Coupling Graph Neural Networks with Fractional Order Continuous Dynamics: A Robustness Study

Qiyu Kang[1]*, Kai Zhao[1]*†, Yang Song[2], Yihang Xie[1], Yanan Zhao[1], Sijie Wang[1], Rui She[1], Wee Peng Tay[1]

[1]Nanyang Technological University
[2]C3 AI, Singapore

Abstract

In this work, we rigorously investigate the robustness of graph neural fractional-order differential equation (FDE) models. This framework extends beyond traditional graph neural (integer-order) ordinary differential equation (ODE) models by implementing the time-fractional Caputo derivative. Utilizing fractional calculus allows our model to consider long-term memory during the feature updating process, diverging from the memoryless Markovian updates seen in traditional graph neural ODE models. The superiority of graph neural FDE models over graph neural ODE models has been established in environments free from attacks or perturbations. While traditional graph neural ODE models have been verified to possess a degree of stability and resilience in the presence of adversarial attacks in existing literature, the robustness of graph neural FDE models, especially under adversarial conditions, remains largely unexplored. This paper undertakes a detailed assessment of the robustness of graph neural FDE models. We establish a theoretical foundation outlining the robustness characteristics of graph neural FDE models, highlighting that they maintain more stringent output perturbation bounds in the face of input and graph topology disturbances, compared to their integer-order counterparts. Our empirical evaluations further confirm the enhanced robustness of graph neural FDE models, highlighting their potential in adversarially robust applications.

1 Introduction

Graph Neural Networks (GNNs) (Kipf and Welling 2017; Veličković et al. 2018; Ji et al. 2023; Lee, Ji, and Tay 2022; She et al. 2023) have emerged as an influential tool capable of extracting meaningful representations from intricate datasets, such as social networks (Huang et al. 2021) and molecular structures (Guo et al. 2023). Despite their impressive capability, GNNs have been found susceptible to adversarial attacks (Dai et al. 2018; Ma, Ding, and Mei 2020; Zügner, Akbarnejad, and Günnemann 2018), with modifications or injections into the graph often causing significant degradation in performance. In real-world scenarios, it is common for data to be perturbed during the training or testing phases (Dai et al. 2023; Wang et al. 2019), highlighting the importance of studying the robustness of GNNs. For instance, in financial systems, fraudulent activities may introduce slight perturbations into transactional data, making it paramount for the underlying models to remain robust against these adversarial changes. Similarly, in social networks, misinformation or the presence of bots can skew the data, which can subsequently impact the insights drawn from it. Therefore, the robustness of GNNs is not just a theoretical concern but a practical necessity. Several defensive strategies have been established to counteract the damaging implications of adversarial attacks on graph data. Approaches such as GARNET (Deng et al. 2022), GNN-Guard (Zhang and Zitnik 2020), RGCN (Zhu et al. 2019), and Pro-GNN (Jin et al. 2020) are grounded in preprocessing techniques that aim to remove adversarial alterations to the structure before GNN training commences. Nonetheless, these methods often necessitate the exploration of graph structure properties, leading to higher computational costs. Furthermore, these strategies are more suitably tailored to combat poisoning attacks.

Recent advances have witnessed a growing use of dynamical system theory in designing and understanding GNNs. Models like CGNN (Xhonneux, Qu, and Tang 2020), GRAND (Chamberlain et al. 2021b), GRAND++ (Thorpe et al. 2021), GraphCON (Rusch et al. 2022b), HANG (Zhao et al. 2023a) and CDE (Zhao et al. 2023b) employ ordinary differential equations (ODEs) to offer a dynamical system perspective on graph node feature evolution. Typically, these dynamics can be described by:

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = \mathcal{F}(\mathbf{W}, \mathbf{X}(t)). \tag{1}$$

In this formulation, $\mathbf{X}(t)$ represents the evolving node features with $\mathbf{X}(0)$ as the initial input node features, while $\mathbf{W}$ is the graph's adjacency matrix. The function, $\mathcal{F}$, is specifically tailored for graph dynamics. As a case in point, GRAND (Chamberlain et al. 2021b) deploys an attention-based aggregation mechanism akin to heat diffusion on the graph. Motivated by the Beltrami diffusion equation (Sochen, Kimmel, and Malladi 1998), the paper (Song et al. 2022) introduces a model based on the Beltrami flow (abbreviated as GraphBel) and designed for enhanced robustness, particularly in the face of topological perturbations. In the study (Zhao et al. 2023a), graph feature updates are conceptualized as a Hamiltonian flow, endowed with Lyapunov stability, to effectively counter

*These authors contributed equally.
†Correspondence to: Kai Zhao <kai.zhao@ntu.edu.sg>

adversarial perturbations. GraphCON (Rusch et al. 2022b) presents a approach by introducing a second-order graph coupled oscillator for modeling feature updates. This model can be decomposed into two first-order equations, aligning with the principle that higher integer-order ODEs can be expressed as a system of first-order ODEs through auxiliary variables, effectively encapsulated in (1).

Recent studies have ventured into the intersection of GNNs and fractional calculus (Diethelm and Ford 2010; Kang et al. 2023c). One prominent example is the FRactional-Order graph Neural Dynamical network (FROND) framework (Kang et al. 2023a). Distinct from conventional graph neural ODE models, FROND leverages fractional-order differential equations (FDEs), with dynamics represented as:

$$D_t^\beta \mathbf{X}(t) = \mathcal{F}(\mathbf{W}, \mathbf{X}(t)),\ \beta > 0. \tag{2}$$

The function $\mathcal{F}(\mathbf{W}, \mathbf{X}(t))$ maintains its form as in (1). Typically, we set $\beta \in (0, 1]$. The Caputo fractional derivative, denoted by D_t^β, infuses memory into the temporal dynamics (see Section 3.3 for more details). For $\beta = 1$, the equation reverts to the familiar first-order dynamics as in (1). 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 is influenced by the entire historical trajectory of the function,* which substantially diverges from the localized impact found in integer-order derivatives.

Incorporating a fractional derivative provides GNNs an avenue to mitigate the prevalent oversmoothing problems by enabling slow algebraic convergence (Kang et al. 2023a), different from the standard fast exponential convergence. Further, with the integration of fractional dynamics, FROND can effortlessly merge with existing graph neural ODE frameworks, potentially increasing their effectiveness, especially with diverse β values, without incorporating any additional training parameters to the underlying graph neural ODE models. Critically, β acts as a proxy for the extent of memory in the feature dynamics: a value of $\beta = 1$ corresponds to memoryless Markovian dynamics, while $\beta < 1$ denotes non-Markovian dynamics with memory. This nuance is further visualized in Fig. 1, where a $\beta < 1$ signifies nontrivial dense connections across model discretization timestamps.

Though FROND showcases proficiency in decoding complex graph data patterns, its robustness against adversarial perturbations remains an area of exploration. By broadening the order of time derivatives from integers to real numbers, fractional calculus can encapsulate more intricate dynamics and data relationships, such as long-range memory effects, where the system's current state is influenced by its comprehensive history, not merely its recent states. This capability augments a GNN's ability to more accurately represent the node features across layers, rendering them less susceptible to noise and perturbations. In this work, we delve deeply into the ramifications of the fractional order parameter β on the robustness attributes of FROND. Our analysis suggests a monotonic relationship between the model's perturbation bounds and the parameter β, with smaller β values indicating augmented robustness.

