
On Vanishing Gradients in GNNs: Bridging Recurrent and Graph Learning - Supplementary Material

Anonymous Author(s)

Affiliation

Address

email

1 A Theoretical Results

2 A.1 Proofs of Jacobian Theorems

3 **Definition A.1** (Vectorization and Kronecker product). Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be a real matrix. The
4 *vectorization* of \mathbf{X} , denoted $\text{vec}(\mathbf{X})$, is the (mn) -dimensional column vector obtained by stacking
5 the columns of \mathbf{X} :

$$\text{vec}(\mathbf{X}) = \begin{bmatrix} \mathbf{X}_{:,1} \\ \mathbf{X}_{:,2} \\ \vdots \\ \mathbf{X}_{:,n} \end{bmatrix} \in \mathbb{R}^{mn}.$$

6 One key property of the vectorization operator is its relationship to the Kronecker product. In
7 particular, for compatible matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, we have

$$\text{vec}(\mathbf{A} \mathbf{B} \mathbf{C}) = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}).$$

8 Here, \otimes denotes the Kronecker product.

9 **Definition A.2** (Wishart matrix). Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ be a matrix with i.i.d. entries $X_{ij} \sim \mathcal{N}(0, \sigma^2)$. The
10 random matrix $\mathbf{X}^T \mathbf{X} \in \mathbb{R}^{p \times p}$ is called a *Wishart matrix* (up to a scaling factor). In particular, such a
11 matrix follows the Wishart distribution $\mathcal{W}_p(n, \sigma^2)$ in certain parametrizations.

12 **Definition A.3** (Marchenko–Pastur distribution. [41]). In the high-dimensional limit ($n, p \rightarrow \infty$ at
13 a fixed ratio $p/n \rightarrow c$), the empirical eigenvalue distribution of the (properly normalized) Wishart
14 matrix $\mathbf{X}^T \mathbf{X}$ converges to the *Marchenko–Pastur distribution*. Concretely, if $\mathbf{X} \in \mathbb{R}^{n \times p}$ has entries
15 $\mathcal{N}(0, 1)$, then the eigenvalues of $\mathbf{X}^T \mathbf{X}$ lie within $[(1 - \sqrt{c})^2, (1 + \sqrt{c})^2]$ for large n, p , and their
16 density converges to

$$f_{\text{MP}}(x) = \frac{1}{2\pi c x} \sqrt{(x - a_{\min})(a_{\max} - x)}, \quad x \in [a_{\min}, a_{\max}],$$

17 with $a_{\min} = (1 - \sqrt{c})^2$ and $a_{\max} = (1 + \sqrt{c})^2$. If the entries of \mathbf{X} have variance $\sigma^2 \neq 1$, then the
18 support is rescaled by σ^2 .

19 **Lemma A.4** (Spectrum of the Jacobian’s singular values). Let $\mathbf{H}^{(k)} = \tilde{\mathbf{A}} \mathbf{H}^{(k-1)} \mathbf{W}$ be a linear
20 GCN layer, where $\tilde{\mathbf{A}}$ has eigenvalues $\{\lambda_1, \dots, \lambda_n\}$ and $\mathbf{W} \mathbf{W}^T$ has eigenvalues $\{\mu_1, \dots, \mu_{d_k}\}$.
21 Consider the layer-wise Jacobian $\mathbf{J} = \partial \text{vec}(\mathbf{H}^{(k)}) / \partial \text{vec}(\mathbf{H}^{(k-1)})$. Then the squared singular
22 values of \mathbf{J} are given by the set

$$\{\lambda_i^2 \mu_j \mid i = 1, \dots, n, \quad j = 1, \dots, d_k\}.$$

23 *Proof.* By the property of vectorization (Definition A.1), we have

$$\text{vec}(\tilde{\mathbf{A}} \mathbf{H}^{(k-1)} \mathbf{W}) = (\mathbf{W}^T \otimes \tilde{\mathbf{A}}) \text{vec}(\mathbf{H}^{(k-1)}).$$

24 Hence

$$\mathbf{J} = \mathbf{W}^T \otimes \tilde{\mathbf{A}}.$$

25 By properties of the Kronecker product, the eigenvalues of $\mathbf{J} \mathbf{J}^T$ are the products of the eigenvalues
26 of $\mathbf{W}^T \mathbf{W}$ and $\tilde{\mathbf{A}}^2$. Equivalently,

$$\text{spec}(\mathbf{J} \mathbf{J}^T) = \text{spec}(\mathbf{W}^T \mathbf{W}) \otimes \text{spec}(\tilde{\mathbf{A}}^2),$$

27 where spec is the vectorized version of the set of eigenvalues of a matrix. If $\mathbf{W}^T \mathbf{W}$ has eigenvalues
28 $\{\mu_j\}_{j=1}^{d_k}$ and $\tilde{\mathbf{A}}^2$ has eigenvalues $\{\lambda_i^2\}_{i=1}^n$, then the squared singular values of \mathbf{J} are precisely $\lambda_i^2 \mu_j$
29 for $i \in \{1, \dots, n\}$, $j \in \{1, \dots, d_k\}$. \square

30 **Theorem A.5** (Jacobian singular-value distribution). *Assume the setting of Lemma ??, and let*
31 *$\mathbf{W} \in \mathbb{R}^{d_{k-1} \times d_k}$ be initialized with i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. Denote the squared singular values of*
32 *the Jacobian by $\gamma_{i,j}$. Then, for sufficiently large d_k the empirical eigenvalue distribution $\mathbf{W} \mathbf{W}^T$*
33 *converges to the Marchenko-Pastur distribution. Then, the mean and variance of each $\gamma_{i,j}$ are*

$$\mathbb{E}[\gamma_{i,j}] = \lambda_i^2 \sigma^2, \quad (1)$$

$$\text{Var}[\gamma_{i,j}] = \lambda_i^4 \sigma^4 \frac{d_k}{d_{k-1}}. \quad (2)$$

34 *Proof.* In this setting, $\mathbf{W} \mathbf{W}^T$ is Wishart if \mathbf{W} has i.i.d. Gaussian entries. Its eigenvalues μ_j thus
35 converge to the Marchenko–Pastur distribution for large d_k . From standard results on the moments of
36 Wishart eigenvalues,

$$\mathbb{E}(\mu_j) = \sigma^2, \quad \text{Var}(\mu_j) = \sigma^4 \frac{d_k}{d_{k-1}}.$$

37 Since $\gamma_{i,j} = \lambda_i^2 \mu_j$, we obtain

$$\mathbb{E}[\gamma_{i,j}] = \lambda_i^2 \mathbb{E}[\mu_j] = \lambda_i^2 \sigma^2,$$

38

$$\text{Var}[\gamma_{i,j}] = \lambda_i^4 \text{Var}(\mu_j) = \lambda_i^4 \sigma^4 \frac{d_k}{d_{k-1}}.$$

39 This completes the proof. \square

40 **Proposition A.6** (Effect of state-space matrices). *Consider the setting in (??) and $\Gamma =$*
41 *$\partial \text{vec}(\mathbf{F}_\theta(\mathbf{H}^{(k)}))/\partial \text{vec}(\mathbf{H}^{(k)})$. Let \otimes denote the Kronecker product. Then, the norm of the vectorized*
42 *Jacobian \mathbf{J} is bounded as:*

$$\begin{aligned} \|\mathbf{J}\|_2 &\leq \|I_{d_k} \otimes \mathbf{A}\|_2 + \|I_{d_k} \otimes \mathbf{B}\|_2 \|\Gamma\|_2 \\ &= \|\mathbf{A}\|_2 + \|\mathbf{B}\|_2 \|\Gamma\|_2, \end{aligned} \quad (3)$$

