Unleashing the Potential of Fractional Calculus in Graph Neural Networks

Qiyu Kang[*]
Nanyang Technological University

Kai Zhao[*]
Nanyang Technological University

Qinxu Ding
Singapore University of Social Sciences

Feng Ji
Nanyang Technological University

Xuhao Li
Anhui University

Wenfei Liang
Nanyang Technological University

Yang Song
C3 AI, Singapore

Wee Peng Tay
Nanyang Technological University

Abstract

We introduce the FRactional-Order graph Neural Dynamical network (FROND), a learning framework that augments traditional graph neural ordinary differential equation (ODE) models by integrating the time-fractional Caputo derivative. Thanks to its non-local characteristic, fractional calculus enables our framework to encapsulate long-term memories during the feature-updating process, diverging from the Markovian updates inherent in conventional graph neural ODE models. This capability could substantially enhance graph representation learning by introducing more nuanced feature updating dynamics. Analytically, we exhibit that the over-smoothing issue is mitigated when feature updating is regulated by a fractional diffusion process. Additionally, our framework affords a fresh dynamical system perspective to comprehend various skip or dense connections situated between GNN layers in existing literature.

1 Introduction

Graph Neural Networks (GNNs) [1–9] have excelled in diverse domains, e.g., chemistry [1], finance [2], and social media [3–5]. The neural message passing scheme, where features are propagated along edges and optimized for a specific downstream task, is crucial for the success of GNNs. Over the past few years, numerous types of GNNs have been proposed, including Graph Convolutional Networks (GCN) [3], Graph Attention Networks (GAT) [10], and GraphSAGE [11]. Recent works, such as [12–20], have incorporated various continuous dynamics to propagate information over the graph nodes, inspiring a new class of GNNs based on ordinary differential equations (ODEs)[2] [21–23].

Within these graph neural ODE models, the differential operator $\mathrm{d}^\beta / \mathrm{d}t^\beta$ is conventionally constrained to *integer values* of β, primarily 1 or 2. However, over recent decades, the wider scientific community has delved into the domains of fractional-order differential operators, where β can be any *real number*. These expansions have proven pivotal in various applications characterized by

[*]First two authors contributed equally to this work. Contact: kang0080@e.ntu.edu.sg, kai.zhao@ntu.edu.sg.

[2]Models like GRAND [12] primarily utilize ODEs on graphs, albeit inspired by partial differential equations. We consistently refer to such models as graph neural ODE models.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

nonlocal and memory-dependent behaviors, with prime examples including viscoelastic materials [24], anomalous transport mechanisms [25], and fractal media [26]. The distinction lies in the fact that the conventional integer-order derivative measures the function's *instantaneous change rate*, concentrating on the proximate vicinity of the point. *In contrast, the fractional-order derivative [27] is influenced by the entire historical trajectory of the function,* which substantially diverges from the localized impact found in integer-order derivatives.

In this study, we introduce the FRactional-Order graph Neural Dynamical network (FROND) framework, a new approach that broadens the capabilities of traditional graph neural ODE models by incorporating fractional calculus. It naturally generalizes the integer-order derivative $\mathrm{d}^\beta / \mathrm{d}t^\beta$ in graph neural ODE models to accommodate any positive real number β. This adaptation enables FROND to incorporate memory-dependent dynamics for information propagation and feature updating, potentially enhancing graph representations and performance. Notably, this technique ensures at least equivalent performance to integer-order models, as it reverts to conventional graph ODE models without memory when β is an integer.

Main contributions. Our main contributions are summarized as follows:

- We propose a novel, generalized graph framework that incorporates time-fractional derivatives. This framework generalizes prior graph neural ODE models [12–18], subsuming them as special instances. This approach also lays the groundwork for a diverse new class of GNNs that can accommodate a broad array of learnable feature-updating processes with memory.
- We have implemented and open-sourced a suite of neural fractional differential equation (FDE) solvers. We anticipate these solvers to be of significant value to the GNN and physic community. Certain time-discretization strategies employed in these solvers can be viewed as layers in a deep neural network with dense/skip connections [28]. This provides a fascinating analogy to the residual characteristic of Euler solvers in conventional neural ODEs [21] and give a new perspective to understand various skip or dense connections used between layers in prior literature [29–32].
- We highlight FROND's compatibility and its seamless integration potential to enhance existing graph ODE models across varied datasets. This work primarily showcases FROND's performance with feature-updating dynamics derived from the *fractional heat diffusion process*. We demonstrate analytically that over-smoothing can be mitigated in this setting. The fractional differential extension of other graph neural ODE models [13–18] is left for future exploration.

