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 enhances graph representation learning. Analytically, we exhibit that over-smoothing issues are mitigated when feature updating is regulated by a 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) 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) [6], and GraphSAGE [7]. Recent works, such as [8–13], have incorporated various continuous dynamical processes to propagate information over the graph nodes, inspiring a new class of GNNs based on ordinary differential equations (ODEs)[2] on graphs which enables the interpretation of GNNs as evolutionary dynamical systems.

Einstein's analysis of Brownian motion derived the heat equation with the presumption of instantaneous memory loss in particle motion direction [14]. This leads to the Markovian nature of heat diffusion on graphs, with subsequent motion depending solely on current location and graph structure [15, 9]. Researchers have also explored scenarios where Einstein's assumptions do not apply. For

[*]First two authors contributed equally to this work.

[2]Models like GRAND [8] 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.

example, various instances of anomalous diffusions have been observed in nature, such as protein diffusion in membranes [16], animal travel strategies [17], human mobility network[18], and biological applications in respiratory tissue and neuroscience [19]. In such cases, fractional-order differential equations (FDEs) [20] have been developed as generalizations of traditional integer-ordered differential equations to model these systems. The derivative order can be a real number (not necessarily an integer) and the fractional derivative's value depends on the function's global domain, not just the infinitesimal neighborhood of a traditional derivative. This global characteristic allows fractional calculus to better model non-Markovian dynamical processes with memory.

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 modification gives FROND the ability to incorporate *memory-dependent dynamics* for information propagation and feature updating, enabling refined graph representations and improved performance potentially. Importantly, this technique assures at least equivalent performance to integer-order models, as, when β assumes integer values, the models revert to conventional graph ODE models without memory.

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 [8–13], 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 [21]. This provides a fascinating analogy to the residual characteristic of Euler solvers in conventional neural ODEs [22] and give a new perspective to understand various skip or dense connections used between layers in prior literature [23–26].
- 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 ODE models [9–13] is left for future exploration.

2 Preliminaries

2.1 The Caputo Time-Fractional Derivative

Various fractional derivative definitions exist in literature, such as those by Riemann, Liouville, Chapman, and Caputo [27]. We employ the Caputo fractional derivative in this work. The conventional first-order derivative, $\mathrm{d}f(t)/\,\mathrm{d}t$, represents the *local rate of change* and has the Laplace transform:

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

where $F(s)$ is the Laplace transform of the function $f(t)$. The Caputo fractional derivative of order $\beta \in (0, 1]$ for a function $f(t)$ is defined as follows:

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

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 [28].

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 feature vectors with $W_{ij} = W_{ji}$. Inspired by the standard heat diffusion equation, GRAND [8] 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 random walk Laplacian is then represented as $\mathbf{L} = \mathbf{I} - \mathbf{W}\mathbf{D}^{-1}$. In a simplified context, we employ the following linear dynamical system:

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = (\mathbf{W}\mathbf{D}^{-1} - \mathbf{I})\mathbf{X}(t) = -\mathbf{L}\mathbf{X}(t). \tag{5}$$

The feature updating in (5) is the GRAND-l version, which is a time-invariant linear FDE.

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 techniques for the numerical FDE solver 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 [8–13]. 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 [8–13], 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 [8–13], we can derive fractional GNN variants such as F-GRAND, F-GRAND++, F-GREAD, F-CDE, and F-GraphCON, which act as fractional extensions of graph ODE models. Due to space constraints, this work primarily focuses on F-GRAND with 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 [29]. In parallel, the F-GRAND-l version stands as the fractional equivalent of (5):

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

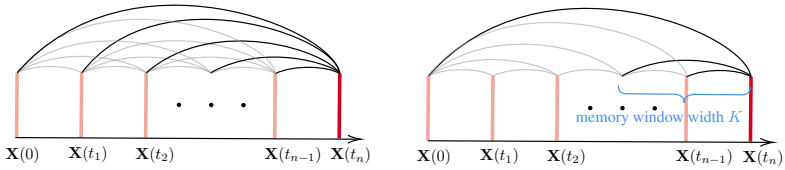

Figure 1: Diagrams of fractional Adams–Bashforth–Moulton method with full (left) and short (right) memory.