43 *Proof.* We start by writing

$$\mathbf{J} = (I_{d_k} \otimes \mathbf{A}) + (I_{d_k} \otimes \mathbf{B}) \Gamma.$$

44 Using the triangle inequality for the spectral norm,

$$\|\mathbf{J}\|_2 = \|(I_{d_k} \otimes \mathbf{A}) + (I_{d_k} \otimes \mathbf{B}) \Gamma\|_2 \leq \|I_{d_k} \otimes \mathbf{A}\|_2 + \|(I_{d_k} \otimes \mathbf{B}) \Gamma\|_2.$$

45 By the submultiplicative property of the spectral norm,

$$\|(I_{d_k} \otimes \mathbf{B}) \Gamma\|_2 \leq \|I_{d_k} \otimes \mathbf{B}\|_2 \|\Gamma\|_2.$$

46 Since $\|I_{d_k} \otimes \mathbf{M}\|_2 = \|\mathbf{M}\|_2$ for any matrix \mathbf{M} , we obtain

$$\|I_{d_k} \otimes \mathbf{A}\|_2 = \|\mathbf{A}\|_2 \quad \text{and} \quad \|I_{d_k} \otimes \mathbf{B}\|_2 = \|\mathbf{B}\|_2.$$

47 Hence,

$$\|\mathbf{J}\|_2 \leq \|\mathbf{A}\|_2 + \|\mathbf{B}\|_2 \|\Gamma\|_2.$$

48 \square

49 A.2 Proofs to Smoothing Theorems

50 **Definition A.7** (Lipschitz continuity). A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is Lipschitz continuous if there
 51 exists an $L \geq 0$ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we have that:

$$\|f(\mathbf{x}) - f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|,$$

52 where we equip \mathbb{R}^n and \mathbb{R}^m with their respective norms. The minimal such L is called the Lipschitz
 53 constant of f .

54 The notion of Lipschitz continuity is effectively a bound on the rate of change of a function. It is
 55 therefore not surprising that one can relate the Lipschitz constant to the Jacobian of f . In particular,
 56 we state a useful and well-known result [32] that relates the (continuous) Jacobian map \mathbf{J}_f of a
 57 continuous function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ to its Lipschitz constant $L \geq 0$. In particular, the Lipschitz
 58 constant is the supremum of the (induced) norm of the Jacobian taken over its domain.

59 **Lemma A.8** ([32]). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ be continuous, with continuous Jacobian \mathbf{J}_f . Con-
 60 sider a convex set $U \subseteq \mathbb{R}^n$. If there exists $L \geq 0$ such that $\|\mathbf{J}_f(\mathbf{x})\| \leq L$ for all $\mathbf{x} \in U$, then
 61 $\|f(\mathbf{x}) - f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|$. In particular, we have that the Lipschitz constant of f is L :

$$L = \sup_{\mathbf{x} \in U} \|\mathbf{J}_f(\mathbf{x})\|.$$

62 The condition of U being convex is a technicality that is easily achieved in practice with the assump-
 63 tion that input features are bounded and that therefore they live in a convex hull U . In particular, at
 64 each layer k one can also find a convex hull U_k such that the image of the layer $k - 1$ is contained
 65 within U_k . We highlight that for non-linearities such as ReLU, there are technical difficulties when
 66 taking this supremum as there is a non-differentiable point at 0. This can be circumvented by consid-
 67 ering instead a supremum of the (Clarke) generalized Jacobian [35]. We ignore this small detail in
 68 this work for simplicity as for ReLU this is equivalent to considering the supremum over $U/\mathbf{0}$, i.e.
 69 simply ignoring the problematic point $\mathbf{0}$.

70 **Lemma A.9.** Consider a GNN layer f_ℓ as in Equation ??, with non-linearity σ such that $\sigma(0) = 0$
 71 (e.g. ReLU or tanh). Then, $f(\mathbf{0}) = \mathbf{0}$, i.e. $\mathbf{0}$ is a fixed point of f .

72 *Proof.* $f_\ell(\mathbf{0}) = \sigma(\hat{\mathbf{A}}\mathbf{0}\mathbf{W}) = \sigma(\mathbf{0}) = \mathbf{0}$. □

73 **Proposition A.10** (Convergence to unique fixed point.). Let $\|f_\ell\|_{\text{Lip}} \leq 1 - \epsilon$ for some $\epsilon > 0$ for all
 74 $\ell = 1 \dots L$. Then, for $\mathbf{H} \in U \subseteq \mathbb{R}^{nd}$, we have that:

$$\|f(\mathbf{H})\| \leq (1 - \epsilon)^L \|\mathbf{H}\| < \|\mathbf{H}\|. \quad (4)$$

75 In particular, as $L \rightarrow \infty$, $f(\mathbf{H}) \rightarrow \mathbf{0}$.

76 *Proof.* By Lipschitz regularity of f over U , we have that $\|f(\mathbf{x}) - f(\mathbf{y})\| \leq \|f\|_{\text{Lip}} \|\mathbf{x} - \mathbf{y}\|$. Recall
 77 that by Lemma ??, we have that $f(\mathbf{0}) = \mathbf{0}$. This implies:

$$\begin{aligned} \|f(\mathbf{H}) - f(\mathbf{0})\| &= \|f(\mathbf{H})\| \\ &\leq \|f\|_{\text{Lip}} \|\mathbf{H}\| \\ &\leq \prod_{\ell=1}^L \|f_\ell\|_{\text{Lip}} \|\mathbf{H}\| \\ &< \|\mathbf{H}\|, \end{aligned}$$

78 where in the last step we use the fact that Lipschitz constants are submultiplicative and that for all ℓ
 79 we have that $\|f_\ell\|_{\text{Lip}} < 1$ by assumption. The final statement is immediate by the Banach fixed point
 80 theorem and by noting that f_ℓ all share the same fixed point $\mathbf{0}$ by Lemma ??. □

81 **Proposition A.11** (Contractions decrease Dirichlet energy.). *Let f be a GNN, $|E|$ be the number of*
 82 *edges in G , and $\mathbf{H} \in \mathbb{R}^{nd}$. We have the following bound:*

$$\mathcal{E}(f(\mathbf{H})) \leq 2|E| \prod_{\ell=1}^L \|f_\ell\|_{\text{Lip}}^2 \|\mathbf{H}\|^2. \quad (5)$$

83 *In particular, if $\|f_\ell\|_{\text{Lip}} \leq 1 - \epsilon$ for some $\epsilon > 0$ for all $\ell = 1 \dots L$, then as $L \rightarrow \infty$, $\mathcal{E}(f(\mathbf{H})) \rightarrow 0$.*

84 *Proof.* We denote by $f(\mathbf{H})|_i \in \mathbb{R}^d$, the d -dimensional evaluation of $f(\mathbf{H})$ at node i . We make use of
 85 the inequality $\|f(\mathbf{H})|_i\| \leq \|\mathbf{H}\|$.

