomly remove nodes Feng et al. (2020), while DropEdge and DropMessage modify edges or messages Rong et al. (2020); Fang et al. (2023). These operations can reduce repeated local mixing, but they also modify or discard parts of the original graph signal. Their propagation remains tied to local similarity assumptions and does not explicitly describe how feature states succeed across layers.

Normalization and regularization control the geometry or spectral behavior of node representations. PairNorm regulates node-wise feature distances Zhao & Akoglu (2020), DGN uses community structure to normalize group representations Zhou et al. (2020), and Oono et al. Oono & Suzuki (2020) analyze the spectral distribution of the normalized Laplacian to constrain GNN weights. These methods stabilize representation variation, while access to distant information remains determined by the underlying propagation process.

Alternative propagation dynamics replace standard recursive aggregation with different update mechanisms. SkipNode samples nodes and skips selected convolutions Lu et al. (2024); random-path aggregation combines features from multiple orders Bose & Das (2023); fractional graph Laplacian neural ODEs model nonlocal dynamics Maskey et al. (2023); SimCaliby calibrates local and global similarity Tang et al. (2024); and GREAD uses reaction–diffusion dynamics Choi et al. (2023). These methods broaden the design space for deep propagation, but the evolution of node features is still specified by a particular operator or sampling scheme.

Across these approaches, depth is handled through state preservation, path modification, representation regulation, or alternative update dynamics. We take a network-succession view of the process: as depth advances, node features evolve through successive interactions across the graph. We instantiate this view with Minimal Substitution, which provides a new rule for selecting nonlocal features at each layer. This leads to Message-substitution, a propagation mechanism that introduces graph-wide feature interactions while retaining the local feature.

3 PRELIMINARIES

3.1 MESSAGE-PASSING

Let $G = (V, E)$ be a graph. The representation of node v_i at layer l is h_i^l, the i-th row of $H^l \in \mathbb{R}^{|V|\times d}$. We write $H^0 = X$ for the features entering the propagation layers, with an input projection to width d when required. Message passing updates a node using messages from its neighbors:

$$h_i^l = \mathrm{Aggregate}^l\big(h_i^{l-1}, \{p^l(h_i^{l-1}, h_j^{l-1}) : v_j \in \mathrm{N}(v_i)\}\big)\,. \tag{1}$$

For GCN, the aggregated messages have the matrix form

$$\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{l-1}, \qquad \hat{A} = A + I, \quad \hat{D} = \mathrm{diag}(\hat{A}\mathbf{1}). \tag{2}$$

Here $\hat{A}$ includes self-loops and $\hat{D}$ is its degree matrix. In the method below, A and D denote the adjacency and degree matrices used for local propagation after this preprocessing. Node degrees D_j in the substitution rule refer to the receiving nodes in the input graph.

3.2 OVER-SMOOTHING

In deep message-passing GNNs, repeated neighborhood aggregation can progressively remove differences among node representations, causing them to converge toward the low-frequency subspace of the graph. For the normalized propagation operator

$$S = D^{-1/2}AD^{-1/2}, \tag{3}$$

this process corresponds to node representations approaching the nullspace of the normalized Laplacian $\widetilde{\Delta} = I - S$. We further measure equal-row representation variation by

$$\phi(H) = |H - \mathbf{1}\gamma_H|_F, \qquad \gamma_H = \frac{\mathbf{1}^T H}{|V|}. \tag{4}$$

A smaller $\phi(H)$ indicates less variation among node representations. Appendix A gives the corresponding spectral formulation and analyzes how DMSGL retains representation variation over finite depths.

3.3 Minimal substitution

Minimal substitution views information diffusion as dynamic succession: an incoming entity replaces an existing one according to similarity, popularity, and age Jin et al. (2019). Its substitution probability is given by

$$p_{n\to m} = \frac{\lambda_{n,m} D_m}{t_m}, \tag{5}$$

where $\lambda_{n,m}$ measures similarity, D_m denotes popularity, and t_m represents age. This formulation has been used to model information diffusion and network label propagation Wang et al. (2025; 2024). We transfer the unit of succession from information entities to node features. In DMSGL, the three factors are instantiated as feature similarity, receiving-node degree, and propagation depth. Depth therefore becomes the succession axis of feature propagation, and the resulting *Message-substitution* mechanism allows nonlocal features to participate in the evolution of node representations at each layer.

4 Method

DMSGL uses Message-substitution to learn from global features and retains a local branch to embed graph structure. The propagation process first collects features from sampled hubs and then combines them with local features through an adaptive coefficient. A layer of DMSGL is shown in Figure 2.

4.1 Feature propagation

Global feature learning. We map the three factors of minimal substitution to the compatibility between node features, the receiving node's degree, and the propagation round. The substitution weight from v_i to v_j at layer $l \geq 1$ is

$$p^l_{i\to j} = \sigma\left(\frac{\lambda_{i,j} D_j}{\log(\max\{2, l\})}\right), \tag{6}$$

where D_j is the degree of v_j and σ is the sigmoid function. The similarity $\lambda_{i,j}$ is the inner product of the L_2-normalized features h^{l-1}_i and h^{l-1}_j; a small positive denominator handles zero feature norms. The time factor is $\log 2$ in the first layer and $\log l$ for $l \geq 2$. Increasing depth reduces the contrast of the weights.

At each forward pass, we sample K distinct nodes uniformly without replacement to form the stochastic macroscopic hub set $\mathcal{C}_l$. The sets are sampled independently across layers. A node receives global features through

$$\mathfrak{g}^i_l = F\left(h^{l-1}_i, \sum_{v_j \in \mathcal{C}_l} p^l_{j\to i} h^{l-1}_j\right), \qquad |\mathcal{C}_l| = K. \tag{7}$$

The function F uses a linear layer followed by a sigmoid activation. Its outputs form $\mathfrak{g}_l \in \mathbb{R}^{|V|\times d}$. The hubs supply features directly to every node, including nodes outside their observed neighborhoods. Each node–hub pair receives a separate sigmoid weight. The aggregate is a sum, so its scale depends on K.

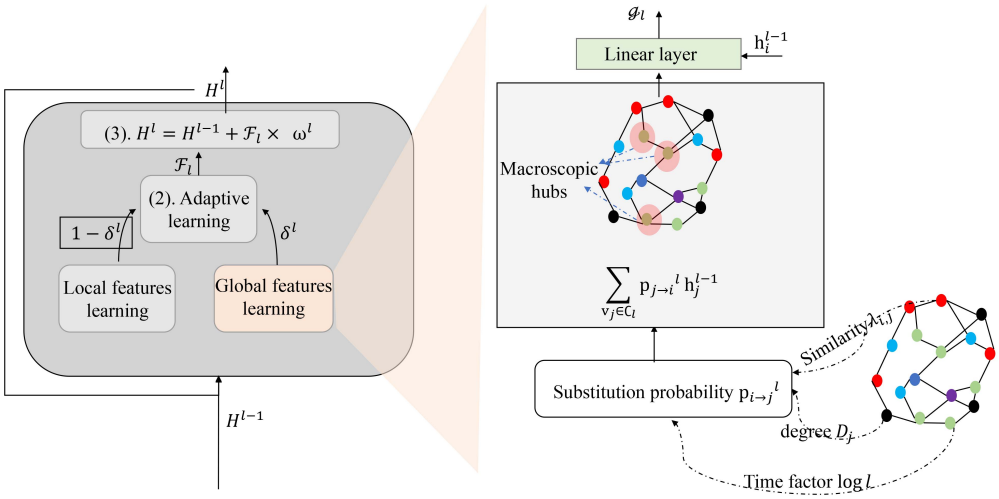


Figure 2: A layer of DMSGL, combining global feature learning, adaptive learning, and a residual update.

Adaptive learning. The local branch uses degree-normalized propagation to learn graph structure. The global and local features are combined as

$$\mathcal{F}_l = (1-\delta^l)D^{-1/2}AD^{-1/2}H^{l-1} + \delta^l \mathfrak{g}_l, \tag{8}$$

where $\delta^l \in \mathbb{R}$ is learned at each layer and $\mathcal{F}_l \in \mathbb{R}^{|V|\times d}$ contains the incoming messages. Equivalently,

$$\mathcal{F}_l = SH^{l-1} + \delta^l(\mathfrak{g}_l - SH^{l-1}), \qquad S = D^{-1/2}AD^{-1/2}. \tag{9}$$

Thus, δ^l scales the correction from the local update toward the global update. Values in $[0,1]$ interpolate between the two branches, whereas values outside this interval extrapolate along the same difference direction. In particular, a negative value moves the update away from $\mathfrak{g}_l$ relative to the local baseline; it does not represent a negative probability. Appendix A gives a finite-depth norm bound for either sign of δ^l.