2 Preliminaries

2.1 The Caputo Time-Fractional Derivative

The traditional first-order derivative of a scalar function $f(t)$ represents the local rate of change of the function at a point, defined as: $\frac{\mathrm{d}f(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{f(t+\Delta t) - f(t)}{\Delta t}$. Its Laplace transform is:

$$\mathcal{L}\left\{\frac{\mathrm{d}f(t)}{\mathrm{d}t}\right\} = sF(s) - f(0). \tag{1}$$

The Caputo fractional derivative of order $\beta \in (0, 1]$ for a function $f(t)$ is defined as follows [27]:

$$D_t^\beta f(t) = \frac{1}{\Gamma(1-\beta)} \int_0^t (t-\tau)^{-\beta} f'(\tau)\, \mathrm{d}\tau, \tag{2}$$

where $\Gamma(\cdot)$ denotes the gamma function, and $f'(\tau)$ is the first-order derivative of f. The Caputo fractional derivative inherently integrates the entire history of the system through the integral term, emphasizing its non-local nature. The Laplace transform of the Caputo fractional derivative is:

$$\mathcal{L}\left\{D_t^\beta f(t)\right\} = s^\beta F(s) - s^{\beta-1} f(0). \tag{3}$$

Comparing the Laplace transforms in (1) and (3) of the traditional first-order derivative and the Caputo fractional derivative respectively, it becomes clear that the latter generalizes the former. When $\beta = 1$, $D_t^1 f = f'$ is uniquely determined through the inverse Laplace transform [33].

2.2 Diffusion Equation and Its Application to GNNs

We denote an undirected graph as $G = (\mathbf{X}, \mathbf{W})$, where $\mathbf{X} = \left(\left[\mathbf{x}^{(1)}\right]^\intercal, \cdots, \left[\mathbf{x}^{(N)}\right]^\intercal\right)^\intercal \in \mathbb{R}^{N \times d}$ consists of rows $\mathbf{x}^{(i)} \in \mathbb{R}^d$ as node feature vectors and i is the node index. The $N \times N$ matrix

$\mathbf{W} := (W_{ij})$ has elements W_{ij} indicating the edge weight between the i-th and j-th nodes with $W_{ij} = W_{ji}$. Inspired by the standard heat diffusion equation, GRAND [12] utilizes the following nonlinear autonomous dynamical system for node feature updating in GNNs:

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = (\mathbf{A}(\mathbf{X}(t)) - \mathbf{I})\mathbf{X}(t). \tag{4}$$

where $\mathbf{A}(\mathbf{X}(t))$ is a learnable, time-variant attention matrix, calculated using the features $\mathbf{X}(t)$, and $\mathbf{I}$ denotes the identity matrix. The feature update outlined in (4) is referred to as the GRAND-nl version (due to the nonlinearity in $\mathbf{A}(\mathbf{X}(t))$). We define $d_i = \sum_{j=1}^{n} W_{ij}$ and let $\mathbf{D}$ be a diagonal matrix with $D_{ii} = d_i$. The normalized Laplacian matrix is then represented as $\mathbf{L} = \mathbf{I} - \mathbf{W}\mathbf{D}^{-1}$ or $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$, following column or row normalization, respectively. In a simplified context, we employ the following linear dynamical system:

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = -\mathbf{L}\mathbf{X}(t). \tag{5}$$

The feature updating in (5) is the GRAND-l version, which is a time-invariant linear FDE. For implementations of (5), one may direct set $\mathbf{W}\mathbf{D}^{-1}$(or $\mathbf{D}^{-1}\mathbf{W}$) $= \mathbf{A}(\mathbf{X}(0))$ as column- or row-stochastic attention matrix, rather than using a plain weight. Notably, in this time-invariant setting, the attention weight matrix, reliant on the initial node features, stays unchanged throughout the feature evolution period.

3 Fractional-Order Graph Neural Dynamical Network

In this section, we introduce the FROND framework, a novel approach that augments traditional graph neural ODE models by incorporating fractional calculus. We feature one specific model, wherein the feature-updating dynamics are derived from the fractional heat diffusion process. Analytically, we show that over-smoothing can be effectively mitigated in this context. Finally, we outline the numerical techniques to solve FDEs pertinent to FROND.