$$\begin{aligned} \mathcal{E}(f(\mathbf{H})) &= \sum_{i \sim j} \|f(\mathbf{H})|_i - f(\mathbf{H})|_j\|^2 \\ &\leq \sum_{i \sim j} \|f(\mathbf{H})|_i\|^2 + \|f(\mathbf{H})|_j\|^2 \\ &\leq 2 \sum_{i \sim j} \|f(\mathbf{H})\|^2 \\ &\leq 2 \|f\|_{\text{Lip}}^2 \sum_{i \sim j} \|\mathbf{H}\|^2 \\ &= 2 \|f\|_{\text{Lip}}^2 |E| \|\mathbf{H}\|^2 \\ &\leq 2 \prod_{\ell=1}^L \|f_\ell\|_{\text{Lip}}^2 |E| \|\mathbf{H}\|^2. \end{aligned}$$

86 It is then clear that, if $\|f_\ell\|_{\text{Lip}} \leq 1 - \epsilon$ for some $\epsilon > 0$ for all $\ell = 1 \dots L$,
 87 $\prod_{\ell=1}^L \|f_\ell\|_{\text{Lip}}^2 \leq (1 - \epsilon)^{2L} \rightarrow 0$ as $L \rightarrow \infty$.
 88

89 We note that a similar procedure was used in [46, 8] for the specific case of GCNs. Our procedure is
 90 more general, as we use the Lipschitz constant of the network, which only requires knowledge of the
 91 input-output Jacobian of each layer of the network. In the case of GCN, this would encapsulate the
 92 dynamics of the adjacency and weight matrix, and also allows us to understand how any GNN (no
 93 matter how complex its internal structure) affects the Dirichlet energy, without requiring the use of
 94 heavy assumptions or simplifications for mathematical tractability. \square

95 **B kGNN-SSM: A simple method to combine high connectivity and** 96 **non-dissipativity.**

97 To test our assumption on more complex downstream tasks, we construct a minimal model that
 98 combines high connectivity with non-dissipativity. To guarantee high connectivity, we employ a
 99 k -hop aggregation scheme. In particular, each node i at layer k will aggregate information as

$$a_{i,k}^{(k)} = \psi^k \left(\{h_j^{(k)} : j \in \mathcal{N}_k(i)\} \right), \quad (6)$$

100 where

$$\mathcal{N}_k(i) := \{j \in V : d_G(i, j) = k\}$$

101 and $d_G : V \times V \rightarrow \mathbb{R}_{\geq 0}$ is the length of the minimal walk connecting nodes i and j . This approach
 102 avoids a large amount of information being squashed into a single vector, and is more in line with the
 103 recurrent paradigm. We note that this scheme is similar to [18], but in this case we do not consider
 104 different block or parameter sharing, and our recurrent mechanism is based on an untrained SSM
 105 layer.

106 We denote a GNN endowed with this rewiring scheme and wrapped with our SSM layer as kGNN-SSM.

107 C Experimental Details

108 In this section, we provide additional experimental details, including dataset and experimental setting
109 description and employed hyperparameters.

110 **Over-smoothing task.** In this task, we aim to analyze the dynamics of the Dirichlet energy across
111 three different graph topologies: Cora [63], Texas [48], and a grid graph. The Cora dataset is a
112 citation network consisting of 2,708 nodes (papers) and 10,556 edges (citations). The Texas dataset
113 represents a webpage graph with 183 nodes (web pages) and 499 edges (hyperlinks). Lastly, the grid
114 graph is a two-dimensional 10×10 regular grid with 4-neighbor connectivity. For all three graphs,
115 node features are randomly initialized from a normal distribution with a mean of 0 and variance of
116 1. These node features are then propagated over 80 layers (or iterations) using untrained GNNs to
117 observe the energy dynamics.

118 **Graph Property Prediction.** This experiment consists of predicting two node-level (i.e., eccentric-
119 ity and single source shortest path) and one graph-level (i.e., graph diameter) properties on synthetic
120 graphs sampled from different distribution, i.e., Erdős-Rényi, Barabasi-Albert, grid, caveman, tree,
121 ladder, line, star, caterpillar, and lobster. Each graph contains between 25 and 35 nodes, with nodes
122 assigned with input features sampled from a uniform distribution in the interval $[0, 1)$. The target
123 values correspond to the predicted graph property. The dataset consists of 5,120 graphs for training,
124 640 for validation and 1,280 for testing.

125 We employ the same experimental setting and data outlined in [25]. Each model is designed as
126 three components: the encoder, the graph convolution, and the readout. We perform hyperparameter
127 tuning via grid search, optimizing the Mean Square Error (MSE). The models are trained using the
128 Adam optimizer for a maximum of 1500 epochs, with early stopping based on the validation error,
129 applying a 100 epochs patience. For each model configuration, we perform 4 training runs with
130 different weight initializations and report the average results. We report in Table 1 the employed grid
131 of hyperparameters.

132 **Long-Range Graph Benchmark.** We consider the peptides-func and peptides-struct
133 datasets from [22]. Both datasets consist of 15,535 graphs, where each graph corresponds to
134 1D amino acid chain (i.e., peptide), where nodes are the heavy atoms of the peptide and edges are the
135 bonds between them. peptides-func is a multi-label graph classification dataset whose objective
136 is to predict the peptide function, such as antibacterial and antiviral function. peptides-struct is a
137 multi-label graph regression dataset focused on predicting the 3D structural properties of peptides,
138 such as the inertia of the molecule and maximum atom-pair distance.

139 We use the same experimental setting and splits from [22]. We perform hyperparameter tuning
140 via grid search, optimizing the Average Precision (AP) in the Peptides-func and Mean Absolute
141 Error (MAE) in the Peptide-struct. The models are trained using the AdamW optimizer for a
142 maximum of 300 epochs. For each model configuration, we perform four training runs with different
143 weight initializations and report the average results. We report in Table 1 the employed grid of
144 hyperparameters.

145 **Tested Hyperparameters.** In Table 1 we report the grid of hyperparameters employed in our
146 experiments by our method.

147 All experiments were run in a single NVIDIA RTX4090 GPU.

148 D Additional empirical results

149 In this section, we propose additional empirical results on over-smoothing and over-squashing, as
150 well as the eigendistribution of the layerwise Jacobians of various standard GNNs.

151 D.1 Additional MPNN Jacobians

152 Here, we present in Figure 1 the eigendistribution of the layerwise Jacobians of GCN, GIN [62]
153 and Gated-GCN [7]. Across the board, we observe similar contraction effects in the Jacobian as

Table 1: The grid of hyperparameters employed during model selection for the graph property prediction tasks (*GraphProp*), and *peptides-func* and *peptides-struct*.

Hyperparameters	Values	
	<i>GraphProp</i>	peptides- (func, struct)
Optimizer	Adam	AdamW
Learning rate	0.003	0.001
Weight decay	10^{-6}	-
N. Layers	10	40, 17
embedding dim	20, 30	105
σ	tanh	ReLU
eig(Λ)	0.5, 0.75, 1.0	1.0

those presented in the main paper, with a long number of eigenvalues accumulating at zero, with no significant changes in the distribution during training. However, the maximum eigenvalues for both GIN and Gated-GCN are much larger than those of GCN. We also compare a nonlinear feedforward network and a nonlinear GCN in Figure 2.