4.2 Aggregate and Update

After obtaining $\mathcal{F}_l$, we retain the previous representation through a residual connection and scale the incoming messages by ω^l:

$$\begin{aligned} H^l &= H^{l-1} + \mathcal{F}_l\omega^l, \\ \omega^l &= 1 + \rho(1-\rho)^l, \qquad 0 < \rho < 1, \quad l = 1, \ldots, L. \end{aligned} \tag{10}$$

The hyperparameter ρ controls the geometric decay of the excess gain $\omega^l - 1 = \rho(1-\rho)^l$. Thus, $\omega^l > 1$ and approaches one with depth; it is a layer-dependent residual gain, not a coefficient that decays to zero. The input is specified by $H^0 = X$.

4.3 Optimization

A fully connected layer maps H^L from width d to N class logits. We use cross-entropy on the labeled training nodes, denoted by $\mathcal{L}_{\text{cross}}$, and optimize

$$\mathcal{L}(H^L \mid \theta) = \mathcal{L}_{\text{cross}} - \alpha_1 \frac{\left\|(I - D^{-1/2}AD^{-1/2})H^L\right\|_F^2}{\|H^L\|_F^2 + \varepsilon}, \tag{11}$$

where $\varepsilon > 0$, θ contains the trainable parameters, and $\alpha_1 \geq 0$ weights the scale-normalized Laplacian term. In the symmetric setting, this term rewards graph-frequency components in proportion to the squared Laplacian eigenvalues without increasing solely through a rescaling of H^L. Appendix A.3 gives its gradient and lower bound.

Table 1: Node classification accuracy (%) at 64 propagation layers (mean ± standard deviation over ten runs). Best means are bold and second-best means are brown. Improvement denotes the margin of DMSGL over the strongest baseline in percentage points.

Method	Cora	Citeseer	Photo	Computer	PubMed	ogbn-arxiv
GCN	28.04±3.60	22.14±1.00	64.62±6.60	59.06±8.90	38.79±1.65	21.64±0.23
GAT	29.22±6.00	24.45±4.50	21.03±6.80	37.45±0.20	41.74±2.93	15.10±8.17
GraphSage	31.90±0.50	18.69±1.80	25.35±7.20	37.45±4.70	40.70±0.00	13.71±7.85
A-DGN	62.37±5.00	48.09±7.20	90.45±2.20	83.39±1.80	75.66±1.84	49.09±0.57
GT	31.95±0.20	18.91±2.50	25.36±17.00	36.36±4.90	40.71±0.03	16.88±7.75
GATV2R	72.88±6.10	58.48±4.30	41.38±5.00	37.57±0.50	77.88±0.88	49.79±0.27
GPRGNN	82.19±0.70	70.58±0.60	90.80±0.60	82.84±1.00	79.54±0.45	49.28±0.41
APPNP	81.69±0.70	71.48±0.80	87.52±1.00	67.19±0.60	80.13±0.33	47.98±0.19
AERO-GNN	81.76±0.70	68.80±0.90	91.80±0.40	86.33±0.50	78.92±0.66	—
DAGNN	81.56±0.60	70.28±0.60	91.51±0.80	82.18±1.50	79.77±0.54	41.61±2.18
SkipNode	79.80±0.70	70.12±0.50	75.18±2.30	69.05±1.00	77.55±0.36	47.68±0.33
GRAND	81.65±0.31	69.95±0.94	68.56±0.61	57.92±1.04	76.87±0.59	9.67±2.37
PairNorm	39.75±3.04	31.39±3.74	58.87±16.15	44.06±9.92	65.09±7.16	8.92±4.95
WeightRep	23.46±7.18	25.65±0.80	40.46±0.19	39.63±1.9	40.96±1.08	12.34±7.93
DMSGL	**82.62±0.70**	**71.62±0.70**	**96.50±0.50**	**90.22±0.50**	**80.21±0.18**	**60.17±0.50**
Improvement	+0.43	+0.14	+3.70	+3.89	+0.08	+10.38

4.4 Representation variation and computational cost

The residual update gives a direct expression for the change in node variation. Let $P = I - \mathbf{1}\mathbf{1}^T/|V|$. Then

$$\phi(H^l)^2 - \phi(H^{l-1})^2 = 2\omega^l \langle PH^{l-1}, P\mathcal{F}_l \rangle_F + (\omega^l)^2 \|P\mathcal{F}_l\|_F^2. \tag{12}$$

The interaction between the previous representation and the incoming messages determines the change in variation. Appendix A derives this identity and a finite-depth lower bound for the component outside the normalized Laplacian nullspace. These results relate representation retention to the learned local–global update.

Computing substitution weights and hub aggregates costs $\mathcal{O}(K|V|d)$ per layer. Local sparse propagation costs $\mathcal{O}(|E|d)$, and linear feature transformations cost $\mathcal{O}(|V|d^2)$. The total is therefore

$$\mathcal{O}(K|V|d + |E|d + |V|d^2). \tag{13}$$

Global interaction is linear in $|V|$ for fixed K and d; the full layer is linear when the graph is also sparse. The global weight matrix occupies $\mathcal{O}(K|V|)$ memory.

5 Experiments

5.1 Experimental design

Datasets and implementation. We use six node classification datasets. Cora, Citeseer, and PubMed are citation networks with document features (Yang et al., 2016); Photo and Computer are Amazon co-purchase networks (Shchur et al., 2018); and ogbn-arxiv is a larger directed citation network with 169,343 papers and 40 subject areas (Hu et al., 2020). Cora, Citeseer, and PubMed use the fixed public Planetoid split: 20 labeled nodes per class for training, 500 nodes for validation, and 1,000 nodes for testing. Photo and Computer use 60%/20%/20% train/validation/test splits. ogbn-arxiv uses its official chronological split: papers published through 2017 for training, papers from 2018 for validation, and papers from 2019 onward for testing. Following the official GNN protocol, we symmetrize the ogbn-arxiv edges before propagation and use the same processed graph for every method. Dataset statistics are given in Appendix B.

The implementation uses PyTorch and PyTorch Geometric. Each experimental group is repeated with ten random seeds. Further settings are given in Appendix B.

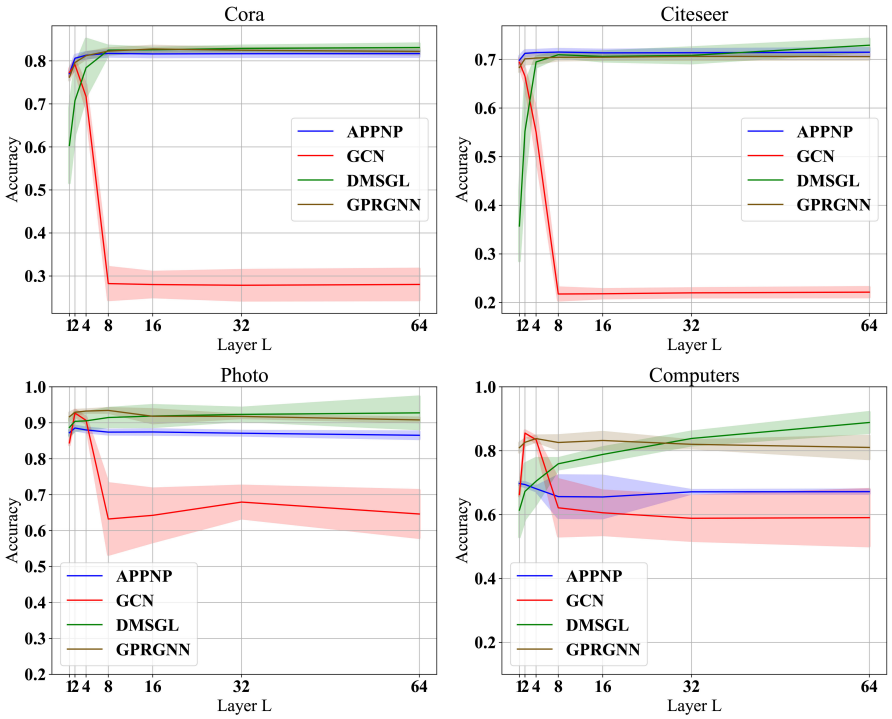


Figure 3: Impact of increasing model depth on classification accuracy. The vertical spans show the variation across repeated experiments.

Baselines. The comparison covers local aggregation (GCN, GAT, and GraphSage) (Kipf & Welling, 2017; Veličković et al., 2018; Hamilton et al., 2017), attention and residual models (GT and GATV2R) (Shi et al., 2021; Lee et al., 2023), stable graph dynamics (A-DGN) (Gravina et al., 2023), multi-step propagation (GPRGNN, APPNP, and DAGNN) (Chien et al., 2021; Gasteiger et al., 2019; Liu et al., 2020), and adaptive attention (AERO-GNN) (Lee et al., 2023). SkipNode randomly bypasses graph convolutions for sampled nodes (Lu et al., 2024); GRAND combines stochastic propagation with consistency regularization (Feng et al., 2020); PairNorm preserves pairwise feature spread by normalization (Zhao & Akoglu, 2020); and WeightRep reparameterizes trainable weights to maintain non-collapsed representations (Zhuo et al., 2025). Appendix B.1 specifies their controlled backbones and hyperparameters.