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. *In [10], the phenomenon of over-smoothness is defined through the exponential convergence of Dirichlet energy to zero.* Dirichlet Energy provides quantitative insights into the variability of features across nodes and their neighbors. Higher Dirichlet energy implies greater diversity in node features, suggesting lower over-smoothing levels, while lower energy points to the contrary, indicating a possible risk of information loss through excessive smoothing. GRAND-l has been proven to exhibit exponentially decreasing $\Theta(e^{-rt})$ Dirichlet energy according to [9]. In contrast, the following Theorem 1 establishes that the Dirichlet energy of F-GRAND-l converges algebraically to zero at a significantly slower rate, thereby mitigating over-smoothness issues.

Theorem 1. *Under the assumption that the graph is strongly connected and aperiodic, the Dirichlet energy* $\mathbf{E}(\mathbf{X}(t)) \coloneqq \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}} \left\|\mathbf{x}^{(i)}(t) - \mathbf{x}^{(j)}(t)\right\|_2^2$ *with* $\mathbf{X}(t)$ *being the solution to* (8)*, has the convergence rate* $\Theta(t^{-2\beta})$.

3.2 Solving FROND

The studies by [22, 30, 31] 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 [32], 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 [32][eq.17]. For $\beta = 1$, this method reduces to the Euler solver [22], 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, [33, 34] 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 generalizes existing GNN architectures. As the time fractional order β nears 1, D_t^β becomes the local first-order derivative $\frac{\mathrm{d}}{\mathrm{d}t}$, aligning FROND with conventional graph ODE frameworks [8–13]. The literature demonstrates numerous inter-layer connections [23–26]. By integrating fractional-order dynamics and memory effects, FROND offers insights into GNN architectures and fosters the development of advanced graph representation learning.

4 Experiments

In this paper, we 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 [9–13] for future work.

4.1 Node Classification of F-GRAND

Datasets and splitting. We utilize datasets with varied topologies, including citation networks (Cora [35], Citeseer [36], Pubmed [37]), tree-structured datasets (Disease and Airport [38]), coauthor and co-purchasing graphs (CoauthorCS [39], Computer and Photo [40]), and the ogbn-arxiv dataset [41]. We follow the same data splitting and pre-processing in [38] for Disease and Airport datasets. The other experiment settings are the same as in GRAND [8].

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. Our tests reveal a preference for smaller β, indicating enhanced dynamic memory, on such fractal-structured datasets.

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

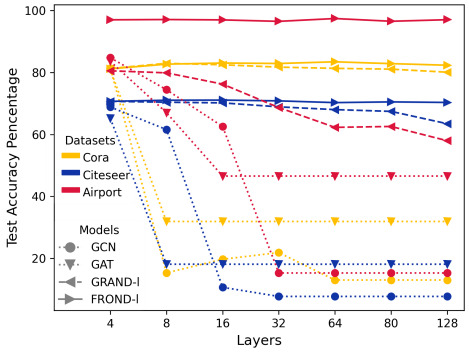

Figure 2: Over-smoothing mitigation.

4.2 Over-smoothing of F-GRAND

We validated F-GRAND's resilience to over-smoothing and its multilayer performance using the solver (9) and adhering to the fixed data split defined in [38]. It rendered optimal results on Cora at 64 layers. F-GRAND-l's performance remains consistent up to 128 layers across datasets, corroborating the slow algebraic convergence foreseen in Theorem 1, unlike GRAND's quicker performance decline noted particularly on the Airport dataset.

5 Conclusions

We introduced FROND, a novel graph learning framework that incorporates time-fractional Caputo derivatives to capture long-term memory in the graph feature updating dynamics. This approach may improve performance over the graph neural ODE models like GRAND. 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 [42].

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. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–12.

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

[8] 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.

[9] 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.

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

[11] 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.

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

[13] 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.

[14] A. Einstein, "Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen," *Annalen der physik*, vol. 4, 1905.

[15] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 912–919.

[16] D. Krapf, "Mechanisms underlying anomalous diffusion in the plasma membrane," *Current Topics Membranes*, vol. 75, pp. 167–207, 2015.

[17] D. Brockmann, L. Hufnagel, and T. Geisel, "The scaling laws of human travel," *Nature*, vol. 439, no. 7075, pp. 462–465, 2006.

[18] K. B. Gustafson, B. S. Bayati, and P. A. Eckhoff, "Fractional diffusion emulates a human mobility network during a simulated disease outbreak," *Frontiers Ecology Evol.*, vol. 5, p. 35, 2017.

[19] C. Ionescu, A. Lopes, D. Copot, J. T. Machado, and J. H. Bates, "The role of fractional calculus in modeling biological phenomena: A review," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 51, pp. 141–159, 2017.