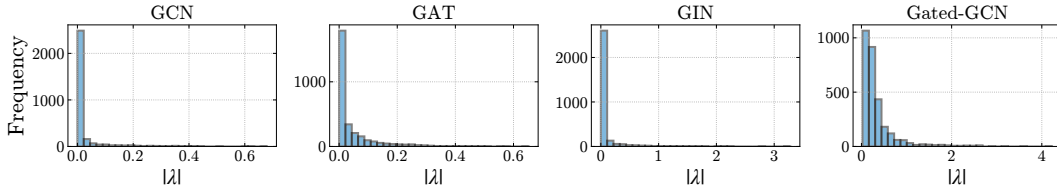


Figure 1: Eigenvalues of layer-to-layer Jacobian of different GNN models.

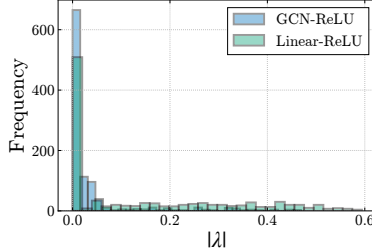


Figure 2: Histogram of eigenvalue modulus of the layerwise Jacobian for a nonlinear convolutional and a nonlinear feedforward layer.

D.2 Additional Over-Smoothing Results

Here, we include additional results related to over-smoothing experiments. Figure 3 shows the effect of $\|\Lambda\|_2$ in GCN-SSM on different graph structures, showing that lower Jacobian norms leads to a rapid decay of the Dirichlet energy, whereas values closer to one result in a more stable energy evolution. This result is also confirmed by Figure 5 and Figure 6. The former presents the vectorized Jacobian for ADGN [25], SWAN [26], and PHDGN [33] on Cora, while the latter the Dirichlet energy evolution of different models on different topologies. Notably, in Figure 6, ADGN, SWAN, and PHDGN exhibit stable Dirichlet energy across layers, and Figure 5 reveals that these Jacobian norms are close to one. These results confirm that stable dynamics also ensure a non-decaying Dirichlet energy, effectively preventing over-smoothing.

D.3 Link between delay and vanishing gradients

Here, we show how the delay term in [30] is directly related to preventing vanishing gradients. We do so by showing that adding the delay term to a GCN is effective at preventing over-smoothing, see Figure 7, as well as by checking the histogram of eigenvalues of the Jacobian, see Figure 9.

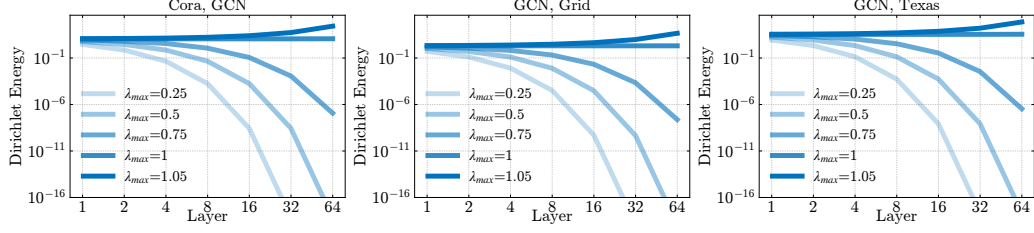


Figure 3: Dirichlet Energy evolution of GCN-SSM for different $\|\Lambda\|_2$ on different graph topologies. **Left:** Cora. **Middle:** Grid graph. **Right:** Texas.

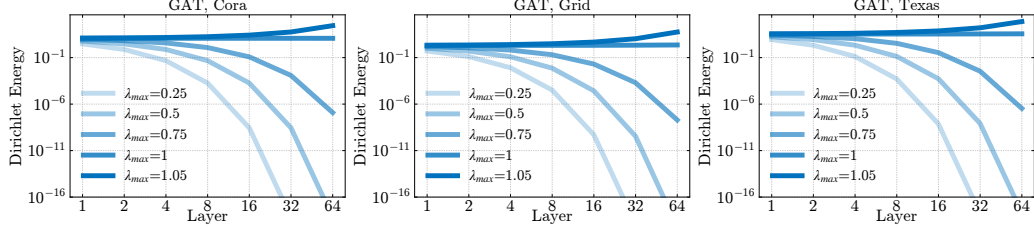


Figure 4: Dirichlet Energy evolution of GAT-SSM for different $\|\Lambda\|_2$ on different graph topologies. **Left:** Cora. **Middle:** Grid graph. **Right:** Texas.

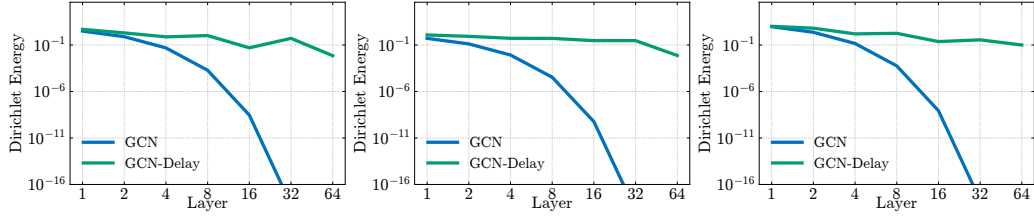


Figure 7: Dirichlet Energy evolution of GCN (+delay mechanism) on different topologies. **Left:** Cora. **Middle:** Grid graph. **Right:** Texas.

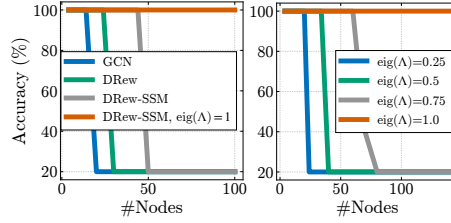


Figure 8: **Left:** Performance on the RingTransfer task for DRew [30]. **Right:** Effect of dissipativity on performance.

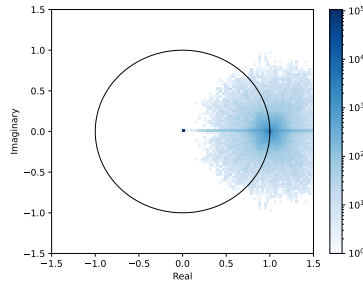


Figure 9: Eigenvalue distribution of DRew-GCN+delay on the ring transfer task.

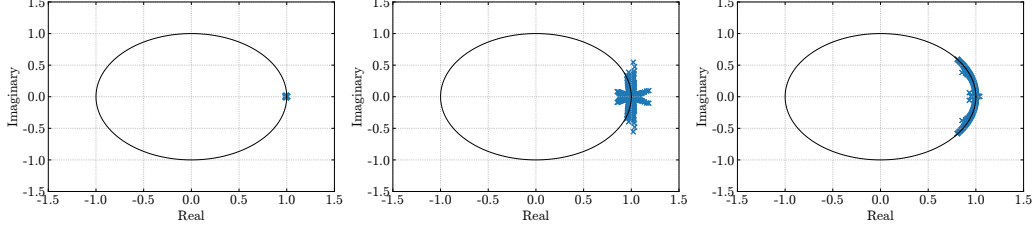


Figure 5: Vectorized Jacobian for ADGN [25], SWAN [26], and PHDGN [33] on Cora. **Left:** ADGN. **Middle:** SWAN. **Right:** PHDGN.