Metrics. We use classification accuracy as the primary performance metric. We further examine representation smoothness using a scale-normalized Dirichlet energy (SNDE) in Appendix C, following normalized Dirichlet-energy analyses, and representation distinctness using class-centroid distances and t-SNE visualizations in Appendix D.

5.2 Comparison Results

Table 1 shows that DMSGL achieves the highest mean accuracy across the six datasets, with different gains across different datasets. On Cora, Citeseer, and PubMed, APPNP and GPRGNN remain competitive. Their results suggest that controlling how information is retained across propagation steps may already capture much of the useful citation structure. Message-substitution adds access to features beyond these local connections. This may be useful on Citeseer, which has lower adjusted homophily than Cora and PubMed (Lim et al., 2021).

The advantage is more pronounced on Photo and Computer. These may be related to the meaning and density of co-purchase connections. Products bought together may belong to different categories. Repeated aggregation over dense neighborhoods may therefore mix category information with purchase associations that are less useful for classification. AERO-GNN is the strongest baseline on both datasets, suggesting that selecting among propagation ranges is helpful in this setting.

Table 2: Component ablation at 64 layers. Classification accuracy (%) is reported as mean ± standard deviation. Bold values indicate the highest value in each column.

Variant	Cora	Citeseer	Photo	Computer	PubMed	ogbn-arxiv
Full DMSGL	**82.62±0.70**	**71.62±0.70**	**96.50±0.50**	**90.22±0.50**	**80.21±0.18**	**60.17±0.50**
No Global	80.89±0.26	69.17±0.62	95.74±0.14	89.62±0.19	76.35±12.05	54.47±4.64
No Local	59.88±2.30	57.47±2.84	91.85±0.26	85.56±0.26	58.27±9.27	50.23±3.28
Uniform Hubs	80.96±0.40	69.22±0.82	95.93±0.18	89.78±0.21	76.30±12.04	38.78±2.47
Fixed Hubs	80.94±0.34	69.13±0.72	95.87±0.13	89.82±0.27	76.10±12.30	38.20±1.50
Attention Hubs	79.08±0.17	65.82±1.22	78.04±1.19	72.70±0.99	78.62±0.39	43.98±4.24

DMSGL also weights nonlocal feature sources by their compatibility with each receiving node. This may help capture product similarities that are not fully reflected in co-purchase connections. The clearest gain appears on ogbn-arxiv, the largest benchmark in the evaluation. DMSGL obtains 60.17%, compared with 49.79% for GATV2R, a gain of 10.38 percentage points on ogbn-arxiv. Thus, Message-substitution may retain its benefit even when the graph, label space, and temporal evaluation are substantially larger than in the other benchmarks. The sampled-hub design provides graph-wide feature access.

The 64-layer results distinguish representation diversity from classification utility. WeightRep constructs input-dependent weights from the feature covariance to prevent full and dimensional collapse (Zhuo et al., 2025). This operation preserves feature variation, but it does not change the repeated neighborhood mixing or ensure that the retained variation follows class boundaries. After many propagation layers, reparameterization can rescale the remaining features but cannot by itself recover class-relevant information that has been mixed across nodes. The original study also reports decreasing accuracy as depth increases, despite improved resistance to representation collapse. GRAND, PairNorm, and SkipNode face a related issue: they regularize, normalize, or bypass local propagation without changing the information available to each node. DMSGL introduces similarity-conditioned nonlocal features and retains representations from different depths, allowing feature diversity to remain useful for classification.

5.3 Analysis of deep models

Figure 3 examines whether additional propagation layers provide useful information or simply repeat the same diffusion process. GCN shows a consistent loss of accuracy as depth increases, reflecting the progressive removal of discriminative feature differences through repeated neighborhood aggregation. APPNP remains stable across the tested depths, but additional propagation produces little further improvement. Its initial-feature injection controls smoothing, while the information source remains fixed. GPRGNN learns how to combine different propagation steps and remains stable beyond 16 layers on Photo. Its decline at greater depths on Computer shows that extending the propagation range does not always introduce useful class information. These results distinguish resistance to degradation from the ability to benefit from depth.

DMSGL exhibits a different depth profile, with accuracy remaining stable or following an overall upward trend. Each layer samples a new set of hubs and introduces nonlocal features according to their compatibility with the receiving node. Residual updates and multiscale fusion retain information learned at earlier depths. Additional layers therefore create new feature interactions while preserving previously learned structure, instead of repeatedly diffusing the same local signal. This allows a deeper model to expand its information range without losing the distinctions needed for classification. The learned coefficients analyzed below show how DMSGL adjusts the local and global contributions during this process. Appendix E reports the sensitivity to α_1 and ρ.

The representation analyses further support this depth advantage. On Cora and Computer, deeper DMSGL models retain class-related clusters while improving classification accuracy, whereas deep GCN representations exhibit substantial class mixing (Appendix D). The SNDE results complement these observations by showing that strong classification performance can coexist with smoother representations along graph edges (Appendix C).

5.4 Ablation study

We examine how Message-substitution contributes to deep graph learning by combining global and local feature learning. Table 2 compares the full model with five variants. **No Global** removes Message-substitution and retains local graph propagation. **No Local** removes local propagation and relies on Message-substitution for feature evolution. **Uniform Hubs** retains the same nonlocal sources and layer-wise source renewal as DMSGL, but removes the substitution-dependent preference among these sources by assigning them equal contributions.**Fixed Hubs** preserves the substitution weighting rule but uses a fixed set of nonlocal sources throughout propagation. **Attention Hubs** replaces the Minimal Substitution-based weighting rule with a standard attention mechanism while retaining the same sampled global nodes, local propagation branch, and layer-wise source renewal.

Does Message-substitution help? Removing Message-substitution consistently reduces accuracy across all datasets, showing that local propagation alone cannot account for the performance of DMSGL. The No Local results further indicate that Message-substitution complements rather than replaces graph structure: local propagation preserves relational information, while Message-substitution introduces additional nonlocal interactions.

Is generic global aggregation sufficient? Uniform Hubs performs close to No Global on several datasets despite retaining access to nonlocal features. This shows that simply enlarging the information range provides limited benefit. Effective global interaction requires determining how different nonlocal sources contribute to each node.

Does the Minimal Substitution rule matter? Attention Hubs provides a further comparison because it retains content-dependent weighting. On ogbn-arxiv, attention improves upon Uniform Hubs, from 38.78% to 43.98%, but remains below Full DMSGL at 60.17%. Full DMSGL also obtains higher mean accuracy on the other five datasets, with gaps of 18.46 and 17.52 points on Photo and Computer. These results favor the proposed rule, which combines feature similarity, receiving-node degree, and propagation depth, over the tested attention formulation. They support the rule as a whole, beyond the use of content-dependent global weighting alone.

Does adaptive substitution weighting matter? Full DMSGL consistently outperforms Uniform Hubs. The substitution rule therefore provides more than global feature access: it distinguishes the contribution of different sources according to their interaction with the receiving node, allowing nonlocal information to be incorporated selectively during representation evolution.

Does learning from different hubs across layers matter? Full DMSGL also outperforms Fixed Hubs, showing that repeatedly interacting with the same hubs is insufficient. Renewing the hub set across layers introduces new nonlocal interactions as node representations evolve, allowing depth to progressively expand the information involved in feature succession. Overall, these results show that Message-substitution turns depth into a process of progressively discovering and integrating relevant nonlocal information.

6 Conclusion

We introduced DMSGL, which views feature propagation across model depth from the perspective of network succession. Based on Minimal Substitution, we developed Message-substitution to describe how node features evolve during deep learning. In this Message-substitution mechanism, Feature similarity, the receiving node's degree, and propagation depth jointly determine the contribution of each source. We resample the hubs to renew these sources across layers, allowing nodes to learn from different parts of the graph through direct nonlocal interactions. We use an Adaptive fusion that combines this global information with local propagation, preserving the role of observed graph relationships. This design gives depth two connected roles: exploring additional feature sources and building on information learned at earlier stages. Our analysis identifies conditions under which representation variation is retained during deep propagation. Experiments across six node classification benchmarks support the effectiveness of the framework in deep settings. By connecting network succession with feature learning, DMSGL provides a propagation mechanism that broadens access to information while retaining distinctions useful for downstream. This is the central goal of learning through Minimal Substitution: going deeper while keeping node representations distinct.

