Adaptive Multi-view Graph Contrastive Learning via Fractional-order Neural Diffusion Networks

Yanan Zhao, Feng Ji, Jingyang Dai, Jiaze Ma, Keyue Jiang, Kai Zhao, Wee Peng Tay

arXiv:2511.06216v3 [cs.LG] 18 Mar 2026

***Abstract*—Graph contrastive learning (GCL) learns node and graph representations by contrasting multiple views of the same graph. Existing methods typically rely on fixed, handcrafted views-usually a local and a global perspective—which limits their ability to capture multi-scale structural patterns. We present an augmentation-free, multi-view GCL framework grounded in fractional-order continuous dynamics. By varying the fractional derivative order $\alpha \in (0, 1]$, our encoders produce a continuous spectrum of views: small α yields localized features, while large α induces broader, global aggregation. We treat α as a learnable parameter so the model can adapt diffusion scales to the data and automatically discover informative views. This principled approach generates diverse, complementary representations without manual augmentations. Extensive experiments on standard benchmarks demonstrate that our method produces more robust and expressive embeddings and outperforms state-of-the-art GCL baselines.**

***Index Terms*—Multi-view graph contrastive learning, fractional differential equations**

I. Introduction

Contrastive learning is a self-supervised paradigm that trains representations by pulling semantically similar pairs together and pushing dissimilar pairs apart in the embedding space. It uncovers useful structure without labels and has produced strong results in computer vision, natural language processing, and recommendation systems [1]–[3]. Applied to graph data, this paradigm is known as *Graph Contrastive Learning (GCL)*, where positive pairs are typically derived from different views (e.g., node-, subgraph-, or graph-level perturbations) of the same instance and negatives from other, unrelated instances.

A central challenge in GCL is how to produce distinct, semantically meaningful views. Prior approaches follow two broad strategies. Augmentation-based methods synthesize multiple views by perturbing the graph (e.g., edge deletion, feature masking), while augmentation-free methods obtain different views by applying distinct encoders to the same input. Despite these differences, most methods still rely on a small set of fixed, handcrafted views—commonly a "local" and a "global" perspective—which constrains their ability to represent multi-scale structure. For example, BGRL [4] creates two views via structural and feature perturbations, and PolyGCL [5] constructs low-pass and high-pass views through spectral filtering. Such designs increase view diversity but remain rigid and often require manual tuning. This motivates two questions:

Q1 How can we adaptively generate a diverse set of views that capture multi-scale semantics in GCL, moving beyond fixed, handcrafted local and global perspectives?
Q2 Can this be achieved without relying on heuristic augmentations?

To answer these questions, we propose a novel **multi-view augmentation-free** framework grounded in fractional-order continuous dynamics. Our key insight is to leverage *fractional-order differential equations (FDEs)* [6] to model graph diffusion and generate a continuum of semantically diverse views, each parameterized by a distinct fractional derivative order. Prior works, which rely on ordinary differential equations (ODEs) [7]–[13], simulate graph diffusion as continuous-time heat flow and serve as smooth analogues of message passing in models like GCN. Unlike such ODE-based approaches, FDEs model graph diffusion by introducing the fractional derivative operator $\frac{\mathrm{d}^\alpha}{\mathrm{d}t^\alpha}$, where $\alpha \in (0, 1]$ [14]–[19]. When $\alpha = 1$, the standard ODE is recovered; however, for $\alpha < 1$, the operator incorporates non-local information through memory effects, allowing the diffusion to reflect long-range temporal dependencies and attenuated propagation.

By leveraging FDEs, **multi-view** GCL emerges naturally: the fractional order $\alpha \in (0, 1]$ provides a continuous control of diffusion scale. Varying α yields a continuum of views — small α emphasizes local components, while large α produces broader global aggregation. This gives a principled, augmentation-free mechanism to generate diverse and complementary views without handcrafted filters (e.g., filter-based methods such as PolyGCL) or task-specific perturbations (e.g., BGRL). Moreover, α is a learnable continuous parameter, so the model can automatically adapt diffusion scales to dataset heterogeneity. To our knowledge, this is the first work to exploit the multi-scale behaviour of fractional-order dynamics for contrastive learning. Our main contributions are summarized as follows:

- We propose a novel multi-view, augmentation-free GCL framework based on neural diffusion encoders governed by *fractional-order dynamics*. This approach is conceptually simple, computationally efficient, and highly flexible, with each feature view controlled by a continuous parameter $\alpha \in (0, 1]$, naturally spanning a spectrum of diffusion scales.
- Our main theoretical result shows that embeddings produced by FDE encoders at different fractional orders are provably distinct, and that the separation between these views grows as the gap between the orders α increases. To our knowledge, this is the **first** formal analysis of this multi-scale distinguishability in the context of contrastive

Yanan Zhao, Feng Ji, Jingyang Dai, Jiaze Ma, Kai Zhao, and Wee Peng Tay are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.
Keyue Jiang is with University College London, London, U.K.

TABLE I
FD-MVGCL (OURS) VS. SOTA GCL METHODS.

Method	>2 views	Aug-Free	Neg-Free	Hetero.	Adapt.
BGRL ('22)	✗	✗	✓	✓	✓
AFGRL ('22)	✗	✓	✓	✗	✗
GraphACL ('23)	✗	✓	✗	✓	✗
GraphECL ('24)	✗	✓	✗	✓	✗
PolyGCL ('24)	✗	✓	✗	✓	✗
LOHA ('25)	✗	✓	✗	✓	✗
AFECL ('25)	✗	✓	✗	✗	✗
EPAGCL ('25)	✗	✗	✗	✗	✓
FD-MVGCL	✓	✓	✓	✓	✓

learning, despite prior empirical use of FDEs in GNNs.

- We tackle two core challenges in GCL: (i) dimension collapse — mitigated by including encoders with small fractional orders that yield less energy-concentrated, higher-rank embeddings; and (ii) view collapse — prevented via a regularized contrastive objective that penalizes alignment of dominant directions across views, promoting diverse, complementary representations without requiring negative samples.
- The fractional orders α are treated as learnable parameters, allowing the model to discover informative diffusion scales from data instead of relying on manual tuning or grid search. To control complexity, we propose a view-learning strategy that prunes redundant scales while retaining diverse, complementary views.
- We present a rigorous stability analysis, deriving explicit perturbation bounds that quantify how input, parameter, and topological disturbances affect encoder outputs. Empirically, our approach consistently outperforms state-of-the-art robust GCL methods under both black-box and white-box attack scenarios.
- We evaluate on a broad set of homophilic and heterophilic graph benchmarks. Empirical results demonstrate that our method consistently outperforms state-of-the-art GCL baselines, confirming the effectiveness of the proposed multi-scale view generation mechanism.

II. Related work

GCL methods can be categorized into augmentation-based and augmentation-free approaches. Augmentation-based methods synthesize multiple views by perturbing the input graph (e.g., edge deletion, feature masking) [20]–[25]. Augmentation-free methods instead obtain different views by applying distinct encoders or filters to the same graph [26], [27]. Many augmentation-free approaches (e.g., GraphACL [28], SP-GCL [29], GraphECL [30], PolyGCL [5], LOHA [31], AFECL [32]) still depend on negative samples for contrastive training. SDMG [33] follows a diffusion-based, self-supervised design using two low-frequency encoders to emphasize global signals. By contrast, methods such as BGRL [4] and CCA-SSG [34] remove negative sampling but keep augmentations, while AFGRL [35] removes both negatives and augmentations yet can suffer from limited view diversity and fixed propagation depth, which hurts performance on heterophilic graphs.

In contrast, our method operates in an augmentation-free and negative-sample-free manner and performs well on both homophilic and heterophilic graphs (see Table I). We propose a different multi-view mechanism based on fractional-order neural diffusion: each view is generated by an FDE with its own fractional order $\alpha \in (0, 1]$, providing continuous, learnable control over the diffusion scale. This yields diverse, complementary views without handcrafted filters or graph augmentations. By learning the α values from data, the model removes the need for manual tuning or grid search. To our knowledge, this is the first work to generate multi-scale contrastive views via fractional-order dynamics on graphs, and it achieves strong results across both homophilic and heterophilic settings.

III. Preliminaries

A. Problem formulation

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is a finite set of N nodes and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges. The raw features of the nodes are represented by the matrix $\mathbf{X}$, with the i-th *row* corresponding to the feature vector $\mathbf{x}_i$ of node v_i. The weighted symmetric adjacency matrix $\mathbf{A} = (a_{ij})$ has size $N \times N$, where a_{ij} denotes the edge weight between nodes v_i and v_j. We denote the complete graph information by $\mathcal{X} = (\mathbf{A}, \mathbf{X})$.

In unsupervised feature learning, an encoder f_θ maps each raw feature $\mathbf{x}_i$ to a latent representation $\mathbf{y}_i = f_\theta(\mathcal{X}, v_i) \in \mathbb{R}^F$, forming the embedding matrix $\mathbf{Y} \in \mathbb{R}^{N \times F}$ for downstream tasks such as node classification. Multi-view GCL introduces self-supervision by encouraging consistency among node embeddings $\mathbf{z}_i$ produced by multiple encoders f_{θ_k}, $k \geq 2$.

B. Graph neural diffusion models: from GCN to FDE

Traditional graph neural network (GNN) models, such as GCN and GAT [36]–[38], rely on (discrete) graph message passing for feature aggregation. In the k-th iteration, the message passing step is given by $\mathbf{Y}^{(k)} = \overline{\mathbf{A}}\mathbf{Y}^{(k-1)}$, where $\overline{\mathbf{A}}$ is the normalized adjacency matrix of $\mathbf{A}$, and $\mathbf{Y}^{(k-1)}$ is the output from the $(k-1)$-th iteration. In contrast, [7] introduces a continuous formulation based on heat diffusion, where the feature evolution $\mathbf{Y}(t)$ is governed by an ordinary differential equation (ODE):

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

with the initial condition $\mathbf{Y}(0)$ set to $\mathbf{X}$ or its transformation. Here, t serves a role analogous to the layer index in GCNs, and $\mathcal{F}(\mathbf{W}, \mathbf{Y}(t))$ defines the spatial diffusion. Let $\overline{\mathbf{L}} = \mathbf{I} - \overline{\mathbf{A}}$ denote the *normalized Laplacian*. The GRAND model [7] is given by $\mathcal{F}(\mathbf{W}, \mathbf{Y}(t)) = -\overline{\mathbf{L}}\mathbf{Y}(t)$.

This formulation is further extended in [15] by incorporating FDEs. Specifically, for each *order parameter* $\alpha \in (0, 1)$, the corresponding fractional-order differential operator D_t^α is defined as

$$D_t^\alpha f(t) = \frac{1}{\Gamma(-\alpha)} \int_0^\infty \frac{f(t-\tau) - f(t)}{\tau^{1+\alpha}} \, \mathrm{d}\tau,$$

where $\Gamma(\cdot)$ denotes the Gamma function.

The operator D_t^α extends the concept of $\frac{\mathrm{d}}{\mathrm{d}t}$ (from (1)) such that as $\alpha \to 1$, D_t^α converges to $\frac{\mathrm{d}}{\mathrm{d}t}$. Thus, it is natural to define $D_t^1 = \frac{\mathrm{d}}{\mathrm{d}t}$, allowing α to be selected from the interval $(0, 1]$. By generalizing (1), we define the dynamics of the features $\mathbf{Y}(t)$ under an FDE as:

$$D_t^\alpha \mathbf{Y}(t) = \mathcal{F}(\mathbf{W}, \mathbf{Y}(t)). \tag{2}$$

The precise definition of D_t^α may vary slightly depending on the context (see Appendix A), but all formulations share a common characteristic: for $\alpha \in (0, 1)$, D_t^α is defined via an integral. This implies that $\mathbf{Y}(t)$ depends on the entire history of $\mathbf{Y}(t')$ for $t' < t$. Consequently, unlike the solution to (1), the dynamics of $\mathbf{Y}(t)$ exhibit a "memory effect."

Our *main insight* for GCL lies in leveraging the fractional order parameter α to systematically generate a diverse set of distinct encoders, enabling augmentation-free GCL. The multi-view framework is naturally facilitated by the continuous and parameterizable nature of α, which allows for seamless integration and optimization through gradient descent. Our theoretical analysis focuses on the F-GRAND-l model [15] given by

$$D_t^\alpha \mathbf{Y}(t) = -\overline{\mathbf{L}}\mathbf{Y}(t). \tag{3}$$

The analysis shall be performed within the framework of graph signal processing (GSP), which is reviewed next.

C. Graph Signal Processing

GSP [39] provides a useful framework to analyze the impact of different α values on the generated views. We present a concise overview here. Since the normalized Laplacian $\overline{\mathbf{L}}$ is symmetric, it admits an orthogonal decomposition $\overline{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\intercal$. Here, the diagonal entries $\lambda_1 \leq \cdots \leq \lambda_N$ of $\mathbf{\Lambda}$ are the eigenvalues of $\overline{\mathbf{L}}$, also known as the *graph frequencies*, and the i-th column $\mathbf{u}_i$ of $\mathbf{U}$ is the eigenvector corresponding to λ_i. Each $\mathbf{u}_i$ represents a signal on the graph $\mathcal{G}$. For small λ_i, $\mathbf{u}_i$ is smooth, meaning the signal values vary gradually across the edges. Conversely, for large λ_i, $\mathbf{u}_i$ tends to be spiky, emphasizing local variations.

Any signal $\mathbf{x}$ (e.g., a column of the feature matrix $\mathbf{X}$) can be expressed as a *spectral decomposition*:

$$\mathbf{x} = \sum_{i=1}^{N} c_i \mathbf{u}_i, \quad \text{where } c_i = \langle \mathbf{x}, \mathbf{u}_i \rangle.$$

In GSP, the coefficients c_i are referred to as *Fourier coefficients*, which quantify the response of $\mathbf{x}$ to the basis vector $\mathbf{u}_i$. A signal $\mathbf{x}$ is said to have *large smooth components* if $|c_i|$ is relatively large for small indices i, and it is considered *energy concentrated* if $|c_i|$ is small for most indices i.

IV. The proposed model: FD-MVGCL

In this section, we introduce the adaptive **F**ractional **D**iffusion-based **M**ulti-view **G**raph **C**ontrastive **L**earning model (FD-MVGCL). We begin by presenting the multi-view generation mechanism inspired by FDEs, supported by rigorous theoretical analysis and numerical validations. Next, we describe the FD-MVGCL architecture and address two key challenges in graph contrastive learning: dimension collapse and view collapse. Additionally, we propose an adaptive algorithm that automatically determines the optimal number of views and their corresponding fractional orders. Finally, we provide a stability analysis under graph perturbations, highlighting the robustness of FD-MVGCL.