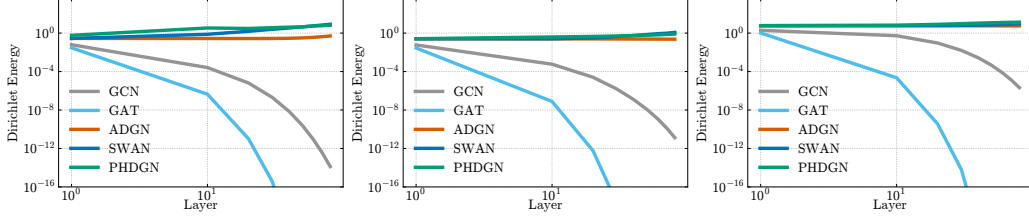


Figure 6: Dirichlet Energy evolution of different models on different topologies. **Left:** Cora. **Middle:** Grid graph. **Right:** Texas.

172 D.4 Graph Property Prediction

173 **Edge-of-chaos behavior and long-range propagation.** To further support our claim that mitigating
 174 gradient vanishing is key to strong long-range performance, Figure 10 shows each method’s average
 175 Jacobian eigenvalue distance to the edge-of-chaos (EoC) region. The figure demonstrates that methods
 176 such as ADGN [25] and SWAN [26], which remain closer to EoC, effectively propagate information
 177 over large graph radii, resulting in superior performance across all three tasks. Figure 11 presents
 178 an ablation study on multiple ADGN variants, controlled by the hyperparameter γ , which governs
 179 the positioning of the Jacobian eigenvalues ($\gamma < 0$ places them outside the stability region, $\gamma > 0$
 180 inside, and $\gamma = 0$ on the unit circle). Notably, regardless of the initial value of γ , ADGN consistently
 181 converges towards the EoC region as performance improves.

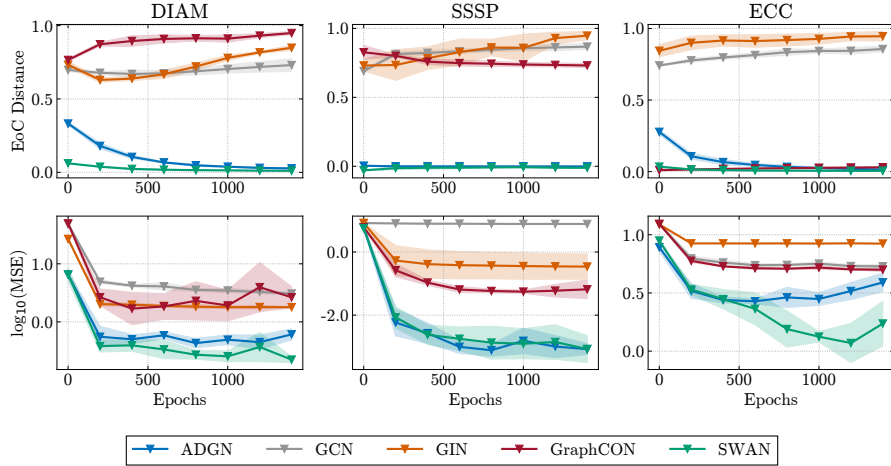


Figure 10: Performance on Graph Property Prediction tasks and average Jacobian eigenvalue distance to the edge of chaos (EoC) region for different GNN models.

182 **Complete results.** Table 2 compares our method on graph property prediction tasks against a range of
 183 state-of-the-art approaches, including GCN [37], GAT [58], GraphSAGE [31], GIN [62], GCNII [16],
 184 DGC [59], GRAND [13], GraphCON [52], ADGN [25], SWAN [26], PH-DGN [33], and DRew [30].
 185 Our method achieves exceptional results across all three tasks, consistently surpassing MPNN

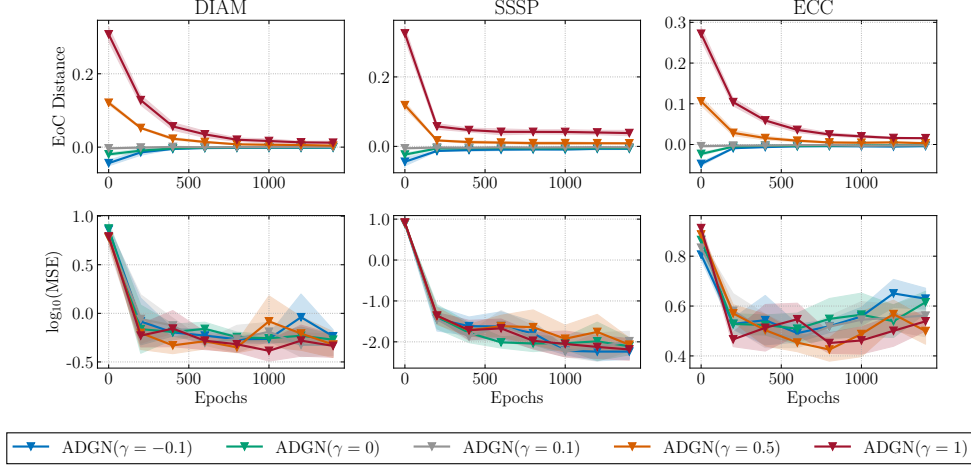


Figure 11: Performance on Graph Property Prediction tasks and average Jacobian eigenvalue distance to the edge of chaos (EoC) region for different ADGN dynamics, i.e., $\gamma \in [-0.1, 1]$. Negative values of γ places the eigenvalues of the ADGN Jacobian outside the stability region, otherwise for positive values.

186 baselines, differential equation-inspired GNNs, and multi-hop GNNs. These findings underscore how
 187 combining powerful model dynamics with improved connectivity provides substantial benefits in
 188 tasks that require long-range information propagation.

Table 2: Mean test set $\log_{10}(\text{MSE})(\downarrow)$ and std averaged on 4 random weight initializations on Graph Property Prediction tasks. The lower, the better. Baseline results are reported from [25, 26, 33].

Model	Diameter	SSSP	Eccentricity
MPNNs			
GCN	0.7424 \pm 0.0466	0.9499 \pm 0.0001	0.8468 \pm 0.0028
GAT	0.8221 \pm 0.0752	0.6951 \pm 0.1499	0.7909 \pm 0.0222
GraphSAGE	0.8645 \pm 0.0401	0.2863 \pm 0.1843	0.7863 \pm 0.0207
GIN	0.6131 \pm 0.0990	-0.5408 \pm 0.4193	0.9504 \pm 0.0007
GCNII	0.5287 \pm 0.0570	-1.1329 \pm 0.0135	0.7640 \pm 0.0355
Differential Equation inspired GNNs			
DGC	0.6028 \pm 0.0050	-0.1483 \pm 0.0231	0.8261 \pm 0.0032
GRAND	0.6715 \pm 0.0490	-0.0942 \pm 0.3897	0.6602 \pm 0.1393
GraphCON	0.0964 \pm 0.0620	-1.3836 \pm 0.0092	0.6833 \pm 0.0074
ADGN	-0.5188 \pm 0.1812	-3.2417 \pm 0.0751	0.4296 \pm 0.1003
SWAN	-0.5981 \pm 0.1145	-3.5425 \pm 0.0830	-0.0739 \pm 0.2190
PH-DGN	-0.5473 \pm 0.1074	-4.2993 \pm 0.0721	-0.9348 \pm 0.2097
Graph Transformers			
GPS	-0.5121 \pm 0.0426	-3.5990 \pm 0.1949	0.6077 \pm 0.0282
Multi-hop GNNs			
DRew-GCN	-2.3692 \pm 0.1054	-1.5905 \pm 0.0034	-2.1004 \pm 0.0256
+ delay	-2.4018 \pm 0.1097	-1.6023 \pm 0.0078	-2.0291 \pm 0.0240
Our			
GCN-SSM	-2.4312 \pm 0.0329	-2.8206 \pm 0.5654	-2.2446 \pm 0.0027
+ eig(Λ) \approx 1	-2.4442 \pm 0.0984	-3.5928 \pm 0.1026	-2.2583 \pm 0.0085
+ k-hop	-3.0748 \pm 0.0545	<u>-3.6044</u> \pm 0.0291	-4.2652 \pm 0.1776