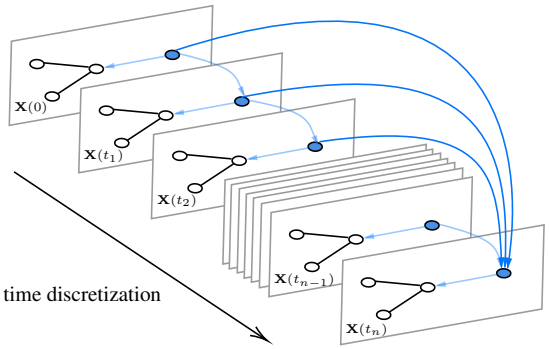


Figure 1: Model discretization in FROND. Unlike the Euler discretization in graph neural ODE models, FROND incorporates connections to historical times, introducing memory effects. Specifically, the dark blue connections observed in FROND at $\beta < 1$ are absent in ODEs (corresponding to $\beta = 1$). The weight of these skip connections correlates with $b_{j,k+1}(\beta)$ as detailed in (12).

Our contributions are summarized as follows:

- We rigorously investigate the robustness characteristics of graph neural FDE models, i.e., FROND models. We show that FROND models exhibit tighter output perturbation bounds compared to their integer-order counterparts in the presence of input and topology perturbations.
- Through extensive experimental evaluations, including graph modifications and injection attacks, we empirically demonstrate the superior robustness of FROND models in contrast to conventional graph neural ODE models.

2 Related Work

2.1 Graph Neural ODE Models

The interplay between ODEs and neural networks has recently shed light on the potential of continuous dynamical systems in deep learning frameworks. This concept is initially explored in the work of (Weinan 2017). A landmark study by (Chen et al. 2018b) further cements this notion, presenting neural ODEs equipped with open-source solvers. This methodology allows for a more precise alignment of the inputs and outputs of neural networks with established physical principles, thereby increasing the networks' interpretability. This field continues to evolve, with notable advancements in enhancing neural network efficiency (Dupont, Doucet, and Teh 2019), bolstering robustness (Yan et al. 2018; Kang et al. 2021), and stabilizing gradient functions (Haber and Ruthotto 2017). In parallel, (Avelar et al. 2019; Poli et al. 2021) demonstrate the utilization of continuous residual GNN layers, leveraging neural ODE solvers to optimize output. Recently, GraphCON (Rusch et al. 2022b) has implemented the coupled oscillator model, which effectively maintains the Dirichlet energy of graphs over time, addressing the prevalent issue of over-smoothing in these networks. The diffusion theory, as conceptualized in (Chamberlain et al.

2021b), likens information propagation to the diffusion process of substances. The Beltrami diffusion models, as utilized in (Chamberlain et al. 2021a; Song et al. 2022), have been pivotal in improving the rewiring and robustness of graphs. Concurrently, ACMP (Wang et al. 2022) draws inspiration from particle reaction-diffusion processes, accounting for both repulsive and attractive interactions among particles. The graph CDE model, as outlined in (Zhao et al. 2023b), addresses heterophilic graph challenges, inspired by convection-diffusion processes. Similarly, GREAD (Choi et al. 2023) introduces an approach based on reaction-diffusion equations, tailored to effectively manage heterophilic datasets. GRAND++ (Thorpe et al. 2021) employs heat diffusion with sources for training models more efficiently, especially when there is a scarcity of labeled data. Further enriching this field, recent research (Zhao et al. 2023a; Kang et al. 2023b) adopts the Hamiltonian mechanism for updating node features, thereby augmenting the networks' adaptability to graph structures and enhancing their robustness.

2.2 Adversarial Attacks and Defenses on Graphs

A plethora of research has consistently underscored the vulnerability of graph deep learning models to adversarial perturbations. Essentially, even inconspicuous alterations to the input data can misdirect a graph neural network into producing fallacious predictions. Adversarial attacks on GNNs typically fall into two categories based on the method of perturbation: Graph Modification Attacks (GMA) and Graph Injection Attacks (GIA). GMA involves manipulating the topology of a graph, primarily by adding or removing edges (Chen et al. 2018a; Waniek et al. 2018; Du et al. 2018; Ma et al. 2021; Geisler et al. 2021). This category also encompasses perturbations to node features (Zügner, Akbarnejad, and Günnemann 2018; Zügner and Günnemann 2019; Ma et al. 2021; Ma, Deng, and Mei 2022; Finkelshtein et al. 2022). In contrast, Graph Injection Attacks (GIA) permit adversaries to incorporate malicious nodes into the original graph (Wang et al. 2020; Zou et al. 2021; Sun et al. 2020; Hussain et al. 2022; Chen et al. 2022). GIA is considered a stronger form of attack on graph data (Chen et al. 2022) as it introduces both structural and feature perturbations to the graph.

The defensive strategies employed in GNNs can be broadly categorized into pre-processing methods and the design of robust architectures. Methods such as GNN-GUARD (Zhang and Zitnik 2020), Pro-GNN (Jin et al. 2020),GARNET(Deng et al. 2022) and GCN-SVD (Entezari et al. 2020) focus on cleansing or pruning the graph, with the aim of maintaining the integrity of the original adjacency matrix, thereby mitigating perturbations. On another front, methods like RGCN (Zhu et al. 2019) and Soft-Median-GCN (Geisler et al. 2021) are tailored to enhance the inherent architecture of GNNs, making them more resilient to feature perturbations. Distinctly, our approach diverges from these conventional defense mechanisms. *Instead of proposing an entirely new defensive technique, our focus is on bolstering the robustness of existing graph neural ODE models by seamlessly integrating the principles of FDEs.*

3 Preliminaries

3.1 Notation

Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathbf{W})$, in which $\mathcal{V} = \{1, \ldots, N\}$ represents a set of N nodes. The $N \times N$ matrix $\mathbf{W} := (W_{ij})$ has elements W_{ij} indicating the original edge weight between the i-th and j-th nodes with $W_{ij} = W_{ji}$. The node features at any given time t can be denoted by $\mathbf{X}(t) \in \mathbb{R}^{|\mathcal{V}| \times N}$, where N corresponds to the dimension of the node feature. In this matrix, the feature vector for the i-th node in $\mathcal{V}$ at time t can be represented as the i-th row of $\mathbf{X}(t)$, indicated by $\mathbf{x}_i^\intercal(t)$.

3.2 Graph Neural ODE Models

Existing research encompasses various continuous dynamics-informed GNNs, with unique configurations of $\mathcal{F}$ in (1) tailored for graph dynamics. This section provides a succinct overview of several graph neural ODE models that we will employ in this study. For an extensive review of these and related GNNs, readers are referred to a recent comprehensive survey (Han et al. 2023).

GRAND (Chamberlain et al. 2021b) incorporates the following dynamical system for graph learning:

$$\begin{aligned} \frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} &= \mathrm{div}(D(\mathbf{X}(t), t) \odot \nabla \mathbf{X}(t)) \\ &= (\mathbf{A}(\mathbf{X}(t)) - \mathbf{I})\mathbf{X}(t) \end{aligned} \tag{3}$$

The initial condition $\mathbf{X}(0)$ is provided by the graph input features. Here, $\odot$ represents the element-wise product, and D is a diagonal matrix with elements $\mathrm{diag}(a(\mathbf{x}_i(t), \mathbf{x}_j(t)))$. The function $a(\cdot)$ serves as a measure of similarity for node pairs (i, j) linked by an edge, that is, when $W_{ij} \neq 0$. As such, the diffusion equation can be reframed as (3), where $\mathbf{A}(\mathbf{X}(t)) = (a(\mathbf{x}_i(t), \mathbf{x}_j(t)))$ constitutes a learnable attention matrix to depict the graph structure. $\mathbf{I}$ is the identity matrix. One way to calculate $a(\mathbf{x}_i, \mathbf{x}_j)$ is based on the Transformer attention (Vaswani et al. 2017):

$$a(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{softmax}\left(\frac{(\mathbf{W}_K \mathbf{x}_i)^\top \mathbf{W}_Q \mathbf{x}_j}{d_k}\right) \tag{4}$$

where $\mathbf{W}_K$ and $\mathbf{W}_Q$ are learned matrices, and d_k is a hyperparameter determining the dimension of $\mathbf{W}_K$.