REPRODUCIBILITY STATEMENT

Section 4 specifies the propagation and training objective. Appendix A gives the mathematical assumptions and proofs. Appendix B records dataset statistics and training settings, and Appendix E specifies the hub-count evaluation. Anonymous source code for reproducing the main experiments is provided in the supplementary material.

AI USE STATEMENT

Generative AI tools assisted with manuscript polish. Experimental measurements were supplied by the authors. The authors are responsible for verifying the mathematical arguments, implementation, references, and final manuscript.

REFERENCES

Kushal Bose and Swagatam Das. Can Graph Neural Networks Go Deeper Without Over-Smoothing? Yes, With a Randomized Path Exploration! *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(5):1595–1604, 2023. doi: 10.1109/tetci.2023.3249255. URL https://doi.org/10.1109/tetci.2023.3249255.

Hao Chen, Yuanchen Bei, Qijie Shen, Yue Xu, Sheng Zhou, Wenbing Huang, Feiran Huang, Senzhang Wang, and Xiao Huang. Macro Graph Neural Networks for Online Billion-Scale Recommender Systems. In *Proceedings of the ACM Web Conference 2024*, pp. 3598–3608, 2024. doi: 10.1145/3589334.3645517. URL https://doi.org/10.1145/3589334.3645517.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive Universal Generalized PageRank Graph Neural Network. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2006.07988.

Jeongwhan Choi, Seoyoung Hong, Noseong Park, and Sung-Bae Cho. GREAD: Graph Neural Reaction-Diffusion Networks. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 5722–5747. PMLR, 2023. URL https://proceedings.mlr.press/v202/choi23a.html.

Tien Huu Do, Duc Minh Nguyen, Giannis Bekoulis, Adrian Munteanu, and Nikos Deligiannis. Graph convolutional neural networks with node transition probability-based message passing and DropNode regularization. *Expert Systems with Applications*, 174:114711, 2021. doi: 10.1016/j.eswa.2021.114711. URL https://doi.org/10.1016/j.eswa.2021.114711.

Taoran Fang, Zhiqing Xiao, Chunping Wang, Jiarong Xu, Xuan Yang, and Yang Yang. DropMessage: Unifying Random Dropping for Graph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4267–4275, 2023. doi: 10.1609/aaai.v37i4.25545. URL https://doi.org/10.1609/aaai.v37i4.25545.

Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph Random Neural Networks for Semi-Supervised Learning on Graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22092–22103, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/fb4c835feb0a65cc39739320d7a51c02-Abstract.html.

Lukas Fesser and Melanie Weber. Mitigating Over-Smoothing and Over-Squashing Using Augmentations of Forman-Ricci Curvature. In *Learning on Graphs Conference*, volume 231 of *Proceedings of Machine Learning Research*, pp. 19:1–19:28. PMLR, 2024. URL https://proceedings.mlr.press/v231/fesser24a.html.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1810.05997.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

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. doi: 10.1145/3583780.3614997. URL https://doi.org/10.1145/3583780.3614997.

Alessio Gravina, Davide Bacciu, and Claudio Gallicchio. Anti-Symmetric DGN: a stable architecture for Deep Graph Networks. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.09789.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html.

Ching Jin, Chaoming Song, Johannes Bjelland, Geoffrey Canright, and Dashun Wang. Emergence of scaling in complex substitutive systems. *Nature Human Behaviour*, 3(8):837–846, 2019. doi: 10.1038/s41562-019-0638-y. URL https://doi.org/10.1038/s41562-019-0638-y.

Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, Junwei Yang, Jingyang Yuan, Yusheng Zhao, Yifan Wang, Xiao Luo, and Ming Zhang. A Comprehensive Survey on Deep Graph Representation Learning. *Neural Networks*, 173:106207, 2024. doi: 10.1016/j.neunet.2024.106207. URL https://doi.org/10.1016/j.neunet.2024.106207.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.02907.

Soo Yong Lee, Fanchen Bu, Jaemin Yoo, and Kijung Shin. Towards Deep Attention in Graph Neural Networks: Problems and Remedies. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18774–18795. PMLR, 2023. URL https://proceedings.mlr.press/v202/lee23b.html.

Jin Li, Qirong Zhang, Shuling Xu, Xinlong Chen, Longkun Guo, and Yang-Geng Fu. Curriculum-Enhanced Residual Soft An-Isotropic Normalization for Over-Smoothness in Deep GNNs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13528–13536, 2024. doi: 10.1609/aaai.v38i12.29256. URL https://doi.org/10.1609/aaai.v38i12.29256.

Derek Lim, Xiuyu Li, Felix Hohne, and Ser-Nam Lim. New Benchmarks for Learning on Non-Homophilous Graphs. In *Workshop on Graph Learning Benchmarks at The Web Conference*, 2021. URL https://arxiv.org/abs/2104.01404.

Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards Deeper Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 338–348, 2020. doi: 10.1145/3394486.3403076. URL https://doi.org/10.1145/3394486.3403076.

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. URL https://doi.org/10.1109/tkde.2024.3374701.

Sohir Maskey, Raffaele Paolino, Aras Bacho, and Gitta Kutyniok. A Fractional Graph Laplacian Approach to Oversmoothing. In *Advances in Neural Information Processing Systems*, volume 36, pp. 13022–13063, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/2a514213ba899f2911723a38be8d4096-Abstract-Conference.html.

Kenta Oono and Taiji Suzuki. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1905.10947.

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. URL https://arxiv.org/abs/1907.10903.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of Graph Neural Network Evaluation. *arXiv preprint arXiv:1811.05868*, 2018. URL https://arxiv.org/abs/1811.05868.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 1548–1554, 2021. doi: 10.24963/ijcai.2021/214. URL https://doi.org/10.24963/ijcai.2021/214.

Boshi Tang, Zhiyong Wu, Xixin Wu, Qiaochu Huang, Jun Chen, Shun Lei, and Helen Meng. SimCalib: Graph Neural Network Calibration Based on Similarity between Nodes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15267–15275, 2024. doi: 10.1609/aaai.v38i14.29450. URL https://doi.org/10.1609/aaai.v38i14.29450.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1710.10903.

Ranran Wang, Yin Zhang, and Limei Peng. Anomaly Detection Service for Blockchain Transactions Using Minimal Substitution-Based Label Propagation. *IEEE Transactions on Services Computing*, 17(5):2054–2066, 2024. doi: 10.1109/tsc.2024.3407601. URL https://doi.org/10.1109/tsc.2024.3407601.

Ranran Wang, Xing Xu, and Yin Zhang. Multiscale Information Diffusion Prediction With Minimal Substitution Neural Network. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):1069–1080, 2025. doi: 10.1109/tnnls.2023.3331159. URL https://doi.org/10.1109/tnnls.2023.3331159.

Xinyi Wu, Amir Ajorlou, Zihui Wu, and Ali Jadbabaie. Demystifying Oversmoothing in Attention-Based Graph Neural Networks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 35084–35106, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/6e4cdfdd909ea4e34bfc85a12774cba0-Abstract-Conference.html.

Yangkai Xue, Jindou Dai, Zhipeng Lu, Yuwei Wu, and Yunde Jia. Residual Hyperbolic Graph Convolution Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16247–16254, 2024. doi: 10.1609/aaai.v38i15.29559. URL https://doi.org/10.1609/aaai.v38i15.29559.

Liang Yang, Weihang Peng, Wenmiao Zhou, Bingxin Niu, Junhua Gu, Chuan Wang, Yuanfang Guo, Dongxiao He, and Xiaochun Cao. Difference Residual Graph Neural Networks. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 3356–3364, 2022. doi: 10.1145/3503161.3548111. URL https://doi.org/10.1145/3503161.3548111.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting Semi-Supervised Learning with Graph Embeddings. In *International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 40–48. PMLR, 2016. URL https://proceedings.mlr.press/v48/yanga16.html.

Lingxiao Zhao and Leman Akoglu. PairNorm: Tackling Oversmoothing in GNNs. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1909.12223.

Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, and Xia Hu. Towards Deeper Graph Neural Networks with Differentiable Group Normalization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4917–4928, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/33dd6dba1d56e826aac1cbf23cdcca87-Abstract.html.

Zhijian Zhuo, Yifei Wang, Jinwen Ma, and Yisen Wang. Graph Neural Networks (with Proper Weights) Can Escape Oversmoothing. In *Asian Conference on Machine Learning*, volume 260 of *Proceedings of Machine Learning Research*, pp. 17–32. PMLR, 2025. URL https://proceedings.mlr.press/v260/zhuo25a.html.

