Advancing Graph Neural Networks Through Joint Time-Space Dynamics

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce the GeneRAlized Fractional Time-space graph diffusion network (GRAFT), a framework combining temporal and spatial nonlocal operators on graphs to effectively capture long-range interactions across time and space. Leveraging time-fractional diffusion processes, GRAFT encompasses a system's full historical context, while the d-path Laplacian diffusion ensures extended spatial interactions based on shortest paths. Notably, GRAFT mitigates the over-squashing problem common in graph networks. Empirical results show its prowess on self-similar, tree-like data due to its fractal-conscious design with fractional time derivatives. We delve deeply into the mechanics of GRAFT, emphasizing its distinctive ability to encompass both time and space diffusion processes through a random walk perspective.

1 Introduction

Graph Neural Networks (GNNs), notably used in bioinformatics [1], finance [2], and social networks [3–5], harness message passing for adaptability. Variants include the Graph Convolutional Networks (GCN) [3], Graph Attention Networks (GAT) [6], and GraphSAGE [7]. Incorporating neural ordinary differential equations (ODEs) into GNNs [8], as evidenced in GRAND [9], GRAND++ [10], GraphCON [11], CDE [12], and GraphBel [13], provides a novel dynamical systems perspective on graph feature evolution. However, GNNs often struggle with over-squashing [14] related to long-range interactions. For tasks dependent on long-range node interactions with distance r, the GNN layer count, K, should match the span of these interactions, necessitating $K \geq r$ layers. This results in a node's receptive field growing exponentially with K, making it intricate to encapsulate the vast information within a fixed-length vector.

In our study, we leverage a dynamical systems approach with a generalized fractional diffusion equation for graphs. In GNNs such as GRAND and GRAND++, standard diffusion equations are expressed as $\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = \mathbf{L}\mathbf{X}(t)$, with $\mathbf{L}$ being a potential adaptive graph Laplacian and $\mathbf{X}(t)$ capturing node features. These equations highlight local characteristics in time and space. In the temporal context, they suggest a short-lived motion direction, translating to a memoryless Markovian graph random walk [10]. Spatially, the scope of particle movement is confined to neighboring nodes. *In divergence, our generalized fractional diffusion equation emphasizes nonlocality in both temporal and spatial domains.* Temporally, the equation embodies a non-Markovian random walk through the more general fractional time-order derivative D^β with $\beta \in (0, 1]$ (notably, when $\beta = 1$, $D^\beta = \frac{\mathrm{d}}{\mathrm{d}t}$). Spatially, it allows jumps beyond immediate neighbors using the Mellin-transformed d-path Laplacian, $\mathbf{L}_s$. In essence, by integrating fractional calculus into our formulation, we arrive at the generalized diffusion equation $D^\beta \mathbf{X}(t) = \mathbf{L}_s \mathbf{X}(t)$, which reverts to the conventional form if $\beta = 1$ and $s = \infty$.

Main contributions. In this paper, our prime focus is on devising a generalized fractional diffusion-based GNN that exudes global characteristics in both time and space domains. We christen our model

Submitted to the DLDE-III Workshop in the 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

the GeneRAlized Fractional Time-space diffusion network (GRAFT). Our main contributions are summarized as follows:

1. We introduce a generalized fractional diffusion graph neural network that manifests nonlocal dynamics in both time (layer-wise) and space (the graph domain).
2. We furnish a detailed random walk interpretation of the generalized diffusion equation, wherein the fractional-time derivative denotes a memory-influenced jump — implying that jumps between consecutive layers are influenced by prior layers upon discretization — and the Mellin-transformed d-path Laplacian operator suggests long-range hops within the graph.
3. We demonstrate that GRAFT can alleviate the over-squashing predicament due to its inherent long-range interactions. Moreover, given the link between fractional dynamics on networks and fractal geometry, we demonstrate GRAFT's commendable performance on tree-structured datasets.

2 Preliminaries and Framework

2.1 Temporal Dynamics with Graph Neural FDEs

There are multiple definitions for D^β in the literature, including those by Riemann, Liouville, Chapman, and Caputo, which explore temporal nonlocality [15]. Note that the temporal domain in our paper refers to the "time" over which node feature evolves, drawing a parallel to layer analogies [8], different from the temporal domain in spatio-temporal GNNs like [16, 17]. In this study, we mainly employ the *Marchaud–Weyl* fractional derivative ${}_{\mathrm{M}}D^\alpha$, recognized for its efficacy in elucidating the fading memory phenomena [18–20]. On the other hand, the *Caputo* derivative ${}_{\mathrm{C}}D^\beta$ is favored in engineering contexts [21] and is used in Appendix E. Due to space considerations, a comprehensive discussion on these derivatives is reserved for supplementary materials.

Definition 1 (Marchaud–Weyl Fractional Derivative)**.** *Given a scalar function f over real numbers and satisfying specific assumptions [22], the* Marchaud–Weyl *fractional derivative at point t is:*

$$ {}_{\mathrm{M}}D^\beta f(t) = \frac{\beta}{\Gamma(1-\beta)} \int_0^\infty \frac{f(t) - f(t-\tau)}{\tau^{1+\beta}} \, \mathrm{d}\tau, \tag{1} $$

where $\Gamma(\cdot)$ denotes the Gamma function. For functions that are sufficiently smooth, according to [22], we have

$$ \lim_{\beta \to 1^-} {}_{\mathrm{M}}D^\beta f(t) = \frac{\mathrm{d}f(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{f(t+\Delta t) - f(t)}{\Delta t}. \tag{2} $$

It is seen from (1) that the *Marchaud–Weyl* fractional derivative, a nonlocal operator, accounts for the past values of f within the range (∞, t), indicative of its temporal memory effect. For a vector-valued function, the fractional derivative is defined component-wise for each dimension.

2.2 Space-Fractional Operator: Path Laplacian

Laplacian operators are derived from the divergence of the gradient of functions defined over the nodes of a graph $\mathcal{G}$ with node set $\mathcal{V}$ comprising $|\mathcal{V}| = N$ nodes. Given $L^2(\mathcal{V})$ as the Hilbert space of functions on $\mathcal{V}$, we introduce the Mellin-transformed d-path Laplacian operator, underscoring its nonlocal long-range interactions over the space domain and its ties with the local Laplacian operator.