By extending the concepts of Beltrami flow (Sochen, Kimmel, and Malladi 1998; Song et al. 2022), a stable graph neural flow GraphBel is formulated as:

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = (\mathbf{A}_\mathbf{S}(\mathbf{X}(t)) \odot \mathbf{B}_\mathbf{S}(\mathbf{X}(t)) - \Psi(\mathbf{X}(t)))\mathbf{X}(t) \tag{5}$$

where $\odot$ represents element-wise multiplication. Both $\mathbf{A}_\mathbf{S}(\cdot)$ and $\mathbf{B}_\mathbf{S}(\cdot)$ serve distinct purposes: the former acts as a learnable attention function, while the latter operates as a normalized vector map. $\mathbf{\Psi}(\mathbf{X}(t))$ is a diagonal matrix where $\Psi(\mathbf{x}_i, \mathbf{x}_i) = \sum_{\mathbf{x}_j} (\mathbf{A}_\mathbf{S} \odot \mathbf{B}_\mathbf{S})(\mathbf{x}_i, \mathbf{x}_j)$.

Using a graph coupled dynamical system, GraphCON (Rusch et al. 2022a) is given by

$$\begin{aligned} \frac{\mathrm{d}\mathbf{Y}(t)}{\mathrm{d}t} &= \sigma(\mathbf{F}_\theta(\mathbf{X}(t), t)) - \gamma\mathbf{X}(t) - \alpha\mathbf{Y}(t) \\ \frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} &= \mathbf{Y}(t) \end{aligned} \tag{6}$$

where $\mathbf{F}_\theta(\cdot)$ is a learnable 1-neighborhood coupling function, σ denotes an activation function, γ and α are adjustable parameters.

Remark 1. *By leveraging the numerical solvers introduced in (Chen et al. 2018b), one can efficiently solve* (3)*,* (5)*, and* (6) *where the initial* $\mathbf{X}(0)$ *represents the input features. This yields the terminal node embeddings, denoted as* $\mathbf{X}(T)$*, at time* T*. Subsequently,* $\mathbf{X}(T)$ *can be utilized for downstream tasks such as node classification or link prediction.*

3.3 Fractional-Order Differential Equation

Within the FROND framework, the fractional time derivative is typically characterized using the Caputo derivative (Caputo 1967), a prevalent choice for modeling real-world phenomena (Diethelm and Ford 2010). It is expressed as:

$$D_t^\beta f(t) = \frac{1}{\Gamma(n-\beta)} \int_0^t (t-\tau)^{n-\beta-1} \frac{\mathrm{d}^n f}{\mathrm{d}\tau^n} \, \mathrm{d}\tau, \quad (7)$$

where β is the fractional order, n is the smallest integer greater than β, Γ is the gamma function, f is a scalar function defined over some interval that includes $[0, t]$, and $\frac{\mathrm{d}^n f}{\mathrm{d}\tau^n}$ is the standard n-th order derivative. A distinguishing trait of the Caputo derivative is its capability to incorporate *memory effects*. This is underscored by observing that the fractional derivative at time t in (7) *aggregates historical states spanning the interval* $0 \leq \tau \leq t$. For the special case where $\beta = 1$, the definition collapses to the standard first-order derivative as $D_t^\beta f = \frac{\mathrm{d}f}{\mathrm{d}\tau}$. For a vector-valued function, the fractional derivative is defined component-wise for each dimension, similar to the integer-order derivative. Thus, while our discussion centers on scalar functions in Sections 3.3 and 3.4, its extension to vector-valued functions is straightforward. A more detailed, self-contained exposition of the Caputo derivative can be found in the supplementary material.

A crucial concept in fractional calculus and its applications is the Mittag-Leffler function $E_\beta(z)$ (Diethelm and Ford 2010). Recognized as a natural extension of the exponential function within fractional domains, it enables the modeling of complex phenomena with increased sophistication. The Mittag-Leffler function is a crucial component in the solutions to numerous fractional differential equations, thus playing an essential role in the analysis and application of such systems. Specifically, as per (Diethelm and Ford 2010)[Theorem 4.3], given $y(t) := E_n\left(\lambda t^\beta\right)$, $x \geq 0$, then

$$D_t^\beta y(t) = \lambda y(t). \quad (8)$$

We present the formal definition of the Mittag-Leffler function below:

Definition 1 (Mittag-Leffler function)**.** *Let* $\beta > 0$*. The function* E_β *defined by*

$$E_\beta(z) \coloneqq \sum_{j=0}^{\infty} \frac{z^j}{\Gamma(j\beta + 1)}, \quad (9)$$

whenever the series converges, is called the Mittag-Leffler function of order β*.*

The extension of the Mittag-Leffler function from the exponential function is apparent when considering the case $\beta = 1$, which simplifies to the well-known exponential function:

$$E_1(z) = \sum_{j=0}^{\infty} \frac{z^j}{\Gamma(j+1)} = \sum_{j=0}^{\infty} \frac{z^j}{j!} = \exp(z). \quad (10)$$

It is also well-recognized that $\exp(z)$ acts as the eigenfunction for ODEs. Specifically, $\exp(\lambda t)$ solves (8) for $\beta = 1$, assuming appropriate initial conditions are met.

3.4 Numerical Solvers for FDEs

In the context of discrete numerical solvers, FDEs can be solved analogously to ODEs as illustrated in (Chen et al. 2018b). Particularly noteworthy is the fractional Adams–Bashforth–Moulton method solver, which shares similarities with the Adams–Bashforth–Moulton technique for ODEs, as expounded in (Diethelm, Ford, and Freed 2004).

For clarity, consider an FDE characterized by $\beta \in (0, 1]$:

$$D_t^\beta y(t) = f(t, y(t)), \quad y(0) = y_0. \quad (11)$$

Here, $f(t, y(t))$ delineates the dynamics of the system, and y_0 specifies the initial condition at $t = 0$.

Delving into numerical approximations and drawing upon (Diethelm, Ford, and Freed 2004), the basic predictor solution y_{k+1}^P (where ^{P} signifies the concept of the "Predictor") derives from the fractional Adams–Bashforth method as:

$$y^P(t_{k+1}) = y_0 + \frac{1}{\Gamma(\beta)} \sum_{j=0}^{k} b_{j,k+1}(\beta) f(t_j, y_j). \quad (12)$$

In this context, k stands for the current iteration or time step in the discretization sequence. Further, with h denoting the step size or time interval between subsequent approximations, t_j is given by $t_j = hj$. The coefficients $b_{j,k+1}(\beta)$, expressed as functions of β, are elaborated in the supplementary material. The reader is directed to Fig. 1, where the role of $b_{j,k+1}(\beta)$ as a weighted skip connection in time discretization, underscoring memory effects, is evident.