3.1 Framework

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathbf{W})$ composed of $|\mathcal{V}| = N$ nodes and $\mathbf{W}$ the set of edge weights as defined in Section 2.2. Analogous to the implementation in traditional graph neural ODE models, a preliminary learnable encoder function $\varphi : \mathcal{V} \to \mathbb{R}^d$ that maps each node to a feature vector can be applied. Stacking all the feature vectors together, we obtain $\mathbf{X} \in \mathbb{R}^{N\times d}$. Employing the Caputo time fractional derivative outlined in Section 2.1, the information propagation and feature updating dynamics in FROND are characterized by the following graph neural FDE:

$$D_t^\beta \mathbf{X}(t) = \mathcal{F}(\mathbf{W}, \mathbf{X}(t)), \quad \beta > 0, \tag{6}$$

where β denotes the fractional order of the derivative, and $\mathcal{F}$ is a dynamic operator on the graph like the graph neural ODE models [12–18]. The initial condition for (6) is set as $\mathbf{X}^{(\lceil\beta\rceil-1)}(0) = \ldots = \mathbf{X}(0) = \mathbf{X}$ consisting of the preliminary node features, where $\lceil\beta\rceil$ denotes the smallest integer greater than or equal to β, akin to the initial conditions seen in ODEs. In this work, we mainly consider $\beta \in (0, 1]$ and the initial condition is $\mathbf{X}(0) = \mathbf{X}$. In alignment with the graph neural ODE models [12–18], we set an integration time parameter T to yield $\mathbf{X}(T)$. The final node embedding for subsequent tasks may be decoded as $\psi(\mathbf{X}(T))$ with ψ being a learnable decoder.

Specifying the operator $\mathcal{F}$ to the dynamics employed from the literature [12–18], fractional extensions of graph ODE models can be derived. Due to space constraints, this paper primarily focuses on the $\mathcal{F}$ dynamics as described in (4) and (5).

3.1.1 F-GRAND: Fractional Diffusion GNN

F-GRAND: Mirroring the GRAND model, the fractional GRAND (F-GRAND) is divided into two versions. The F-GRAND-nl employs a time-variant FDE as follows:

$$D_t^\beta \mathbf{X}(t) = (\mathbf{A}(\mathbf{X}(t)) - \mathbf{I})\mathbf{X}(t), \quad 0 < \beta \leq 1. \tag{7}$$

It is computed using $\mathbf{X}(t)$ and the attention mechanism $\mathbf{A}(\cdot)$ derived from the Transformer model [34]. In parallel, the F-GRAND-l version stands as the fractional extension of (5):

$$D_t^\beta \mathbf{X}(t) = -\mathbf{L}\mathbf{X}(t), \quad 0 < \beta \leq 1. \tag{8}$$

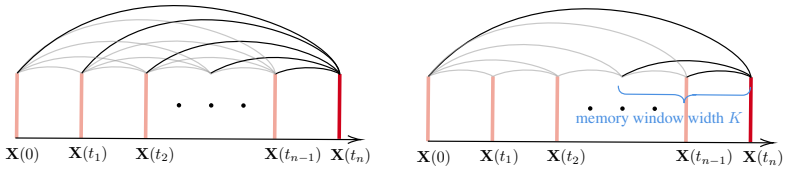

Figure 1: Diagrams of fractional Adams–Bashforth–Moulton method with full (left) and short (right) memory.

3.1.2 Over-smoothing Mitigation of F-GRAND

The efficacy of GNNs in node classification diminishes exponentially owing to intrinsic constraints in their architecture and capabilities. The seminal research [35][Corollary 3. and Remark 1] has highlighted that, when considering a GNN as a layered dynamical system, over-smoothing is a broad expression of the *exponential convergence* to stationary states that only retain information about graph connected components and node degrees. The memoryless graph neural ODE model, GRAND-l, is demonstrated to approach asymptotic stationary states as shown in [13], with a known *exponentially rapid convergence rate* [36]. In contrast, the fractional memory-dependent dynamic model, F-GRAND-l, as described in (8), converges to stationary states at a *slow algebraic rate*, thereby helping to mitigate over-smoothing. As $\beta \to 0$, the convergence is expected to be *arbitrarily slow*. In real-world scenarios where we operate within a finite horizon, this slower rate of convergence may be sufficient to alleviate over-smoothing, particularly when it is imperative for a deep model to extract distinctive features instead of achieving exponentially fast convergence to stationary states.

Theorem 1. *Assuming the graph is strongly connected and aperiodic, the solution* $\mathbf{X}(t)$ *to* (8) *converges to a stationary state at the rate of* $\Theta(t^{-\beta})$*, provided that the initial condition* $\mathbf{X}(0)$ *differs from the stationary state.*

3.2 Solving FROND

The studies by [21, 37, 38] introduce numerical solvers specifically designed for neural ODE models when β is an integer in the FROND framework. Our research, in contrast, engages with FDEs, entities inherently more intricate than ODEs. To address the scenario *where* β *is non-integer*, we introduce the *fractional explicit Adams–Bashforth–Moulton method*, incorporating two variants employed in this study: the **basic predictor**, and the **short memory principle**. These methods exemplify how time persistently acts as a continuous analog to the layer index and elucidate how resultant memory dependence manifests as nontrivial dense or skip connections between layers (see Fig. 1), stemming from the non-local properties of fractional derivatives.