Definition 2 (Mellin-transformed d-path Laplacian Operator)**.** *The Mellin-transformed d-path Laplacian operator in $L^2(\mathcal{V})$ is defined as*

$$ \big(\mathbf{L}_s f\big)(i) := \sum_{w \in \mathcal{V}: d(i,j) = d_{ij}} \frac{f(i) - f(j)}{(d_{ij})^s}, \tag{3} $$

where $f \in L^2(\mathcal{V})$, d_{ij} is the shortest path distance between node i and node j, and $0 \le s \le \infty$ represents the nonlocal parameter. Additionally, the Mellin-transformed d-path Laplacian can be defined as a matrix form: $\mathbf{L}_s := \mathbf{D}_s - \mathbf{A}_s$, where $\mathbf{A}_s = [a_{ij}(s)]_{|\mathcal{V}| \times |\mathcal{V}|}$ is a d-path adjacency matrix by taking the shortest path distance d_{ij} into consideration with $a_{ij}(s) = (d_{ij})^{-s}$ if $i \neq j$, and $a_{i,j} = 0$ if $i = j$. The term $(-s)$ represents the negative entrywise power, and $\mathbf{D}_s$ is the node degree matrix defined as: $\mathbf{D}_s := \operatorname{diag}(\mathbf{A}_s \mathbf{1})$, which is a diagonal matrix with $(D_s)_{ii} = \sum_j a_{ij}(s)$. Here $\mathbf{1}$ denotes the all-one vector. Furthermore, the normalized Mellin-transformed d-path Laplacian can be further defined as $\widetilde{\mathbf{L}}_s := \mathbf{I} - \widetilde{\mathbf{A}}_s$ with $\widetilde{\mathbf{A}}_s = \mathbf{A}_s (\mathbf{D}_s)^{-1}$.

Remark 1. *The Mellin-transformed d-path Laplacian operator $\mathbf{L}_s$ incorporates nonlocal/long-range interactions between node pairs via the shortest paths connecting them. Pairs of directly-connected nodes in the graph interact locally with a 'strength' of one ($d_{ij} = 1$), whereas pairs of non-directly connected nodes (i, j) exhibit nonlocal/long-range interactions with a 'strength' of $(d_{ij})^{-s}$. The parameter s modulates the extent of these long-range interactions in the space domain, analogous to the role of β in the time fractional derivative ${}_{\mathrm{M}}D^\beta \mathbf{X}(t)$. Compared to the conventional Laplacian operator:*

$$ \mathbf{L} f(v) := \sum_{(v,w) \in E} f(v) - f(w), \quad f \in L^2(\mathcal{V}), \tag{4} $$

It is evident that as $s \to \infty$, the Mellin-transformed d-path Laplacian operator, $\mathbf{L}_s$, converges to the standard Laplacian operator, $\mathbf{L}$.

3 Generalized Fractional Time-Space Diffusion Equation Graph Network

Exploring the temporal and spatial nonlocal operators ${}_{\mathrm{M}}D^{\beta}$ and $\mathbf{L}_s$ highlighted in Section 2, this section introduces GRAFT. This generalized diffusion equation on graphs enriches GNN frameworks beyond the conventional GRAND. We explore the GRAFT model and its random walk interpretation, with details on numerical solvers in Appendix E.

3.1 Model

GRAFT embodies a generalized fractional diffusion process on graphs. The incorporation of the time-fractional diffusion mechanism embeds a memory mechanism, taking into account the entire evolutionary history of a system rather than solely its present state. Meanwhile, the d-path Laplacian reflects the long-range interactions between nodes, gauged by their shortest connecting path.

Consider an undirected graph $G = (\mathbf{X}, \mathbf{W})$, where $\mathbf{X} = \left(\left[\mathbf{x}^{(1)}\right]^{\intercal}, \cdots, \left[\mathbf{x}^{(N)}\right]^{\intercal}\right)^{\intercal} \in \mathbb{R}^{N\times d}$ where each row $\mathbf{x}^{(i)} \in \mathbb{R}^d$ represents the i-th node feature vector. The $N \times N$ matrix $\mathbf{W} := (W_{ij})$ is the adjacency matrix of the graph whose elements W_{ij} indicating the edge weight between the i-th and j-th nodes with $W_{ij} = W_{ji}$. We set the feature updating equation as

$$ {}_{\mathrm{M}}D^{\beta}\mathbf{X}(t) = -\mathbf{L}_s\mathbf{X}(t) = -\sum_{d=1}^{\Delta} d^{-s}\mathbf{L}_d\mathbf{X}(t), \tag{5}$$

where $0 < \beta \leq 1$ and $0 < s < \infty$. Here $\mathbf{X}(t) = \left(\left[\mathbf{x}^{(1)}(t)\right]^{\intercal}, \ldots, \left[\mathbf{x}^{(N)}(t)\right]^{\intercal}\right)^{\intercal} \in \mathbb{R}^{N\times d}$ is the features at time t with $\mathbf{x}^{(i)}(0) = \mathbf{x}^{(i)}$ serving as the initial features for $i = 1, \ldots, N$. It is evident that node features in a graph are influenced not only by their immediate neighbors but also through space-based long-range interactions. The coefficients, d^{-s}, depict the rate at which these interactions decay based on the power law of path-length d. The parameter s is designed to be learnable.

Remark 2. *GRAFT's dynamics, highlighted in* (5)*, present a holistic time-space diffusion equation. When $s \to \infty$, the equation converges to $D_t^{\beta}\mathbf{X}(t) = -\mathbf{L}\mathbf{X}(t)$, reflecting the standard graph Laplacian with just the time-fractional process and no long-range spatial interactions [23]. Meanwhile, as $\beta \to 1$, we derive $\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = -\mathbf{L}_s\mathbf{X}(t)$, denoting the typical d-path Laplacian diffusion, focusing on spatial graph interactions without temporal ones [24].*

3.2 Fractional Graph Random Walk with Memory and Long range Interaction

In this section, we provide a non-Markov graph random walk interpretation for (5), highlighting long-range jumps in both the temporal and spatial realms, each following a power-law decay probability. For clarity, without loss of generality, we interpret $\mathbf{X}(t)$ as a $|\mathcal{V}|$-dimensional probability or mass concentration vector $\mathbb{P}(t)$ over the graph nodes $\mathcal{V}$. We consider a random walker navigating over graph $\mathcal{G}$ with an infinitesimal interval of time $\Delta\tau > 0$. We assume that there is no self-loop in the graph topology. The dynamics of the random walk are characterized as follows:

1. The walker is expected to wait at the current location for a random period of time. The distribution of waiting times, $\psi_\beta(\tau)$, is given by a power-law function $d_\beta n^{-(1+\beta)}$ with $d_\beta > 0$ chosen to ensure $\sum_{n=1}^{\infty}\psi_\beta(n) = 1$.
2. Upon deciding to make a jump, the walker can either move from the current node i to node j with a power-law probability of $(\Delta\tau)^\beta d_\beta|\Gamma(-\beta)|\frac{(d_{ij})^{-s}}{\sum_j (d_{ij})^{-s}}$ if $i \neq j$. Alternatively, with a probability of $1-(\Delta\tau)^\beta d_\beta|\Gamma(-\beta)|$, it will remain at the current node i.