4 Methodology

In this section, we present the theoretical analysis of the output boundary of (13) under specific perturbation scenarios. By leveraging the characteristics of the Mittag-Leffler function, we detail the response of the FROND, noting that it undergoes smaller output perturbations compared to the graph neural ODE framework when exposed to the same disturbances. We furnish three pivotal theorems underscoring the inherent resilience of the FROND paradigm:

- Theorem 1 (Diethelm and Ford 2010) establishes the output perturbation bounds of the FDEs under small perturbations in the initial conditions, which, in our case, correspond to input feature changes of FROND models.
- Theorem 2 (Diethelm and Ford 2010) extends the discussion to include perturbations in the function that governs the system's dynamics. In the context of graph learning, such perturbations include the changes in the topology of the graph, which can occur due to the addition, deletion, or modification of edges.

- Finally, Theorem 3 provides important insights into how the choice of the fractional order β can influence the system's robustness. Our analysis suggests a monotonic relationship between the model's perturbation bounds and the parameter β, with smaller β values indicating augmented robustness.

Together, these results provide a strong theoretical basis for the robustness of FROND, setting the stage for its deployment in various practical applications.

4.1 FROND: Graph Neural FDE Framework

Building upon the foundation of FDEs, recall that FROND incorporates the Caputo fractional-order time derivative into the model for feature evolution:

$$D_t^\beta \mathbf{X}(t) = \mathcal{F}(\mathbf{W}, \mathbf{X}(t)),\ \mathbf{X}(0) = \mathbf{X}_0,\ 0 < \beta \leq 1. \quad (13)$$

In equation (13), $D_t^\beta \mathbf{X}(t)$ represents the fractional derivative of the state $\mathbf{X}(t)$ with respect to feature evolution time t, where β is a real number in the interval $(0, 1]$. This fractional derivative introduces memory effects, which enrich the capacity of the model to interpret complex patterns in data. The term $\mathcal{F}(\mathbf{W}, \mathbf{X}(t))$ denotes the dynamic function modeling the interactions between nodes given graph topology $\mathbf{W}$. With the system initialized from input node features as $\mathbf{X}(0) = \mathbf{X}_0$, the model's output is $\mathbf{X}(T)$ at a specified time T.

One intrinsic characteristic of FROND is the long-memory property from fractional derivative. This property encapsulates the system's ability to "remember" its historical states. This inherent memory effect contributes significantly to the robustness of the system, particularly when faced with perturbations. When the system encounters disturbances or noise, the extensive memory of FROND serves as a protective buffer. It mitigates the immediate effects of these disruptions by integrating the system's past states into its response, rather than amplifying the disturbances.

4.2 Robustness of FROND under Perturbation

In this subsection, we delve into a theoretical analysis of the model output perturbations. We begin by highlighting two central theorems from (Diethelm and Ford 2010), which provide bounds on output perturbations, grounded in the properties of the Mittag-Leffler function. Following this, we analyze the bound outlined in Theorem 3, shedding light on the notion that a smaller β contributes to enhanced robustness of the model, particularly when faced with perturbations in input node features and graph topology.

Theorem 1. *(Diethelm and Ford 2010, Theorem 6.20) Let $\mathbf{X}(t)$ be the solution of the initial value problem* (13)*, and let $\tilde{\mathbf{X}}(t)$ be the solution of the initial value problem*

$$\begin{aligned} D_t^\beta \tilde{\mathbf{X}}(t) &= \mathcal{F}(\mathbf{W}, \tilde{\mathbf{X}}(t)), \\ \tilde{\mathbf{X}}(0) &= \tilde{\mathbf{X}}_0, \end{aligned} \quad (14)$$

where $\varepsilon := \|\mathbf{X}_0 - \tilde{\mathbf{X}}_0\|$. Then, if ε is sufficiently small, there exists some $h > 0$ such that both the functions $\mathbf{X}$ and $\tilde{\mathbf{X}}$ are defined on $[0, h]$, and

$$\sup_{0 \leq t \leq h} \|\mathbf{X}(t) - \tilde{\mathbf{X}}(t)\| = c_1 \varepsilon E_\beta(L h^\beta) \quad (15)$$

where L is the Lipschitz constant of $\mathcal{F}$ and c_1 is a constant.

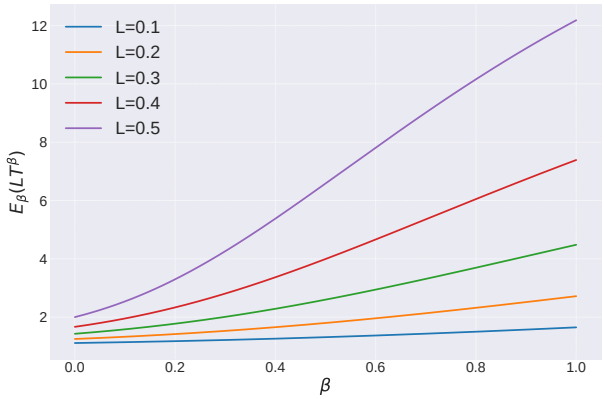


Figure 2: Plot of the Mittag-Leffler function $E_\beta(LT^\beta)$ against β with $T = 10$. Distinctively, for varying L, it displays monotonic increase over interval $[\epsilon, 1]$.

Remark 2. *In our FROND framework, Theorem 1 provides an upper bound for the perturbation in system trajectory. This encompasses the perturbation of the FROND output $\mathbf{X}(T)$ at time T if $T < h$, in situations of slight perturbations to the input features, specifically when the system's initial input features shift to $\tilde{\mathbf{X}}_0$.*

Theorem 2. *(Diethelm and Ford 2010, Theorem 6.21) Let $\mathbf{X}(t)$ be the solution of the initial value problem* (13)*, and let $\tilde{\mathbf{X}}(t)$ be the solution of the initial value problem*

$$\begin{aligned} D_t^\beta \tilde{\mathbf{X}}(t) &= \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \mathbf{X}(t)), \\ \tilde{\mathbf{X}}(0) &= \mathbf{X}_0 \end{aligned} \quad (16)$$

Moreover, let $\varepsilon := \sup_{\mathbf{X} \in A} \|\mathcal{F}(\mathbf{W}, \mathbf{X}) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \mathbf{X})\|$, with A being an appropriate compact set where solutions for both systems exist. Then, if ε is sufficiently small, there exists some $h > 0$ such that both the functions $\mathbf{X}$ and $\tilde{\mathbf{X}}$ are defined on $[0, h]$, and

$$\sup_{0 \leq t \leq h} \|\mathbf{X}(t) - \tilde{\mathbf{X}}(t)\| = c_2 \varepsilon E_\beta(L h^\beta), \quad (17)$$

where L is the Lipschitz constant of $\mathcal{F}$ and c_2 is a constant.