Basic predictor. Referencing [39], we first employ a preliminary numerical solver called "predictor" through time discretisation $t_j = jh$, where the discretisation parameter h is a small positive value:

$$\mathbf{X}^{\mathrm{P}}(t_n) = \sum_{j=0}^{\lceil\beta\rceil-1} \frac{t_n^j}{j!}\mathbf{X}^{(k)}(0) + \frac{1}{\Gamma(\beta)}\sum_{j=0}^{n-1} \mu_{j,n}\mathcal{F}(\mathbf{W}, \mathbf{X}(t_j)), \tag{9}$$

with coefficients $\mu_{j,n}$ outlined in [39][eq.17]. For $\beta = 1$, this method reduces to the Euler solver [21], where $\mu_{j,n} \equiv h$, resulting in $\mathbf{X}^{\mathrm{P}}(t_n) = \mathbf{X}^{\mathrm{P}}(t_{n-1}) + h\mathcal{F}(\mathbf{W}, \mathbf{X}(t_{n-1}))$.

Short memory principle. For large T, the non-locality of fractional derivatives complicates computations. To counter this, [40, 41] recommend applying the short memory principle, modifying the summation in (9) to $\sum_{j=n-K}^{n-1}$, representing a shifting memory window of fixed width K. See Fig. 1.

3.3 Connection to Existing Architectures

The FROND framework provides a generalized dynamical system perspective on existing GNN architectures. As the time fractional order β approaches integer values, D_t^β becomes the local integer-order derivative, aligning FROND seamlessly with conventional graph ODE frameworks [12–18]. The various skip/dense connections used between layers in existing literature [29–32] can be viewed as the discretization of FROND. By incorporating fractional-order dynamics and memory effects, FROND not only provides fresh insights into GNN architectures but also promotes the advancement of graph representation learning.

4 Experiments

In this paper, we mainly highlight F-GRAND's superior results and validate the slow algebraic convergence for deeper GNNs with non-integer $\beta < 1$, as per Theorem 1. We leave fractional differential extension of other graph ODE models like [13–18] for future work.

4.1 Node Classification of F-GRAND

Datasets and splitting. We utilize datasets with varied topologies, including citation networks (Cora [42], Citeseer [43], Pubmed [44]), tree-structured datasets (Disease and Airport [45]), coauthor and co-purchasing graphs (CoauthorCS [46], Computer and Photo [47]), and the ogbn-arxiv dataset [48]. We follow the same data splitting and pre-processing in [45] for Disease and Airport datasets. The other experiment settings are the same as in GRAND [12].

Performance. As summarized in Table 1 and aligned with expectations, F-GRAND consistently outperforms GRAND, its special case with $\beta = 1$, across all datasets, emphasizing the benefits of integrating memorized dynamics. The advantage is especially pronounced on tree-structured datasets like Airports and Disease, where it significantly surpasses baselines. For example, F-GRAND-l exceeds GRAND and GIL by roughly 7% on the Airport dataset. Intriguingly, our experiments suggest a preference for a smaller β, which implies enhanced dynamic memory, in these fractal-structured datasets. This observation aligns with prior research [49, 26, 50, 51], which has established that dynamical processes exhibiting self-similarity in fractal media are more precisely characterized by fractional differential equations.

Table 1: Node classification results(%) for random train-val-test splits. The best and the second-best result are highlighted in **red** and **blue**, respectively.