A MATHEMATICAL ANALYSIS

A.1 MEASURES OF REPRESENTATION VARIATION

Let $P = I - \mathbf{1}\mathbf{1}^T/|V|$. This is an orthogonal projection, so Eq. (**??**) can be written as $\phi(H) = \|PH\|_F$. It follows that $\phi(aH) = |a|\phi(H)$ and $\phi(H + H') \leq \phi(H) + \phi(H')$. Furthermore, $PH = 0$ if and only if every row equals γ_H. Applying P to Eq. (10) and expanding the squared Frobenius norm proves Eq. (12).

For spectral analysis, assume that A is symmetric with nonnegative entries and that $D = \mathrm{diag}(A\mathbf{1})$ has positive diagonal entries. Write

$$S = D^{-1/2}AD^{-1/2}, \qquad \tilde{\Delta} = I - S, \qquad \mathcal{E}(H) = \mathrm{Tr}(H^T\tilde{\Delta}H). \tag{14}$$

The eigenvalues of $\tilde{\Delta}$ lie in $[0, 2]$. Let $Q = I - \Pi_{\ker\tilde{\Delta}}$ be the orthogonal projection onto its positive eigenspace. On a connected graph, the nullspace is spanned by $D^{1/2}\mathbf{1}$; on a disconnected graph, there is one such vector per connected component. Thus, P measures equal-row collapse and Q measures approach to the normalized-Laplacian nullspace. The spectral statements below use this symmetric setting.

Lemma 1 (Energy and distance from the nullspace)**.** *Suppose $\tilde{\Delta}$ has a positive eigenvalue and let λ_+ be its smallest positive eigenvalue. For any representation H,*

$$\lambda_+\|QH\|_F^2 \leq \mathcal{E}(H) \leq 2\|QH\|_F^2, \qquad \|\tilde{\Delta}H\|_F^2 \leq 2\mathcal{E}(H). \tag{15}$$

Proof. Let (λ_i, u_i) be an orthonormal eigendecomposition of $\tilde{\Delta}$. Then

$$\|QH\|_F^2 = \sum_{\lambda_i>0} \|u_i^T H\|_2^2, \tag{16}$$

$$\mathcal{E}(H) = \sum_{\lambda_i>0} \lambda_i\|u_i^T H\|_2^2, \tag{17}$$

$$\|\tilde{\Delta}H\|_F^2 = \sum_{\lambda_i>0} \lambda_i^2\|u_i^T H\|_2^2. \tag{18}$$

The first two inequalities follow from $\lambda_+ \leq \lambda_i \leq 2$ on the positive eigenspace. The last follows from $\lambda_i^2 \leq 2\lambda_i$. □

A.2 FINITE-DEPTH RETENTION OF VARIATION

The combined update can be written as

$$H^l = B_l H^{l-1} + \omega^l\delta^l\mathfrak{g}_l, \qquad B_l = I + \omega^l(1 - \delta^l)S. \tag{19}$$

Define the smallest singular value of B_l on the positive Laplacian eigenspace by

$$a_l = \min_{\lambda_i>0} \left|1 + \omega^l(1 - \delta^l)(1 - \lambda_i)\right|, \qquad b_l = |\omega^l\delta^l|\,\|Q\mathfrak{g}_l\|_F. \tag{20}$$

These quantities are defined for the learned coefficients and the realized hub sets. The resulting statement is pathwise: it holds for every sampled sequence $(\mathcal{C}_1, \ldots, \mathcal{C}_L)$ and does not replace the sampled weights by a maximum probability.

Theorem 1 (Finite-depth variation bound)**.** *Under the conditions of Lemma 1, the update in Eq.* (19) *satisfies, for every depth $L \geq 1$,*

$$\|QH^L\|_F \geq \left[\left(\prod_{l=1}^{L} a_l\right)\|QH^0\|_F - \sum_{t=1}^{L} b_t \prod_{s=t+1}^{L} a_s\right]_+, \tag{21}$$

where $[x]_+ = \max\{x, 0\}$ and an empty product equals one. If the right-hand side is $r_L > 0$, then $\mathcal{E}(H^L) \geq \lambda_+ r_L^2$.

Proof. Since S and Q have the same eigenvectors, $QB_l = B_lQ$. Applying Q to Eq. (19) gives

$$QH^l = B_lQH^{l-1} + \omega^l\delta^lQ\mathfrak{g}_l. \tag{22}$$

The singular-value bound and the reverse triangle inequality imply

$$\|QH^l\|_F \geq a_l\|QH^{l-1}\|_F - b_l. \tag{23}$$

Successive substitution, using $a_l \geq 0$, yields the expression inside the brackets in Eq. (21). Combining it with the nonnegativity of a norm proves the bound. Lemma 1 gives the energy inequality. □

Equation (21) supplies a checkable finite-depth certificate. It does not assume a positive constant: when the expression inside $[\cdot]_+$ is positive, the representation has a corresponding positive energy bound; otherwise the theorem makes no non-collapse claim for that realization.

The alignment of the global update gives a more direct description of an individual layer. Expanding Eq. (19) yields

$$\begin{aligned}\|QH^l\|_F^2 = \|B_lQH^{l-1}\|_F^2 + 2\omega^l\delta^l\langle B_lQH^{l-1}, Q\mathfrak{g}_l\rangle_F \\ + (\omega^l\delta^l)^2\|Q\mathfrak{g}_l\|_F^2.\end{aligned} \tag{24}$$

For example, $a_l \geq 1$ and $\omega^l\delta^l\langle B_lQH^{l-1}, Q\mathfrak{g}_l\rangle_F \geq 0$ are sufficient for $\|QH^l\|_F \geq \|QH^{l-1}\|_F$. Theorem 1 and Eq. (24) specify finite-depth conditions in terms of the update itself.

The same update also gives an explicit norm bound for signed fusion. Because F ends with a sigmoid, every entry of $\mathfrak{g}_l$ lies in $(0, 1)$ and $\|\mathfrak{g}_l\|_F \leq \sqrt{|V|d}$.

Proposition 1 (Finite-depth norm bound)**.** *Under the symmetric-graph assumptions above, define*

$$c_l = 1 + \omega^l|1 - \delta^l|, \qquad q_l = \omega^l|\delta^l|\sqrt{|V|d}. \tag{25}$$

For every finite sequence of learned coefficients, regardless of their signs,

$$\|H^L\|_F \leq \left(\prod_{l=1}^{L} c_l\right)\|H^0\|_F + \sum_{t=1}^{L} q_t \prod_{s=t+1}^{L} c_s. \tag{26}$$

Proof. For a symmetric normalized adjacency, $\|S\|_2 \leq 1$. Applying the triangle inequality to Eq. (19) therefore gives

$$\|H^l\|_F \leq \big(1 + \omega^l|1 - \delta^l|\big)\|H^{l-1}\|_F + \omega^l|\delta^l|\sqrt{|V|d}. \tag{27}$$

Iterating this inequality proves Eq. (26). □

Proposition 1 makes the role of negative coefficients explicit: their effect is controlled by their magnitude.

A.3 Scale-normalized Laplacian objective

Let $M = \tilde{\Delta}^T\tilde{\Delta}$ and define

$$R(H) = \frac{\|\tilde{\Delta}H\|_F^2}{\|H\|_F^2 + \varepsilon}. \tag{28}$$

For any fixed graph operator,

$$\nabla_H[-\alpha_1 R(H)] = -\frac{2\alpha_1}{\|H\|_F^2 + \varepsilon}\big(MH - R(H)H\big). \tag{29}$$

In the symmetric setting, $M = \tilde{\Delta}^2$. Since $\|\tilde{\Delta}\|_2 \leq 2$,

$$0 \leq R(H) \leq \|\tilde{\Delta}\|_2^2\frac{\|H\|_F^2}{\|H\|_F^2 + \varepsilon} < 4. \tag{30}$$

Cross-entropy is nonnegative, so Eq. (11) satisfies

$$\mathcal{L}(H^L \mid \theta) > -4\alpha_1 \tag{31}$$

without imposing a norm constraint on H^L. The normalization also makes the regularizer invariant to representation scale in the limit $\varepsilon \to 0$. The analysis concerns representation variation and hub sampling; it does not use convexity of the deep-network objective or assert an optimizer-specific learning-rate bound.

A.4 HUB SAMPLING

Condition on the features entering a layer and consider uniform sampling of K distinct hubs from $n = |V| > 1$ nodes. For a receiving node v_i, write $x_j = p^l_{j\to i} h^{l-1}_j$ and $\bar{x} = n^{-1} \sum_{j=1}^n x_j$.