Remark 3. *Within the framework of FROND, the discussion relates to how the model's output reacts to perturbations in functional elements (such as learned parameters in $\mathcal{F}$) and changes in graph structure $\mathbf{W}$. In our paper, we consider topological changes in the graph structure, such as edge additions, deletions, or modifications. Further investigation into attacks related to functional perturbations that directly alter the parameters of the neural network $\mathcal{F}$ is earmarked for future work.*

Theorem 3. *Let $f(\beta) = E_\beta(LT^\beta)$. For any $\epsilon > 0$, if T is sufficiently large and $L < 1$, $f(\beta)$ is monotonically increasing on the interval $[\epsilon, 1]$.*

Dataset	Ptb(%)	F-GRAND	GRAND	F-GraphBel	GraphBel	F-GraphCON	GraphCON	GAT	GCN
Cora	0	81.25±0.89	82.24±1.82	79.05±0.73	80.28±0.87	80.91±0.54	83.10±0.79	**83.97±0.65**	83.50±0.44
	5	78.84±0.57	78.97±0.49	76.10±0.74	77.70±0.66	77.80±0.44	77.90±1.14	**80.44±0.74**	76.55±0.79
	10	**76.61±0.68**	75.02±1.25	74.03±0.47	74.30±0.88	74.63±1.42	72.53±1.08	70.39±1.28	70.39±1.28
	15	**73.42±0.97**	71.43±1.09	73.01±0.75	72.14±0.69	73.01±0.78	69.83±0.68	65.10±0.71	65.10±0.71
	20	69.27±2.10	60.53±1.99	**69.35±1.23**	65.41±0.99	69.23±1.35	57.28±1.62	59.56±2.72	59.56±2.72
	25	64.47±1.83	55.26±2.14	**67.63±0.93**	62.31±1.13	65.27±1.33	53.17±1.52	47.53±1.96	47.53±1.96
Citeseer	0	71.37±1.34	71.50±1.10	68.90±1.15	69.46±1.15	71.49±0.71	70.48±1.18	**73.26±0.83**	71.96±0.55
	5	71.47±0.96	71.04±1.15	68.36±0.93	68.45±1.02	70.77±1.15	69.75±1.63	**72.89±0.83**	70.88±0.62
	10	69.76±0.71	68.88±0.60	67.22±1.52	66.72±1.31	69.54±0.82	67.40±1.78	**70.63±0.48**	67.55±0.89
	15	67.94±1.42	66.35±1.37	63.56±1.95	63.63±1.67	67.37±0.87	65.78±1.97	**69.02±1.09**	64.52±1.11
	20	64.18±0.93	58.71±1.42	63.38±0.96	58.90±0.84	**66.52±0.68**	56.79±1.46	61.04±1.52	62.03±3.49
	25	65.46±1.12	60.15±1.37	64.60±0.48	61.24±1.28	**66.72±1.12**	57.30±1.38	61.85±1.12	56.94±2.09
Pubmed	0	**87.28±0.23**	85.06±0.26	86.34±0.15	84.02±0.26	87.12±0.21	84.65±0.13	83.73±0.40	87.19±0.09
	5	**87.05±0.17**	84.11±0.30	86.17±0.12	83.91±0.26	86.72±0.23	83.06±0.22	78.00±0.44	83.09±0.13
	10	**86.74±0.23**	84.24±0.18	86.01±0.18	84.62±0.26	86.64±0.20	82.25±0.12	74.93±0.38	81.21±0.09
	15	86.51±0.14	83.74±0.34	85.92±0.13	84.83±0.20	**86.40±0.14**	81.26±0.33	71.13±0.51	78.66±0.12
	20	**86.50±0.12**	83.58±0.20	85.73±0.18	84.89±0.45	86.32±0.12	81.58±0.41	68.21±0.96	77.35±0.19
	25	**86.47±0.15**	83.66±0.25	86.11±0.30	85.07±0.15	86.15±0.26	80.75±0.32	65.41±0.77	75.50±0.17

Table 1: Node classification accuracy (%) under modification, poisoning, non-targeted attack (Metattack) in transductive learning. The best and the second-best results for each criterion are highlighted in bold and underlined, respectively.

Proof. See the supplementary material for the proof. □

Remark 4. *In conjunction with Theorems 1 and 2, Theorem 3 shows that the fractional order β of the FROND plays a crucial role in the model's robustness. With an appropriately chosen β, the model can reduce the discrepancy between the clean and perturbed states, thereby improving the robustness. Particularly, a smaller β is associated with a smaller discrepancy, signifying enhanced robustness of FROND against perturbations when $\beta < 1$ compared to graph neural ODE models with $\beta = 1$. Please refer to Fig. 3 for an illustration. The monotonicity suggests that a larger β results in a larger perturbation bound for the FROND solution at time T, thus expecting a larger perturbed output at the same time under identical input/graph topology perturbations.*

4.3 Algorithms

Our proposed approach enhances the robustness of integer-order graph neural ODE models by introducing fractional-order derivatives into the model framework. Specifically, we extend three prominent graph neural ODE models, GRAND, GraphBel, and GraphCON through this method.

We upgrade the GRAND framework with a fractional-order derivative, resulting in the Fractional-GRAND (F-GRAND) model. The F-GRAND formulation is as follows:

$$D_t^\beta \mathbf{X}(t) = (\mathbf{A}(\mathbf{X}(t)) - \mathbf{I})\mathbf{X}(t). \tag{18}$$

Following a similar approach, the GraphBel model is modified to incorporate a fractional-order derivative, resulting in the Fractional-GraphBel (F-GraphBel) model. This model is expressed as:

$$D_t^\beta \mathbf{X}(t) = (\mathbf{A}_\mathbf{S}(\mathbf{X}(t)) \odot \mathbf{B}_\mathbf{S}(\mathbf{X}(t)) - \Psi(\mathbf{X}(t)))\mathbf{X}(t). \tag{19}$$

Additionally, we introduce the Fractional-GraphCON (F-GraphCON) model, described by the following equations:

$$\begin{aligned} D_t^\beta \mathbf{Y}(t) &= \sigma(\mathbf{F}_\theta(\mathbf{X}(t), t)) - \gamma \mathbf{X}(t) - \alpha \mathbf{Y}(t), \\ D_t^\beta \mathbf{X}(t) &= \mathbf{Y}(t). \end{aligned} \tag{20}$$

The order β of these fractional derivatives serves as a hyperparameter, introducing extra flexibility to these models. This flexibility allows for adaptation to specific data characteristics, enhancing the robustness of the learning process.

5 Experiments

To empirically validate the robustness of FROND, we carry out a series of experiments where real-world graphs are subjected to various attack methods. The objective of these experiments is to showcase that FROND models, even in the face of such adversarial perturbations, maintain stable performance in downstream tasks, without the need for any additional preprocessing steps to handle the perturbed data. For a comprehensive and fair evaluation, we perform two distinct evaluations: a poisoning Graph Modification Attack (GMA), where training occurs directly on the perturbed graph; and an evasion attack for Graph Injection Attack (GIA), taking place during the inference phase.

We emphasize that the primary objective of this paper is to investigate the robustness imparted by FROND and to establish that fractional methods exhibit superior robustness compared to GNNs governed by integer-order dynamical systems. Accordingly, our comparisons focus primarily on standard integer-order graph neural ODE models, along with several specific non-ODE-based methods, including GCN (Kipf and Welling 2017), GAT (Vaswani et al. 2017), and GraphSAGE (Hamilton, Ying, and Leskovec 2017). It is worth noting that FROND models can be further integrated with other defense techniques, including adversarial training