Method	Cora	Citeseer	Pubmed	CoauthorCS	Computer	Photo	CoauthorPhy	ogbn-arxiv	Airport	Disease
GCN	81.5±1.3	71.9±1.9	77.8±2.9	91.1±0.5	82.6±2.4	91.2±1.2	92.8±1.0	72.2±0.3	81.6±0.6	69.8±0.5
GAT	81.8±1.3	71.4±1.9	78.7±2.3	90.5±0.6	78.0±19.0	85.7±20.3	92.5±0.90	**73.7±0.1**	81.6±0.4	70.4±0.5
HGCN	78.7±1.0	65.8±2.0	76.4±0.8	90.6±0.3	80.6±1.8	88.2±1.4	90.8±1.5	59.6±0.4	85.4±0.7	89.9±1.1
GIL	82.1±1.1	71.1±1.2	77.8±0.6	89.4±1.5	–	89.6±1.3	–	–	91.5±1.7	**90.8±0.5**
GRAND-l	**83.6±1.0**	73.4±0.5	78.8±1.7	92.9±0.4	83.7±1.2	92.3±0.9	93.5±0.9	71.9±0.2	80.5±9.6	74.5±3.4
GRAND-nl	82.3±1.6	70.9±1.0	77.5±1.8	92.4±0.3	82.4±2.1	92.4±0.8	91.4±1.3	71.2±0.2	90.9±1.6	81.0±6.7
F-GRAND-l	**84.8±1.1**	**74.0±1.5**	**79.4±1.5**	**93.0±0.3**	**84.4±1.5**	**92.8±0.6**	**94.5±0.4**	**72.6±0.1**	**98.1±0.2**	**92.4±3.9**
β for F-GRAND-l	0.9	0.9	0.9	0.7	0.98	0.9	0.6	0.7	0.5	0.6
F-GRAND-nl	83.2±1.1	**74.7±1.9**	**79.2±0.7**	**92.9±0.4**	**84.1±0.9**	**93.1±0.9**	**93.9±0.5**	71.4±0.3	**96.1±0.7**	85.5±2.5
β for F-GRAND-nl	0.9	0.9	0.4	0.6	0.85	0.8	0.4	0.7	0.1	0.7

4.2 Over-smoothing of F-GRAND

The capability of F-GRAND to mitigate over-smoothing and sustain strong performance across various depths is demonstrated in Fig. 2 using the solver (9), adhering to the fixed data split in [45]. F-GRAND-l achieves its best performance with 64 layers on the Cora dataset. Consistently across various datasets, F-GRAND-l's performance remains stable up to 128 layers, supporting the slow algebraic convergence in Theorem 1. In contrast, GRAND demonstrates a quicker decline in performance, especially notable on the Airport dataset.

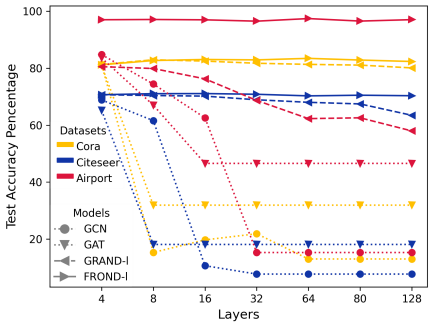

Figure 2: Over-smoothing mitigation.

5 Conclusions

We introduced FROND, a novel graph learning framework that incorporates time-fractional Caputo derivatives to infuse memory into graph feature updating dynamics. This novel approach holds potential for outperforming existing graph neural ODE models, as exemplified by the comparison between F-GRAND and GRAND in this work. The resulting framework paves the way for a new class of GNNs capable of addressing key challenges in the field, such as over-smoothing. Our results signify a promising step towards more effective graph representation learning by capitalizing on the power of fractional calculus.

6 Acknowledgments and Disclosure of Funding

This research is supported by the Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE-T2EP20220-0002, and the National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research and Development Programme. To improve the readability, parts of this paper have been grammatically revised using ChatGPT [52].

References

[1] X. Yue, Z. Wang, J. Huang, S. Parthasarathy, S. Moosavinasab, Y. Huang, S. M. Lin, W. Zhang, P. Zhang, and H. Sun, "Graph embedding on biomedical networks: methods, applications and evaluations," *Bioinformatics*, vol. 36, no. 4, pp. 1241–1251, 2019.

[2] H. Ashoor, X. Chen, W. Rosikiewicz, J. Wang, A. Cheng, P. Wang, Y. Ruan, and S. Li, "Graph embedding and unsupervised learning predict genomic sub-compartments from hic chromatin interaction data," *Nat. Commun.*, vol. 11, 2020.

[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[4] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 249–270, Jan 2022.

[5] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.

[6] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.

[7] F. Ji, S. H. Lee, H. Meng, K. Zhao, J. Yang, and W. P. Tay, "Leveraging label non-uniformity for node classification in graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 202, Jul. 2023, pp. 14 869–14 885.

[8] S. H. Lee, F. Ji, and W. P. Tay, "SGAT: Simplicial graph attention network," in *Proc. Inter. Joint Conf. Artificial Intell.*, Jul. 2022.

[9] R. She, Q. Kang, S. Wang, W. P. Tay, Y. L. Guan, D. N. Navarro, and A. Hartmannsgruber, "Image patch-matching with graph-based learning in street scenes," *IEEE Trans. Image Process.*, vol. 32, pp. 3465–3480, 2023.

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.

[11] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances Neural Inf. Process. Syst.*, 2017.