Proposition 2 (Uniform hub sampling)**.** *The expected sampled aggregate is*

$$\mathbb{E}_{\mathcal{C}_l}\left[\sum_{j\in\mathcal{C}_l} x_j\right] = K\bar{x}. \tag{32}$$

The sample mean and sampled sum satisfy

$$\begin{aligned}\mathbb{E}\left\|\frac{1}{K}\sum_{j\in\mathcal{C}_l} x_j - \bar{x}\right\|_2^2 &= \frac{n-K}{K(n-1)}\frac{1}{n}\sum_{j=1}^n \|x_j - \bar{x}\|_2^2,\\ \mathbb{E}\left\|\sum_{j\in\mathcal{C}_l} x_j - K\bar{x}\right\|_2^2 &= \frac{K(n-K)}{n(n-1)}\sum_{j=1}^n \|x_j - \bar{x}\|_2^2.\end{aligned} \tag{33}$$

Proof. Let I_j indicate whether node j is sampled. Then $\mathbb{E}I_j = K/n$ and $\mathbb{E}[I_jI_t] = K(K-1)/(n(n-1))$ for $j \neq t$. The first identity follows by linearity of expectation. For the variance, set $y_j = x_j - \bar{x}$ and use $\sum_j y_j = 0$. Expanding the squared norm gives

$$\begin{aligned}\mathbb{E}\left\|\frac{1}{K}\sum_j I_j y_j\right\|_2^2 &= \frac{1}{K^2}\left(\frac{K}{n}\sum_j \|y_j\|_2^2 + \frac{K(K-1)}{n(n-1)}\sum_{j\neq t}\langle y_j, y_t\rangle\right)\\ &= \frac{n-K}{Kn(n-1)}\sum_j \|y_j\|_2^2.\end{aligned} \tag{34}$$

Multiplying the sample-mean identity by K^2 gives the sampled-sum identity. □

The sampled sum in Eq. (7) has expectation K/n times the full sum. Its scale and fluctuation therefore depend on K. Theorem 1 conditions on each realized hub sequence, while Proposition 2 describes variation across sequences; neither result assumes that a maximum substitution probability remains fixed across iterations.

B DATASETS AND IMPLEMENTATION DETAILS

Table 3: Dataset statistics. Edge counts are reported before adding reverse directions or self-loops; ogbn-arxiv is directed and the other graphs are treated as undirected.

Dataset	Nodes	Edges	Features	Classes	Split
Cora	2,708	5,429	1,433	7	Public fixed
Citeseer	3,327	4,732	3,703	6	Public fixed
PubMed	19,717	44,338	500	3	Public fixed
Photo	7,650	119,081	745	8	60/20/20
Computer	13,752	245,861	767	10	60/20/20
ogbn-arxiv	169,343	1,166,243	128	40	Official time

Cora, Citeseer, and PubMed use papers as nodes and citation links as edges. PubMed contains biomedical papers represented by 500-dimensional document features and three classes. Photo and Computer use products as nodes, co-purchases as edges, and product categories as labels. In ogbn-arxiv, nodes are computer-science papers, directed edges are citations, 128-dimensional features summarize titles and abstracts, and labels are 40 subject areas (Hu et al., 2020). We retain the official temporal split and symmetrize the adjacency before adding self-loops. The same symmetrized operator is used by all baselines and by the spectral analysis in Appendix A.

All methods use identical split indices within a run, and hyperparameters are selected only from validation performance. For the fixed Planetoid and OGB splits, the ten runs change initialization and training randomness but not the labeled nodes.

B.1 Experimental Configurations

For a controlled comparison, all methods are trained with a hidden width of 64 and are evaluated at a depth of 64 propagation layers or steps. Each experiment is repeated ten times. All methods use identical training, validation, and test indices, and both hyperparameters and checkpoints are selected exclusively according to validation performance. Unless otherwise stated, the baselines use Adam, a dropout rate of 0.5, a learning rate of 10^{-2}, and a weight decay of 5×10^{-4}. Table 4 summarizes the configurations.

Table 4: Configurations of all compared methods. Dataset-wise tuples follow the order (Cora, Citeseer, Photo, Computers, PubMed, ogbn-arxiv).

Method	Important configuration
GCN	A 64-layer GCN backbone. ReLU and dropout are applied after every hidden graph-convolution layer.
GAT	A 64-layer GAT with one attention head per layer, non-concatenated head output, and dropout applied to both hidden representations and attention computation.
GraphSAGE	A 64-layer GraphSAGE network using mean neighborhood aggregation, with ReLU and dropout after every hidden layer.
A-DGN	A 64-step anti-symmetric deep graph network with integration step $\epsilon_{\mathrm{ADGN}} = 0.1$ and diffusion constant $\gamma = 10^{-3}$.
GT	A 64-layer graph Transformer using one attention head, non-concatenated outputs, and the non-beta TransformerConv formulation.
GATv2R	A 64-layer GATv2 network with one attention head, shared source and target projection weights, and an initial-residual coefficient $\alpha = 0.5$. On PubMed, dropout is 0.7 and weight decay is 10^{-3}.
GPRGNN	A two-layer MLP followed by 64 graph-propagation steps. The propagation coefficients are trainable and initialized with generalized PageRank coefficients using $\alpha = 0.1$; no weight decay is applied to the propagation coefficients.
APPNP	A two-layer MLP followed by 64 personalized PageRank propagation steps with teleport probability $\alpha = 0.1$.
AERO-GNN	A 64-layer AERO-GNN with $\lambda = 0.25$, dropout 0.5, and weight decay 5×10^{-4} for both dense and propagation parameters. And no additional classifier dropout is used.
DAGNN	A two-layer MLP followed by 64 propagation steps. Representations from different propagation depths are combined using learned sigmoid-based retention scores.
SkipNode	A 64-step propagation network with node-skip probability $p_{\mathrm{skip}} = 0.5$ and no skip-rate schedule. An independent Bernoulli mask determines whether each node is updated or retains its preceding-layer representation. Inference uses the expected mixture of the updated and preceding-layer representations.
GRAND	Propagation order 64, two stochastic views per optimization step, DropNode probability 0.5, sharpening temperature $T = 0.5$, and consistency-loss weight $\lambda_{\mathrm{con}} = 1.0$.
PairNorm	A 64-step GCN-style backbone using standard PairNorm (PN) with scale $s = 1.0$ and numerical constant $\epsilon_{\mathrm{PN}} = 10^{-6}$. PairNorm is applied after every hidden propagation block.
WeightRep	A 64-step GCN-style backbone with covariance power $p = -0.5$ and numerical constant $\epsilon_{\mathrm{WR}} = 10^{-6}$. The covariance transformation is computed without gradient tracking and applied after every hidden propagation block.
DMSGL	**Optimization:** learning rate $(10^{-2}, 10^{-3}, 10^{-3}, 10^{-2}, 1.25 \times 10^{-3}, 10^{-2})$; Adam weight decay $(5 \times 10^{-4}, 5 \times 10^{-4}, 5 \times 10^{-4}, 5 \times 10^{-4}, 6 \times 10^{-3}, 5 \times 10^{-4})$; explicit parameter penalty $\lambda_2 = (0, 10^{-6}, 10^{-6}, 10^{-6}, 5 \times 10^{-8}, 0)$; and dropout $(0.6, 0.6, 0.6, 0.6, 0.1, 0.5)$. **Hub configuration:** $K = (128, 1000, 1000, 1000, 256, 64)$. **Adaptive and decay coefficients:** $\rho = (0.2, 0.2, 0.2, 0.2, 0.05, 0.1)$. The adaptive coefficient is initialized as $\delta_0^l = \rho(1-\rho)^l$ for $1 \leq l < L$ and $\delta_0^L = (1-\rho)^L$, and is subsequently optimized as an unconstrained scalar.

C SNDE Analysis

C.1 Scale-normalized Dirichlet energy

We use scale-normalized Dirichlet energy (SNDE) to examine representation smoothness across model depths. For an undirected, unweighted graph, let A_0 denote the adjacency matrix without self-loops, $d_i = \sum_j (A_0)_{ij}$, $A = A_0 + I$, and $D = \mathrm{diag}(A\mathbf{1})$. The normalized Laplacian is $\widetilde{\Delta} = I - D^{-1/2} A D^{-1/2}$. The Dirichlet energy of the node representations H is

$$\mathcal{E}(H) = \mathrm{Tr}(H^T \widetilde{\Delta} H) = \frac{1}{2} \sum_{i,j} (A_0)_{ij} \left\| \frac{H_i}{\sqrt{1+d_i}} - \frac{H_j}{\sqrt{1+d_j}} \right\|_2^2, \tag{35}$$

where H_i denotes the representation of node i. We normalize this energy by the total representation energy:

$$\mathrm{SNDE}(H) = \frac{\mathcal{E}(H)}{\|H\|_F^2} = \frac{\mathrm{Tr}(H^T \widetilde{\Delta} H)}{\|H\|_F^2}, \qquad H \neq 0. \tag{36}$$

We set $\mathrm{SNDE}(0) = 0$. For every nonzero scalar c, $\mathrm{SNDE}(cH) = \mathrm{SNDE}(H)$, so the metric is unaffected by global rescaling. Since the eigenvalues of $\widetilde{\Delta}$ lie in $[0, 2]$, SNDE also lies in $[0, 2]$.