189 D.5 Additional comments on LRGB tasks

190 In our experiments with the LRGB tasks, we observe that the peptides-func task exhibits signifi-
 191 cantly longer-range dependencies than the peptides-struct task. Notably, the peptides-struct

task performs best when the model is not initialized at the edge of chaos and requires fewer layers. Conversely, on `peptides-struct` the model performs best when it is set to be at the edge of chaos, and shows a monotonic performance increase with additional layers, with optimal results achieved when using forty layers.

Furthermore, we highlight that while our experiments with a small hidden dimension adhere to the parameter budget established in [22], increasing the hidden dimension ($d \uparrow$) to 256 causes us to exceed the 500k parameter budget limit, even though our model maintains the same number of parameters as a regular GCN. While this budget is a useful tool to benchmark different models, we highlight that this restriction results in models running with fewer layers and small hidden dimensions. However, a large number of layers is crucial for effective long-range learning in graphs that are not highly connected, while increasing the hidden dimension also directly affects the bound in Theorem ???. As such, we believe that this parameter budget indirectly benefits models with higher connectivity graphs, inadvertently hindering models that do not perform edge addition.

D.6 Scalability Results

In terms of runtime, GNN-SSM has two additional (fixed) matrices to store w.r.t. its backbone, and involves only an addition and two additional matrix multiplications. This has a negligible effect on training time, thus our model retains the complexity of its backbone, see Table 3 below.

Table 3: Epoch Time (sec.) for GCN and GCN-SSM when performing node classification of the Cora dataset.

Layers	GCN	GCN-SSM
5	0.009	0.009
10	0.015	0.017
20	0.025	0.031
30	0.041	0.046
40	0.053	0.051
50	0.066	0.075
60	0.078	0.089

D.7 State-Space Matrices Sensitivity

Empirically, we have that sharing Λ across layers did not alter performance: using a single fixed Λ yielded the same accuracy as training each Λ_i with identical dynamics. Fixing Λ ensures the system remains at the edge of chaos during training. Preserving this prior under a trainable Λ would require additional constraints, in line with stabilization techniques used in RNN architectures, see Table 4 below.

Table 4: Performance comparison for different design choices when performing node classification of the Cora dataset.

Layers	GCN-SSM (shared)	GCN-SSM (no sharing)	GCN-SSM (trained Λ)
5	76.3	78.3	71.3
10	78.5	78.5	71.1
20	81.2	77.8	49.9
30	78.1	79.6	33.8
40	77.5	78.5	31.9
50	76.4	74.6	31.9
60	77.8	77.4	31.9

E Additional Details and Comments on Over-Smoothing

E.1 Choice of Feature Distance Measure

Throughout the paper we adopt the *unnormalized Dirichlet energy*

$$\mathcal{E}(\mathbf{H}) = \text{tr}(\mathbf{H}^\top \mathbf{L} \mathbf{H}) = \sum_{(u,v) \in \mathcal{E}} \|\mathbf{h}_u - \mathbf{h}_v\|_2^2,$$

This choice aligns with several well cited papers in the over-smoothing literature [52, 51]. Moreover, the connection we unravel between vanishing gradients and over-smoothing also explains why techniques borrowed from recurrent architectures [52, 25] are expected to mitigate feature collapse in GNNs.

While our theoretical analysis focuses on $\mathcal{E}(\mathbf{H})$ for mathematical simplicity, we will now also evaluate an alternative smoothness measure to ensure our insights generalize beyond this choice. In particular, we use the smoothness measure employed in other over-smoothing works [61, 54]

$$\mu(\mathbf{H}) = \|\mathbf{H} - \mathbf{1} \gamma_{\mathbf{H}}\|_F, \quad \gamma_{\mathbf{H}} = \frac{\mathbf{1}^\top \mathbf{H}}{N},$$

Next, we report empirical experiments on this measure, which empirically shows that the qualitative trends predicted by our unnormalized-energy theory also manifest under this alternative metric. Although formal equivalence between these energies and our collapse proofs is not explored, this empirical alignment provides strong justification for the broader applicability of our analysis to the broader literature on oversmoothing.

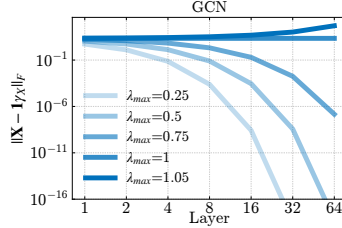


Figure 12: GCN-SSM on Cora with $\mu(\mathbf{H})$ smoothing measure.

E.2 The effect of the Jacobian spectrum on node classification performance

In order to assess how the spectral properties of the layer-wise Jacobian influence node-classification performance, we carried out the following two experiments on the Cora citation network.

First, we systematically varied the spectrum of the diagonal Λ matrix in our GCN-SSM backbone. For each number of layers $n \in \{5, 10, 20, 30, 40, 50, 60\}$, we set the maximal eigenvalue of Λ to one of $\{1, 0.66, 0.33\}$, retrained the model, and recorded the best test accuracy. As shown in Table 5, moving the spectrum of Λ away from unity leads to a pronounced degradation in accuracy, indicating that keeping the Jacobian eigenvalues near one is crucial for stable and effective information propagation across many layers.

Second, to isolate the contribution of the SSM backbone itself, we performed an analogous spectrum-shaping experiment directly on the weight matrix \mathbf{W} of a vanilla GCN (i.e. without any SSM components). We scaled \mathbf{W} so that its spectral radius lay near the edge of stability (spectral norm ≈ 1), but otherwise left the model architecture and training unchanged. Despite matching the Jacobian stability regime, these “spectrally tuned” vanilla GCNs failed to achieve the accuracy improvements seen with the full GCN-SSM backbone (last column of Table 5). This confirms that merely tuning \mathbf{W} ’s spectrum is insufficient: the structured state-space dynamics provided by the SSM backbone are essential for the observed performance gains.