A. Contrasting encoders

1) Multi-view generation: We adopt a non-Markovian random-walk interpretation [15] to illustrate how varying the fractional order α produces multi-scale views. Under the FDE model (3), let ${}_i\mathbf{P}(t) \in \mathbb{R}^N$ be the probability distribution of a fractional-order random walker over the node set $\mathcal{V}$ when initialized at v_i; its j-th entry is the probability of finding the walker at node v_j at time t. The node embeddings are recovered as $\mathbf{Y}(t) = \sum_i {}_i\mathbf{P}(t)\mathbf{Y}(0)_i$, where $\mathbf{Y}(0)_i$ is the initial feature (row) vector of v_i. For $\alpha \in (0, 1]$, the random walk on the graph $\mathcal{G}$ evolves in infinitesimal time intervals $\Delta\tau > 0$ according to the following dynamics:

1) The walker remains at the current node for a random duration sampled from a heavy-tailed distribution $\psi_\alpha(n) = d_\alpha n^{-(1+\alpha)}$, where $d_\alpha > 0$ ensures normalization. Smaller values of α result in heavier tails, leading to longer expected waiting times and a stronger memory effect.
2) After waiting, the walker either transitions from node i to a neighboring node j with probability $(\Delta\tau)^\alpha d_\alpha |\Gamma(-\alpha)| \frac{W_{ij}}{d_i}$ for $i \neq j$, or remains at node i with probability $1 - (\Delta\tau)^\alpha d_\alpha |\Gamma(-\alpha)|$.

This formulation provides an intuitive understanding of how the fractional parameter α governs the scale of information propagation over the graph:

- **Local view**: As $\alpha \to 0$, the walker exhibits long waiting times and infrequent transitions, resulting in highly localized behavior and aggregating information primarily from nearby nodes.
- **Global view**: As $\alpha \to 1$, the walker transitions more frequently and explores a wider portion of the graph, enabling global aggregation and capturing long-range dependencies.

To further elucidate how multi-scale views naturally arise from fractional-order derivatives, we provide a mathematical analysis of the distinct representations generated by FDE encoders with varying fractional orders α. For clarity and conciseness, we present an informal summary of the results here, while a *detailed and rigorous* explanation is deferred to Appendix B. To the best of our knowledge, this is the *first* formal mathematical analysis of this phenomenon in the context of contrastive learning.

Theorem 1 (Informal; see Theorem 4 in Appendix B-A)**.** *Assuming the model* (3)*, for* $0 < \alpha_l < \alpha_g \leq 1$*, the following hold for features* $\mathbf{Y}_{\alpha_l}(t)$ *and* $\mathbf{Y}_{\alpha_g}(t)$ *when* t *is large:*

(a) $\mathbf{Y}_{\alpha_g}(t)$ *contains more large smooth components as compared with* $\mathbf{Y}_{\alpha_l}(t)$*.*
(b) $\mathbf{Y}_{\alpha_l}(t)$ *is less energy concentrated as compared with* $\mathbf{Y}_{\alpha_g}(t)$*.*

Moreover, the contrast in (a) and (b) becomes more pronounced as the difference $\alpha_g - \alpha_l$ *increases.*

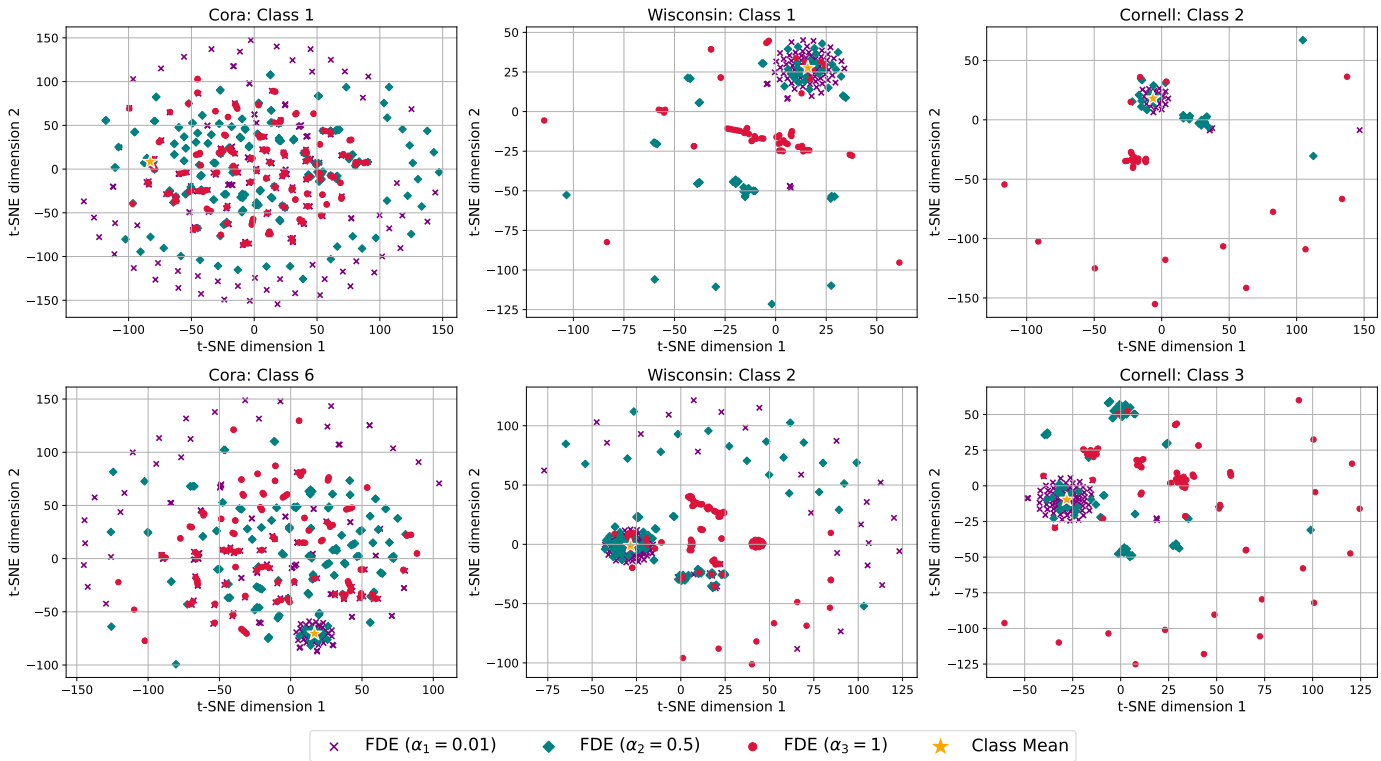


Fig. 1. t-SNE visualizations of single-class node embeddings from encoders with different FDE orders are shown. Class means are aligned for fair comparison. Results on Cora (homophilic) and Wisconsin and Cornell (heterophilic) reveal distinct embedding behaviors: smaller α values yield compact, core-concentrated clusters, whereas larger α values produce more evenly dispersed feature distributions. Additional visualizations for other classes are provided in Appendix D.

As discussed in Section I, the effectiveness of augmentation-free GCL relies on the ability of different encoders to generate feature representations with distinct views. We argue that $\mathbf{Y}_{\alpha_g}(t)$ emphasizes *global* structural patterns as $\alpha_g \to 1$, while $\mathbf{Y}_{\alpha_l}(t)$ captures finer *local* details as $\alpha_l \to 0$. Specifically, by Theorem 1(a), the smooth components (associated with small graph frequencies λ_i) dominate in $\mathbf{Y}_{\alpha_g}(t)$. Each such component is spanned by a vector $\mathbf{u}_i$, which represents a signal with minimal variation across edges, thereby encoding global structural information. Conversely, in $\mathbf{Y}_{\alpha_l}(t)$, non-smooth components $\mathbf{u}_i$ (corresponding to large graph frequencies λ_i) have higher coefficients and thus contribute more significantly. Since these $\mathbf{u}_i$ are localized (spiky), $\mathbf{Y}_{\alpha_l}(t)$ effectively captures local details. In summary, we establish the correspondence: $\alpha_l \longleftrightarrow$ "local" and $\alpha_g \longleftrightarrow$ "global."

By selecting distinct α values, we construct a set of FDEs, each capturing a unique propagation scale. This approach naturally generates diverse, multi-scale graph views (i.e., embeddings $\mathbf{Y}_\alpha(t)$), enriching graph contrastive learning. Numerical experiments demonstrate that embeddings $\mathbf{Y}_{\alpha_i}(t)$ and $\mathbf{Y}_{\alpha_j}(t)$ exhibit distinct patterns when $\alpha_i \neq \alpha_j$, as illustrated in Fig. 1 for the Cora, Wisconsin, and Cornell datasets. For small α values (e.g., 0.01), features are tightly clustered near the class mean, while larger α values (e.g., 1) produce more dispersed feature distributions. Intermediate α values (e.g., 0.5) yield distributions that balance these extremes. This demonstrates that each FDE provides a semantically distinct and complementary view, enhancing the effectiveness of contrastive learning.

2) Dimension collapse: Dimension collapse occurs when features are confined to a low-dimensional subspace within the full embedding space [40], which is a critical issue to address in contrastive learning. We argue that $\mathbf{Y}_{\alpha_l}(t)$, generated with a small α_l, effectively mitigates this problem. As established in Theorem 1(b), $\mathbf{Y}_{\alpha_l}(t)$ exhibits lower energy concentration compared to $\mathbf{Y}_{\alpha_g}(t)$ when $\alpha_l < \alpha_g$. This means that the columns of $\mathbf{Y}_{\alpha_l}(t)$ can be expressed as $\sum_{1\leq i\leq N} c_i \mathbf{u}_i$, with a relatively larger number of significant $|c_i|$ values. As a result, the features are distributed across a higher-dimensional space, avoiding collapse into a low-dimensional subspace.

To support this claim, we perform principal component analysis (PCA) on $\mathbf{Y}_{\alpha_l}(t)$ and $\mathbf{Y}_{\alpha_g}(t)$ for the Cora and Wisconsin datasets. The results, shown in Fig. 2, confirm our assertions.

3) Encoder Quality: The performance of the model improves when the encoders effectively cluster similar nodes. Our approach leverages the proven capabilities of FDE-based encoders in supervised settings [7], [15]. To evaluate their unsupervised clustering ability, we compute the following metrics for each label class c: the average intra-class feature distance d_c^{intra} among nodes within the same class, and the average inter-class feature distance d_c^{inter} between nodes of class c and nodes of other classes. The clustering quality is quantified by the ratio $r_c = d_c^{\text{inter}}/d_c^{\text{intra}}$, where higher values indicate better clustering. From Fig. 3, we observe that our proposed FDE-based encoder yields consistently higher r_c values than PolyGCL, whose two filter-based encoders (low-pass and high-pass, analogous to $\alpha_g = 1$ and $\alpha_l = 0.01$, respectively) are

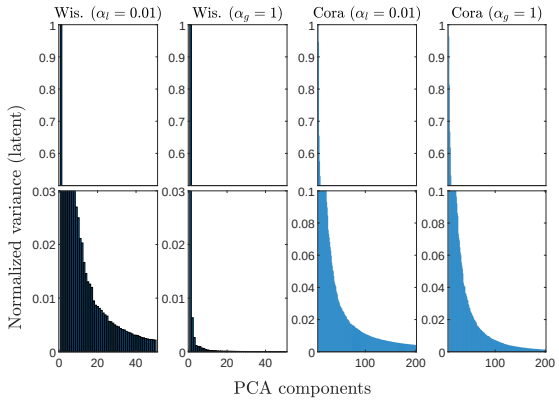


Fig. 2. The PCA components of features for different datasets and choices of FDE order parameters. We see that for at least the small order α_l, the bar chart is more spread out, which prevents dimension collapse.

used for comparison. These results demonstrate that FDE-based encoders produce more discriminative and better-clustered embeddings across both homophilic and heterophilic graphs.

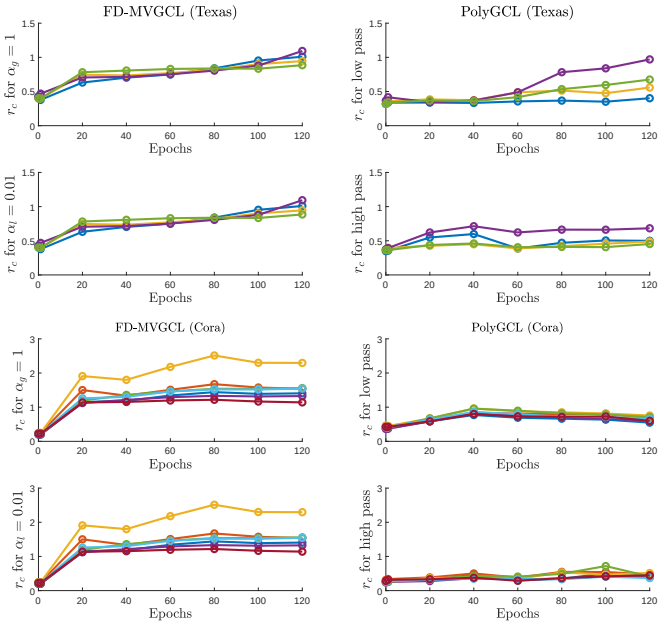


Fig. 3. The variation in the ratio r_c during training. Each line plot corresponds to a label class. The ratio r_c for the input features is shown at epoch 0. We see that the ratio generally increases at the beginning of the training and stabilizes. For example, FD-MVGCL produces higher r_c values than the strong benchmark PolyGCL [5], suggesting better clustered embeddings.

B. The adaptive FD-MVGCL model

We now present the full details of the proposed adaptive FD-MVGCL model. Given a graph $\mathcal{G}$ with adjacency matrix $\mathbf{A}$ and initial node features $\mathbf{X}$, we denote the input as $\mathcal{X} = (\mathbf{A}, \mathbf{X})$. Our objective is to learn multiple FDE encoders f_{θ_k} for $k \geq 2$ in an unsupervised manner, with each encoder capturing structural patterns at a specific scale. In the adaptive FD-MVGCL model, all encoders f_{θ_k} share the same architecture (see Fig. 4). For clarity, we describe the structure of a single encoder below:

- **Feature projection:** Apply a linear transformation with learnable weights $\mathbf{W}_k$ to the input features, i.e., $\mathbf{Z}_k = \mathbf{W}_k\mathbf{X}$. This step typically increases the feature dimension to enhance the model's expressiveness and ability to capture complex patterns.
- **Fractional diffusion:** Initialize the diffusion process with $\mathbf{Y}_{\alpha_k}(0) = \mathbf{Z}_k$, where α_k is the fractional order. Compute $\mathbf{Y}_{\alpha_k}(t)$ using the FDE-based encoder, as described in (2).
- **Activation and output:** Stop the diffusion process at time T and apply a non-linear activation function σ (e.g., ReLU) to obtain the final output, denoted by $\mathbf{Y}_k = \sigma(\mathbf{Y}_{\alpha_k}(T))$.

The model parameters θ_k include the learnable weight matrix $\mathbf{W}_k$, the fractional order α_k, and the diffusion time T. Unlike prior FDE-based models such as FROND [15] and DRAGON [14], which rely on manually tuning α via grid search, we treat the α values as **learnable parameters**. This allows the model to adaptively generate informative views tailored to the input data.