It should be noted that the jump to node i itself in the second option is conceptually distinct from waiting at node i as per the first option, despite the resultant observation appearing identical—i.e., the walker remaining at the current node. Our goal is to compute $\mathbb{P}_j(t;\beta)$, the probability of the walker being at node j at time t. The law of total probability for the above random walk is expressed as:

$$\mathbb{P}_j(t;\beta) = \sum_{n=1}^{\infty}\Big[\sum_{\substack{i\in\mathcal{V}\\ i\neq j}}\mathbb{P}_i(t-n\Delta\tau;\beta)(\Delta\tau)^\beta d_\beta|\Gamma(-\beta)|\frac{(d_{ij})^{-s}}{\sum_j (d_{ij})^{-s}} + \mathbb{P}_j(t-n\Delta\tau;\beta)\left(1-(\Delta\tau)^\beta d_\beta|\Gamma(-\beta)|\right)\Big]\psi_\beta(n).$$

In this equation, the summation over n accounts for the possibility that the walker may have remained stationary for a period of $t - n\Delta\tau$, with a waiting time probability of $\psi_\beta(n)$.

Theorem 1. *Given a specific $\beta \in (0, 1)$ and as $\Delta\tau \to 0$, we have that $\mathbb{P}(t;\beta)$ solves* (5)*, i.e.,*

$$\lim_{\Delta\tau\to 0}\left\{{}_{\mathrm{M}}D^{\beta}\mathbb{P}(t;\beta) + \mathbf{L}_s\mathbb{P}(t;\beta)\right\} = 0.$$

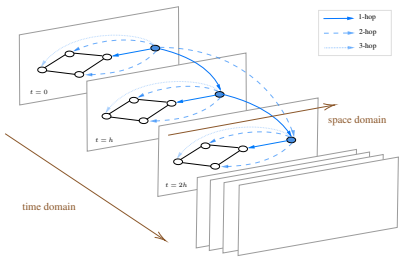

Figure 1: GRAFT's information flow is discretized with a source node marked in blue. Colored arrows show hop distances, linking spatial and temporal neighbors. Layers, as referenced in Appendix E, equate to time, while the graph depicts space. GRAFT ensures bidirectional communication, integrating long-range space-time interactions.

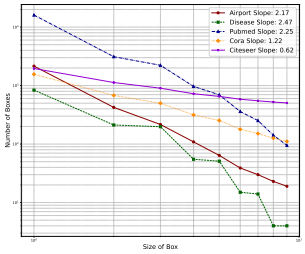

Figure 2: The fractal dim of datasets. We use the Compact-Box-Burning algorithm in [25] to compute the log-log slope (fractal dim) of the box size and the minimum number of boxes needed to cover the graph.

4 Experiments

4.1 Node Classification

For node classification, we employed various datasets including Cora [26], Citeseer [27], Pubmed [28], and tree-structured datasets (Disease and Airport [29]). We processed Disease and Airport datasets following [29], and applied random splits [9] to the others. Our GRAFT model's performance for node classification was evaluated against key GNNs baselines: Euclidean (e.g., GCN [3], GAT [6], and SGC [30]), Hyperbolic (e.g., HGCN [29], HGAT [31], and LGCN [32]); and as well as GIL [33], HamGNN [34], and graph neural Diffusions models GRAND [9] and GraphCON [35].

In Table 1, GRAFT outperforms on citation networks and holds its own on tree-structured data, notably against tree-focused GNNs like HGCN, HGAT, and GIL. This prowess is credited to GRAFT's fractional calculus techniques. Using the Compact-Box-Burning algorithm [25], we determine fractal dimensions for datasets, depicted in Fig. 2. A clear correlation in Table 1 emerges: lower δ-hyperbolicity (indicating more tree-like graphs as per [29]) relates to higher fractal dimensions. According to [15, 36], fractional calculus adeptly captures heat or mass dispersion in fractal mediums. The close tie between fractal dimension and fractional derivative order [15, 36] insinuates the optimal β in $D_t^\beta \mathbf{X}(t)$ may uncover the graph's inherent fractal nature. The consistent performance of GRAFT across various datasets emphasizes its adaptability to different levels of fractalness. In contrast, graph ODE models such as GRAND and GraphCON struggle with datasets like Airport and Disease, which have higher fractal dimensions.

Method	Cora	Citeseer	Pubmed	Airport	Disease
fractal dim	1.22	0.62	2.25	2.17	2.47
δ hyperbolicity	11.0	4.5	3.5	1.0	0
MLP	57.2±1.2	58.1±1.9	72.0±1.4	77.0±1.8	50.0±0.0
GCN	81.5±1.3	71.9±1.9	77.8±2.9	81.6±0.6	69.8±0.5
GAT	81.8±1.3	71.4±1.9	78.7±2.3	81.6±0.4	70.4±0.5
SGC	82.0±1.7	70.9±1.3	76.8±1.1	81.4±2.2	82.8±0.9
HGCN	78.7±1.0	65.8±2.0	76.4±0.8	85.4±0.7	89.9±1.1
HGAT	80.9±0.8	69.2±1.0	78.0±0.5	87.5±1.0	88.7±3.4
LGCN	80.6±0.9	68.1±2.0	77.4±1.4	88.2±0.2	88.5±1.8
GIL	83.6±1.0	73.4±0.5	78.8±1.7	91.5±1.7	90.8±0.5
GRAND	83.6±1.0	73.4±0.5	78.8±1.7	80.5±9.6	74.5±3.4
GraphCON	84.2±1.3	74.2±1.7	79.4±1.3	68.6±2.1	87.5±4.1
HamGNN	82.2±0.8	72.4±0.9	78.1±0.5	96.0±0.1	**91.5±2.1**
GRAFT	**84.4±0.7**	**74.6±1.8**	**79.7±1.8**	**96.6±0.6**	90.7±2.7

Table 1: Node classification results(%) random train-val-test splits

Model	DD	PROTEINS
GCN [3]	75.63±2.95	75.11±4.51
ResGCN [37]	76.65±2.73	75.11±3.22
GCNJK [38]	73.16±5.12	75.24±4.15
DGCNN [39]	61.63±5.33	73.95±3.04
SAGPool [40]	70.52±5.48	71.89±4.03
DiffPool [41]	73.16±5.12	75.24±4.15
TwoHop [42]	74.53±5.24	75.30±4.27
GCNFA [43]	OOM	74.31±4.16
GraphTrans [44]	OOM	75.12±4.89
LRGNN [45]	78.18±2.02	75.39±4.04
GRAFT	**79.83±5.45**	**76.28±3.57**

Table 2: Graph Classification Results

4.2 Graph Classification

We incorporated the DD and Proteins datasets from [46] for graph-based protein structure classification. Statistics for these protein graphs can be found in the supplementary material. Notably, such datasets assess a model's proficiency in capturing long-range interactions [45]. As Table 2 reveals, our model thrives on these datasets, underscoring its adeptness at grasping long-range graph interactions.