E.3 On Residual Connections and Gating

Several prior works have employed residual connections in graph neural networks (GNNs) to counteract over-smoothing—for example, see [39]. In fact, these residual-GNN designs can be

Table 5: Node-classification accuracy on Cora when varying the spectrum of the backbone Jacobian (Λ) and comparing to vanilla GCN models whose weight matrix \mathbf{W} is spectrally tuned near the edge of stability.

n_{layers}	$\text{eig}(\Lambda) = 1$	$\text{eig}(\Lambda) = 0.66$	$\text{eig}(\Lambda) = 0.33$	$\text{eig}(W) = 1$ (GCN)
5	81.30	78.10	74.00	71.90
10	78.70	61.90	56.00	33.80
20	78.90	48.00	30.20	31.80
30	80.00	39.70	18.00	22.30
40	77.90	34.60	20.30	25.60
50	77.70	29.10	24.90	23.80
60	77.70	20.50	20.40	29.10

viewed as a special case of our approach, corresponding to the choice $\Lambda = \mathbf{I}$. Under this constraint, the model outperforms a standard, memoryless GCN (see Fig. ??), but only by accumulating node features in an unstructured way. By contrast, our experiments demonstrate that freeing Λ from the identity constraint yields substantially richer state dynamics, which in turn improves both information retention and retrieval. Imposing $\Lambda = \mathbf{I}$ severely restricts this capacity, as shown empirically in Fig. ?. Moreover, the spectral properties of the propagation matrix \mathbf{B} play an important role: by appropriately “damping” incoming signals, one can stabilize the system’s behavior and further mitigate feature collapse.

F Supplementary Related Work and Limitations

Long-range propagation and depth GNNs. Learning long-range dependencies on graphs involves effectively propagating and preserving information across distant nodes. Despite recent advancements, ensuring effective long-range communication between nodes remains an open problem [55]. Several techniques have been proposed to address this issue, including graph rewiring methods, such as [24, 57, 36, 2, 30, 4], which modify the graph topology to enhance connectivity and facilitate information flow. Similarly, Graph Transformers enhance the connectivity to capture both local and global interactions, as demonstrated by [64, 21, 56, 38, 49, 60]. Other approaches incorporate non-local dynamics by using a fractional power of the graph shift operator [43], leverage quantum diffusion kernels [42], regularize the model’s weight space [25, 26], or exploit port-hamiltonian dynamics [33]. Some methods which have increased the depth of GNNs include [39, 40].

Despite the effectiveness of these methods in learning long-range dependencies on graphs, they primarily introduce solutions to mitigate the problem rather than establishing a unified theoretical framework that defines its underlying cause.

Vanishing gradients in sequence modelling and deep learning. One of the primary challenges in training recurrent neural networks lies in the vanishing (and sometimes exploding) gradient problem, which can hinder the model’s ability to learn and retain information over long sequences. In response, researchers have proposed numerous architectures aimed at preserving or enhancing gradients through time. Examples include Unitary RNNs [1], Orthogonal RNNs [34], Linear Recurrent Units [47], and Structured State Space Models [28, 27]. By leveraging properties such as orthogonality, carefully designed parameterizations, or alternative update mechanisms, these models seek to alleviate gradient decay and capture longer-range temporal relationships more effectively.

Dynamical systems and physics inspired neural networks. Since the introduction of Neural ODEs in [17], there have been various methods that employ ideas of dynamical systems within neural networks, including continuous-time methods [50, 45, 9, 10, 3, 44, 11] or state-space approaches [15, 19, 20]. Within graph neural networks, we highlight PDE-GCN [23], GRAND [13], BLEND [12] and Neural Sheaf Diffusion [5]. Other approaches which leverage other type of physics-inspired inductive biases such as topological latent space modelling include [14, 29, 6, 53]

Broader Impact, Limitations and Future Work. We believe our work opens up a number of interesting directions that aim to bridge the gap between graph and sequence modeling. In particular,

288 we hope that this work will encourage researchers to adapt vanishing gradient mitigation methods
289 from the sequence modeling community to GNNs, and conversely explore how graph learning ideas
290 can be brought to recurrent models. In our work, we mostly focused on GCN and GAT type updates,
291 but we believe that our analysis can be extended to understand how different choices of updates and
292 non-linearities affect training dynamics, which we leave for future work.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: All the claims found in the abstract and introduction are supported by the results and theorems discussed in the paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Yes, this is included in the supplementary material.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Yes, all assumptions are stated and proofs can be found in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, datasets are publicly available and the hyperparameters used in the code are in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will provide the code in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, we follow the standard procedure for each individual dataset.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provide the results in the format “mean \pm standard error”.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provided information on the GPU used in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Our submission abides by the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss potential positive and negative societal impacts after the conclusion section.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The release of our data and models does not pose a direct risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We provide citations for all sources of code and/or data used, both in the paper and in the accompanying repository.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new datasets are introduced.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No crowdsourcing or human subjects were involved in the experiments conducted for this paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No crowdsourcing or human subjects were involved in the experiments conducted for this paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLMs are not a central component of the paper in terms of methodological advancements.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

References

- [1] Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International conference on machine learning*, pages 1120–1128. PMLR, 2016.
- [2] Federico Barbero, Ameya Velingker, Amin Saberi, Michael Bronstein, and Francesco Di Giovanni. Locality-aware graph-rewiring in gnns. *arXiv preprint arXiv:2310.01668*, 2023.
- [3] Richard Bergna, Sergio Calvo-Ordonez, Felix L Opolka, Pietro Liò, and Jose Miguel Hernandez-Lobato. Uncertainty modeling in graph neural networks via stochastic differential equations. *arXiv preprint arXiv:2408.16115*, 2024.
- [4] Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org, 2023.
- [5] Cristian Bodnar, Francesco Di Giovanni, Benjamin P. Chamberlain, Pietro Liò, and Michael M. Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs, 2022.
- [6] Haitz Sáez de Ocáriz Borde, Álvaro Arroyo, and Ingmar Posner. Projections of model spaces for latent graph inference. *arXiv preprint arXiv:2303.11754*, 2023.
- [7] Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.
- [8] Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.
- [9] Sergio Calvo-Ordonez, Jiahao Huang, Lipei Zhang, Guang Yang, Carola-Bibiane Schonlieb, and Angelica I Aviles-Rivero. Beyond u: Making diffusion models faster & lighter. *arXiv preprint arXiv:2310.20092*, 2023.
- [10] Sergio Calvo-Ordonez, Matthieu Meunier, Francesco Piatti, and Yuantao Shi. Partially stochastic infinitely deep bayesian neural networks. *arXiv preprint arXiv:2402.03495*, 2024.
- [11] Sergio Calvo-Ordoñez, Jonathan Plenk, Richard Bergna, Alvaro Cartea, Jose Miguel Hernandez-Lobato, Konstantina Palla, and Kamil Ciosek. Observation noise and initialization in wide neural networks. *arXiv preprint arXiv:2502.01556*, 2025.
- [12] Benjamin Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Xiaowen Dong, and Michael Bronstein. Beltrami flow and neural diffusion on graphs. *Advances in Neural Information Processing Systems*, 34:1594–1609, 2021.
- [13] Benjamin Paul Chamberlain, James Rowbottom, Maria Gorinova, Stefan Webb, Emanuele Rossi, and Michael M Bronstein. GRAND: Graph neural diffusion. In *International Conference on Machine Learning (ICML)*, pages 1407–1418. PMLR, 2021.