We utilize K encoders $f_{\theta_k}, k = 1, 2, \ldots, K$, each generating an output $\mathbf{Y}_k$. To optimize the model parameters, we employ a *contrastive loss* $\mathcal{L}_0(\mathbf{Y}_k, \mathbf{Y}_{k'})$, where $k' = k + 1 \bmod K$, i.e., this loss is applied to each pair of consecutive outputs $(\mathbf{Y}_k, \mathbf{Y}_{k+1})$, with wrap-around from $\mathbf{Y}_K$ to $\mathbf{Y}_1$. For simplicity, $\mathcal{L}_0$ is defined as the mean cosine similarity, referred to as *cosmean* (see (24)). This consecutive-pair scheme balances representational alignment and computational efficiency by enforcing consistency only between successive views. Additionally, the loss applied to $(\mathbf{Y}_K, \mathbf{Y}_1)$ ensures global consistency across all encoders. This approach reduces the computational complexity from comparing all possible pairs ($\mathcal{O}(K^2)$) to only consecutive pairs ($\mathcal{O}(K)$), maintaining scalability without compromising representation quality.

1) Model/View Collapse: As discussed in Section IV-A2, directly minimizing $\mathcal{L}_0$ can lead to view collapse, where two encoders produce nearly identical representations. To address this, we leverage the observation in Fig. 2 that both $\mathbf{Y}_k$ and $\mathbf{Y}_{k'}$ exhibit dominant components. Let $\mathbf{c}_k$ and $\mathbf{c}_{k'}$ represent the unit directional vectors of these components. To prevent collapse, we introduce a penalty term that discourages alignment between these vectors by incorporating their inner product $\langle \mathbf{c}_k, \mathbf{c}_{k'} \rangle$. The resulting *regularized cosmean* loss is defined as:

$$\mathcal{L}_{\mathcal{R}}(\mathbf{Y}_k, \mathbf{Y}_{k'}) = \mathcal{L}_0(\mathbf{Y}_k, \mathbf{Y}_{k'}) + \eta |\langle \mathbf{c}_k, \mathbf{c}_{k'} \rangle|,$$

where η is a regularization weight. This regularization term encourages diversity between the two representations, ensuring they remain distinct without relying on negative samples, unlike traditional contrastive objectives [5], [28]. The impact of this regularization is further analyzed in Section V-C5. Finally, the overall contrastive loss across all encoders is given by:

$$\mathcal{L} = \sum_{k=1}^{K} \mathcal{L}_{\mathcal{R}}(\mathbf{Y}_k, \mathbf{Y}_{k'}). \tag{4}$$

2) Encoder reduction strategy: To manage training costs as K increases, we propose an encoder reduction strategy (Algorithm 1). Initially, K encoders are created with fractional orders $\{\alpha_k\}_{k=1}^{K}$ uniformly sampled from the range $(0.01, 1]$. Both the weights $\mathbf{W}_k$ and the fractional orders α_k are optimized

Algorithm 1 Adaptive View Learning Algorithm (AVLA)

Input: Initial number of encoders as K, initial fractional orders $\{\alpha_k^{(0)}\}_{k=1}^{K}$, initial weights $\{\mathbf{W}_k^{(0)}\}_{k=1}^{K}$ randomly, diffusion step T, learning rate η and η_{α_k}, number of training epochs N, threshold ϵ and δ.

Output: Optimized number of encoders $\tilde{K}$, fractional orders $\{\tilde{\alpha}_k\}_{k=1}^{\tilde{K}}$

Set **Repeat** $\leftarrow$ `True`
while Repeat is `True` **do**
  for $e = 1$ to N **do**
    Compute $\mathbf{Y}_k(T)$ via (2) for each encoder k
    Compute contrastive loss $\mathcal{L}$ via (4)
    Update $\mathbf{W}_k^{(e)} \leftarrow \mathbf{W}_k^{(e-1)} - \eta \nabla_{\mathbf{W}_k} \mathcal{L}$
    Update $\alpha_k^{(e)} \leftarrow \alpha_k^{(e-1)} - \eta_{\alpha_k} \nabla_{\alpha_k} \mathcal{L}$
    if $\alpha_k^{(e)} \notin (0, 1]$ **then**
      $\alpha_k^{(e)} \leftarrow \min(1, \max(\alpha_k^{(e)}, \epsilon))$
    end if
  end for
  Let $\bar{\boldsymbol{\alpha}} = \big\{\alpha_i \in \{\alpha_k^{(N)}\}_{k=1}^{K} \mid$
    $\max_{\alpha_j, \alpha_\ell \in \bar{\boldsymbol{\alpha}}} |\log(\alpha_j) - \log(\alpha_\ell)| < \delta\big\}$
  Let $\tilde{\boldsymbol{\alpha}} = \Big(\{\alpha_k^{(N)}\}_{k=1}^{K} \setminus \bar{\boldsymbol{\alpha}}\Big) \cup \{\alpha_r\}$, $\alpha_r \sim \text{Unif}(\bar{\boldsymbol{\alpha}})$
  if $|\tilde{\boldsymbol{\alpha}}| < K$ **then**
    $K \leftarrow |\tilde{\boldsymbol{\alpha}}|$
    Reinitialize $\{\alpha_k^{(0)}\}_{k=1}^{K} \leftarrow \tilde{\boldsymbol{\alpha}}$ and $\{\mathbf{W}_k^{(0)}\}_{k=1}^{K}$
    Repeat $\leftarrow$ `True`
  else
    $\tilde{K} = K$ and $\{\tilde{\alpha}_k\}_{k=1}^{K} = \tilde{\boldsymbol{\alpha}}$
    Repeat $\leftarrow$ `False`
  end if
end while
return $\tilde{K}$ and $\{\tilde{\alpha}_k\}_{k=1}^{\tilde{K}}$

jointly using gradient descent on the loss $\mathcal{L}$ defined in (4). After each training phase, if any α_k values are too similar in log-scale, only one is randomly retained, while the corresponding encoders are reinitialized. The number of encoders K is then reduced accordingly. This process is repeated until all α_k values are sufficiently distinct. The final number of encoders is denoted by $\tilde{K}$.

Finally, for downstream tasks, we compute a weighted average of the multiple views as $\mathbf{Y} = \sum_{k=1}^{\tilde{K}} \beta_k \mathbf{Y}_k$, where $\sum_{k=1}^{\tilde{K}} \beta_k = 1$. The weights β_k are tuned on a validation set for each downstream task.

C. Stability

Graph data in real-world applications often contain noisy node features or missing edges [41]. To assess the robustness of FD-MVGCL, we perform a unified stability analysis that accounts for perturbations in input features, model parameters, and graph structure. Theorem 2 establishes the stability of the FDE encoder, while the overall stability of the model is formalized in Theorem 3. Detailed proofs are provided in Appendix B-B.

Theorem 2. *Let $\mathbf{Y}(t)$ denote the solution to the original FDE encoder in* (2)*, and let $\tilde{\mathbf{Y}}(t)$ be the solution to the perturbed system:*

$$\begin{aligned} D_t^{\alpha} \tilde{\mathbf{Y}}(t) &= \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \tilde{\mathbf{Y}}(t)), \\ \tilde{\mathbf{Y}}(0) &= \tilde{\mathbf{Y}}_0. \end{aligned} \tag{5}$$

Define the total perturbation magnitude as

$$\varepsilon := \|\mathbf{Y}_0 - \tilde{\mathbf{Y}}_0\| + \sup_{\mathbf{Y} \in A} \|\mathcal{F}(\mathbf{W}, \mathbf{Y}) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \mathbf{Y})\|, \tag{6}$$

where A denotes a compact set where the solutions of both systems are well defined. Suppose that $\mathcal{F}$ is Lipschitz continuous with constant L. Then, for sufficiently small ε, there exist a sufficiently large positive constant T such that both the functions $\mathbf{Y}(t)$ and $\tilde{\mathbf{Y}}(t)$ are defined on $[0, T]$, and the following perturbation bound holds:

$$\|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| = O\left(\varepsilon T^{\alpha - 1}\right), \tag{7}$$

for $t \in [0, T]$.

Theorem 2 extends both Theorem 6.20 and Theorem 6.21 from [42] by providing a unified bound that accounts for both initial state and functional perturbations. When the perturbation is solely due to the initial state, i.e., $\varepsilon := \|\mathbf{Y}_0 - \tilde{\mathbf{Y}}_0\|$, the theorem simplifies to Theorem 6.20. Similarly, when the perturbation arises only from the governing function, i.e., $\varepsilon := \sup_{\mathbf{Y} \in A} \|\mathcal{F}(\mathbf{W}, \mathbf{Y}) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \mathbf{Y})\|$, it reduces to Theorem 6.21. Thus, Theorem 2 unifies these scenarios, offering a comprehensive stability result for simultaneous perturbations.

Theorem 3. *Suppose $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_{\tilde{K}} \leq 1$ and let*

$$\mathbf{Y}(t) = \sum_{k=1}^{\tilde{K}} \beta_k \mathbf{Y}_k(t), \tag{8}$$

where each $\mathbf{Y}_k(t)$ is the solution to the feature dynamics in (2)*, denote the solution at evolution time t of the FD-MVGCL model with K encoders. Let $\tilde{\mathbf{Y}}(t)$ be the perturbed solution of FD-MVGCL where each encoder is a perturbed system in* (5)*. For sufficiently small combined perturbation ε in* (6)*, and a sufficiently large time horizon $T > 0$ such that both the functions $\mathbf{Y}(t)$ and $\tilde{\mathbf{Y}}(t)$ are defined on $[0, T]$, then we have*

$$\|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| = O\left(\varepsilon T^{\alpha_{\tilde{K}} - 1}\right) \tag{9}$$

for $t \in [0, T]$.

Theorem 3 highlights the inherent robustness of FD-MVGCL: smaller α_k values result in reduced feature discrepancies in (9), making the diffusion process more resilient to perturbations when $\alpha_k < 1$. This property underscores FD-MVGCL's robustness against adversarial attacks, as further demonstrated in Section V-D.

V. Experiments

A. Experimental setup

1) Datasets and splits: We evaluate FD-MVGCL on both heterophilic and homophilic datasets. The heterophilic benchmarks include Wisconsin, Cornell, Texas, Actor, Crocodile, and the filtered versions of Squirrel and Chameleon, where

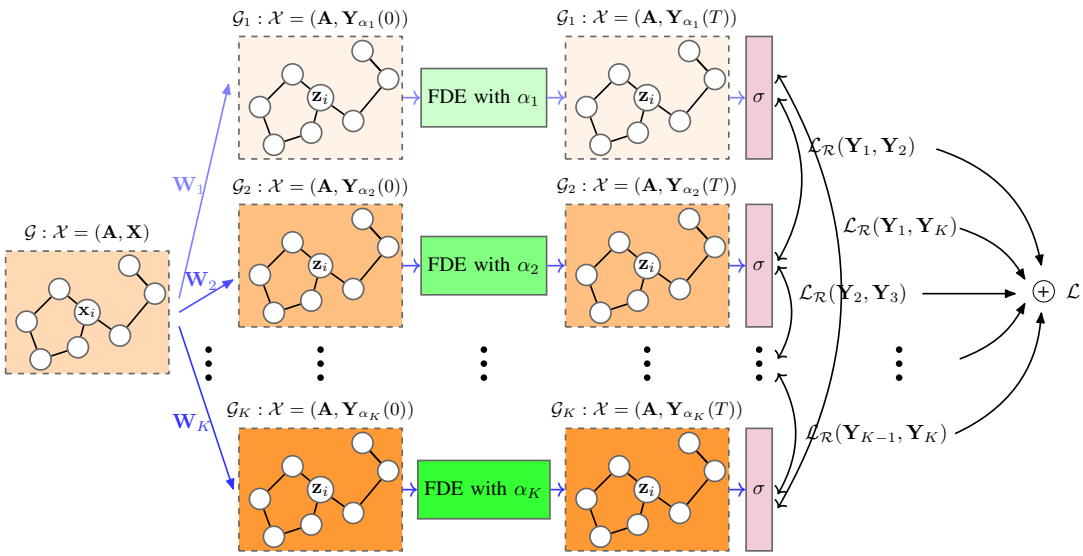


Fig. 4. Overview of the proposed adaptive FD-MVGCL framework. The fractional orders are selected such that $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_K$, enabling contrastive learning across multiple distinct views.

duplicate nodes are removed to prevent training–test leakage [43].[1] In addition, we consider three larger heterophilic datasets: Roman-empire (Roman), Amazon-ratings (Amazon), and Arxiv-year. The homophilic datasets include Cora, Citeseer, Pubmed, Computer, Photo, and the large-scale Ogbn-Arxiv. We adopt the standard public splits from [28], with further details provided in Appendix C-A.

2) Baselines: We compare FD-MVGCL with recent state-of-the-art unsupervised learning methods: DGI [44], GMI [26], MVGRL [45], GRACE [21], CCA-SSG [34], BGRL [4], AFGRL [35], HGRL [24], DSSL [23], SP-GCL [29], GraphACL [28], GraphECL [30], PolyGCL [5], LOHA [31], EPAGCL [25], AFECL [32] and SDMG [33].

3) Evaluation protocol: We mainly evaluate representation quality via node classification using the standard linear protocol: a linear classifier is trained on frozen embeddings, and test accuracy is reported. We further assess FD-MVGCL on the graph classification task (see Section V-F) to demonstrate its generalization beyond node-level settings.

4) Setup: For FD-MVGCL, we adopt the basic GRAND form of $\mathcal{F}$ in (2). Model parameters are randomly initialized and optimized using Adam. Each experiment is run with 10 random seeds (as in [28]), and we report the mean and standard deviation. All methods are tuned solely on validation accuracy for fairness. For baselines without results on certain datasets or using nonstandard splits [5], [25], [31], we reproduce results using the authors' official code. Implementation and hyperparameter details are in Appendix C-B.

B. Overall performance

Node classification results on heterophilic and homophilic datasets are reported in Tables II and III, respectively. FD-MVGCL consistently achieves state-of-the-art or highly competitive performance across both graph types. On heterophilic datasets, it substantially outperforms baseline methods, securing the best or second-best results on most benchmarks. On homophilic datasets, FD-MVGCL remains competitive with leading approaches such as GraphACL and PolyGCL. We also report the average ranking, computed as the arithmetic mean of its valid per-dataset ranks (excluding OOM and missing results), where lower values indicate better overall performance. FD-MVGCL attains the lowest average rank on heterophilic datasets and shares the lowest rank with GraphACL on homophilic datasets, demonstrating its strong generalization and consistency across diverse graph types.

The stronger performance gains on heterophilic graphs reflect the inherent difficulty of unsupervised learning in such settings. Unlike homophilic graphs, where simple clustering methods often suffice, heterophilic graphs require more sophisticated multi-scale representations to capture complex structural patterns. These results demonstrate that FD-MVGCL effectively learns expressive embeddings across diverse graph structures by adaptively balancing local and global diffusion scales. Details on the final number of encoders $\tilde{K}$ and the learned fractional orders $\{\tilde{\alpha}_k\}_{k=1}^{\tilde{K}}$ for each dataset are provided in Appendix C-B.