[12] B. P. Chamberlain, J. Rowbottom, M. Goronova, S. Webb, E. Rossi, and M. M. Bronstein, "Grand: Graph neural diffusion," in *Proc. Int. Conf. Mach. Learn.*, 2021.

[13] M. Thorpe, H. Xia, T. Nguyen, T. Strohmer, A. Bertozzi, S. Osher, and B. Wang, "Grand++: Graph neural diffusion with a source term," in *Proc. Int. Conf. Learn. Representations*, 2022.

[14] T. K. Rusch, B. Chamberlain, J. Rowbottom, S. Mishra, and M. Bronstein, "Graph-coupled oscillator networks," in *Proc. Int. Conf. Mach. Learn.*, 2022.

[15] Y. Song, Q. Kang, S. Wang, K. Zhao, and W. P. Tay, "On the robustness of graph neural diffusion to topology perturbations," in *Advances Neural Inf. Process. Syst.*, 2022.

[16] J. Choi, S. Hong, N. Park, and S.-B. Cho, "Gread: Graph neural reaction-diffusion networks," in *Proc. Int. Conf. Mach. Learn.*, 2023.

[17] K. Zhao, Q. Kang, Y. Song, R. She, S. Wang, and W. P. Tay, "Graph neural convection-diffusion with heterophily," in *Proc. Inter. Joint Conf. Artificial Intell.*, Macao, China, 2023.

[18] ——, "Adversarial robustness in graph neural networks: A Hamiltonian energy conservation approach," in *Advances in Neural Information Processing Systems*, New Orleans, USA, Dec. 2023.

[19] R. She, Q. Kang, S. Wang, Y.-R. Yáng, K. Zhao, Y. Song, and W. P. Tay, "Robustmat: Neural diffusion for street landmark patch matching under challenging environments," *IEEE Trans. Image Process.*, 2023.

[20] S. Wang, Q. Kang, R. She, W. P. Tay, A. Hartmannsgruber, and D. N. Navarro, "RobustLoc: Robust camera pose regression in challenging driving environments," in *Proc. AAAI Conference on Artificial Intelligence*, Feb. 2023.

[21] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," in *Advances Neural Inf. Process. Syst.*, 2018.

[22] Q. Kang, Y. Song, Q. Ding, and W. P. Tay, "Stable neural ODE with Lyapunov-stable equilibrium points for defending against adversarial attacks," in *Advances Neural Inf. Process. Syst.*, 2021.

[23] I. D. J. Rodriguez, A. Ames, and Y. Yue, "Lyanet: A lyapunov framework for training neural odes," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 18 687–18 703.

[24] R. L. Bagley and P. Torvik, "A theoretical basis for the application of fractional calculus to viscoelasticity," *J. Rheology*, vol. 27, no. 3, pp. 201–210, 1983.

[25] J. F. Gómez-Aguilar, M. Miranda-Hernández, M. López-López, V. M. Alvarado-Martínez, and D. Baleanu, "Modeling and simulation of the fractional space-time diffusion equation," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 30, no. 1-3, pp. 115–127, 2016.

[26] B. B. Mandelbrot and B. B. Mandelbrot, *The fractal geometry of nature*. WH freeman New York, 1982, vol. 1.

[27] V. E. Tarasov, *Fractional dynamics: applications of fractional calculus to dynamics of particles, fields and media*. Springer Science & Business Media, 2011.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[29] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5453–5462.

[30] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1725–1735.

[31] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 9267–9276.

[32] G. Li, C. Xiong, A. Thabet, and B. Ghanem, "Deepergcn: All you need to train deeper gcns," *arXiv preprint arXiv:2006.07739*, 2020.

[33] A. M. Cohen, *Inversion Formulae and Practical Results*. Boston, MA: Springer US, 2007, pp. 23–44.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[35] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," in *Proc. Int. Conf. Learn. Representations*, 2020.

[36] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.

[37] A. Quaglino, M. Gallieri, J. Masci, and J. Koutník, "Snode: Spectral discretization of neural odes for system identification," in *Proc. Int. Conf. Learn. Representations*, 2019.

[38] H. Yan, J. Du, V. Y. Tan, and J. Feng, "On robustness of neural ordinary differential equations," in *Advances Neural Inf. Process. Syst.*, 2018, pp. 1–13.

[39] K. Diethelm, N. J. Ford, and A. D. Freed, "Detailed error analysis for a fractional adams method," *Numer. Algorithms*, vol. 36, pp. 31–52, 2004.

[40] W. Deng, "Short memory principle and a predictor–corrector approach for fractional differential equations," *J. Comput. Appl. Math.*, vol. 206, no. 1, pp. 174–188, 2007.