5 Conclusion

We presented GRAFT, a groundbreaking framework blending temporal and spatial nonlocal operators for graphs. Through time-fractional diffusion and the d-path Laplacian, GRAFT addresses feature dynamics and the over-squashing challenge in GNNs. Our empirical results highlight its potency, especially with fractal-like data, marking a new avenue in GNN research.

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

[9] 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.

[10] 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.

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

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

[13] 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.*, New Orleans, USA, Nov. 2022.

[14] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," *arXiv preprint arXiv:2006.05205*, 2020.

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

[16] J. Kan, K. Hu, M. Hagenbuchner, A. C. Tsoi, M. Bennamounm, and Z. Wang, "Sign language translation with hierarchical spatio-temporal graph neural network," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 3367–3376.

[17] B.-H. Kim, J. C. Ye, and J.-J. Kim, "Learning dynamic graph representation of brain connectome with spatio-temporal attention," in *Advances Neural Inf. Process. Syst.*, 2021, pp. 4314–4327.

[18] S. G. Samko, "Fractional integrals and derivatives," *Theory Appl.*, 1993.

[19] A. Bernardis, F. J. Martín-Reyes, P. R. Stinga, and J. L. Torrea, "Maximum principles, extension problem and inversion for nonlocal one-sided equations," *J. Differ. Equ.*, vol. 260, no. 7, pp. 6333–6362, 2016.

[20] P. R. Stinga, "Fractional derivatives: Fourier, elephants, memory effects, viscoelastic materials and anomalous diffusions," *arXiv preprint arXiv:2212.02279*, 2022.

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

[22] F. Ferrari, "Weyl and marchaud derivatives: A forgotten history," *Mathematics*, vol. 6, no. 1, p. 6, 2018.

[23] Anonymous, "Fractional-order graph neural diffusion," in *Submitted paper*, 2023.

[24] E. Estrada, E. Hameed, N. Hatano, and M. Langer, "Path laplacian operators and superdiffusive processes on graphs. i. one-dimensional case," *Linear Algebra appl.*, vol. 523, pp. 307–334, 2017.

[25] C. Song, L. K. Gallos, S. Havlin, and H. A. Makse, "How to calculate the fractal dimension of a complex network: the box covering algorithm," *J. Stat. Mech. Theory Exp.*, vol. 2007, no. 03, p. P03006, 2007.

[26] 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.

[27] 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.

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

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

[30] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2019, pp. 6861–6871.

[31] Y. Zhang, X. Wang, C. Shi, X. Jiang, and Y. F. Ye, "Hyperbolic graph attention network," *IEEE Tran. Big Data*, vol. 63, no. 1, 2021.

[32] Y. Zhang, X. Wang, C. Shi, N. Liu, and G. Song, "Lorentzian graph convolutional networks," in *Proc. Web Conf.*, 2021.

[33] S. Zhu, S. Pan, C. Zhou, J. Wu, Y. Cao, and B. Wang, "Graph geometry interaction learning," in *Advances Neural Inf. Process. Syst.*, 2020.

[34] Q. Kang, K. Zhao, Y. Song, S. Wang, and W. P. Tay, "Node embedding from neural Hamiltonian orbits in graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, Haiwaii, USA, Jul. 2023.

[35] T. K. Rusch, B. Chamberlain, J. Rowbottom, S. Mishra, and M. Bronstein, "Graph-coupled oscillator networks," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2022, pp. 18 888–18 909.

[36] R. Nigmatullin, "Fractional integral and its physical interpretation," *Theoretical and mathematical physics*, vol. 90, no. 3, pp. 242–251, 1992.

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

[38] 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.

[39] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proc. AAAI Conf. Artif. Intell.*, 2018.

[40] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proc. Int. Conf. Mach. Learn.*, 2018.

[41] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Advances Neural Inf. Process. Syst.*, 2018.

[42] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. V. Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *Int. Conf. Mach. Learn.*, 2019.

[43] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," in *Int. Conf. Learn. Represent.*, 2021.

[44] P. Jain, Z. Wu, M. Wright, A. Mirhoseini, J. E. Gonzalez, and I. Stoica, "Representing long-range context for graph neural networks with global attention," in *Advances Neural Inf. Process. Syst.*, 2021.

[45] L. Wei, Z. He, H. Zhao, and Q. Yao, "Search to capture long-range dependency with stacking gnns for graph classification," in *Proc. ACM Web Conf.*, Apirl 2023, p. 588–598.

[46] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *J. Molecular Biology*, vol. 330, no. 4, pp. 771–783, 2003.

[47] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 21–29.

[48] S. Maskey, R. Paolino, A. Bacho, and G. Kutyniok, "A fractional graph laplacian approach to oversmoothing," *arXiv preprint arXiv:2305.13084*, 2023.

[49] G. Fu, P. Zhao, and Y. Bian, "p-laplacian based graph neural networks," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2022, pp. 6878–6917.

[50] B. Gutteridge, X. Dong, M. M. Bronstein, and F. Di Giovanni, "Drew: Dynamically rewired message passing with delay," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2023, pp. 12 252–12 267.

[51] E. Estrada, "Path laplacians versus fractional laplacians as nonlocal operators on networks," *New J. Phys.*, vol. 23, no. 7, p. 073049, 2021.

[52] F. Di Giovanni, L. Giusti, F. Barbero, G. Luise, P. Lio, and M. M. Bronstein, "On over-squashing in message passing neural networks: The impact of width, depth, and topology," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2023, pp. 7865–7885.

[53] 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.

[54] 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.

[55] 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.

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

[57] S. B. Yuste and L. Acedo, "An explicit finite difference method and a new von neumann-type stability analysis for fractional diffusion equations," *SIAM J. Numerical Analysis*, vol. 42, no. 5, pp. 1862–1874, 2005.

[58] B. D. Coleman and W. Noll, "Foundations of linear viscoelasticity," *Reviews of modern physics*, vol. 33, no. 2, p. 239, 1961.

[59] R. Almeida, N. R. Bastos, and M. T. T. Monteiro, "Modeling some real phenomena by fractional differential equations," *Mathematical Methods in the Applied Sciences*, vol. 39, no. 16, pp. 4846–4855, 2016.

[60] I. Podlubny, "Fractional-order systems and fractional-order controllers," *Institute of Experimental Physics, Slovak Academy of Sciences, Kosice*, vol. 12, no. 3, pp. 1–18, 1994.