C. Ablation studies

1) Multi-view vs two-view: We evaluate the performance of FD-MVGCL under various configurations of $\{\alpha_k\}_{k=1}^{K}$: (i) a fixed two-view setting with distinct fractional orders ($\alpha 1 = 0.01$, $\alpha_2 = 1$), (ii) a fixed three-view setting ($\alpha_1 = 0.01$, $\alpha_2 = 0.51$, $\alpha_3 = 1$), (iii) a fixed five-view setting ($\alpha_1 = 0.01$, $\alpha_2 = 0.26$, $\alpha_3 = 0.51$, $\alpha_4 = 0.75$, $\alpha_5 = 1$), and (iv) AVLA (Algorithm 1) that dynamically selects the number and values of α_k during training. These are compared on the Cora and Cornell datasets using classification accuracy, as shown in Table IV.

2) Feature dimension: In FD-MVGCL model, we apply a linear encoder that typically increases the feature dimension to enhance the model's expressiveness and ability to capture

[1]Note that this typically leads to lower performance of baseline methods in our presented results compared to those reported in the baseline papers.

TABLE II
NODE CLASSIFICATION RESULTS(%) ON HETEROPHILIC DATASETS. THE BEST AND THE SECOND-BEST RESULT UNDER EACH DATASET ARE HIGHLIGHTED IN **RED** AND **BLUE**, RESPECTIVELY. OOM REFERS TO OUT OF MEMORY ON AN NVIDIA A5000 GPU (24GB).

Method	DGI	CCA-SSG	BGRL	DSSL	SP-GCL	GraphACL	PolyGCL	GraphECL	LOHA	EPAGCL	SDMG	FD-MVGCL
Squirrel	40.60±0.35	**41.23±1.77**	40.13±2.16	37.69±2.07	40.11±2.20	35.51±2.03	33.07±0.94	35.89±2.62	34.46±1.69	40.28±1.59	36.82±0.77	**41.47±1.54**
Chameleon	42.57±0.71	39.46±3.10	42.55±2.35	40.95±3.35	44.49±2.59	38.59±2.81	41.79±2.45	36.11±1.15	**45.45±1.83**	35.43±1.28	41.55±0.73	**45.57±2.00**
Crocodile	51.25±0.51	56.77±0.39	53.87±0.65	62.98±0.51	61.72±0.21	**66.17±0.24**	65.95±0.59	52.52±3.01	66.09±0.69	66.14±0.62	65.38±0.37	**68.41±0.57**
Actor	28.30±0.76	27.82±0.60	28.80±0.54	28.15±0.31	28.94±0.69	30.03±0.13	34.37±0.69	**35.80±0.89**	33.69±0.73	30.02±0.91	26.74±0.13	**35.91±0.92**
Wisconsin	55.21±1.02	58.46±0.96	51.23±1.17	62.25±0.55	60.12±0.39	69.22±0.40	76.92±1.48	**79.41±0.65**	74.27±1.99	63.73±3.95	52.68±1.21	**78.04±0.08**
Cornell	45.33±6.11	52.17±1.04	50.33±2.29	53.15±1.28	52.29±1.21	59.33±1.48	44.11±1.12	**66.24±1.82**	55.35±1.32	53.78±3.72	45.59±0.67	**72.70±0.57**
Texas	58.53±2.98	59.89±0.78	52.77±1.98	62.11±1.53	59.81±1.33	71.08±0.34	72.51±1.95	**73.76±2.02**	67.75±2.70	68.92±4.05	53.60±2.67	**80.19±0.81**
Roman	63.71±0.63	67.35±0.61	68.66±0.39	71.70±0.54	70.88±0.35	**74.91±0.28**	**72.97±0.25**	45.05±1.57	OOM	47.11±0.87	49.20±0.51	72.45±0.73
Amazon	42.72±0.42	41.23±0.25	41.17±0.25	42.12±0.78	42.04±0.68	OOM	**44.29±0.43**	36.88±1.25	38.45±0.20	OOM	43.70±0.36	**43.86±0.77**
Arxiv-year	39.26±0.72	37.38±0.41	43.02±0.62	45.80±0.57	44.11±0.35	**47.21±0.39**	43.07±0.23	OOM	OOM	OOM	OOM	**47.35±0.22**
Avg. rank.	8.00	7.60	7.50	6.30	6.70	**4.38**	5.20	5.00	5.43	6.33	8.22	**1.40**

TABLE III
NODE CLASSIFICATION RESULTS(%) ON HOMOPHILIC DATASETS.

Method	DGI	GMI	MVGRL	GRACE	BGRL	AFGRL	GraphACL	PolyGCL	LOHA	EPAGCL	AFECL	SDMG	FD-MVGCL
Cora	82.30±0.60	82.70±0.20	82.90±0.71	76.80±0.90	82.70±0.60	82.31±0.42	**84.20±0.31**	82.74±0.14	81.22±0.17	82.14±0.89	82.10±1.30	83.60±0.60	**84.40±0.04**
Citeseer	71.80±0.70	73.01±0.30	72.61±0.70	71.72±0.62	71.10±0.80	68.70±0.30	**73.63±0.22**	71.82±0.45	71.89±0.63	71.94±0.57	67.14±1.87	73.20±0.50	**73.32±0.41**
Pubmed	76.80±0.60	80.11±0.22	79.41±0.31	79.51±1.10	79.60±0.50	79.71±0.21	**82.02±0.15**	77.31±0.27	78.09±0.29	81.28±0.62	81.20±1.70	80.00±0.40	**81.32±0.22**
Computer	83.95±0.47	82.21±0.34	87.52±0.11	86.51±0.32	89.69±0.37	89.90±0.31	89.80±0.25	86.54±0.45	79.05±0.32	76.81±0.79	- -	**90.40±0.20**	**90.38±0.86**
Photo	91.61±0.22	90.72±0.21	91.72±0.10	92.50±0.22	92.90±0.30	93.25±0.33	93.31±0.19	91.45±0.35	86.46±0.41	93.05±0.23	89.20±1.20	**94.10±0.20**	**94.27±0.85**
Ogbn-arxiv	70.32±0.25	OOM	OOM	65.10±0.50	**71.64±0.24**	OOM	**71.72±0.26**	OOM	OOM	69.29±0.01	- -	70.60±0.20	70.46±0.06
Avg. rank.	9.00	6.40	6.83	8.83	5.83	6.00	**2.00**	8.67	9.40	6.33	9.75	**3.50**	**2.00**

TABLE IV
NODE CLASSIFICATION RESULTS (%) ACROSS DIFFERENT DATASETS AND PARAMETER CONFIGURATIONS.

Method	Cora	Cornell
(i) Two-view	83.56±0.22	**65.14±0.54**
(ii) Three-view	**83.67±0.23**	62.16±0.38
(iii) Five-view	83.61±0.25	63.51±0.82
(iv) AVLA	**84.40±0.04**	**72.70±0.57**
$\{\tilde{\alpha}_k\}_{k=1}^{\tilde{K}}$	$\{0.001, 0.529, 0.736, 1\}$	$\{0.001, 0.018, 0.137, 1\}$

TABLE V
CLASSIFICATION ACCURACY W.R.T DIFFUSION DEPTHS (T) ACROSS DIFFERENT DATASETS.

T	5	15	20	25
Cora	81.48±0.12	83.38±0.37	84.20±0.26	84.14±0.07
Ogbn-arxiv	63.84±0.12	66.43±0.14	66.46±0.08	69.76±0.04
Cornell	62.97±0.52	63.70±0.71	68.38±0.54	67.97±0.82
Wisconsin	71.57±0.19	74.12±0.54	78.04±0.08	78.06±0.12

complex patterns. As verified in Fig. 6a, increasing the feature dimension consistently enhances performance across both homophilic and heterophilic graphs, with particularly notable gains on heterophilic graphs.

3) Diffusion depths: As observed in [15], increasing the diffusion depth T improves the solution accuracy of (2) and generally enhances view diversity. Empirical results in Table V validate that larger T (up to a threshold) yields better performance on both homophilic and heterophilic datasets. The selected T values per dataset are listed in Appendix C-B.

4) Different options for $\mathcal{F}$: We further evaluate FD-MVGCL under alternative formulations of $\mathcal{F}$ in (2), including CDE, GREAD and GraphCON. As reported in Table VI, FD-MVGCL consistently achieves competitive performance across these variants, demonstrating its generality and adaptability to diverse diffusion dynamics.

TABLE VI
NODE CLASSIFICATION RESULTS (%) ACROSS DIFFERENT MODELS IN FD-MVGCL.

	Model	Cora	Cornell	Roman
FD-MVGCL	GRAND [7]	84.40±0.04	72.70±0.57	72.47±0.70
	CDE [46]	75.02±0.26	60.54±3.92	OOM
	GREAD [47]	81.02±0.15	66.24±0.67	71.48±0.36
	GraphCON [11]	80.64±0.33	67.03±0.33	68.20±0.65

5) Loss functions: To assess the performance of FD-MVGCL across various contrastive learning paradigms, we compare several widely adopted *contrastive loss* functions and their impact on two FDE encoder feature representations, where the local view $\mathbf{Y}_l$ is obtained with fractional order $\alpha_l = 0.01$ and and the global view $\mathbf{Y}_g$ with $\alpha_g = 1$. We measure classification accuracy on two benchmark datasets: Cora (homophilic) and Wisconsin (heterophilic). The loss functions considered include Euclidean loss, Cosmean loss, Barlow Twins loss [40], VICReg loss [48], CCA [34], and our proposed Regularized Cosmean loss. As illustrated in Fig. 5, the Regularized Cosmean loss consistently outperforms the others, demonstrating stable accuracy throughout the training epochs. This robustness is attributed to its effectiveness in mitigating dimensional collapse, which ensures reliable performance over extended training periods. Detailed definitions of these loss functions and additional results on other datasets can be found in Appendix C-C.

TABLE VII
BLACK-BOX ATTACK ROBUST ACCURACY (%) UNDER 20% PERTURBATION FOR NODE CLASSIFICATION.

Dataset	Attack	FROND	GCL-Jac	Ariel	Res-GRACE	GraphACL	PolyGCL	LOHA	EPAGCL	FD-MVGCL
Cora	*clean*	81.25±0.89	76.80±0.90	79.80±0.60	79.70±1.00	**84.20±0.31**	82.74±0.14	81.22±0.17	82.10±1.30	**84.40±0.04**
	Random	78.65±1.15	74.10±2.00	76.30±0.60	**78.89±1.17**	76.92±0.14	78.75±0.77	75.74±1.14	75.33±0.34	**79.16±0.12**
	PRBCD	79.17± 0.57	75.30±1.20	75.60±0.20	78.80±1.16	**80.29±2.13**	79.78±1.08	79.70±0.35	77.94±1.11	**81.06±1.34**
	Metattack	73.42±0.97	67.49±2.23	63.50±1.50	67.82±1.36	53.24±2.66	70.11±1.53	**74.69±0.41**	70.65±1.27	**80.28±0.77**
	Nettack	80.11±1.22	72.90±1.39	70.30±0.90	76.36±1.47	**81.39±0.73**	81.45±0.47	79.26±0.86	79.54±0.66	**82.05±0.55**
Citeseer	*clean*	71.37±1.34	70.52±0.65	50.89±3.76	71.72±0.62	**73.60±0.70**	71.82±0.45	71.95±0.45	71.94±0.57	**73.32±0.41**
	Random	70.23±1.40	57.26±4.20	44.98±3.45	56.69±2.63	68.13±0.44	**71.58±0.24**	**71.70±0.29**	63.10±1.08	71.23±0.55
	PRBCD	71.47±1.29	58.30±4.11	46.02±3.16	58.86±266	70.52±1.16	71.19±0.61	**71.60±0.63**	64.54±2.00	**71.57±1.17**
	Metattack	67.94±1.42	57.51±5.21	36.68±3.76	36.20±5.62	20.50±0.28	**71.78±0.42**	42.99±4.02	47.24±2.67	**68.99±2.62**
	Nettack	70.05±1.10	59.40±4.17	46.45±3.16	58.18±2.65	**71.93±1.10**	70.33±0.50	71.01±0.34	65.27±1.21	**71.09±1.01**
Cornell	*clean*	**63.24±9.38**	42.97±6.78	51.89±6.71	51.08±5.19	59.33±1.48	44.10±1.12	55.35±1.35	53.78±3.72	**72.70±0.57**
	Random	**63.24±7.27**	37.30±4.49	40.00±4.95	49.19±4.15	42.97±8.10	43.78±5.14	45.68±3.51	54.32±6.33	**64.62±0.41**
	PRBCD	**64.86±5.27**	41.62±9.83	48.38±6.33	48.92±5.98	46.22±9.66	44.59±4.05	51.08±3.24	53.24±6.74	**67.32±1.50**
	Metattack	**67.03±5.51**	38.65±6.63	49.73±7.85	49.73±6.07	45.14±6.87	42.43±4.87	48.11±5.14	55.68±5.86	**67.95±0.53**
	Nettack	**66.49±6.53**	41.08±7.03	50.54±6.95	49.19±5.24	49.73±7.45	43.78±3.24	52.43±3.51	51.89±3.59	**72.62±0.82**
Actor	*clean*	**35.08±1.08**	29.25±1.21	24.36±1.11	30.72±0.72	30.03±0.13	34.37±0.69	33.69±0.73	30.02±0.91	**35.91±0.92**
	Random	**35.15±0.78**	27.59±1.12	25.64±1.02	30.16±1.09	28.36±1.95	25.41±0.72	34.19±0.59	28.92±1.03	**35.29±0.82**
	PRBCD	**35.04±0.90**	27.76±1.66	24.95±0.89	30.48±1.28	28.37±1.95	27.21±0.64	26.23±0.79	28.66±2.01	**35.49±1.13**
	Metattack	**32.63±0.60**	28.00±1.10	25.54±0.75	30.34±1.04	28.45±1.26	28.29±0.42	26.97±0.65	29.65±1.12	**33.02±0.10**
	Nettack	**34.97±0.88**	28.87±0.73	25.51±0.95	30.86±0.96	28.60±1.20	25.96±0.86	27.20±0.74	30.05±0.81	**35.36±1.31**

TABLE VIII
WHITE-BOX ATTACK ROBUST ACCURACY (%) UNDER 15% PERTURBATION FOR NODE CLASSIFICATION.