Dataset	Attack	F-GRAND	GRAND	F-GraphBel	GraphBel	F-GraphCON	GraphCON	GAT	GCN
Cora	*clean*	**86.44±0.31**	85.87±0.59	77.55±0.79	79.07±0.46	82.42±0.89	83.10±0.63	86.37±0.56	85.09±0.26
	PGD	56.38±6.39	36.80±1.86	**69.50±2.83**	63.93±3.88	56.70±4.36	48.38±2.44	38.82±2.48	40.11±0.70
	TDGIA	54.88±6.72	40.0±3.52	**56.94±1.82**	53.22±2.95	54.24±2.54	46.43±2.82	32.76±3.30	40.43±1.76
	MetaGIA	53.36±5.31	37.89±1.56	**71.98±1.32**	66.74±3.23	63.97±2.09	52.21±2.71	42.23±4.19	42.52±0.90
Citeseer	*clean*	71.91±0.43	72.52±0.73	71.09±0.30	**74.75±0.28**	73.50±0.43	72.07±0.93	73.10±0.39	74.48±0.66
	PGD	**61.26±1.23**	42.20±2.77	60.78±2.37	47.73±5.87	54.47±1.0	37.71±7.0	35.12±12.44	30.49±0.80
	TDGIA	50.74±1.20	30.02±1.33	**65.52±0.55**	47.88±1.83	54.71±1.69	30.93±3.00	28.64±4.05	28.88±2.07
	MetaGIA	55.50±1.72	30.42±1.87	**60.85±1.88**	39.13±1.19	48.82±3.27	29.09±2.01	30.17±2.71	32.74±1.00
Computers	*clean*	**92.61±0.20**	92.53±0.34	88.02±0.24	88.12±0.33	91.86±0.38	91.30±0.20	91.42±0.22	91.83±0.25
	PGD	89.90±1.33	70.45±11.03	87.60±0.33	87.38±0.37	**91.36±0.74**	81.28±7.99	38.82±5.53	33.43±0.21
	TDGIA	84.71±1.52	65.45±14.30	87.81±0.28	87.67±0.40	**90.45±0.71**	68.70±15.67	42.04±9.01	39.83±3.15
	MetaGIA	87.50±3.17	70.01±9.32	87.37±0.23	87.77±0.22	**90.51±0.88**	82.43±8.42	41.86±8.33	34.03±0.36
Pubmed	*clean*	88.39±0.47	88.44±0.34	89.51±0.12	88.18±1.89	**90.30±0.11**	88.09±0.32	87.41±1.73	88.46±0.20
	PGD	59.62±11.66	44.61±2.78	**82.09±0.83**	67.81±12.23	51.16±6.04	45.85±1.97	48.94±12.99	39.03±0.10
	TDGIA	54.31±2.38	46.26±1.32	**82.72±0.47**	68.66±10.64	55.50±4.03	45.57±2.02	47.56±3.11	42.64±1.41
	MetaGIA	61.62±9.05	44.07±2.11	**79.16±0.87**	64.64±9.70	52.03±5.53	45.81±2.81	44.75±2.53	40.42±0.17

Table 2: Node classification accuracy (%) on graph injection, evasion, non-targeted attack in inductive learning. The best and the second-best results for each criterion are highlighted in bold and underlined, respectively.

and pre-processing strategies. We delve into this aspect in the supplementary material.

5.1 GMA

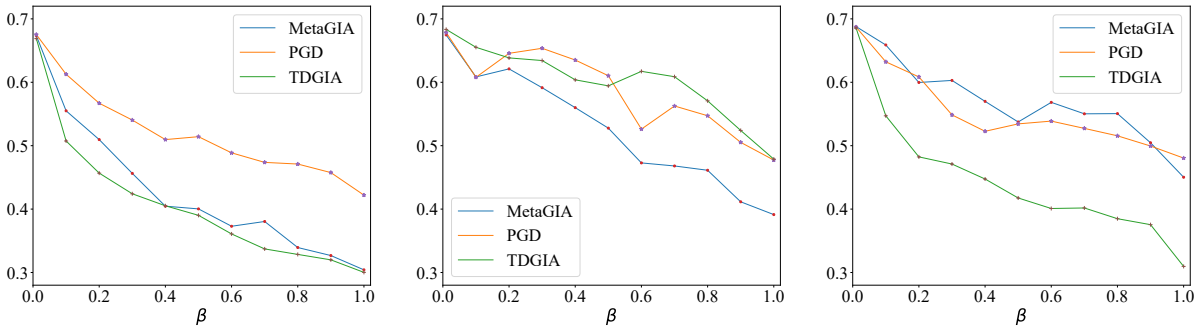
Our experimental setup involves the execution of graph modification adversarial attacks employing the Metattack method (Zügner and Günnemann 2019). Within the Metattack paradigm, the graph's adjacency matrix is perceived not just as a static structure but as a malleable hyperparameter. This perspective allows for attack optimization through meta-gradients to effectively address the inherent bilevel problem. For the sake of ensuring a consistent and unbiased comparative landscape, our experiments strictly conform to the attack parameters as outlined in the paper (Jin et al. 2020). To achieve a comprehensive evaluation, we vary the perturbation rate, representing the proportion of edge modifications. We source the perturbed graph data from the comprehensive DeepRobust library (Li et al. 2020). The perturbation rate is adjusted in consistent increments of 5%, starting from an untouched graph (0%) and extending up to significant alterations at 25%.

5.2 GIA

As elucidated in the paper (Chen et al. 2022), GIA presents a considerably potent challenge to GNNs because of its ability to introduce new nodes and establish new edges within the original graph. Executing a GIA entails a two-step process: the injection of nodes and the subsequent update of features. During the node injection phase, new edges are established for the inserted nodes, driven by either gradient data or heuristic methods. Drawing inspiration from the methods proposed in (Chen et al. 2022), we have incorporated three distinct GIA techniques: PGD-GIA, TDGIA(Zou et al. 2021), and MetaGIA. The PGD-GIA method predominantly relies on a randomized approach for node injection. Once these nodes are in place, their features are meticulously curated using the Projected Gradient Descent (PGD) algorithm (Mądry et al. 2018). The Topological Deficiency Graph Injection Attack (TDGIA) (Zou et al. 2021) exploits inherent topological weaknesses in graph structures. This approach harnesses these vulnerabilities to guide edge creation, optimizing a specific loss function to devise suitable features. MetaGIA (Chen et al. 2022) continually refines the adjacency matrix and node features, leaning heavily on gradient information to guide these refinements. We conduct inductive learning for GIA in line with the data partitioning approach of the GRB framework (Zheng et al. 2021), allocating 60% for training, 10% for validation, and 20% for testing purposes. To maintain a balanced attack landscape, we pre-process the data using methods from (Zheng et al. 2021), which involve excluding the 5% of nodes with the lowest degrees (more susceptible to attacks) and the 5% with the highest degrees (more resistant to attacks).

5.3 Results

Table 1 presents the results of GMA for transductive learning. As can be observed from the table, our proposed fractional-order methods outperform the original GRAND, GraphBel, and GraphCON in terms of robustness accuracy. These results validate and resonate with our theoretical findings, as discussed in Theorem 3. Notably, these empirical observations underscore the capability of the FROND paradigm in enhancing a system's resilience, particularly when faced with input perturbations. The GIA results are presented in Table 2. We note that the fractional-order approach significantly improves post-attack accuracy relative to its integer-order graph neural ODE counterparts. Among these neural ODE models, (Song et al. 2022) demonstrated that GraphBel possesses superior robustness, which is further amplified by our fractional-order differential technique.



(a) F-GRAND (b) F-GraphBel (c) F-GraphCON

Figure 3: The impact of β on the robust test accuracy.

Dataset	Attack	F-GRAND	GRAND	F-GraphBel	GraphBel	F-GraphCON	GraphCON
Computers	*clean*	90.0±0.05	**92.78±0.13**	88.36±1.05	90.14±0.27	89.99±0.28	91.70±0.25
	PGD	75.29±1.17	16.44±0.11	**86.35±0.10**	67.04±1.28	71.64±2.33	13.11±4.73
	TDGIA	71.99±0.73	15.10±0.76	**86.21±0.21**	53.75±2.84	66.35±1.94	4.33±4.21

Table 3: Node classification accuracy (%) on graph injection, evasion, non-targeted, white-box attack in inductive learning.