[61] J. T. Machado, V. Kiryakova, and F. Mainardi, "Recent history of fractional calculus," *Commun. Nonlinear Sci.*, vol. 16, no. 3, pp. 1140–1153, 2011.

[62] E. Scalas, R. Gorenflo, and F. Mainardi, "Fractional calculus and continuous-time finance," *Physica A*, vol. 284, no. 1-4, pp. 376–384, 2000.

[63] 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.

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

[65] 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.

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

[67] Z. Liu, Y. Wang, Y. Luo, and C. Luo, "A regularized graph neural network based on approximate fractional order gradients," *Mathematics*, vol. 10, no. 8, p. 1320, 2022.

[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[69] H. Antil, R. Khatri, R. Löhner, and D. Verma, "Fractional deep neural network via constrained optimization," *Mach. Learn.: Sci. Tech.*, vol. 2, no. 1, p. 015003, 2020.

[70] G. Pang, L. Lu, and G. E. Karniadakis, "fpinns: Fractional physics-informed neural networks," *SIAM J. Sci. Comput.*, vol. 41, no. 4, pp. A2603–A2626, 2019.

[71] L. Guo, H. Wu, X. Yu, and T. Zhou, "Monte carlo fpinns: Deep learning method for forward and inverse problems involving high dimensional fractional partial differential equations," *Comput. Methods Appl. Mech. Eng.*, vol. 400, p. 115523, 2022.

[72] S. Wang, H. Zhang, and X. Jiang, "Fractional physics-informed neural networks for time-fractional phase field models," *Nonlinear Dyn.*, vol. 110, no. 3, pp. 2715–2739, 2022.

[73] J. Topping, F. Di Giovanni, B. P. Chamberlain, X. Dong, and M. M. Bronstein, "Understanding over-squashing and bottlenecks on graphs via curvature," *arXiv preprint arXiv:2111.14522*, 2021.

[74] M. Black, Z. Wan, A. Nayyeri, and Y. Wang, "Understanding oversquashing in gnns through the lens of effective resistance," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2023, pp. 2528–2547.

[75] M. Poli, S. Massaroli, J. Park, A. Yamashita, H. Asama, and J. Park, "Graph neural ordinary differential equations," *arXiv preprint arXiv:1911.07532*, 2019.

[76] L.-P. Xhonneux, M. Qu, and J. Tang, "Continuous graph neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10 432–10 441.

[77] B. Chamberlain, J. Rowbottom, M. I. Gorinova, M. Bronstein, S. Webb, and E. Rossi, "Grand: Graph neural diffusion," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2021, pp. 1407–1418.

[78] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14.

[79] 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.

[80] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances neural inf. process. syst.*, vol. 30, 2017.

[81] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "Tudataset: A collection of benchmark datasets for learning with graphs," *ArXiv*, vol. abs/2007.08663, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:220633407

[82] K. Karhadkar, P. K. Banerjee, and G. Montúfar, "Fosr: First-order spectral rewiring for addressing oversquashing in gnns," *ArXiv*, vol. abs/2210.11790, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253080708

[83] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," *arXiv preprint arXiv:2106.07594*, 2021.

[84] K. Diethelm and N. Ford, "The analysis of fractional differential equations," *Lect. Notes Math*, vol. 2004, pp. 3–12, 2010.

[85] D. Baleanu, V. E. Balas, and P. Agarwal, *Fractional order systems and applications in engineering*. Academic Press, 2022.

[86] D. Baleanu, K. Diethelm, E. Scalas, and J. J. Trujillo, *Fractional calculus: models and numerical methods*. World Scientific, 2012, vol. 3.

[87] H. Scher and E. W. Montroll, "Anomalous transit-time dispersion in amorphous solids," *Phys. Rev. B*, vol. 12, no. 6, p. 2455, 1975.

[88] K. Atkinson, W. Han, and D. E. Stewart, *Numerical solution of ordinary differential equations*. John Wiley & Sons, 2011.

[89] 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.

A Supplementary materials

A.1 Related work

A.1.1 Long Range Interaction and Skip Connection

Numerous works, including [47–50], have investigated the k-hop interaction between graph nodes within the context of GNN design. This is often facilitated by leveraging either a polynomial or a specific order of the Laplacian matrix, effectively enabling a form of random walk characterized by space-based long-range interactions. Such k-hop engagements have the potential to address challenges like over-squashing. The work on the fractional graph Laplacian by [48] employs a real order of the graph Laplacian to mitigate the oversmoothing issue, necessitating the use of singular value decomposition (SVD). A foundational comparison between the fractional graph Laplacian and the Mellin-transformed d-path Laplacian operator is provided in [51]. The study reveals that path-based diffusion consistently exhibits a reduced average commute time, potentially indicating diminished oversquashing, as elucidated by [52, Theorem 5.5]. The significant computational overhead introduced by SVD poses challenges to its applicability to large-scale graphs.

In parallel, the incorporation of various skip or dense connections across layers, as evidenced in [53–56], adopts diverse memory utilization strategies. These strategies can, to an extent, be conceptualized as the discretization of certain fractional differential equations (FDEs). A notable contribution in [50] introduces an innovative layer-dependent rewiring mechanism, progressively encompassing high-order neighbors. Their approach, which establishes a skip connection from the current layer back to a preceding one, embodies a unique form of memory utilization. This contrasts with the memory mechanisms explored in our study.

Our work distinctively frames a graph neural FDE approach, marked by its intrinsic nonlocal dynamics, both temporally (across layers) and spatially (within the graph). Note that the temporal domain in our paper refers to the "time" over which node feature evolves, different from the temporal domain in spatio-temporal GNNs like [16, 17].

A.1.2 Fractional Calculus and Deep Learning

The field of fractional calculus has seen a notable surge in interest recently due to its wide-ranging applications across various domains. These include but are not limited to, numerical analysis [57], viscoelastic materials [58], population growth models [59], control theory [60], signal processing [61], financial mathematics [62], and particularly in the representation of porous and fractal phenomena [63–65]. Within these contexts, fractional-order differential equations have been developed as a powerful extension to the conventional integer-ordered differential equations, offering a valuable mathematical tool for system modeling [66].

In the landscape of deep learning, [67] introduced an innovative approach for GNN parameter optimization via fractional derivatives. This deviates from the traditional use of integer-order derivatives in optimization algorithms such as SGD or Adam [68]. However, the focus of [67] is fundamentally different from our problem formulation. While [67] uses fractional derivatives for gradient optimization, our emphasis is on the *fractional-derivative evolution of node embeddings*. In another vein, [69] draws from fractional calculus, specifically the L1 approximation of the fractional derivative, to design a densely connected neural network. This design seeks to effectively manage non-smooth data and counter the vanishing gradient problem. Our work is different as we introduce fractional calculus into graph ODE models for evolving node embeddings, making use of its non-Markovian dynamic process nature.