Dataset	Attack	FROND	Res-GRACE	GraphACL	PolyGCL	LOHA	EPAGCL	SDMG	FD-MVGCL
Photo	*clean*	92.03±1.27	92.50±0.17	93.31±0.19	91.45±0.35	86.46±0.41	93.05±0.23	**94.10±0.20**	**94.27±0.85**
	PGD	14.18±5.16	45.46±2.05	30.94±3.62	22.93±2.73	**59.35±0.43**	5.02±2.85	11.98±0.81	**92.58±0.64**
	PRBCD	15.08±7.52	40.72±3.70	28.56±1.16	15.86±1.99	**52.82±3.45**	9.91±1.73	8.71±6.22	**79.24±0.84**
Texas	*clean*	**74.86±3.21**	54.59±5.51	71.08±0.34	72.51±1.95	67.75±2.70	68.92±4.05	53.60±2.67	**80.19±0.81**
	PGD	**69.46±7.16**	28.85±8.55	21.62±7.20	57.84±14.05	47.30±16.45	13.78±6.78	30.49±4.32	**76.49±4.84**
	PRBCD	**65.14±3.91**	26.49±11.22	23.78±8.99	51.35±10.26	32.70±10.50	14.32±6.29	19.43±4.39	**71.62±4.87**

D. Robustness studies

We evaluate the robustness of FD-MVGCL through node classification under both black-box and white-box adversarial attacks on standard benchmarks, including homophilic (Cora, Citeseer, Photo) and heterophilic (Cornell, Actor, Texas) graphs. FD-MVGCL is compared against nine baselines: a robust supervised methods (FROND [15]), three robust GCL methods (GCL-Jac [49], Ariel [50], Res-GRACE [51]), and five state-of-the-art GCL models (GraphACL, PolyGCL, LOHA, EPAGCL, SDMG).

1) Attack methods: We consider four *black-box* topology attacks in the evasion setting: Random, PRBCD [52], Nettack [53], and Metattack [54]. We also evaluate two *white-box* attacks (i.e., PGD [55] and PRBCD) that jointly perturb the graph structure and node features. All models are trained on clean graphs, with adversarial perturbations applied only during inference. The details of these attack methods are as follows:

a) Random attack: Randomly adds noisy edges by selecting node pairs across the graph. The number of added edges is controlled by a perturbation ratio relative to the original edge count.

b) PRBCD: Projected Randomized Block Coordinate Descent attack is a structure-based attack that perturbs the adjacency matrix $\mathbf{A}$ by iteratively adding or removing edges to maximize the classification loss of a surrogate GNN (e.g., GCN). It uses a projected randomized block coordinate descent strategy under a fixed budget of edge modifications, enabling efficient and scalable adversarial perturbations. In the white-box setting, PRBCD further extends to jointly perturb node features by leveraging full access to model parameters and gradients.

c) Netttack: A targeted structure-based attack that perturbs the graph to mislead node classification. It estimates homophily and heterophily probabilities, removing edges to same-class nodes to reduce classification confidence and adding edges to different-class nodes to induce misclassification. Guided by a surrogate GNN, it greedily selects the most influential edges to modify within a given budget.

d) Metattack: A global structure-based attack that perturbs the adjacency matrix by computing meta-gradients of a surrogate GNN. It modifies the graph structure to maximize the overall classification loss, effectively degrading performance across the entire graph.

e) PGD: Projected Gradient Descent is a white-box attack that simultaneously perturbs the graph structure and node features to maximize the classification loss of the target model.

TABLE IX
COMPARISON OF TRAINING TIME (S), INFERENCE TIME (S), AND GPU MEMORY USED (MIB) ACROSS DIFFERENT DATASETS.

Method	Training Time (s)				Inference Time (s)				GPU (MiB)			
	Cora	Wisconsin	Roman	Ogbn-arxiv	Cora	Wisconsin	Roman	Ogbn-arxiv	Cora	Wisconsin	Roman	Ogbn-arxiv
GraphACL	0.22	0.63	248.38	927.48	13.29	38.69	1215.34	44.59	326	204	11999	6114
PolyGCL	0.34	0.23	OOM	OOM	7.46	9.73	OOM	OOM	4098	894	OOM	OOM
EPAGCL	0.13	0.12	0.97	OOM	2.81	2.63	3.14	OOM	649	329	22127	OOM
GraphECL	0.02	0.27	60.58	OOM	0.50	0.04	0.04	OOM	112	336	2306	OOM
SDMG	0.06	0.54	0.36	OOM	1.22	0.28	0.55	OOM	2888	824	3676	OOM
FD-MVGCL	0.37	0.29	18.04	1.69	1.47	1.41	0.18	20.06	1330	906	16490	18207

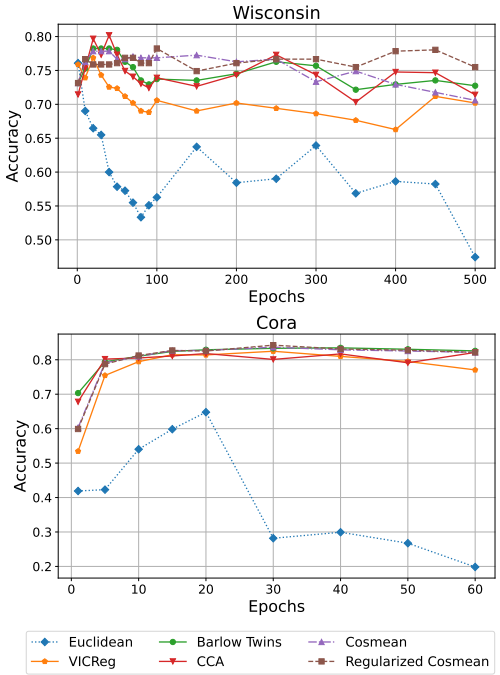


Fig. 5. Accuracy vs. training epochs for various loss functions on the Wisconsin and Cora datasets.

It performs iterative gradient-based updates within a predefined perturbation budget, projecting the modifications back to the feasible space after each step. By leveraging full access to model parameters and gradients, PGD enables strong and precise attacks.

2) Robustness results: We evaluate FD-MVGCL's robustness by varying the perturbation rate from 0% to 25% in 5% increments, using random attacks on Photo (Fig. 6b) and Metattack on Cora (Fig. 6c). FD-MVGCL exhibits the smallest performance degradation and consistently outperforms strong baselines such as GraphACL and PolyGCL. This robustness is attributed to the stability properties of fractional-order diffusion: smaller α_k values lead to reduced feature discrepancies as shown in (9), rendering the diffusion process more resistant to perturbations when $\alpha_k < 1$. The results in Tables VII and VIII further confirm FD-MVGCL's superior robustness compared to baseline methods, particularly under more challenging white-box attack scenarios.

E. Complexity analysis

The training time complexity of FD-MVGCL consists of solving the FDE encoders and computing the contrastive loss. For a graph with N nodes and $|\mathcal{E}|$ edges, each FDE is solved over $E := T/h$ time steps, where h is the discretization step size and T is the total integration time. At each step, the function $\mathcal{F}(\mathbf{W}, \mathbf{Y}_j)$ is evaluated and cached. Assuming $\mathcal{F}$ follows the GRAND model [7], each evaluation costs $C = |\mathcal{E}|d$, where d is the feature dimension. Solving one FDE costs $\mathcal{O}(EC + E\log E)$, with $\log E$ term from fast convolution. For $\tilde{K}$ FDEs, the total cost is $\mathcal{O}(\tilde{K}(EC + E\log E))$. The loss computation adds an additional cost $\mathcal{O}(\tilde{K}N)$, resulting in a total training complexity of $\mathcal{O}(\tilde{K}(EC + E\log E + N))$.

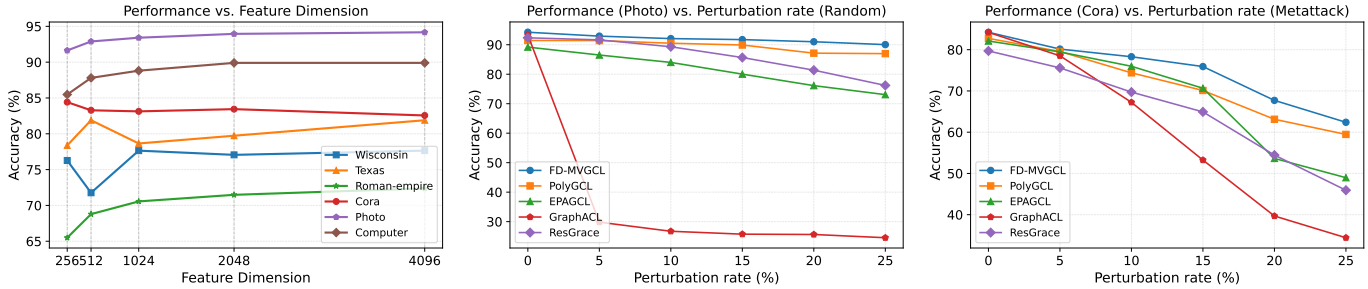
Training/inference time and GPU memory (during training) comparisons with baselines (e.g., GraphACL, PolyGCL and EPAGCL) are shown in Table IX. Although FD-MVGCL requires more GPU memory on large graphs (e.g., Roman and Ogbn-arxiv) due to maintaining intermediate states for multiple FDE solvers, it consistently avoids out-of-memory (OOM) issues and achieves lower overall training and inference costs compared with strong baselines. Moreover, apart from the orders α, each diffusion module of FD-MVGCL introduces *no other* learnable parameters, thereby contributing to its relatively low computational cost.

F. Graph classification results

While most GCL methods target node-level representation learning and do not provide a straightforward graph-level extension, we assess FD-MVGCL's generality by applying a simple, non-parametric readout (MeanPooling) to obtain graph-level embeddings. We evaluate this configuration on two standard graph-classification benchmarks, Proteins and DD, and compare against recent **graph-level** GCL models such as DRGCL and CI-GCL as well as strong node-level baselines adapted to the graph-level setting (e.g., GraphACL). As reported in Table X, FD-MVGCL surpasses all node-level baselines including GraphACL and achieves results on par with **specialized** graph-level contrastive methods, demonstrating that the proposed fractional-order multi-view encoders generalize effectively beyond node-level tasks.

VI. CONCLUSION

We have introduced FD-MVGCL, an adaptive multi-view, augmentation-free graph contrastive learning framework that leverages fractional-order graph neural diffusion to generate



(a) Accuracy vs. feature dimension (b) Accuracy vs Random attack rate (c) Accuracy vs. Metattack rate

Fig. 6. Comparison across feature dimension, Metattack, and Random attack

TABLE X
GRAPH CLASSIFICATION RESULTS (%). THE TOP 3 METHODS ARE ADAPTED FROM NODE-LEVEL GCL MODELS, WHILE THE NEXT 7 MODELS ARE SPECIALIZED GRAPH-LEVEL GCL MODELS.

Method	Proteins	DD
MVGRL [45]	74.02±0.30	75.20±0.40
GraphACL [28]	73.50±0.70	- -
SDMG [33]	73.16±0.16	72.66±3.16
InfoGraph [56]	74.44±0.40	72.85±1.70
GraphCL [57]	74.39±0.45	**78.62±0.40**
JOAO [58]	74.55±0.41	77.32±0.54
JOAO2 [58]	75.35±0.09	77.40±1.11
SimGRACE [59]	75.35±0.09	77.44±1.11
DRGCL [60]	75.20±0.60	78.40±0.70
CI-GCL [61]	**76.50±0.10**	**79.63±0.30**
FD-MVGCL	**75.42±0.29**	78.45±0.43

diverse, multi-scale representations. By treating the fractional order α as a learnable parameter, our approach dynamically adapts diffusion scales to the data, eliminating the need for handcrafted augmentations, negative sampling, or manual tuning. Extensive experiments demonstrate that FD-MVGCL achieves state-of-the-art performance across both homophilic and heterophilic benchmarks, while maintaining superior robustness under adversarial perturbations. This work establishes fractional-order dynamics as a principled and effective paradigm for multi-view contrastive learning on graphs. While FD-MVGCL achieves strong performance, it assumes a static graph structure. Future work will focus on extending our framework to dynamic graphs, enabling fractional diffusion to model evolving temporal and topological patterns.

APPENDIX A
FRACTIONAL-ORDER DERIVATIVES AND GNNS

The works [14], [15] extend the ODE-based approaches by incorporating graph neural FDE, generalizing the order of the derivative to positive real number α. The motivation is the solution to an FDE that encodes historical information and thus retains the "memory" of the evolution.

A. Fractional-order derivatives

Recall that the first-order derivative of a scalar function $f(t)$ is defined as the rate of change of f at a time t:

$$f'(t) = \frac{\mathrm{d}f(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{f(t+\Delta t) - f(t)}{\Delta t}$$

It is local in the sense that $f'(t)$ is determined by the value of f in a small neighborhood of t.

Fractional-order derivatives D_t^α generalize integer derivatives and the domain of α is extended to any positive real number. By the relation $D_t^{\alpha+1} = D_t^\alpha D_t^1$, it suffices to define D_t^α for the order parameter $\alpha \in (0,1)$. For the encoder design in this paper, we only consider $\alpha \in (0,1]$. The domain is large enough for us to choose encoders that generate different views. Moreover, solving a higher-order FDE requires additional information such as the initial derivative, which is usually not available.

Although there are different approaches to defining D_t^α, they all share the common characterization that the derivative is defined using an integral. Hence, the operator is "global" as historical values of the function are used. We show two approaches below.

The *left fractional derivative* [62] of $f(t)$ is defined by the following integral

$$D_t^\alpha f(t) = \frac{1}{\Gamma(-\alpha)} \int_0^\infty \frac{f(t-\tau) - f(t)}{\tau^{1+\alpha}} \, \mathrm{d}\tau.$$

On the other hand, the *Caputo fractional derivative* [63] is defined as

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

One can find other approaches and more discussions in [64].

B. Incorporating FDE

To build a diffusion model based on FDE, it usually suffices to replace the integer order derivative with a fractional-order derivative in the governing differential equation. For example, the ODE in the GraphCON model [11] is equivalent to a system of equations:

$$\frac{\mathrm{d}\mathbf{Z}(t)}{\mathrm{d}t} = \sigma\left(\mathbf{F}_\theta(\mathbf{Y}(t), t)\right) - \gamma \mathbf{Y}(t) - \nu \mathbf{Z}(t)$$
$$\frac{\mathrm{d}\mathbf{Y}(t)}{\mathrm{d}t} = \mathbf{Z}(t).$$

Replacing $\mathrm{d}/\mathrm{d}t$ with D_t^α, one obtains its FDE version as

$$D_t^\alpha \mathbf{Z}(t) = \sigma\left(\mathbf{F}_\theta(\mathbf{Y}(t), t)\right) - \gamma \mathbf{Y}(t) - \nu \mathbf{Z}(t)$$
$$D_t^\alpha \mathbf{Y}(t) = \mathbf{Z}(t).$$

A numerical solver is provided in [15].

Appendix B
Theoretical discussions

A. Formal version of Theorem 1

In this section, we provide a rigorous discussion of the formal version of Theorem 1. To do this, we need the Mittag-Leffler functions, which were introduced by Mittag-Leffler in 1903 in the form of a Maclaurin series. We focus on a special case defined as follows:

$$e_\alpha(\lambda, t) = \sum_{n=0}^{\infty} (-1)^n \frac{\lambda^n t^{\alpha n}}{\Gamma(\alpha n + 1)}, \quad \lambda > 0,\ t \geq 0,$$

where $\Gamma(\cdot)$ is the Gamma function.