[41] I. Podlubny, *Fractional Differential Equations*. Academic Press, 1999.

[42] A. McCallum, K. Nigam, J. D. M. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Inf. Retrieval*, vol. 3, pp. 127–163, 2004.

[43] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, p. 93, Sep. 2008.

[44] G. M. Namata, B. London, L. Getoor, and B. Huang, "Query-driven active surveying for collective classification," in *Workshop Mining Learn. Graphs*, 2012.

[45] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *Advances Neural Inf. Process. Syst.*, 2019.

[46] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," *Relational Representation Learning Workshop, Advances Neural Inf. Process. Syst.,*, 2018.

[47] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2015, p. 43–52.

[48] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *arXiv:2005.00687*, 2020.

[49] R. Nigmatullin, "The realization of the generalized transfer equation in a medium with fractal geometry," *Physica status solidi (b)*, vol. 133, no. 1, pp. 425–430, 1986.

[50] A. Zhokh, A. Trypolskyi, and P. Strizhak, "Relationship between the anomalous diffusion and the fractal dimension of the environment," *Chem. Phys.*, vol. 503, pp. 71–76, 2018.

[51] S. Butera and M. Di Paola, "A physically based connection between fractional calculus and fractal geometry," *Ann. Phys.*, vol. 350, pp. 146–158, 2014.

[52] OpenAI, "Chatgpt-4," 2022, available at: https://www.openai.com (Accessed: 26 September 2023).

[53] R. A. Horn and C. R. Johnson, *Matrix analysis*. New York: Cambridge university press, 2012.

[54] K. Diethelm, *The analysis of fractional differential equations: an application-oriented exposition using differential operators of Caputo type*, 2010, vol. 2004.

[55] W. Feller, *An introduction to probability theory and its applications, Volume 2*. John Wiley & Sons, 1991, vol. 81.

A Proof of Theorem 1

It is evident that for the matrix $\mathbf{W}\mathbf{D}^{-1}$ (or $\mathbf{D}^{-1}\mathbf{W}$), given that it is column (or row) stochastic and the graph is strongly connected and aperiodic, the Perron-Frobenius theorem [53][Lemma 8.4.3., Theorem 8.4.4] confirms that the value 1 is the unique eigenvalue of this matrix that equals its spectral radius, which is also 1. Consequently, it follows that the matrix $\mathbf{L} = \mathbf{I} - \mathbf{W}\mathbf{D}^{-1}$ (or $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$) has an eigenvalue of 0, with all other eigenvalues possessing positive real parts. Considering the Jordan canonical form of $\mathbf{L}$, denoted as $\mathbf{L} = \mathbf{S}\mathbf{J}\mathbf{S}^{-1}$, it is observed that $\mathbf{J}$ contains a block that consists solely of a single 0, while the other blocks are characterized by eigenvalues λ_k possessing positive real parts. WLOG, we assume the feature dimension is one and we rewrite (8) as

$$D_t^{\beta}\mathbf{Y}(t) = -\mathbf{J}\mathbf{Y}(t) \tag{10}$$

where $\mathbf{S}^{-1}\mathbf{X}(t) = \mathbf{Y}(t) \in \mathbb{R}^N$ representing a transformation of the feature space, and the transformed initial condition is defined as $\mathbf{S}^{-1}\mathbf{X}(0) = \mathbf{Y}(0)$.

If the matrix $\mathbf{L}$ is diagonalizable, then its Jordan canonical form $\mathbf{J}$ becomes a diagonal matrix, with the diagonal elements representing the eigenvalues of $\mathbf{L}$. In this scenario, the differential equation can be decoupled into a set of independent equations, each described by

$$D_t^{\beta}\mathbf{Y}_k(t) = -\lambda_k\mathbf{Y}_k(t). \tag{11}$$

Here, $\mathbf{Y}_k$ signifies the k-th component of the vector $\mathbf{Y}$. According to [54][Theorem 4.3.], the solution to each differential equation in the given context is represented as:

$$\mathbf{Y}_k(t) = \mathbf{Y}_k(0)E_{\beta}(-\lambda_k t^{\beta}) \tag{12}$$

where is $E_{\beta}(\cdot)$ is the Mittag-Leffler function define as $E_{\beta}(z) = \sum_{j=0}^{\infty}\frac{z^j}{\Gamma(\beta j+1)}$ and $\Gamma(\cdot)$ is the gamma function. This formulation leads to two important observations:

1. For the index j such that the eigenvalue $\lambda_j = 0$, the solution simplifies to $\mathbf{Y}_j(t) = \mathbf{Y}_j(0)$. This corresponds to a stationary vector in the original space when transformed back to $\mathbf{X}(t)$.
2. According to [41][Theorem 1.4.], for indices $k \neq j$, since λ_k has positive real part, the convergence to zero is characterized by the following order:
$$\mathbf{Y}_k(t) = \Theta(t^{-\beta}).$$

Asymptotically, this indicates that all components $\mathbf{Y}_k(t)$, except $\mathbf{Y}_j(t)$, will converge to zero at an algebraic rate. In terms of $\mathbf{X}(t)$, this translates into a convergence towards a stationary vector in the eigenspace corresponding to the eigenvalue 0, while components associated with other eigenspaces diminish at an algebraic rate.

If the matrix $\mathbf{J}$ is not diagonal, the entries of $\mathbf{Y}(t)$ corresponding to distinct Jordan blocks in $\mathbf{J}$ remain uncoupled. Therefore, it suffices to consider a single Jordan block corresponding to a non-zero eigenvalue λ_k. In this case, employing the Laplace transform technique becomes useful for demonstrating that the algebraic rate of convergence remains valid. We assume the Jordan block $\mathbf{J}(\lambda_k)$, associated with λ_k, is of size m. It follows that for this Jordan block we have

$$\begin{aligned}
D_t^{\beta}\mathbf{Y}_1(t) &= -\lambda_k\mathbf{Y}_1(t) - \mathbf{Y}_2(t),\\
&\vdots\ \ \vdots\\
D_t^{\beta}\mathbf{Y}_{m-1}(t) &= -\lambda_k\mathbf{Y}_{m-1}(t) - \mathbf{Y}_m(t),\\
D_t^{\beta}\mathbf{Y}_m(t) &= -\lambda_k\mathbf{Y}_m(t),
\end{aligned}$$

which can be solved from the bottom up. Beginning with the last equation, we obtain:

$$\mathbf{Y}_m(t) = \mathbf{Y}_m(0)E_{\beta}(-\lambda_k t^{\beta}) = \Theta(t^{-\beta}).$$

Further, the differential equation for $\mathbf{Y}_{m-1}(t)$ is given by:

$$D_t^{\beta}\mathbf{Y}_{m-1}(t) = -\lambda_k\mathbf{Y}_{m-1}(t) - \mathbf{Y}_m(0)E_{\beta}(-\lambda_k t^{\beta})$$

Applying the Laplace transform and referring to (3), we obtain:

$$\mathcal{L}\left\{D_t^{\beta}\mathbf{Y}_{m-1}(t)\right\} = s^{\beta}Y_{m-1}(s) - s^{\beta-1}\mathbf{Y}_{m-1}(0)$$

where $Y_{m-1}(s)$ is the Laplace transform of $\mathbf{Y}_{m-1}(t)$. For the right-hand side of the differential equation, we have $\mathcal{L}\{\lambda_k\mathbf{Y}_{m-1}(t)\} = \lambda_k Y_{m-1}(s)$. Additionally, the Laplace transform of the Mittag-Leffler function $E_{\beta}\left(-\lambda_k t^{\beta}\right)$ known to be $\frac{s^{\beta-1}}{s^{\beta}+\lambda_k}$ [41][eq 1.80]. Consequently, the equation in the

Laplace domain is represented as:

$$s^{\beta}Y_{m-1}(s) - s^{\beta-1}\mathbf{Y}_{m-1}(0) = -\lambda_k Y_{m-1}(s) - \mathbf{Y}_m(0)\frac{s^{\beta-1}}{s^{\beta}+\lambda_k}$$

Rearranging this equation to isolate $Y_{m-1}(s)$ yields:

$$Y_{m-1}(s) = \frac{s^{\beta-1}\mathbf{Y}_{m-1}(0) - \mathbf{Y}_m(0)\frac{s^{\beta-1}}{s^{\beta}+\lambda_k}}{s^{\beta}+\lambda_k}$$

As $s \to 0$, it follows that $Y_{m-1}(s) = \Theta(s^{\beta-1})$. Applying the same process recursively, we find that $Y_i(s) = \Theta(s^{\beta-1})$ for all $i = 1, \ldots, m$. Invoking the Hardy–Littlewood Tauberian theorem [55], we can conclude that for all indices $i = 1, \ldots, m$, the following relationship holds:

$$\mathbf{Y}_i(t) = \Theta(t^{-\beta}). \tag{13}$$

Consequently, we can deduce that, akin to the scenarios involving diagonalizable matrices, the feature components associated with other eigenspaces in non-diagonalizable cases also diminish at an algebraic rate.

The proof now is complete.