From the vantage of physics-informed machine learning, there exists a research trajectory dedicated to the formulation of neural networks anchored in physical principles, specifically tailored for solving fractional PDEs. A trailblazing contribution in this sphere is the Fractional Physics Informed Neural Networks (fPINNs) [70]. Subsequent explorations, including [71,72], have expanded in this trajectory. We emphasize that these endeavors are distinctly different from our proposed methodology.

Our paper introduces a graph fractional differential equation framework to update graph node features, positioning our research distinctly from the aforementioned works.

A.1.3 Over-squashing

The paper by [73] offers a geometric perspective on the bottleneck and over-squashing phenomena in Message Passing Neural Networks. Introducing the novel Balanced Forman curvature, they establish

that edges with negative curvature play a role in bottleneck formation, subsequently leading to over-squashing. Building on this understanding, they present a curvature-informed approach for graph rewiring. In [52], it is observed that increasing the network's width can mitigate over-squashing but heightens the network's sensitivity. On the other hand, enhancing the depth does not alleviate over-squashing and can result in vanishing gradient issues. They underscore that graph topology is the predominant factor in over-squashing, emphasizing its prevalence between nodes with extended commute times. The paper [74] adopts the total effective resistance as a metric to evaluate over-squashing in GNNs. The authors further expand their work by introducing an advanced rewiring algorithm, aimed at reducing the total effective resistance through the strategic addition of edges to the graph. *In contrast, our research harnesses the nonlocal characteristic of the Mellin-transformed d-path Laplacian operator, $\mathbf{L}_s$, as a novel technique to tackle the over-squashing challenge — a perspective not investigated by the aforementioned studies.*

A.1.4 Graph Neural ODEs

Drawing inspiration from the pioneering work of [8], which reinterprets neural networks using the framework of ODEs, subsequent studies like [75–77] have ventured into the domain of graph neural ODEs. This paradigm treats GNNs as dynamical systems, wherein the ODE function can be instantiated through various architectures, including GCNs [78], GATs [79], and even Transformers [80]. Notably, graph neural ODEs inherit distinctive properties from dynamical system theory. For instance, they exhibit stability [13] and present a promising avenue to counteract the over-smoothing challenge [35]. In general, we have

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = \mathcal{F}(\mathbf{W}, \mathbf{X}(t)). \tag{6}$$

In this representation, $\mathbf{X}(t)$ signifies the evolving node features, and $\mathbf{W}$ is the adjacency matrix of the graph. $\mathcal{F}$ serves as the dynamical ODE function tailored for graphs and can be instantiated by varied GNN architectures such as GCN or GAT. It is worth noting that higher-order graph ODE models like GraphCON [11] can be equivalently expressed in the first order by scaling or extending the node dimension. *Our methodology uniquely utilizes non-integer ordered FDEs, pushing boundaries beyond the conventional integer-order ODEs.*

B DD and Protein Datasets

Both DD and Protein Datasets are from TUDdataset [81] where the topology of the graphs in relation to the graph classification tasks [82, 83] has been identified to require long-range interactions. The statistics of these two datasets are shown in Table 3.

Dataset	# Graphs	# Feature	# Classes	Avg. # Nodes	Avg. # Edges
DD	1178	89	2	384.3	715.7
Proteins	1113	3	2	39.1	72.8

Table 3: Statistics of the datasets

C Preliminaries and Framework

C.1 Temporal Dynamics with Graph Neural FDEs

Motivated by the unique modeling capabilities of FDEs [84] in elucidating physical phenomena — especially their adeptness at encapsulating memory over the temporal landscape and hereditary traits across diverse materials and processes beyond traditional ODEs [85] — we extend graph neural ODEs to the model:

$${}_{\mathrm{M}}D^{\beta}\mathbf{X}(t) = \mathcal{F}(\mathbf{W}, \mathbf{X}(t)), \tag{7}$$

where $\beta \in (0, 1]$ denotes the order of the fractional derivative. Interestingly, the function on the right-hand side, $\mathcal{F}(\mathbf{W}, \mathbf{X}(t))$, maintains its structure as in (6). By doing so, we exploit the inherent temporal nonlocality and extensive range dependency features of fractional derivatives [86], rendering

this approach potent for modeling intricate graph-structured data with nuanced temporal dynamics. While multiple definitions exist for fractional derivatives, for mathematical clarity and elegance, we adopt the *Marchaud–Weyl* fractional derivative ${}_{\mathrm{M}}D^{\beta}$, primarily for its effectiveness in capturing the fading memory phenomena [18–20]. The parameter β serves to quantify the degree of memory (i.e., long-range interaction in time) implicated in the feature evolution dynamics.

Definition 3 (Marchaud–Weyl Fractional Derivative)**.** *The* Marchaud–Weyl *fractional derivative of a scalar function f, defined over the real numbers and subjected to particular assumptions, at a specified point t, is given as:*

$$ {}_{\mathrm{M}}D^{\beta}f(t) = \frac{\beta}{\Gamma(1-\beta)} \int_0^{\infty} \frac{f(t) - f(t-\tau)}{\tau^{1+\beta}} \, \mathrm{d}\tau, \tag{8} $$

where $\Gamma(\cdot)$ denotes the Gamma function. For functions that are sufficiently smooth, according to [22], we have

$$ \lim_{\beta \to 1^-} {}_{\mathrm{M}}D^{\beta}f(t) = \frac{\mathrm{d}f(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{f(t+\Delta t) - f(t)}{\Delta t}. \tag{9} $$

It is seen from (8) that the *Marchaud–Weyl* fractional derivative, a nonlocal operator, accounts for the past values of f within the range (∞, t), indicative of its temporal memory effect. In the language of probability, a non-Markovian process' evolution depends not only on the current state but also on its historical states. As $\beta \to 1^-$ (limit from the left) in (9), the operator reverts to the traditional first-order derivative, representing the local rate of change of the function with respect to time, considering only the infinitesimally small neighborhood around the point of interest. Correspondingly, the non-Markovian process may degenerate to a Markovian process, as discussed in Appendix C.2. For a vector-valued function, the fractional derivative is defined component-wise for each dimension, similar to the first-order derivative.