Smaller SNDE values indicate smoother degree-normalized representations along graph edges. For nonzero H, values approaching zero indicate that $H/\|H\|_F$ approaches the normalized-Laplacian nullspace. We interpret SNDE jointly with classification accuracy to examine how edge-wise smoothness relates to predictive performance. SNDE is computed separately for each run before reporting its mean and standard deviation.

We evaluate how representation smoothness changes with model depth.

C.2 SNDE ACROSS MODEL DEPTHS

We evaluate DMSGL at depths ranging from 1 to 64 layers. Figure 4 shows how SNDE changes with

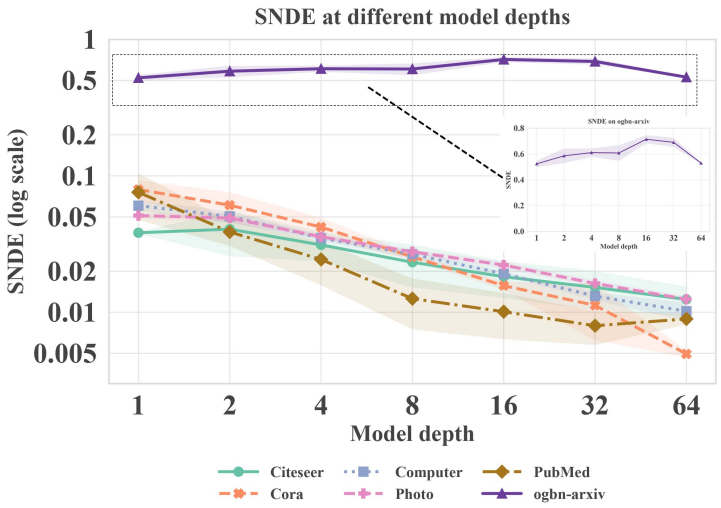


Figure 4: SNDE of DMSGL across model depths on six datasets. Curves show mean values, and shaded regions indicate ± one standard deviation. The main panel uses a logarithmic vertical axis. The inset shows the same ogbn-arxiv results with a linear vertical axis to clarify their variation across depths.

model depth. The inset presents the ogbn-arxiv results on a linear vertical scale. On Cora, Photo, and Computer, SNDE decreases as model depth increases. Citeseer follows an overall downward trend after a small increase at two layers. These results show that DMSGL may continue to learn smoother representations on these graphs. The decrease remains visible at greater depths: from 32 to 64 layers, SNDE changes from 0.01124 to 0.00495 on Cora and from 0.01626 to 0.01248 on Photo. PubMed shows a pronounced reduction at shallow depths, followed by smaller changes. Its SNDE decreases from 0.07588 at one layer to 0.01259 at eight layers, then remains between 0.00797 and 0.01010 across the evaluated depths from 16 to 64 layers. Thus, most smoothing occurs at shallow depths, while deeper layers produce smaller changes in edge-wise variation. On ogbn-arxiv, SNDE follows a different trajectory. It starts at 0.52375, rises to 0.71404 at 16 layers, and decreases to 0.52949 at 64 layers. The deepest model therefore retains a level of edge-wise variation comparable to the shallowest model. Across the tested depths, the values fluctuate without a sustained decline towards zero. This behavior is consistent with nonlocal feature interactions continuing to contribute variation as propagation deepens.

Table 5: SNDE of representative models at 64 propagation layers. Values are reported as mean ± standard deviation. A dash denotes an unavailable result.

Method	Cora	Citeseer	Photo	Computer	PubMed	ogbn-arxiv
GCN	0.09082 ± 0.07793	0.15537 ± 0.09534	0.15886 ± 0.06136	0.06062 ± 0.00908	0.06788 ± 0.02896	0.11133 ± 0.02129
AERO-GNN	0.05656 ± 0.04932	0.02816 ± 0.01774	0.58181 ± 0.27749	0.66812 ± 0.37550	0.05230 ± 0.01643	–
APPNP	0.03431 ± 0.00418	0.01446 ± 0.00310	0.04022 ± 0.00233	0.04771 ± 0.00216	0.04217 ± 0.00112	0.19952 ± 0.00103
GPRGNN	0.02344 ± 0.00803	0.01536 ± 0.00770	0.05272 ± 0.01964	0.05805 ± 0.02924	0.02856 ± 0.00195	0.16737 ± 0.00073
GATv2R	0.08087 ± 0.00730	0.05181 ± 0.00442	0.20139 ± 0.00550	0.24324 ± 0.00669	0.17123 ± 0.00121	0.26762 ± 0.00138
DMSGL	0.00495 ± 0.00027	0.01243 ± 0.00299	0.01248 ± 0.00059	0.01018 ± 0.00154	0.00894 ± 0.00081	0.52949 ± 0.00542

C.3 Comparison with representative models

We select AERO-GNN, APPNP, GPRGNN, and GATv2R for their competitive classification accuracy in the 64-layer experiments, and include GCN as a standard message-passing baseline. Table 5 compares their SNDE with that of DMSGL at the same depth to examine how representation smoothness relates to classification performance. On Cora, Citeseer, Photo, Computer, and PubMed, DMSGL has lower SNDE than the selected baselines while maintaining strong classification performance. For example, GCN has an SNDE of 0.09082 on Cora, compared with 0.00495 for DMSGL, whereas their classification accuracies are 28.04% and 82.62%, respectively (Table 1). This comparison shows that substantial variation along graph edges can coexist with poor classification. Conversely, the low SNDE of DMSGL is compatible with representations that retain information useful for distinguishing classes.

One possible explanation is that smoothing reduces variation within groups of related nodes while preserving differences needed for classification. When many edges connect nodes within these groups, such representations can have low overall SNDE. This interpretation is consistent with the joint accuracy and SNDE results. Appendix D complements this edge-wise analysis by examining class-centroid separation and visualized class structure across model depths.

On ogbn-arxiv, DMSGL has an SNDE of 0.52949, while the available baselines range from 0.11133 to 0.26762. Combined with its classification result, this shows that DMSGL can also support accurate prediction with greater variation along the observed edges. The depth comparison establishes that this variation persists across the tested model depths. These findings connect the two roles of the model: local propagation incorporates graph structure, while Message-substitution provides access to features beyond observed neighborhoods.

D Representation Distinctness Analysis

To further examine node representation distinctness as model depth increases, we visualize Cora and Computer, which contain 7 and 10 classes, respectively. Their moderate numbers of classes facilitate color-based inspection of class clustering and mixing. We compare the node representations learned by GCN and DMSGL at depths of 2, 16, and 64 layers, using t-SNE to visualize their two-dimensional distributions and coloring nodes by class. In Figure 5, *Inter-class Dist.* (i.e. $D_{\text{inter}}^{(L)}$) denotes the average distance between class centroids, computed in the original representation space before dimensionality reduction:

$$\begin{aligned} \boldsymbol{\mu}_c^{(L)} &= \frac{1}{|\mathcal{V}_c|} \sum_{i \in \mathcal{V}_c} \mathbf{h}_i^{(L)}, \\ D_{\text{inter}}^{(L)} &= \frac{2}{C(C-1)} \sum_{c=1}^{C-1} \sum_{c'=c+1}^{C} \left\| \boldsymbol{\mu}_c^{(L)} - \boldsymbol{\mu}_{c'}^{(L)} \right\|_2, \end{aligned} \tag{37}$$

where $\mathbf{h}_i^{(L)}$ is the node representation learned by an L-layer model, $\mathcal{V}_c$ is the set of nodes from class c included in the analysis, and C is the number of classes. This metric characterizes the separation between class centroids and is considered jointly with the visualization and classification accuracy to assess representation distinctness.

The results on Cora show clear differences between the two propagation mechanisms in deep representation learning. GCN forms several distinguishable class regions at shallow depths; how-

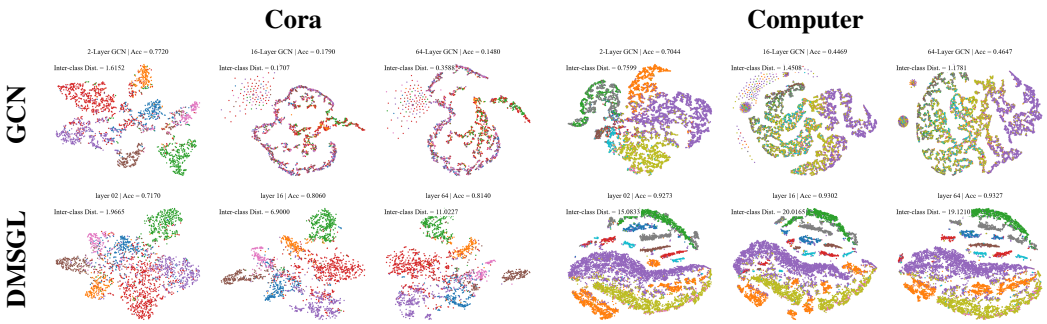


Figure 5: Node representation visualization on Cora (left) and Computer (right). The top and bottom rows show GCN and DMSGL, respectively. Within each dataset, model depths increase from 2 to 16 and 64 layers from left to right. Colors indicate node classes. Each panel reports classification accuracy and the average pairwise Euclidean distance between class centroids, computed in the original representation space before t-SNE.