[20] K. Diethelm and N. J. Ford, "Analysis of fractional differential equations," *J. Math. Anal. Appl.*, vol. 265, no. 2, pp. 229–248, 2002.

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

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

[23] 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.

[24] 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.

[25] 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.

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

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

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

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

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

[39] 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.

[40] 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.

[41] 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.

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

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

[44] F. Mainardi, "On some properties of the mittag-leffler function $\mathbf{E}_\alpha(-\mathbf{t}^\alpha)$, completely monotone for $\mathbf{t} > \mathbf{0}$ with $\mathbf{0} < \alpha < \mathbf{1}$," *Discrete Continuous Dyn. Syst. Ser. B*, vol. 19, no. 7, pp. 2267–2278, 2014.

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

A Proof of Theorem 1

Consider $\mathbf{L} = \mathbf{SJS}^{-1}$ as the Jordan canonical form of $\mathbf{L}$. It is evident that for the matrix $\mathbf{WD}^{-1}$, since $\mathbf{WD}^{-1}$ is left stochastic and the graph is strongly connected and aperiodic, the Perron-Frobenius theorem [43][Lemma 8.4.3., Theorem 8.4.4] confirms that the value 1 is the sole eigenvalue of it that equals the spectral radius 1. Hence, we have that $\mathbf{L} = \mathbf{I} - \mathbf{WD}^{-1}$ possesses an eigenvalue of 0, and all the remaining eigenvalues have positive real parts. Consequently, $\mathbf{J}$ contains a block that consists of only a single 0. WLOG, we assume the feature dim is one and we rewrite (8) as

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

where $\mathbf{S}^{-1}\mathbf{X}(t) = \mathbf{Y}(t) \in \mathbb{R}^N$ and the inital condition is $\mathbf{S}^{-1}\mathbf{X}(0) = \mathbf{Y}(0)$.

If $\mathbf{L}$ is diagonalizable, then $\mathbf{J}$ is a diagonal matrix with the diagonal elements being the eigenvalues. We have an uncoupled set of equations in the form $D_t^{\beta}\mathbf{Y}_k(t) = -\lambda_k\mathbf{Y}_k(t)$, where $\mathbf{Y}_k$ is the k-th component of $\mathbf{Y}$. According to [34], the solution is given by

$$\mathbf{Y}_k(t) = \mathbf{Y}_k(0)E_{\beta}(-\lambda_k t^{\beta}) \tag{11}$$

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. For the index w s.t. the eigenvalue $\lambda_w = 0$, we have the solution $\mathbf{Y}_k(t) = \mathbf{Y}_k(0)$ which corresponds to the stationary probability vector if we transform it back to $\mathbf{X}(t)$. From [44], we have that for $k \neq w$, the convergence to 0 is in the following order

$$\mathbf{Y}_k(t) = \Theta(t^{-\beta}).$$

If $\mathbf{J}$ is not diagonal, the entries of $\mathbf{Y}(t)$ that correspond to distinct Jordan blocks in $\mathbf{J}$ are not coupled. W.L.O.G, we assume the first Jordan block is associated to eigenvalue $\lambda_1 = 0$, while all eigenvalues $\lambda_k > 0$, for $k = 2, \ldots, N$. A consideration of the Jordan block corresponding to one λ_k, $k = 2, \ldots, N$, is adequate. We assume the Jordan block $\mathbf{J}(\lambda_k)$ corresponding to λ_k has size m. It follows that for this Jordan block we have

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

which can be solved from the bottom up. Starting with the last equation, we have that

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

Furthermore, we have

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

Take the Laplace transform, we have

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

where $Y_{m-1}(s)$ is the Laplace transform of $\mathbf{Y}_{m-1}(t)$ according to (3). Now, for the right hand side, we have $L\{\lambda\mathbf{Y}_{m-1}(t)\} = \lambda_k Y_{m-1}(s)$ and we know that the Laplace transform of $E_{\beta}\left(-\lambda_k t^{\beta}\right)$ is $\frac{s^{\beta-1}}{(s^{\beta}+\lambda_k)}$. Therefore, the equation in the Laplace domain becomes:

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

Rearranging this equation to solve for $Y_{m-1}(s)$ gives:

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

It follows that $Y_{m-1}(s) \sim Cs^{\beta-1}$ when $s \to 0$. We can repeat the above process to show $Y_i(s) \sim Cs^{\beta-1}$ when $s \to 0$ for all $i = 1, \ldots, m-2$. According to the Hardy–Littlewood Tauberian theorem [45], we have that, for all $i = 1, \ldots, m$,

$$\mathbf{Y}_i(t) = \Theta(t^{-\beta}). \tag{12}$$

We use notation $\mathbf{e}_i$ to denote the one-hot vector where the i-th component stands at 1. The set $\{\mathbf{e}_i\}_{i=1}^N$ is linearly independent and span the full $\mathbb{R}^N$ space. It is equivalent to getting the transformed solution $\mathbf{Y}_{(i)}(t)$ with the initial condition $\mathbf{S}^{-1}\mathbf{e}_i$ in (10). The entries of $\mathbf{Y}(t)$ that correspond to distinct Jordan blocks in $\mathbf{J}$ are not coupled. We denote the solution to (10) with initial condition $\mathbf{e}_k$ as $\bar{\mathbf{Y}}_{(k)}(t)$. We have that the solution corresponds to the unique eigenvalue 0 to matrix $\mathbf{J}$ keep a constant. If we assume eigenvalue 0 is the first Jodan block, we have $\bar{\mathbf{Y}}_{(1)}(t) = \bar{\mathbf{Y}}_{(1)}(0)$ for all time $t \geq 0$. While all the other solutions $\bar{\mathbf{Y}}_{(k)}(t)$, $k = 2, ..., N$, corresponding to the other Jordan blocks, converge to 0 in $\Theta\left(t^{-\beta}\right)$ rate. From the linearity, we then have $\mathbf{Y}_{(i)}(t)$ are the linear combination of the N independent solutions $\{\bar{\mathbf{Y}}_{(k)}(t)\}_{i=1}^N$. More specifically, we have that

$$\mathbf{Y}_{(i)}(t) = [\mathbf{S}^{-1}\mathbf{e}_i]_0 \bar{\mathbf{Y}}_1(0) + \sum_{k=2}^{N} [\mathbf{S}^{-1}\mathbf{e}_i]_k \Theta\left(t^{-\beta}\right)$$

where $[\mathbf{S}^{-1}\mathbf{e}_i]_k$ is the k-component of matrix $\mathbf{S}^{-1}\mathbf{e}_i$. We can prove that the first row of $\mathbf{S}^{-1}$ is $a\mathbf{1}^\top$ with a being a scalar and $\mathbf{1}$ is an all-ones vector (it is based on [43][Theorem 3.2.5.2.], see Lemma 1). It follows that $[\mathbf{S}^{-1}\mathbf{e}_i]_0$ is the same for all i. We therefore have that for some i and j

$$\left\|\mathbf{x}^{(i)}(t) - \mathbf{x}^{(j)}(t)\right\|_2^2 = \left\|\mathbf{S}\mathbf{Y}_{(i)}(t) - \mathbf{S}\mathbf{Y}_{(j)}(t)\right\|^2 = \Theta\left(t^{-2\beta}\right)$$

The proof is now complete.

Lemma 1. *The first row of* $\mathbf{S}^{-1}$ *is* $a\mathbf{1}^\top$ *with* a *being a scalar and* $\mathbf{1}$ *is an all-ones vector.*

Proof. The Jordan canonical form of $\mathbf{W}\mathbf{D}^{-1}$ is represented as $\mathbf{S}\bar{\mathbf{J}}\mathbf{S}^{-1}$ where $\bar{\mathbf{J}} = \mathbf{J} + \mathbf{I}$ with the first Jordan block being 1 and the rest having eigenvalues *strictly less than* 1. Based on [43][Theorem 3.2.5.2.], we observe that $\lim_{k\to\infty}(\mathbf{W}\mathbf{D}^{-1})^k = \lim_{k\to\infty} \mathbf{S}\bar{\mathbf{J}}^k\mathbf{S}^{-1} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$, where $\mathbf{\Lambda}$ is a diagonal matrix with the first element as 1 and all the others as 0:

$$\mathbf{\Lambda} = \begin{pmatrix} 1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{pmatrix}$$

Since $\lim_{k\to\infty}(\mathbf{W}\mathbf{D}^{-1})^k$ maintains its column stochasticity and the rank of $\mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$ is 1, we deduce that the first row of $\mathbf{S}^{-1}$ is $a\mathbf{1}^\intercal$ with a being a scalar and $\mathbf{1}$ an all-ones vector. □