- [14] Ines Chami, Rex Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks, 2019.
- [15] Peter G Chang, Gerardo Durán-Martín, Alexander Y Shestopaloff, Matt Jones, and Kevin Murphy. Low-rank extended kalman filtering for online learning of neural networks from streaming data. *arXiv preprint arXiv:2305.19535*, 2023.
- [16] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and Deep Graph Convolutional Networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1725–1735. PMLR, 13–18 Jul 2020.
- [17] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- [18] Yuhui Ding, Antonio Orvieto, Bobby He, and Thomas Hofmann. Recurrent distance filtering for graph representation learning. In *Forty-first International Conference on Machine Learning*, 2024.
- [19] Gerardo Duran-Martin, Matias Altamirano, Alexander Y Shestopaloff, Leandro Sánchez-Betancourt, Jeremias Knoblauch, Matt Jones, François-Xavier Briol, and Kevin Murphy. Outlier-robust kalman filtering through generalised bayes. *arXiv preprint arXiv:2405.05646*, 2024.
- [20] Gerardo Duran-Martin, Leandro Sánchez-Betancourt, Alexander Y Shestopaloff, and Kevin Murphy. Bone: a unifying framework for bayesian online learning in non-stationary environments. *arXiv preprint arXiv:2411.10153*, 2024.
- [21] Vijay Prakash Dwivedi and Xavier Bresson. A Generalization of Transformer Networks to Graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.
- [22] Vijay Prakash Dwivedi, Ladislav Rampásek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22326–22340. Curran Associates, Inc., 2022.
- [23] Moshe Eliasof, Eldad Haber, and Eran Treister. Pde-gcn: Novel architectures for graph neural networks motivated by partial differential equations. *Advances in neural information processing systems*, 34:3836–3849, 2021.
- [24] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. *Advances in neural information processing systems*, 32, 2019.
- [25] Alessio Gravina, Davide Bacciu, and Claudio Gallicchio. Anti-Symmetric DGN: a stable architecture for Deep Graph Networks. In *The Eleventh International Conference on Learning Representations*, 2023.
- [26] Alessio Gravina, Moshe Eliasof, Claudio Gallicchio, Davide Bacciu, and Carola-Bibiane Schönlieb. On oversquashing in graph neural networks through the lens of dynamical systems. In *The 39th Annual AAAI Conference on Artificial Intelligence*, 2025.
- [27] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- [28] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2021.
- [29] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *ICLR*, 2019.
- [30] Benjamin Gutteridge, Xiaowen Dong, Michael M Bronstein, and Francesco Di Giovanni. DRew: dynamically rewired message passing with delay. In *International Conference on Machine Learning*, pages 12252–12267. PMLR, 2023.
- [31] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

- [32] K Khalil Hassan et al. Nonlinear systems. *Departement of Electrical and computer Engineering, Michigan State University*, 2002.
- [33] Simon Heilig, Alessio Gravina, Alessandro Trenta, Claudio Gallicchio, and Davide Bacciu. Port-hamiltonian architectural bias for long-range propagation in deep graph networks. 2025.
- [34] Mikael Henaff, Arthur Szlam, and Yann LeCun. Recurrent orthogonal networks and long-memory tasks. In *International Conference on Machine Learning*, pages 2034–2042. PMLR, 2016.
- [35] Matt Jordan and Alexandros G Dimakis. Exactly computing the local lipschitz constant of relu networks. *Advances in Neural Information Processing Systems*, 33:7344–7353, 2020.
- [36] Kedar Karhadkar, Pradeep Kr Banerjee, and Guido Montúfar. Fosr: First-order spectral rewiring for addressing oversquashing in gnns. *arXiv preprint arXiv:2210.11790*, 2022.
- [37] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.
- [38] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.
- [39] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9267–9276, 2019.
- [40] Juncheng Liu, Kenji Kawaguchi, Bryan Hooi, Yiwei Wang, and Xiaokui Xiao. Eignn: Efficient infinite-depth graph neural networks. *Advances in Neural Information Processing Systems*, 34:18762–18773, 2021.
- [41] Vladimir Alexandrovich Marchenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 114(4):507–536, 1967.
- [42] Thomas Markovich. Qdc: Quantum diffusion convolution kernels on graphs, 2023.
- [43] Sohir Maskey, Raffaele Paolino, Aras Bacho, and Gitta Kutyniok. A fractional graph laplacian approach to oversmoothing. *Advances in Neural Information Processing Systems*, 36, 2024.
- [44] Fernando Moreno-Pino, Álvaro Arroyo, Harrison Waldon, Xiaowen Dong, and Álvaro Cartea. Rough transformers: Lightweight and continuous time series modelling through signature patching. *Advances in Neural Information Processing Systems*, 37:106264–106294, 2024.
- [45] Alexander Norcliffe, Cristian Bodnar, Ben Day, Nikola Simidjievski, and Pietro Liò. On second order behaviour in augmented neural odes. *Advances in neural information processing systems*, 33:5911–5921, 2020.
- [46] Kenta Oono and Taiji Suzuki. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations*, 2020.
- [47] Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR, 2023.
- [48] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. 2020.
- [49] Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. *Advances in Neural Information Processing Systems*, 35, 2022.
- [50] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

- 747 [51] T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. A survey on oversmoothing
748 in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.
- 749 [52] T Konstantin Rusch, Ben Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael
750 Bronstein. Graph-coupled oscillator networks. In *International Conference on Machine*
751 *Learning*, pages 18888–18909. PMLR, 2022.
- 752 [53] Haitz Sáez de Ocáriz Borde, Alvaro Arroyo, Ismael Morales, Ingmar Posner, and Xiaowen Dong.
753 Neural latent geometry search: product manifold inference via gromov-hausdorff-informed
754 bayesian optimization. *Advances in Neural Information Processing Systems*, 36, 2024.
- 755 [54] Michael Scholkemper, Xinyi Wu, Ali Jadbabaie, and Michael T Schaub. Residual con-
756 nections and normalization can provably prevent oversmoothing in gnns. *arXiv preprint*
757 *arXiv:2406.02997*, 2024.
- 758 [55] Dai Shi, Andi Han, Lequan Lin, Yi Guo, and Junbin Gao. Exposition on over-squashing problem
759 on gnns: Current methods, benchmarks and challenges, 2023.
- 760 [56] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked
761 Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In
762 Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Arti-
763 ficial Intelligence, IJCAI-21*, pages 1548–1554. International Joint Conferences on Artificial
764 Intelligence Organization, 8 2021. Main Track.
- 765 [57] Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and
766 Michael M Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature.
767 *arXiv preprint arXiv:2111.14522*, 2021.
- 768 [58] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua
769 Bengio. Graph attention networks. *ArXiv*, 2018.
- 770 [59] Yifei Wang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Dissecting the Diffusion Process
771 in Linear Graph Convolutional Networks. In *Advances in Neural Information Processing*
772 *Systems*, volume 34, pages 5758–5769. Curran Associates, Inc., 2021.
- 773 [60] Qitian Wu, Chenxiao Yang, Wentao Zhao, Yixuan He, David Wipf, and Junchi Yan. DIFFormer:
774 Scalable (Graph) Transformers Induced by Energy Constrained Diffusion. In *The Eleventh*
775 *International Conference on Learning Representations*, 2023.
- 776 [61] Xinyi Wu, Amir Ajorlou, Zihui Wu, and Ali Jadbabaie. Demystifying oversmoothing in
777 attention-based graph neural networks. *Advances in Neural Information Processing Systems*,
778 36:35084–35106, 2023.
- 779 [62] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural
780 networks? *arXiv preprint arXiv:1810.00826*, 2018.
- 781 [63] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning
782 with graph embeddings. In *Proceedings of The 33rd International Conference on Machine*
783 *Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 40–48, New York,
784 New York, USA, 20–22 Jun 2016. PMLR.
- 785 [64] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen,
786 and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in*
787 *Neural Information Processing Systems*, 34:28877–28888, 2021.