C.2 Non-Markovian Random Walk Interpretation

This subsection elucidates the significance of fractional-order derivatives in the context of one-dimensional heat diffusion, underlining their deep association with memory-decaying random walks [20]. Consider a scenario where a random walker traverses the x-axis, moving within infinitesimal intervals of space $\Delta x > 0$ and time $\Delta \tau > 0$. The walker moves a distance of Δx from the starting point x in either direction with equal probability and pauses at each location for a random period of time. This introduces randomness in the waiting times between steps, echoing observations from several physical experiments [87]. Our objective is to determine $u(x, t)$, which represents the likelihood/concentration of the walker being at position x at a given time t. The waiting time distribution, symbolized as $\psi_\beta(\tau)$, is shaped by a power-law function $d_\beta n^{-(1+\beta)}$, with $d_\beta > 0$ set such that $\sum_{n=1}^{\infty} \psi_\beta(n) = 1$. The law of total probability can be articulated as:

$$ u(x,t) = \sum_{n=1}^{\infty} \left[\frac{1}{2} u(x - \Delta x, t - n\Delta\tau) + \frac{1}{2} u(x + \Delta x, t - n\Delta\tau) \right] \psi_\beta(n). $$

Within this formulation, the terms enclosed in brackets signify the likelihood of the walker reaching position x from its neighboring locations, either $x - \Delta x$ or $x + \Delta x$, each at a probability of $1/2$. The summation across n encapsulates scenarios where the walker might have been stationary for a duration of $t - n\Delta\tau$, influenced by the waiting time probability $\psi_\beta(n)$. Inserting the expression for $\psi_\beta(n)$, we get:

$$ \sum_{n=1}^{\infty} \frac{u(x,t) - u(x, t - n\Delta\tau)}{(n\Delta\tau)^{1+\beta}} (\Delta\tau) = \frac{(\Delta x)^2}{2 d_\beta (\Delta\tau)^{\beta}} \sum_{n=1}^{\infty} \delta_2 u(x, t - n\Delta\tau) \psi_\beta(n). $$

Here, the second-order incremental quotient is articulated as:

$$ \delta_2 u(x,t) = \frac{u(x - \Delta x, t) + u(x + \Delta x, t) - 2u(x,t)}{(\Delta x)^2}. $$

In the limit as $\Delta x, \Delta\tau \to 0$ and assuming that $\frac{(\Delta x)^2}{d_\beta(\Delta\tau)^\beta} \to k_\beta|\Gamma(-\beta)|$, we obtain the time-fractional diffusion equation:

$$ {}_{\mathrm{M}}D^{\beta}u = \frac{k_\beta}{2}u_{xx}. \tag{10}$$

As the value of β approaches 1^-, this intricate non-Markovian random walk with attenuating memory converges to the simpler Markovian counterpart, thus negating memory influences. As a result, (10) seamlessly transitions to the canonical heat diffusion equation with integer-order time derivative when $\beta = 1$:$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = \frac{k_1}{2}u_{xx}$.

C.3 Space-Fractional Operator: Path Laplacian

Definition 4 (Mellin-transformed d-path Laplacian Operator)**.** *The Mellin-transformed d-path Laplacian operator in $L^2(\mathcal{V})$ is defined as*

$$\big(\mathbf{L}_s f\big)(i) := \sum_{w\in\mathcal{V}:d(i,j)=d_{ij}} \frac{f(i)-f(j)}{(d_{ij})^s}, \tag{11}$$

where $f \in L^2(\mathcal{V})$, d_{ij} is the shortest path distance between node i and node j, and $0 \le s \le \infty$ represents the nonlocal parameter. Additionally, the Mellin-transformed d-path Laplacian can be defined as a matrix form:

$$\mathbf{L}_s := \mathbf{D}_s - \mathbf{A}_s \tag{12}$$

where $\mathbf{A}_s = [a_{ij}(s)]_{|\mathcal{V}|\times|\mathcal{V}|}$ is a d-path adjacency matrix by taking the shortest path distance d_{ij} into consideration with

$$a_{ij}(s) := \begin{cases} (d_{ij})^{-s} & \text{if } i \neq j, \\ 0 & \text{if } i = j, \end{cases} \tag{13}$$

and $(-s)$ represents the negative entrywise power, and $\mathbf{D}_s$ is the node degree matrix defined as:

$$\mathbf{D}_s := \operatorname{diag}(\mathbf{A}_s\mathbf{1}) \tag{14}$$

which is a diagonal matrix with $(D_s)_{ii} = \sum_j a_{ij}(s)$. Here $\mathbf{1}$ denotes the all-one vector. Furthermore, the normalized Mellin-transformed d-path Laplacian can be further defined as $\widetilde{\mathbf{L}}_s := \mathbf{I} - \widetilde{\mathbf{A}}_s$ with $\widetilde{\mathbf{A}}_s = \mathbf{A}_s(\mathbf{D}_s)^{-1}$.

Moreover, the Mellin-transformed d-path Laplacian operator, $\mathbf{L}_s$, can be expressed as:

$$\mathbf{L}_s = \sum_{d=1}^{\Delta} d^{-s}\mathbf{L}_d = \mathbf{L} + \sum_{d=2}^{\Delta} d^{-s}\mathbf{L}_d \tag{15}$$

where Δ is the diameter of the graph, and $\mathbf{L}_d$ is the vanilla d-path Laplacian operator, defined as:

$$\mathbf{L}_d f(i) := \sum_{w\in\mathcal{V}:d(i,j)=d_{ij}} f(i)-f(j), \quad f \in L^2(\mathcal{V}). \tag{16}$$

and $\mathbf{L}$ is the standard Laplacian operator:

$$\mathbf{L}f(v) := \sum_{(v,w)\in E} f(v)-f(w), \quad f \in L^2(\mathcal{V}). \tag{17}$$

It is evident that setting $\Delta = 1$ results in graph nodes connecting only with their immediate neighbors, negating any long-range interactions via the standard Laplacian operator $\mathbf{L}$. When $\Delta > 1$, graph nodes are influenced not just by adjacent neighbors but also by space-determined long-range interactions, and the coefficients d^{-s} determine their interactions' decay, mirroring the power law of path-length d. As $s \to \infty$, the Mellin-transformed d-path Laplacian operator, $\mathbf{L}_s$, converges to the standard Laplacian operator, $\mathbf{L}$.