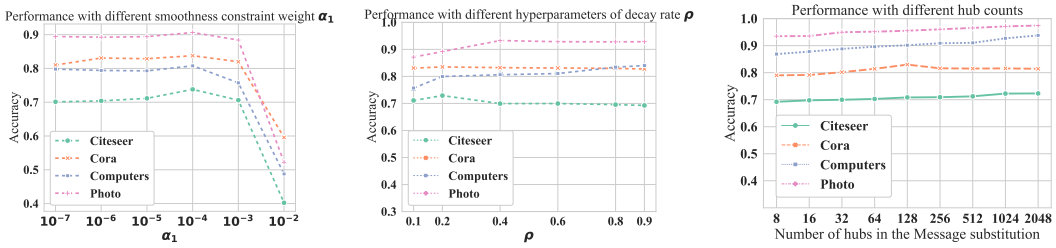
ever, as the number of layers increases, nodes from different classes gradually become intertwined within similar band-shaped structures, with the classification accuracy decreasing from 77.20% to 17.90% and 14.80%. This trend is consistent with the progressive weakening of inter-class distinctions caused by repeated neighborhood smoothing. In contrast, DMSGL preserves class-corresponding clusters as the network becomes deeper, with the classification accuracy increasing from 71.70% to 80.60% and 81.40%. Meanwhile, the inter-class centroid distance increases from 1.9665 to 6.9000 and 11.0227. The joint evolution of class distributions and predictive performance indicates that DMSGL can preserve effective node distinctions during deeper propagation and further improve classification performance.

The results on Computer further demonstrate the ability of DMSGL to preserve classification-relevant distinctions. As GCN is deepened from 2 layers to 16 and 64 layers, the classification accuracy decreases from 70.44% to 44.69% and 46.47%, respectively, while the inter-class centroid distance increases from 0.7599 to 1.4508 before declining to 1.1781. In the visualization, multiple classes in deep GCNs become intertwined within the same regions, reflecting the progressive weakening of class distinctions under repeated neighborhood mixing. In contrast, DMSGL preserves relatively clear class-specific clusters across all three depths, with the classification accuracy gradually increasing from 92.73% to 93.02% and 93.27%. These results indicate that, while adjusting the spatial organization among classes, the deep representations learned by DMSGL consistently preserve distinctions that are beneficial for node classification.

These observations are consistent with the design motivation of Message-substitution. GCN learns structural information through layer-wise neighborhood aggregation along graph edges, whereas continuous local mixing can gradually make the representations of nodes from different classes increasingly similar. Building upon local propagation, DMSGL introduces non-local features through layer-wise sampled macroscopic hubs and determines their contributions during feature propagation according to the Minimal Substitution model. Consequently, increasing the network depth enables each node to interact with features originating from different regions of the graph, allowing local structural learning and class discrimination to complement each other. Together with the preceding ablation studies, the visualization and classification results on both datasets provide consistent evidence for this design: DMSGL preserves the node distinguishability required for classification during deep propagation while continuing to improve predictive performance, highlighting its core advantage of **"Going Deeper, Staying Distinct."**

E HYPERPARAMETER STUDY

Laplacian coefficient α_1. We vary α_1 over $\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. Figure 6a shows the strongest performance around 10^{-4}. A larger value increases the high-frequency term's contribution in Eq. (11). Overemphasizing this term can reduce classification accuracy.



(a) Impact of the Laplacian coefficient α_1.

(b) Impact of the residual-weight parameter ρ.

(c) Impact of the Hub counts K.

Figure 6: Parameter sensitivity analysis of DMSGL.

Residual-weight parameter ρ. We vary ρ over $\{0.1, 0.2, 0.4, 0.6, 0.8, 0.9\}$. A larger ρ gives a faster geometric decay of the additional term in ω^l. As shown in Figure 6b, Cora is relatively insensitive to ρ. The best values in the reported sweep are 0.2 for Citeseer, 0.4 for Photo, and 0.9 for Computer.

Hub-count comparisons We analyze the sensitivity of DMSGL to the number of hubs by varying (K) over {8,16,32,...,2048}. As shown in Figure 6c, accuracy generally improves as (K) increases and then stabilizes with minor fluctuations. The improvement rate and saturation point vary across datasets due to differences in graph scale and structure. A larger hub set covers more nonlocal feature sources, giving each layer more opportunities to incorporate useful information from distant nodes and enrich the learned representations. As this coverage becomes sufficient, additional hubs provide increasingly redundant information, leading to the observed plateau. Meanwhile, the Adaptive Learning module controls the contribution of the global aggregate and limits the influence of noise introduced by the expanded hub set. DMSGL therefore maintains stable performance over a broad range of (K), without requiring precise tuning of the hub count.

F GLOBAL FEATURES VS LOCAL FEATURES

Local propagation embeds the observed graph structure, while Message-substitution incorporates nonlocal node features. Their contributions vary across classification tasks and propagation depths. We examine this variation through the adaptive coefficients δ^l learned by 16- and 32-layer DMSGL models on Cora and Citeseer. Figure 7 shows different coefficient profiles even between these two citation datasets. In the 16-layer Cora model, δ^l is negative between layers 2 and 9. Here, $\delta^l \in \mathbb{R}$ is a learned fusion coefficient. For $\delta^l < 0$, the update can be written as

$$\mathcal{F}_l = D^{-1/2}AD^{-1/2}H^{l-1} + |\delta^l| \left(D^{-1/2}AD^{-1/2}H^{l-1} - \mathfrak{g}_l \right).$$

Thus, the model strengthens the local term and subtracts the global aggregate, moving along the difference between local and global features. The global branch serves as a reference for adjusting the neighborhood representation in these layers. This gives Message-substitution a subtractive role alongside its role in supplying additional feature content.

The difference between the datasets becomes more pronounced in the 32-layer models. Citeseer assigns greater weight to the global branch in shallower layers and shifts toward local propagation after layer 26. Cora shows the opposite trend: δ^l increases with depth, assigning greater weight to global features in later layers. These profiles suggest different sequences of feature integration: Citeseer first incorporates nonlocal context and then emphasizes neighborhood structure, while Cora first emphasizes local structure and subsequently increases nonlocal input. The adaptive coefficients allow the strength and direction of this interaction to change as node representations evolve across layers.

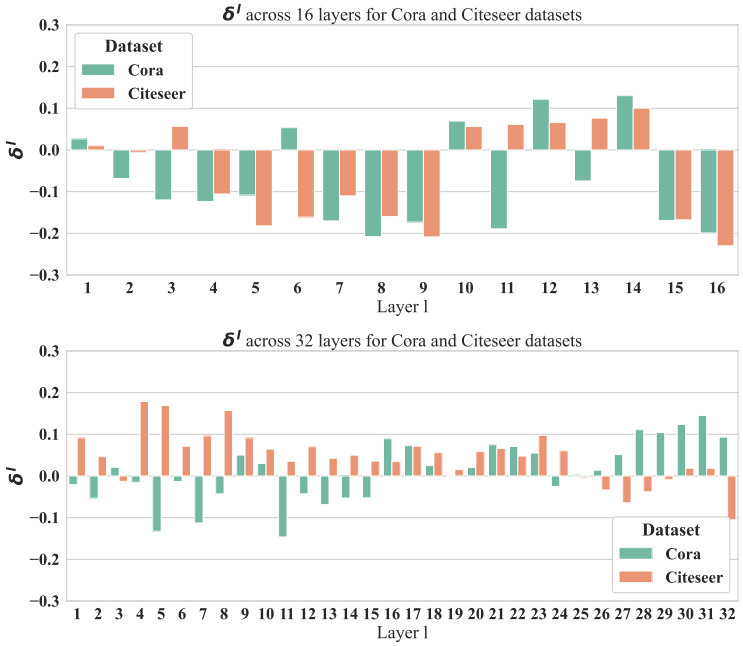


Figure 7: Layer-wise adaptive fusion coefficients δ^l learned by DMSGL on Cora and Citeseer with 16 (top) and 32 (bottom) layers.