We make the following *assumptions* regarding the encoder. The function $\mathcal{F}$ in (2) is the simple $-\overline{\mathbf{L}}\mathbf{X}$ as described after (1). For D_t^α, we consider the left fractional derivative. For the model, we apply skip-connection after a diffusion of length τ, and there are m-iterations so that the total time is $T = m\tau$. We shall consider $\alpha \in (0, 1)$, and this a mild assumption, as we have seen that $D_t^1 = \mathrm{d}/\mathrm{d}t$ is the limit of D_t^α when $\alpha \to 1$. The following formal version of Theorem 1 analyzes the (asymptotic) properties of the Fourier coefficients of the output features.

Theorem 4. *Suppose $\mathcal{G}$ is a connected graph and $0 < \alpha_l < \alpha_g \leq 1$ are the order parameters. For subscript s stands for either l or g, let $n_s \geq 1$ be the integer such that $n_s\alpha_s < 1 \leq (n_s+1)\alpha_s$. Consider a graph signal $\mathbf{x}$ with Fourier coefficients $\{c_i, 1 \leq i \leq N\}$. Let $\mathbf{z}_{\alpha_s}(T)$ be the output features of the encoders with order parameters α_s. Its Fourier coefficients are $\{c_{\alpha_s,i}(T), 1 \leq i \leq N\}$. Then the following holds:*

(a) We have the expression:

$$c_{\alpha_s,i}(T) = \left(\sum_{j=0}^{n_s} b_{\alpha_s,i,j}\tau^{-j\alpha_s} + O\left(\frac{1}{\tau}\right)\right) c_i,$$

and $b_{\alpha_s,i,j} > 0$.

(b) For fixed l and j, the coefficient $b_{\alpha_s,i,j}$ is decreasing w.r.t. i.

(c) For fixed i and j, we have $b_{\alpha_l,i,j} > b_{\alpha_g,i,j}$.

Proof. Let $\overline{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\intercal$ be the orthogonal eigendecomposition of $\overline{\mathbf{L}}$. As $\overline{\mathbf{L}}$ is positive semi-definite, its (ordered) eigenvalues satisfy $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \ldots \leq \lambda_N$. Notice we have $\lambda_2 > 0$ as $\mathcal{G}$ is connected. To solve the FDE $D_t^\alpha \mathbf{z}_\alpha(t) = -\overline{\mathbf{L}}\mathbf{z}_\alpha(t)$, we perform eigendecomposition:

$$D_t^\alpha \mathbf{z}_\alpha(t) = -\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\intercal \mathbf{z}_\alpha(t),$$

and hence

$$D_t^\alpha \mathbf{U}^\intercal \mathbf{z}_\alpha(t) = -\mathbf{\Lambda}\mathbf{U}^\intercal \mathbf{z}_\alpha(t).$$

Therefore, the spectral decomposition of the dynamics $\mathbf{z}_\alpha(t)$ follows the equation $D_t^\alpha \mathbf{c}_\alpha(t) = -\mathbf{\Lambda}\mathbf{c}_\alpha(t)$, where the i-th entry of $\mathbf{c}_\alpha(t)$ is $c_{\alpha,i}(t)$.

For consistency, we set $e_\alpha(0, t) = 1$ the constant function independent of α. Therefore, by [62], we have the solution $c_{\alpha,i}(t) = e_\alpha(\lambda, t)c_{\alpha,i}(0), 1 \leq i \leq N$, where $c_{\alpha,i}(0) = c_i$ depends only on the input feature. Recall we perform diffusion for a time τ followed by a skip-connection for m iterations. The coefficients of the spectral decomposition of the final output features are

$$\begin{aligned} c_{\alpha,i}(T) &= \Big(1 + \ldots + e_\alpha(\lambda, \tau)\big(1 + e_\alpha(\lambda, \tau)\big)\Big) c_{\alpha,i}(0) \\ &= \Big(1 + e_\alpha(\lambda, \tau) + \ldots + e_\alpha(\lambda, \tau)^m\Big) c_{\alpha,i}(0). \end{aligned} \tag{10}$$

The case $c_{\alpha,i}(0) = c_i = 0$ is trivial as we can set $b_{\alpha_s,i,j}$ to be any positive number. We assume $c_{\alpha,i}(0) \neq 0$ for the rest of the proof, and want to estimate $1 + e_\alpha(\lambda, \tau) + \ldots + e_\alpha(\lambda, \tau)^m$ for $\alpha = \alpha_l$ and α_g.

By [65], if $\lambda_i > 0$, $e_{\alpha_s}(\lambda_i, \tau)$ satisfies the following asymptotic estimation:

$$e_{\alpha_s}(\lambda_i, \tau) = \sum_{j=1}^{n_s} \frac{1}{\lambda_i^j \Gamma(1 - j\alpha_s)} \tau^{-j\alpha_s} + O\left(\frac{1}{\tau}\right). \tag{11}$$

The Gamma function $\Gamma(\cdot)$ is positive and strictly decreasing on the interval $(0, 1]$ (e.g., $\Gamma(1) = 1! = 1$). Therefore, as $1 - j\alpha_s > 0$, the coefficient of $\tau^{-j\alpha_s}$ in (11) is positive. Moreover, for s denoting the subscripts l and g and $i \geq 2$, we have

$$\begin{aligned} \frac{1}{\lambda_i^j \Gamma(1 - j\alpha_l)} &> \frac{1}{\lambda_i^j \Gamma(1 - j\alpha_g)} \quad \text{and} \\ \frac{1}{\lambda_i^j \Gamma(1 - j\alpha_s)} &\geq \frac{1}{\lambda_{i+1}^j \Gamma(1 - j\alpha_s)}. \end{aligned} \tag{12}$$

Now to compare the solution for the subscript s being l and g, we express $e_{\alpha_l}(\lambda_i, \tau)^k$ as follows

$$\begin{aligned} e_{\alpha_l}(\lambda_i, \tau)^k = \Big(&\sum_{j=1}^{n_g} \frac{1}{\lambda_i^j \Gamma(1 - j\alpha_l)} \tau^{-j\alpha_l} \\ &+ \sum_{j=n_g+1}^{n_l} \frac{1}{\lambda_i^j \Gamma(1 - j\alpha_l)} \tau^{-j\alpha_l}\Big)^k + O\left(\frac{1}{\tau}\right). \end{aligned}$$

As compared with

$$e_{\alpha_g}(\lambda_i, \tau)^k = \Big(\sum_{j=1}^{n_g} \frac{1}{\lambda_i^j \Gamma(1 - j\alpha_g)} \tau^{-j\alpha_g}\Big)^k + O\left(\frac{1}{\tau}\right),$$

we see that up to $O(\frac{1}{\tau})$, $e_{\alpha_g}(\lambda_i, \tau)^k$ and hence $c_{\alpha_g,i}(T)/c_{\alpha_g,i}(0)$ (cf. (10)) consists of a summation of terms $\tau^{-j\alpha_g}$ for $1 \leq j \leq n_g$, and the positive coefficient $\tau^{-j\alpha_g}$ is smaller than that of $\tau^{-j\alpha_l}$ in $c_{\alpha_l,i}(T)/c_{\alpha_l,i}(0)$. Moreover, $c_{\alpha_l,i}(T)/c_{\alpha_l,i}(0)$ has addition terms $\tau^{-j\alpha_l}, n_g < j \leq n_l$ with positive coefficients. This proves (a) and (c). From the second equality in (12), we obtain the conclusion that $b_{\alpha_s,i,j}$ is decreasing w.r.t. i, as claimed in (b). □

To explain Theorem 1, t in Theorem 1 corresponds to $T = m\tau$ in Theorem 4, and once the number of iterations

m is fixed, $\tau \to \infty$ if and only if $t \to \infty$. The signal $\mathbf{x}$ in Theorem 4 is a component (i.e., a column) of the feature matrix $\mathbf{X}$. By Theorem 4, the Fourier coefficients $c_{\alpha_s,i}(t)$ of $\mathbf{Y}_{\alpha_s}(t)$, subscript s being l or g, relative to the Fourier coefficients of the input features, decrease as the frequency index i increases. Moreover, the decrement for $\mathbf{Y}_{\alpha_l}(t)$ is much slower than that of $\mathbf{Y}_{\alpha_g}(t)$ as t increases. The difference in the rate increases accordingly as $\alpha_g - \alpha_l$ becomes larger. As the leading Fourier coefficients ($i = 0$) are both the same ($= m$), the Fourier coefficients of $\mathbf{Y}_{\alpha_l}(t)$ are more spread, whence the claims of Theorem 1.

For most datasets studied in the paper, we choose $\alpha_l \leq 0.1$ and $\alpha_g > 0.5$, and thus $n_g = 1$. By (10) and (11), the following holds for $c_{\alpha_g,i}(T)$:

$$c_{\alpha_g,i}(T)/c_{\alpha_g,i}(0) = 1 + \frac{1}{\lambda_i \Gamma(1-\alpha_g)} \tau^{-\alpha_g} + O\left(\frac{1}{\tau}\right).$$

For $c_{\alpha_l,i}(T)/c_{\alpha_l,i}(0)$, we can identify a summand $1/\big(\lambda_i \Gamma(1-\alpha_l)\big) \cdot \tau^{-\alpha_g}$. The ratio between $1/\big(\lambda_i \Gamma(1-\alpha_l)\big) \cdot \tau^{-\alpha_l}$ and $1/\big(\lambda_i \Gamma(1-\alpha_g)\big) \cdot \tau^{-\alpha_g}$ is at least $\sqrt{\pi}/1.07 \cdot \tau^{0.4}$. This gives us some understanding of how these two sets of Fourier coefficients differ.

B. Stability

1) Proof of Theorem 2: By taking the difference between the original and perturbed dynamics, we obtain the following equation:

$$D_t^\alpha(\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)) = \mathcal{F}(\mathbf{W}, \mathbf{Y}(t)) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \tilde{\mathbf{Y}}(t)).$$

Integrating both sides with the fractional integral operator I_t^α and then taking norms gives

$$\begin{aligned} &\|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| \\ &= \left\| \mathbf{Y}(0) - \tilde{\mathbf{Y}}(0) + I_t^\alpha(\mathcal{F}(\mathbf{W}, \mathbf{Y}(t)) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \tilde{\mathbf{Y}}(t))) \right\| \\ &\leq \|\mathbf{Y}(0) - \tilde{\mathbf{Y}}(0)\| \\ &\quad + I_t^\alpha \|\mathcal{F}(\mathbf{W}, \mathbf{Y}(t)) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \tilde{\mathbf{Y}}(t))\|. \end{aligned} \tag{13}$$

Using the triangle inequality, we bound the integrand of the second term on the right-hand side of (13) as follows:

$$\begin{aligned} &\|\mathcal{F}(\mathbf{W}, \mathbf{Y}(t)) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \tilde{\mathbf{Y}}(t))\| \\ &\quad \leq \|\mathcal{F}(\mathbf{W}, \mathbf{Y}(t)) - \mathcal{F}(\mathbf{W}, \tilde{\mathbf{Y}}(t))\| \\ &\quad\quad + \|\mathcal{F}(\mathbf{W}, \tilde{\mathbf{Y}}(t)) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \tilde{\mathbf{Y}}(t))\|. \end{aligned} \tag{14}$$

Given that $\mathcal{F}$ is Lipschitz continuous with constant L, it follows that:

$$\begin{aligned} &\|\mathcal{F}(\mathbf{W}, \mathbf{Y}(t)) - \mathcal{F}(\mathbf{W}, \tilde{\mathbf{Y}}(t))\| \\ &\quad \leq L\|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\|. \end{aligned} \tag{15}$$

By definition of the combined perturbation ε, the second term on the right-hand side of (14) is bounded as follows:

$$\begin{aligned} &\|\mathcal{F}(\mathbf{W}, \tilde{\mathbf{Y}}(t)) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \tilde{\mathbf{Y}}(t))\| \\ &\quad \leq \sup_{\mathbf{Y} \in A} \|\mathcal{F}(\mathbf{W}, \mathbf{Y}) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \mathbf{Y})\| \leq \varepsilon. \end{aligned} \tag{16}$$

Combining (13), (15) and (16), we have the inequality

$$\begin{aligned} &\|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| \\ &\quad \leq LI_t^\alpha \|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| + (1 + \frac{t^\alpha}{\Gamma(1+\alpha)})\varepsilon \\ &\quad \leq LI_t^\alpha \|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| + (1 + \frac{T^\alpha}{\Gamma(1+\alpha)})\varepsilon, \end{aligned} \tag{17}$$

where we have used the assumption that the initial perturbation satisfies $\|\mathbf{Y}_0 - \tilde{\mathbf{Y}}_0\| \leq \varepsilon$. Applying the fractional Grönwall inequality [63, Lemma 6.19.] to (17), we obtain

$$\|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| \leq \left(1 + \frac{T^\alpha}{\Gamma(1+\alpha)}\right) \varepsilon e_\alpha(-L, t^\alpha). \tag{18}$$

Since the Mittag-Leffler function $e_\alpha(-L, t^\alpha)$ can be bounded by $O\left(\frac{1}{1+LT^\alpha}\right)$, it follows from (18) that

$$\begin{aligned} \|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| &\leq \left(1 + \frac{T^\alpha}{\Gamma(1+\alpha)}\right) \varepsilon O\left(\frac{1}{1+LT^\alpha}\right) \\ &= O\left(\varepsilon T^{\alpha-1}\right) \end{aligned} \tag{19}$$

for $t \in [0, T]$. This completes the proof of Theorem 2.

2) Proof of Theorem 3: Subtracting the weighted average solution in (8) and the perturbation one yields

$$\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t) = \sum_{k=1}^{\tilde{K}} \beta_k \left(\mathbf{Y}_k(t) - \tilde{\mathbf{Y}}_k(t)\right). \tag{20}$$

Take the norm of (20) and applying the triangle inequality, we obtain

$$\|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| \leq \sum_{k=1}^{\tilde{K}} \beta_k \|\mathbf{Y}_k(t) - \tilde{\mathbf{Y}}_k(t)\|. \tag{21}$$

By applying Theorem 2 separately for each term $\|\mathbf{Y}_k(t) - \tilde{\mathbf{Y}}_k(t)\|$, we have

$$\|\mathbf{Y}_k(t) - \tilde{\mathbf{Y}}_k(t)\| = O\left(\varepsilon T^{\alpha_k - 1}\right). \tag{22}$$

Thus, the final unified perturbation bound is given by:

$$\begin{aligned} \|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| &\leq \sum_{k=1}^{\tilde{K}} \beta_k O\left(\varepsilon T^{\alpha_k - 1}\right) \\ &= O\left(\varepsilon T^{\alpha_{\tilde{K}} - 1}\right), \end{aligned} \tag{23}$$

for $t \in [0, T]$. This completes the proof of Theorem 3.

Appendix C
Experimental details

A. Details of datasets

We refer the reader to Table XI for detailed statistics of the datasets.