5.4 White-Box Attack

White-box attacks, which directly target the model, are stronger than the black-box attacks used in Table 2. To demonstrate that our graph neural FDE model can consistently improve the robustness of graph neural ODE models, we also conducted white-box GIA. The results are presented in Table 3. Although the accuracy under white-box GIA is lower than under black-box GIA, our graph neural FDE models still outperform the graph neural ODE models. This observation aligns with our theoretical findings presented in Section 4 of our main paper. Our graph neural FDE models indeed enhance the robustness of neural ODE models under both black-box and white-box scenarios.

5.5 Ablation Study

Influence of β We assess the robustness accuracy of our fractional-order method across varying β values. The findings are depicted in Fig. 3. A discernible trend emerges: as β increases, the accuracy under the three GIA methods diminishes. This observation aligns with our theoretical insights presented in Theorem 3.

Model Complexity A comparison of inference times between our models and the baseline models is presented in Table 4. The results indicate that fractional-based models have similar inference times to graph neural ODE models. Notably, fractional-based models maintain the same training parameters as integer ODE models, avoiding any extra parameters. These findings highlight the efficiency and flexibility of our approach.

Model	Inf. Time(s)
F-GRAND	18.74
GRAND	16.40
F-GraphBel	64.90
GraphBel	78.51
F-GraphCON	21.33
GraphCON	18.27

Table 4: Inference time of models on the Cora dataset: integral time $T = 10$ and step size of 1.

6 Conclusion

In this paper, we have undertaken a comprehensive exploration of robustness against adversarial attacks within the framework of the graph neural FDE models, i.e., FROND models, leading to substantial insights. Our investigation has yielded significant revelations, notably demonstrating the heightened robustness of the FROND models when compared to existing graph neural ODE models. Moreover, our work has contributed theoretical clarity, shedding light on the underlying reasons behind the heightened robustness of the FROND models in contrast to the graph neural ODE counterparts.

Acknowledgments

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. The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg). To improve the readability, parts of this paper have been grammatically revised using ChatGPT (OpenAI 2022).

References

Avelar, P. H. C.; Tavares, A. R.; Gori, M.; and Lamb, L. C. 2019. Discrete and Continuous Deep Residual Learning Over Graphs. arXiv:1911.09554.

Caputo, M. 1967. Linear models of dissipation whose Q is almost frequency independent—II. *Geophysical Journal International*, 13(5): 529–539.

Chamberlain, B.; Rowbottom, J.; Eynard, D.; Di Giovanni, F.; Dong, X.; and Bronstein, M. 2021a. Beltrami flow and neural diffusion on graphs. In *Adv. Neural Inform. Process. Syst.*, 1594–1609.

Chamberlain, B. P.; Rowbottom, J.; Goronova, M.; Webb, S.; Rossi, E.; and Bronstein, M. M. 2021b. GRAND: Graph Neural Diffusion. In *Proc. Int. Conf. Mach. Learn.*

Chen, J.; Wu, Y.; Xu, X.; Chen, Y.; Zheng, H.; and Xuan, Q. 2018a. Fast Gradient Attack on Network Embedding. arXiv:1809.02797.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2018b. Neural ordinary differential equations. In *Adv. Neural Inform. Process. Syst.*

Chen, Y.; Yang, H.; Zhang, Y.; Ma, K.; Liu, T.; Han, B.; and Cheng, J. 2022. Understanding and Improving Graph Injection Attack by Promoting Unnoticeability. In *Proc. Int. Conf. Learn. Represent.*

Choi, J.; Hong, S.; Park, N.; and Cho, S.-B. 2023. GREAD: Graph Neural Reaction-Diffusion Networks. In *Proc. Int. Conf. Mach. Learn.*

Dai, E.; Zhao, T.; Zhu, H.; Xu, J.; Guo, Z.; Liu, H.; Tang, J.; and Wang, S. 2023. A Comprehensive Survey on Trustworthy Graph Neural Networks: Privacy, Robustness, Fairness, and Explainability. arXiv:2204.08570.

Dai, H.; Li, H.; Tian, T.; Huang, X.; Wang, L.; Zhu, J.; and Song, L. 2018. Adversarial attack on graph structured data. In *Proc. Int. Conf. Mach. Learn.*, 1115–1124.

Deng, C.; Li, X.; Feng, Z.; and Zhang, Z. 2022. GARNET: Reduced-Rank Topology Learning for Robust and Scalable Graph Neural Networks. In *Learning on Graphs Conference*, 3–1. PMLR.

Diethelm, K.; and Ford, N. 2010. The analysis of fractional differential equations. *Lect. Notes Math*, 2004: 3–12.

Diethelm, K.; Ford, N. J.; and Freed, A. D. 2004. Detailed error analysis for a fractional Adams method. *Numer. Algorithms*, 36: 31–52.

Du, J.; Zhang, S.; Wu, G.; Moura, J. M. F.; and Kar, S. 2018. Topology Adaptive Graph Convolutional Networks. arXiv:1710.10370.

Dupont, E.; Doucet, A.; and Teh, Y. W. 2019. Augmented neural odes. In *Adv. Neural Inform. Process. Syst.*, 1–11.

Entezari, N.; Al-Sayouri, S. A.; Darvishzadeh, A.; and Papalexakis, E. E. 2020. All You Need Is Low (Rank): Defending Against Adversarial Attacks on Graphs. In *Proc. Int. Conf. Web Search Data Mining*, 169–177.

Finkelshtein, B.; Baskin, C.; Zheltonozhskii, E.; and Alon, U. 2022. Single-node attacks for fooling graph neural networks. *Neurocomputing*, 513: 1–12.

Geisler, S.; Schmidt, T.; Şirin, H.; Zügner, D.; Bojchevski, A.; and Günnemann, S. 2021. Robustness of Graph Neural Networks at Scale. In *Adv. Neural Inform. Process. Syst.*

Guo, Z.; Guo, K.; Nan, B.; Tian, Y.; Iyer, R. G.; Ma, Y.; Wiest, O.; Zhang, X.; Wang, W.; Zhang, C.; and Chawla, N. V. 2023. Graph-based Molecular Representation Learning. arXiv:2207.04869.

Haber, E.; and Ruthotto, L. 2017. Stable architectures for deep neural networks. *Inverse Problems*, 34(1): 1–23.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Adv. Neural Inform. Process. Syst.*

Han, A.; Shi, D.; Lin, L.; and Gao, J. 2023. From Continuous Dynamics to Graph Neural Networks: Neural Diffusion and Beyond. arXiv:2310.10121.

Huang, C.; Xu, H.; Xu, Y.; Dai, P.; Xia, L.; Lu, M.; Bo, L.; Xing, H.; Lai, X.; and Ye, Y. 2021. Knowledge-aware coupled graph neural network for social recommendation. In *Proc. AAAI Conf. Artificial Intell.*, volume 35, 4115–4122.

Hussain, H.; Cao, M.; Sikdar, S.; Helic, D.; Lex, E.; Strohmaier, M.; and Kern, R. 2022. Adversarial Inter-Group Link Injection Degrades the Fairness of Graph Neural Networks. In *IEEE International Conference on Data Mining (ICDM)*. IEEE.

Ji, F.; Lee, S. H.; Meng, H.; Zhao, K.; Yang, J.; and Tay, W. P. 2023. Leveraging Label Non-Uniformity for Node Classification in Graph Neural Networks. In *Proc. Int. Conf. Mach. Learn.*, volume 202, 14869–14885.

Jin, W.; Ma, Y.; Liu, X.; Tang, X.; Wang, S.; and Tang, J. 2020. Graph structure learning for robust graph neural networks. In *Proc. Int. Conf. Knowl. Discovery Data Mining*, 66–74.