D Fractional Graph Random Walk with Memory and Long range Interaction

Theorem 2. *Given a specific $\beta \in (0,1)$ and as $\Delta\tau \to 0$, we have that $\mathbb{P}(t;\beta)$ solves* (5)*, i.e.,*

$$\lim_{\Delta\tau\to 0}\left\{{}_{\mathrm{M}}D^{\beta}\mathbb{P}(t;\beta)+\mathbf{L}_s\mathbb{P}(t;\beta)\right\}=0.$$

Remark 3. *At its core, this type of random walk is non-Markovian, underscoring the importance of the entire temporal history of the walk (temporal long-range iteration) with spatially long-range iterations at the same time. In contrast to traditional graph diffusion GNNs [9,10] which correspond to $\beta = 1$ and assume transitions between node states to be Markovian (where future states depend only on the present state), GRAFT accommodates non-Markovian dynamics, where future states depend on a continuum of past states. At the same time, GRAFT also takes space-based long-range interactions between the pairs of nodes with strength $\frac{(d_{ij})^{-s}}{\sum_j (d_{ij})^{-s}}$ into consideration thanks to the d-path Laplacian, which facilitates transitions between node states to be nonlocal. This approach enables GRAFT to model more intricate dependencies, achieve richer representations both historically and spatially, and potentially enhance predictive performance. The non-Markovian nature and space-based long-range interactions are also evident in the numerical solution to GRAFT compared to ODE solvers used in GRAND.*

Recall the law of total probability for the random walk is expressed as:

$$\begin{aligned}\mathbb{P}_j(t;\beta)=&\sum_{n=1}^{\infty}\Bigg[\sum_{\substack{i\in\mathcal{V}\\ i\neq j}}\mathbb{P}_i(t-n\Delta\tau;\beta)(\Delta\tau)^{\beta}d_{\beta}|\Gamma(-\beta)|\frac{(d_{ij})^{-s}}{\sum_j(d_{ij})^{-s}}\\ &+\mathbb{P}_j(t-n\Delta\tau;\beta)\left(1-(\Delta\tau)^{\beta}d_{\beta}|\Gamma(-\beta)|\right)\Bigg]\psi_{\beta}(n).\end{aligned}\tag{18}$$

Proof. With notice of $\sum_{n=1}^{\infty}\psi_{\beta_0}(n) = 1$, we set $\beta = \beta_0$ in (18) and subtract $\sum_{n=1}^{\infty}\psi_{\beta_0}(n)\mathbb{P}_j(t-n\Delta\tau;\beta_0)$ from both sides of (18), then (18) yields

$$\begin{aligned}&\sum_{n=1}^{\infty}\left(\mathbb{P}_j(t;\beta_0)-\mathbb{P}_j(t-n\Delta\tau;\beta_0)\right)\psi_{\beta_0}(n)\\ =&(\Delta\tau)^{\beta_0}d_{\beta_0}|\Gamma(-\beta_0)|\sum_{n=1}^{\infty}\Bigg[\sum_{\substack{i\in\mathcal{V}\\ i\neq j}}\mathbb{P}_i(t-n\Delta\tau;\beta_0)\frac{(d_{ij})^{-s}}{\sum_j(d_{ij})^{-s}}\\ &-\mathbb{P}_j(t-n\Delta\tau;\beta_0)\Bigg]\psi_{\beta_0}(n)\\ =&(\Delta\tau)^{\beta_0}d_{\beta_0}|\Gamma(-\beta_0)|\sum_{n=1}^{\infty}\left[-\widetilde{\mathbf{L}}_s\mathbb{P}(t-n\Delta\tau;\beta_0)\right]_j\psi_{\beta_0}(n).\end{aligned}$$

Divide both sides by $(\Delta\tau)^{\beta_0}d_{\beta_0}|\Gamma(-\beta_0)|$, we have

$$\frac{1}{|\Gamma(-\beta_0)|}\sum_{n=1}^{\infty}\frac{\mathbb{P}_j(t;\beta_0)-\mathbb{P}_j(t-n\Delta\tau;\beta_0)}{(n\Delta\tau)^{1+\beta_0}}\Delta\tau=\sum_{n=1}^{\infty}\left[-\widetilde{\mathbf{L}}_s\mathbb{P}(t-n\Delta\tau;\beta_0)\right]_j\psi_{\beta_0}(n).$$

Let $\Delta\tau \to 0$ and switch the limit and the summation according to dominated convergence theorem (we assume the conditions are satisfied), we have

$$\frac{1}{|\Gamma(-\beta_0)|}\int_0^{\infty}\frac{\mathbb{P}_j(t;\beta_0)-\mathbb{P}_j(t-\tau;\beta_0)}{\tau^{1+\beta}}\,\mathrm{d}\tau=\left[-\widetilde{\mathbf{L}}_s\mathbb{P}(t;\beta_0)\right]_j.$$

Since $\Gamma(1-\beta)=\beta\Gamma(-\beta)$, according to (8), we have

$${}_{\mathrm{M}}D^{\beta_0}\mathbb{P}(t;\beta_0)=-\widetilde{\mathbf{L}}_s\mathbb{P}(t;\beta_0).$$

The proof is now complete. □

E Solving GRAFT

The conventional graph diffusion approach (6) discussed in [9, 10, 13] aligns the time parameter t with GNN layers, echoing the neural ODEs' portrayal as uninterrupted residual networks [8]. In many neural ODE solvers, time discretization is vital. The explicit Euler method, for instance, reduces neural ODEs to residual networks [8]. Despite the accuracy of adaptive step size solvers, they are resource-intensive [88]. In our GRAFT solution, we leverage the *Caputo* fractional derivative ${}_{\mathrm{C}}D^{\beta}$ is utilized as:

$$ {}_{\mathrm{C}}D^{\beta}\mathbf{X}(t) = \mathcal{F}(\mathbf{W}, \mathbf{X}(t)), \tag{19} $$

where the dynamic function $\mathcal{F}$ can be either $-\mathbf{L}_s\mathbf{X}(t)$ in (5). To derive numerical solvers for GRAFT, we address the complexity of fractional-order differential equations, differing from previous time discretization methods. Drawing from [89], we employ the *fractional Adams–Bashforth–Moulton method* and use an initial numerical solver termed "predictor" with time discretization given by $t_j = jh$, where h is a small positive increment.

$$ \mathbf{X}(t_n) = \mathbf{X}(0) + \frac{1}{\Gamma(\beta)} \sum_{j=0}^{n-1} \mu_{j,n} \mathcal{F}(\mathbf{W}, \mathbf{X}(t_j)), \tag{20} $$

where $\mu_{j,n} = \frac{h^{\beta}}{\beta}\left((n-j)^{\beta} - (n-1-j)^{\beta}\right)$ and $h = t_n - t_{n-1}$ is the time discretisation. At each time t_n, the node feature vector $\mathbf{X}(t_n)$ is influenced through spaced-based long-range interactions with $-\mathbf{L}_s\mathbf{X}(t)$ and the formulation of the node feature $\mathbf{X}(t_n)$ utilizes the full temporal memory $\{\mathbf{X}(t_j)\}_{j=0}^{n-1}$, which reflects the time-space-based long range iterations at the same time. The visualization of information flow in this discretization in shown in Fig. 1.

Remark 4. *When $\beta = 1$, this method simplifies to the Euler solver in [8, 9] as $\mu_{j,n} \equiv h$, yielding $\mathbf{X}(t_n) = \mathbf{X}(t_{n-1}) + h\mathcal{F}(\mathbf{W}, \mathbf{X}(t_{n-1}))$. Thus, the solver shown in* (20) *can be considered as the fractional Euler method or fractional Adams–Bashforth method, which is a generalization of the Euler method used in [8, 9].*