TABLE XI
STATISTICS OF HOMOPHILIC AND HETEROPHILIC GRAPH DATASETS

Dataset	Nodes	Edges	Classes	Node Features	Data splits
Cora	2708	5429	7	1433	standard
Citeseer	3327	4732	6	3703	standard
PubMed	19717	88651	3	500	standard
Computer	13752	574418	10	767	10%/10%/80%
Photo	7650	119081	8	745	10%/10%/80%
Ogbn-arxiv	169343	1166243	40	128	standard
Texas	183	309	5	1793	48%/32%/20%
Cornel	183	295	5	1703	48%/32%/20%
Wisconsin	251	466	5	1703	48%/32%/20%
Squirrel-filtered	2205	46557	5	2089	48%/32%/20%
Chameleon-filtered	864	7754	5	2325	48%/32%/20%
Crocodile	11631	360040	5	2089	48%/32%/20%
Actor	7600	33391	5	932	48%/32%/20%
Roman-empire	22662	32927	18	300	50%/25%/25%
Amazon-ratings	24492	186100	5	300	50%/25%/25%
Arxiv-year	169343	1166243	5	128	50%/25%/25%

B. Hyperparameter choices

We conduct all experiments on a machine equipped with an NVIDIA A5000 GPU with 24 GB of memory. A small grid search is performed to identify the optimal hyperparameters. Specifically, we search over the following ranges: diffusion depth $T \in \{1, 2, 3, 5, 10, 15, 20, 30\}$, step size $h \in \{0.5, 1, 2, 3, 5, 10\}$, and hidden dimension $d \in \{256, 512, 1024, 2048, 4096, 8192\}$. The initial number of encoders is set to $K = 5$, and threshold ϵ and δ in Algorithm 1 are both set to 0.0001. In addition, the learning rate is selected from $\{0.001, 0.005, 0.01, 0.015, 0.2\}$, the weight decay from $\{0.0001, 0.0005, 0.00001\}$, and the regularization weight η from $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$. The combination weight β for aggregating $\mathbf{Y}_k$ is tuned over $[0, 1]$ with step of 0.01. The optimal configuration is chosen based on the average validation accuracy. A detailed summary of the selected hyperparameters and the final set of learnable parameters (e.g., $\tilde{K}$ and corresonding $\{\alpha_k\}_{k=1}^{\tilde{K}}$) for each dataset is provided in Table XII.

C. Contrastive loss functions

In scenarios where explicit negative samples are absent, non-contrastive methods instead optimize agreement between positive pairs. Representative approaches include knowledge-distillation frameworks such as BGRL [4], which relies on cosine similarity, and redundancy-reduction techniques such as Barlow Twins [40] and VICReg [48]. To illustrate how different loss functions are formulated in this context, we consider two feature representations (or "views"), denoted by $\mathbf{Y}_l$ and $\mathbf{Y}_g$. Each consists of N samples, where $\mathbf{Y}_{l,i}$ and $\mathbf{Y}_{g,i}$ represent the feature vectors of the i-th sample in the two views. Based on this setup, the Euclidean loss, Cosine similarity loss, Barlow Twins loss, VICReg loss and CCA loss are defined as follows.

a) Euclidean Loss: measures the squared ℓ_2-norm between two feature representations $\mathbf{Y}_l$ and $\mathbf{Y}_g$, encouraging them to be as close as possible. It is defined as:

$$\mathcal{L}_{\text{Euclidean}} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{Y}_{l,i} - \mathbf{Y}_{g,i}\|_2^2.$$

b) Cosmean Loss: measures the cosine similarity between two feature representations $\mathbf{Y}_l$ and $\mathbf{Y}_g$, encouraging their alignment. It is mathematically expressed as

$$\mathcal{L}_{\text{Cosmean}} = 1 - \frac{1}{N} \sum_{i=1}^{N} \frac{\langle \mathbf{Y}_{l,i}, \mathbf{Y}_{g,i} \rangle}{\|\mathbf{Y}_{l,i}\|_2 \|\mathbf{Y}_{g,i}\|_2} \tag{24}$$

where $\langle \mathbf{Y}_{l,i}, \mathbf{Y}_{g,i} \rangle$ is the inner product of these two vectors $\mathbf{Y}_{l,i}$ and $\mathbf{Y}_{g,i}$, and $\|\mathbf{Y}_{l,i}\|_2$ and $\|\mathbf{Y}_{g,i}\|_2$ are their respective ℓ_2-norms. The cosmean loss is minimized when the feature vectors are perfectly aligned in the same direction, achieving maximum cosine similarity.

c) Barlow Twins Loss: [40] is designed to encourage similarity between two feature representations $\mathbf{Y}_l$ and $\mathbf{Y}_g$, while reducing redundancy across dimensions within each representation. It is defined as:

$$\mathcal{L}_{\text{Barlow Twins}} = \sum_i \left((1 - \mathbf{C}_{ii})^2 \right) + \lambda \sum_i \sum_{i \neq j} \mathbf{C}_{ii}^2$$

where $\mathbf{C} = \frac{\mathbf{Y}_l^\intercal \mathbf{Y}_g}{N}$ is the cross-correlation matrix of the normalized feature representations $\mathbf{Y}_l$ and $\mathbf{Y}_g$, and λ is a trade-off parameter. The first term minimizes the difference between the diagonal elements of $\mathbf{C}$ and 1, ensuring the features from $\mathbf{Y}_l$ and $\mathbf{Y}_g$ are highly correlated. The second term minimizes the off-diagonal elements, promoting decorrelation between features and reducing redundancy. By balancing these two objectives, Barlow Twins Loss enables learning robust and diverse feature representations.

d) VICReg Loss: (Variance-Invariance-Covariance Regularization Loss) [48] is designed to align two feature representations, $\mathbf{Y}_l$ and $\mathbf{Y}_g$, while ensuring variance preservation and minimizing redundancy. It consists of three components:

TABLE XII
DETAILS OF THE HYPERPARAMETERS TUNED BY GRID SEARCH ON VARIOUS DATASETS, ALONG WITH THE FINAL SET OF LEARNABLE PARAMETERS.

Datasets	T	h	d	η	lr	weight decay	epochs	$\tilde{K}$	$\{\alpha_k\}_{k=1}^{\tilde{K}}$	$\{\beta_k\}_{k=1}^{\tilde{K}}$
Cora	20	1	512	0.05	0.01	0.0005	30	4	$\{0.001, 0.529, 0.736, 1\}$	$\{0.45, 0.02, 0.03, 0.5\}$
Citeseer	5	0.5	2048	0.15	0.02	0.0005	20	5	$\{0.009, 0.158, 0.405, 0.653, 1\}$	$\{0.05, 0.05, 0.3, 0.05, 0.55\}$
Pubmed	10	2	8192	0.1	0.01	0.0001	2	5	$\{0.011, 0.259, 0.506, 0.752, 1\}$	$\{0.01, 0.01, 0.08, 0.45, 0.45\}$
Computer	5	0.5	4096	0.01	0.005	0.0005	2	5	$\{0.001, 0.248, 0.515, 0.763, 1\}$	$\{0.05, 0.05, 0.1, 0.65, 0.15\}$
Photo	5	1	4096	0.05	0.005	0.0005	2	5	$\{0.010, 0.258, 0.510, 0.990, 1\}$	$\{0.45, 0.02, 0.02, 0.01, 0.5\}$
Ogbn-arxiv	30	3	256	0.2	0.01	0.0005	50	2	$\{0.01, 1\}$	$\{0.2, 0.8\}$
Squirrel	3	1	4096	0.05	0.01	0.0005	3	5	$\{0.015, 0.561, 0.587, 0.706, 1\}$	$\{0.5, 0.015, 0.015, 0.005, 0.465\}$
Chameleon	3	1	2048	0.3	0.01	0.0005	2	5	$\{0.012, 0.450, 0.641, 0.654, 1\}$	$\{0.45, 0.01, 0.01, 0.03, 0.5\}$
Crocodile	20	2	2048	0.01	0.01	0.0005	20	4	$\{0.001, 0.589, 0.883, 1\}$	$\{0.4, 0.1, 0.1, 0.4\}$
Actor	15	1	2048	0.01	0.01	0.0005	10	5	$\{0.008, 0.354, 0.554, 0.827, 1\}$	$\{0.7, 0.05, 0.05, 0.01, 0.1\}$
Wisconsin	20	2	2048	0.25	0.01	0.0005	30	5	$\{0.001, 0.009, 0.113, 0.184, 1\}$	$\{0.3, 0.5, 0.05, 0.05, 0.1\}$
Cornell	30	5	2048	0.01	0.01	0.0005	50	4	$\{0.001, 0.018, 0.137, 1\}$	$\{0.1, 0.6, 0.25, 0.05\}$
Texas	30	5	2048	0.01	0.01	0.0005	30	4	$\{0.001, 0.063, 0.848, 1\}$	$\{0.1, 0.3, 0.05, 0.55\}$
Roman	2	1	8192	0.001	0.01	0.0005	2	2	$\{0.01, 1\}$	$\{0.5, 0.5\}$
Amazon	3	0.5	8192	0.001	0.001	0.0005	10	2	$\{0.01, 1\}$	$\{0.99, 0.01\}$
Arxiv-year	2	1	1024	0.001	0.01	0.0005	2	5	$\{0.008, 0.240, 0.523, 0.750, 1\}$	$\{0.025, 0.025, 0.075, 0.025, 0.85\}$

variance regularization, invariance loss, and covariance regularization. The loss is expressed as:

$$\mathcal{L}_{\text{VICReg}} = \eta_1 \mathcal{L}_{\text{inv}} + \eta_2 \mathcal{L}_{\text{var}} + \eta_3 \mathcal{L}_{\text{cov}},$$

where η_1, η_2 and η_3 are hyper-parameters controlling the relative contributions of each term. The invariance component $\mathcal{L}_{\text{inv}}$ is computed as the mean-squared Euclidean distance of the corresponding samples from $\mathbf{Y}_l$ and $\mathbf{Y}_g$:

$$\mathcal{L}_{\text{inv}} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{Y}_{l,i} - \mathbf{Y}_{g,i}\|_2^2.$$

The variance component $\mathcal{L}_{\text{var}}$ ensures that each feature dimension in $\mathbf{Y}_l$ and $\mathbf{Y}_g$ has sufficient variance to potentially prevent collapse. It is given by

$$\mathcal{L}_{\text{var}} = \frac{1}{d} \sum_{j=1}^{d} \max(0, \varepsilon - \sqrt{\text{Var}(\mathbf{Y}_l[:,j])}) + \max(0, \varepsilon - \sqrt{\text{Var}(\mathbf{Y}_g[:,j])})$$

where $\mathbf{Y}_l[:,j]$ and $\mathbf{Y}_g[:,j]$ are the j-th feature columns of $\mathbf{Y}_l$ and $\mathbf{Y}_g$, respectively. $\text{Var}(\mathbf{Y}_l[:,j]) = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{Y}_l[i,j] - \mu_{l,j})^2$ is the variance of the j-th feature in $\mathbf{Y}_l$ (with $\mu_{l,j} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{Y}_l[i,j]$), and ε is a small positive constant to enforce nonzero variance. And the covariance regularization term $\mathcal{L}_{\text{cov}}$ reduces redundancy by decorrelating different feature dimensions within each representation. It is expressed as:

$$\mathcal{L}_{\text{cov}} = \frac{1}{d} \sum_{i \neq j} \left(\text{Cov}(\mathbf{Y}_l)_{j,k}^2 + \text{Cov}(\mathbf{Y}_g)_{j,k}^2 \right)$$

where $\text{Cov}(\mathbf{Y}_l)_{j,k} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{Y}_{l,i,j} - \mu_{l,j})(\mathbf{Y}_{l,i,k} - \mu_{l,k})$ represents the off-diagonal elements of the covariance matrix for $\mathbf{Y}_l$ and likewise for $\mathbf{Y}_g$. The hyperparameters η_1, η_2, and η_3 control the relative contributions of variance regularization, invariance loss, and covariance regularization, respectively. By balancing these three terms, VICReg ensures robust feature alignment while maintaining diversity and decorrelation, making it well-suited for unsupervised learning tasks.

e) CCA Loss: (Canonical Correlation Analysis Loss) [34] is self-supervised objective that balances invariance and decorrelation between two augmented views of the same data. Given two normalized feature matrices $\mathbf{Y}_l$ and $\mathbf{Y}_g$, the CCA-style learning loss is defined as

$$\mathcal{L}_{\text{CCA}} = \|\mathbf{Y}_l - \mathbf{Y}_g\|_F^2 + \lambda \left(\|\mathbf{Y}_l^\intercal \mathbf{Y}_l - \mathbf{I}\|_F^2 + \|\mathbf{Y}_g^\intercal \mathbf{Y}_g - \mathbf{I}\|_F^2 \right)$$

where λ is a non-negative trade-off parameter. The first (invariance) term encourages the two augmented views to produce similar representations, thereby maximizing cross-view correlation. The second (decorrelation) term regularizes each view to be approximately uncorrelated, ensuring that different feature dimensions encode diverse and complementary semantic information.

Fig. 7 presents additional classification accuracy results on two benchmark datasets: Cornell and Squirrel, evaluated above various contrastive loss functions. Consistent with the findings in the main text, these results further underscore the effectiveness of our proposed Regularized Cosmean loss. Unlike other loss functions, which tend to experience performance degradation as training progresses, the Regularized Cosmean loss demonstrates superior stability by maintaining consistent accuracy across epochs. These results provide additional evidence of its ability to mitigate dimension collapse and ensure robust performance.

APPENDIX D
T-SNE VISUALIZATIONS OF NODE FEATURES

We present additional t-SNE visualizations of node features for each class in the Cora (homophilic), Wisconsin and Cornell (heterophilic) datasets, shown in Fig. 8, Fig. 9 and Fig. 10 respectively. These plots are generated using encoders with varying fractional-order parameters, highlighting the distinct embedding patterns produced by different diffusion dynamics.

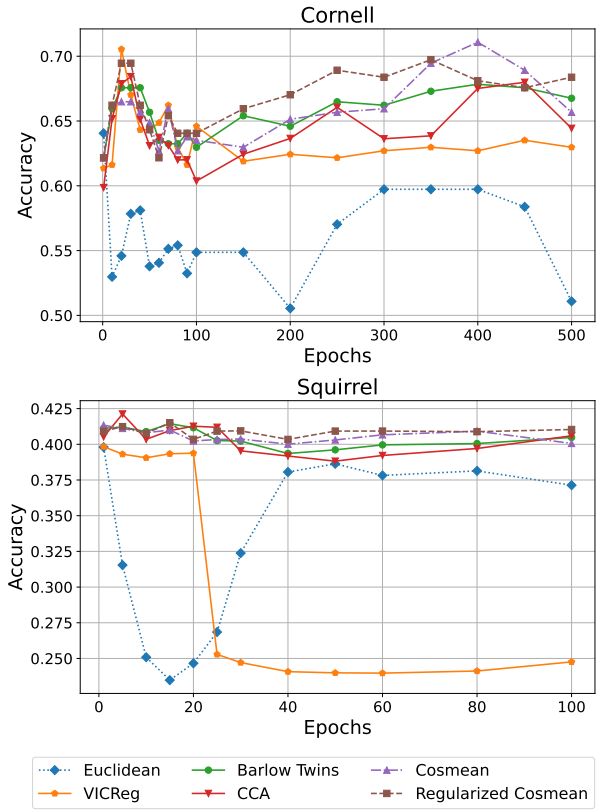


Fig. 7. Accuracy vs. training epochs for various loss functions on the Cornell and Squirrel datasets.