Kang, Q.; Song, Y.; Ding, Q.; and Tay, W. P. 2021. Stable neural ODE with Lyapunov-stable equilibrium points for defending against adversarial attacks. In *Adv. Neural Inform. Process. Syst.*

Kang, Q.; Zhao, K.; Ding, Q.; Ji, F.; Li, X.; Liang, W.; Song, Y.; and Tay, W. P. 2023a. Unleashing the Potential of Fractional Calculus in Graph Neural Networks. In *Adv. Neural Inform. Process. Syst. Workshop on Machine Learning and the Physical Sciences*.

Kang, Q.; Zhao, K.; Song, Y.; Wang, S.; and Tay, W. P. 2023b. Node Embedding from Neural Hamiltonian Orbits in Graph Neural Networks. In *Proc. Int. Conf. Mach. Learn.*, 15786–15808.

Kang, Q.; Zhao, Y.; Zhao, K.; Li, X.; Ding, Q.; Tay, W. P.; and Wang, S. 2023c. Advancing Graph Neural Networks Through Joint Time-Space Dynamics. In *Adv. Neural Inform. Process. Syst. Workshop on The Symbiosis of Deep Learning and Differential Equations III*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proc. Int. Conf. Learn. Represent.*

Lee, S. H.; Ji, F.; and Tay, W. P. 2022. SGAT: Simplicial Graph Attention Network. In *Proc. Inter. Joint Conf. Artificial Intell.*

Li, Y.; Jin, W.; Xu, H.; and Tang, J. 2020. DeepRobust: A PyTorch Library for Adversarial Attacks and Defenses. arXiv:2005.06149.

Ma, J.; Deng, J.; and Mei, Q. 2022. Adversarial Attack on Graph Neural Networks as An Influence Maximization Problem. In *Proc. of the 15th ACM Int. Conf. Web Search and Data Min.*, 675–685.

Ma, J.; Ding, S.; and Mei, Q. 2020. Towards more practical adversarial attacks on graph neural networks. In *Adv. Neural Inform. Process. Syst.*, 4756–4766.

Ma, Y.; Wang, S.; Derr, T.; Wu, L.; and Tang, J. 2021. Graph Adversarial Attack via Rewiring. In *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1161–1169.

Mądry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *Proc. Int. Conf. Learn. Represent.*

OpenAI. 2022. ChatGPT-4. Available at: https://www.openai.com (Accessed: 26 September 2023).

Poli, M.; Massaroli, S.; Park, J.; Yamashita, A.; Asama, H.; and Park, J. 2021. Graph Neural Ordinary Differential Equations. arXiv:1911.07532.

Rusch, T. K.; Chamberlain, B.; Rowbottom, J.; Mishra, S.; and Bronstein, M. 2022a. Graph-coupled oscillator networks. In *Proc. Int. Conf. Mach. Learn.*, 18888–18909. PMLR.

Rusch, T. K.; Chamberlain, B. P.; Rowbottom, J.; Mishra, S.; and Bronstein, M. M. 2022b. Graph-Coupled Oscillator Networks. In *Proc. Int. Conf. Mach. Learn.*

She, R.; Kang, Q.; Wang, S.; Tay, W. P.; Guan, Y. L.; Navarro, D. N.; and Hartmannsgruber, A. 2023. Image Patch-Matching With Graph-Based Learning in Street Scenes. *IEEE Trans. Image Process.*, 32: 3465–3480.

Sochen, N.; Kimmel, R.; and Malladi, R. 1998. A general framework for low level vision. *IEEE Trans. Image Process.*, 7(3): 310–318.

Song, Y.; Kang, Q.; Wang, S.; Zhao, K.; and Tay, W. P. 2022. On the Robustness of Graph Neural Diffusion to Topology Perturbations. In *Adv. Neural Inform. Process. Syst.* New Orleans, USA.

Sun, Y.; Wang, S.; Tang, X.; Hsieh, T.-Y.; and Honavar, V. 2020. Adversarial Attacks on Graph Neural Networks via Node Injections: A Hierarchical Reinforcement Learning Approach. In *Proc. Web Conf.*, 673–683.

Thorpe, M.; Nguyen, T. M.; Xia, H.; Strohmer, T.; Bertozzi, A.; Osher, S.; and Wang, B. 2021. GRAND++: Graph neural diffusion with a source term. In *Proc. Int. Conf. Learn. Represent.*

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *Proc. Int. Conf. Learn. Represent.*, 1–12.

Wang, D.; Lin, J.; Cui, P.; Jia, Q.; Wang, Z.; Fang, Y.; Yu, Q.; Zhou, J.; Yang, S.; and Qi, Y. 2019. A semi-supervised graph attentive network for financial fraud detection. In *IEEE International Conference on Data Mining (ICDM)*, 598–607. IEEE.

Wang, J.; Luo, M.; Suya, F.; Li, J.; Yang, Z.; and Zheng, Q. 2020. Scalable attack on graph data by injecting vicious nodes. *Data Mining and Knowledge Discovery*, 34: 1363–1389.

Wang, Y.; Yi, K.; Liu, X.; Wang, Y. G.; and Jin, S. 2022. ACMP: Allen-Cahn Message Passing with Attractive and Repulsive Forces for Graph Neural Networks. In *Proc.Int. Conf. Learn. Represent.*

Waniek, M.; Michalak, T. P.; Wooldridge, M. J.; and Rahwan, T. 2018. Hiding individuals and communities in a social network. *Nature Human Behaviour*, 2(1): 139–147.

Weinan, E. 2017. A proposal on machine learning via dynamical systems. *Commun. Math. Statist.*, 1(5): 1–11.

Xhonneux, L.-P.; Qu, M.; and Tang, J. 2020. Continuous graph neural networks. In *Proc. Int. Conf. Mach. Learn.*, 10432–10441.

Yan, H.; Du, J.; Tan, V. Y.; and Feng, J. 2018. On robustness of neural ordinary differential equations. In *Adv. Neural Inform. Process. Syst.*, 1–13.

Zhang, X.; and Zitnik, M. 2020. Gnnguard: Defending graph neural networks against adversarial attacks. *Adv. Neural Inform. Process. Syst.*, 33: 9263–9275.

Zhao, K.; Kang, Q.; Song, Y.; She, R.; Wang, S.; and Tay, W. P. 2023a. Adversarial Robustness in Graph Neural Networks: A Hamiltonian Energy Conservation Approach. In *Adv. Neural Inform. Process. Syst.* New Orleans, USA.

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

Zheng, Q.; Zou, X.; Dong, Y.; Cen, Y.; Yin, D.; Xu, J.; Yang, Y.; and Tang, J. 2021. Graph Robustness Benchmark: Benchmarking the Adversarial Robustness of Graph Machine Learning. *Adv. Neural Inform. Process. Syst. Track Datasets Benchmarks*.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust graph convolutional networks against adversarial attacks. In *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1399–1407.

Zou, X.; Zheng, Q.; Dong, Y.; Guan, X.; Kharlamov, E.; Lu, J.; and Tang, J. 2021. TDGIA: Effective Injection Attacks on Graph Neural Networks. In *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2461–2471.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial Attacks on Neural Networks for Graph Data. In *Proc. Int. Conf. Knowl. Discovery Data Mining*.

Zügner, D.; and Günnemann, S. 2019. Adversarial Attacks on Graph Neural Networks via Meta Learning. In *Proc. Int. Conf. Learn. Represent.*