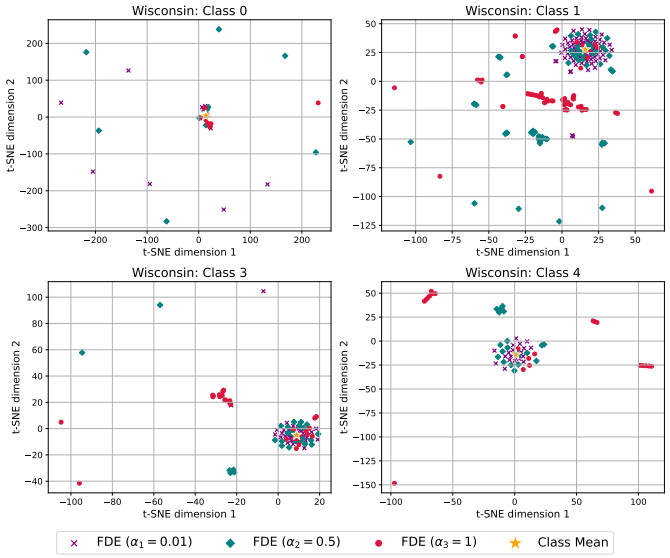


Fig. 8. t-SNE visualizations of Wisconsin node features for different class labels.

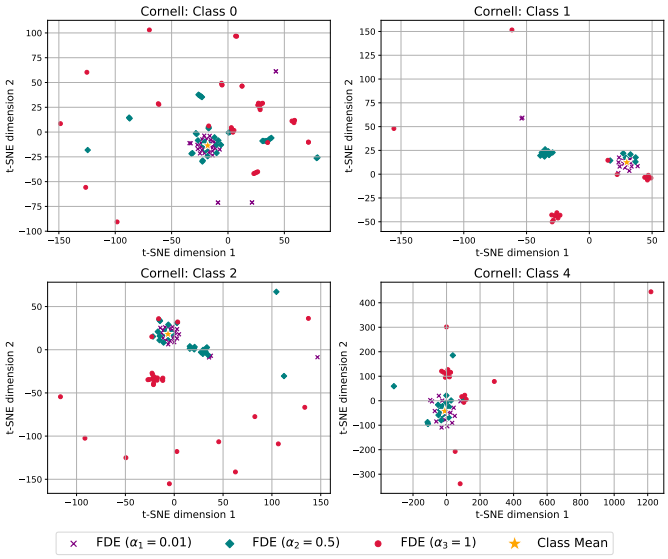


Fig. 9. t-SNE visualizations of Cornell node features for different class labels.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, and J. Clark, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[2] J. B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. Guo, and M. G. Azar, "Bootstrap your own latent-a new approach to self-supervised learning," in *Advances Neural Inf. Process. Syst.*, 2020.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[4] S. Thakoor, C. Tallec, M. G. Azar, M. Azabou, E. L. Dyer, R. Munos, P. Veličković, and M. Valko, "Large-scale representation learning on graphs via bootstrapping," in *Proc. Int. Conf. Learn. Representations*, 2022.

[5] J. Chen, R. Lei, and Z. Wei, "PolyGCL: Graph contrastive learning via learnable spectral polynomial filters," in *Proc. Int. Conf. Learn. Representations*, 2024.

[6] S. M. Sivalingam and V. Govindaraj, "Neural fractional order differential equations," *Expert Systems with Applications*, vol. 267, p. 126041, 2025.

[7] 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.

[8] 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.

[9] B. P. Chamberlain, J. Rowbottom, D. Eynard, F. Di Giovanni, D. Xiaowen, and M. M. Bronstein, "Beltrami flow and neural diffusion on graphs," in *Advances Neural Inf. Process. Syst.*, 2021.

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

[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] 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.*, 2023, pp. 15 786–15 808.

[13] K. Zhao, Q. Kang, Y. Song, R. She, S. Wang, and W. P. Tay, "Adversarial robustness in graph neural networks: A Hamiltonian energy conservation approach," in *Advances Neural Inf. Process. Syst.*, New Orleans, USA, 2023.

[14] K. Zhao, Q. Kang, F. Ji, X. Li, Q. Ding, Y. Zhao, W. Liang, and W. P. Tay, "Distributed-order fractional graph operating network," in *Advances Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2024.

[15] Q. Kang, K. Zhao, Q. Ding, F. Ji, X. Li, W. Liang, Y. Song, and W. P. Tay, "Unleashing the potential of fractional calculus in graph neural networks with FROND," in *Proc. Int. Conf. Learn. Representations*, Vienna, Austria, 2024.

[16] Q. Kang, K. Zhao, Y. Song, Y. Xie, Y. Zhao, S. Wang, R. She, and W. P. Tay, "Coupling graph neural networks with fractional order continuous dynamics: A robustness study," in *Proc. AAAI Conf. Artif. Intell.*, Vancouver, Canada, Feb. 2024.

[17] Q. Kang, X. Li, K. Zhao, W. Cui, Y. Zhao, W. Deng, and W. P. Tay, "Efficient training of neural fractional-order differential equation

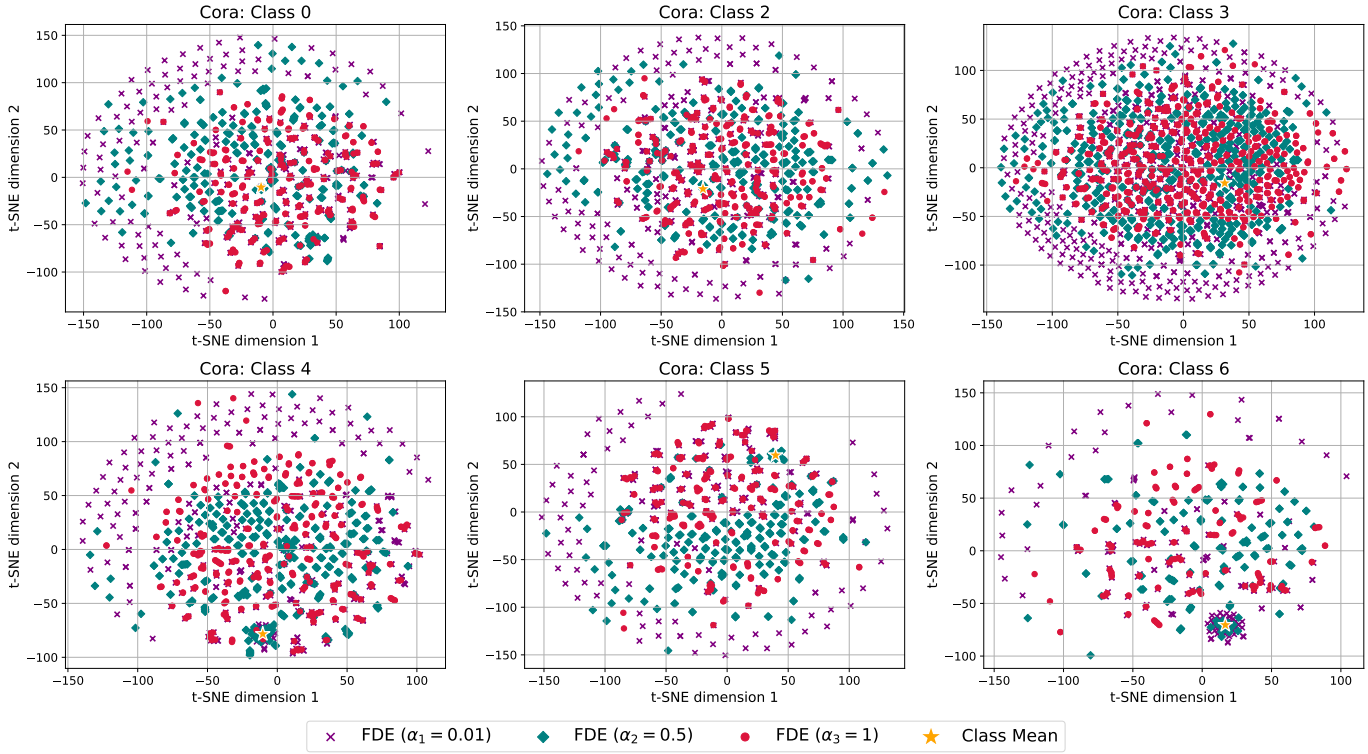


Fig. 10. t-SNE visualizations of node features from selected Cora labels. Each pair shows embeddings from FDE encoders with different α.

via adjoint backpropagation," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*. Philadelphia, USA: arXiv, Feb. 2025.

[18] W. Cui, Q. Kang, X. Li, K. Zhao, W. P. Tay, W. Deng, and Y. Li, "Neural variable-order fractional differential equation networks," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*. Philadelphia, USA: arXiv, Feb. 2025.

[19] W. Cui, X. Li, Y. Li, Y. Ma, Q. Kang, X. Fu, W. P. Tay, and Z.-J. Zha, "Generalized fractional neural differential equation network with learnable attention kernels," in *Advances in Neural Information Processing Systems (NeurIPS)*, San Diego, USA, Dec. 2025.

[20] Y. Zhang, Y. Chen, Z. Song, and I. King, "Contrastive cross-scale graph knowledge synergy," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2023, p. 3422–3433.

[21] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep graph contrastive representation learning," *arXiv preprint arXiv:2006.04131*, 2020.

[22] ——, "Graph contrastive learning with adaptive augmentation," in *Proc. Web Conf.*, 2021, pp. 2069–2080.

[23] T. Xiao, Z. Chen, Z. Guo, Z. Zhuang, and S. Wang, "Decoupled self-supervised learning for graphs," in *Advances Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 620–634.

[24] J. Chen, G. Zhu, Y. Qi, C. Yuan, and Y. Huang, "Towards self-supervised learning on graphs with heterophily," in *Proc. ACM Int. Conf. Inf. & Knowledge Management*, 2022, p. 201–211.

[25] Y. Xu, S. Huang, H. Zhang, and X. Li, "Why does dropping edges usually outperform adding edges in graph contrastive learning?" in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 21 824–21 832.

[26] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, and J. Huang, "Graph representation learning via graphical mutual information maximization," in *Proc. Web Conf.*, 2020.

[27] H. Zhang, Q. Wu, Y. Wang, S. Zhang, J. Yan, and P. S. Yu., "Localized contrastive learning on graphs," *arXiv preprint arXiv:2212.04604*, 2022.

[28] T. Xiao, H. Zhu, Z. Chen, and S. Wang, "Simple and asymmetric graph contrastive learning without augmentations," in *Advances Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 16 129–16 152.

[29] H. Wang, J. Zhang, Q. Zhu, W. Huang, K. Kawaguchi, and X. Xiao, "Single-pass contrastive learning can work for both homophilic and heterophilic graph," *Trans. Mach. Learn. Res.*, 2023.

[30] T. Xiao, H. Zhu, Z. Zhang, Z. Guo, C. C. Aggarwal, S. Wang, and V. G. Honavar, "Efficient contrastive learning for fast and accurate inference on graphs," in *Proc. Int. Conf. Mach. Learn.*, Jul 2024, pp. 54 363–54 381.

[31] Z. Zou, Y. Jiang, L. Shen, J. Liu, and X. Liu, "Loha: Direct graph spectral contrastive learning between low-pass and high-pass views," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, 2025, pp. 13 492–13 500.

[32] Y. Li, H. Zhang, and Y. Yuan, "Edge contrastive learning: An augmentation-free graph contrastive learning model," in *Proc. AAAI Conf. Artif. Intell.*, 2025, pp. 18 575–18 583.

[33] J. Zhu, L. He, C. Gao, D. Hou, Z. Su, P. S. Yu, J. Kurths, and F. Hellmann, "SDMG: Smoothing your diffusion models for powerful graph representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2025.

[34] H. Zhang, Q. Wu, J. Yan, D. Wipf, and P. S. Yu, "From canonical correlation analysis to self-supervised graph neural networks," in *Advances Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 76–89.

[35] N. Lee, J. Lee, and C. Park, "Augmentation-free self-supervised learning on graphs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2022, p. 7372–7380.

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

[37] 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.

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

[39] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[40] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, Jul 2021, pp. 12 310–12 320.

[41] S. Lee, F. Ji, W. P. Tay, and K. Xia, "Graph neural networks with a distribution of parametrized graphs," in *Proc. Int. Conf. Mach. Learn.*, 2024.

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

[43] O. Platonov, D. Kuznedelev, M. Diskin, A. Babenko, and L. Prokhorenkova, "A critical look at evaluation of gnns under heterophily: Are we really making progress?" in *Proc. Int. Conf. Learn. Representations*, 2023.

[44] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. Int. Conf. Learn. Representations*, 2019.

[45] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4116–4126.

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

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

[48] A. Bardes, J. Ponce, and Y. LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2022.

[49] K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin, "Topology attack and defense for graph neural networks: An optimization perspective," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020.

[50] S. Feng, B. Jing, Y. Zhu, and H. Tong, "Adversarial graph contrastive learning with information regularization," in *Proc. Web Conf.*, 2022.

[51] M. Lin, T. Xiao, E. Dai, X. Zhang, and S. Wang, "Certifiably robust graph contrastive learning," in *Advances Neural Inf. Process. Syst.*, 2024.

[52] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2018, pp. 2847–2856.

[53] S. Geisler, T. Schmidt, H. Sirin, D. Zügner, A. Bojchevski, and S. Günnemann, "Robustness of graph neural networks at scale," in *Advances Neural Inf. Process. Syst.*, 2021, pp. 7637–7649.

[54] D. Zügner and S. Günnemann, "Adversarial attacks on graph neural networks via meta learning," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[55] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[56] F.-Y. Sun, J. Hoffman, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *Proc. Int. Conf. Learn. Representations*, 2019.

[57] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Advances Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 5812–5823.

[58] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, Jul 2021, pp. 12 121–12 132.

[59] J. Xia, L. Wu, J. Chen, B. Hu, and S. Z. Li, "Simgrace: A simple framework for graph contrastive learning without data augmentation," in *Proc. the ACM Web Conf.*, 2022, p. 1070–1079.

[60] Q. Ji, J. Li, J. Hu, R. Wang, C. Zheng, and F. Xu, "Rethinking dimensional rationale in graph contrastive learning from causal perspective," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, pp. 12 810–12 820.

[61] S. Tan, D. Li, R. Jiang, Y. Zhang, and M. Okumura, "Community-invariant graph contrastive learning," in *Proc. Int. Conf. Mach. Learn.*, Jul 2024, pp. 47 579–47 606.

[62] P. Stinga, "Fractional derivatives: Fourier, elephants, memory effects, viscoelastic materials, and anomalous diffusions," *Not. Am. Math. Soc.*, vol. 70, no. 4, pp. 576–587, 2023.

[63] K. Diethelm, *The Analysis of Fractional Differential Equations*. Springer Berlin, Heidelberg, 2010.

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

[65] A. Erdelyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi, *Higher Transcendental Functions, Vol. III*. McGraw-Hill Book Company, 1955.