Adaptive Multi-view Graph Contrastive Learning via Fractional Continuous Dynamics

Yanan Zhao [1], Feng Ji[1] [*], Jingyang Dai [1], Jiaze Ma [1], Keyue Jiang [2], Kai Zhao [1], Wee Peng Tay [1]
[1]Nanyang Technological University, [2]University College London

Abstract

Graph contrastive learning (GCL) learns node and graph representations by contrasting multiple views of the same graph. Existing methods predominantly rely on a small set of fixed, handcrafted views—typically a local and a global perspective—which fundamentally constrains their capacity to capture multi-scale structural patterns. We present an augmentation-free, multi-view GCL framework grounded in fractional-order continuous dynamics. By systematically varying the fractional derivative order $\alpha \in (0, 1]$, our encoders produce a continuous spectrum of semantically distinct views: small values of α induce localized, memory-attenuated feature propagation, whereas values approaching 1 recover broader, global aggregation. Crucially, we treat α as a learnable parameter, enabling the model to automatically identify informative views in a data-driven manner, without resorting to manual view engineering. Extensive experiments on standard node- and graph-level benchmarks demonstrate that the resulting representations are more robust and expressive, consistently outperforming state-of-the-art GCL baselines. The implementation code is available at `https://github.com/YananZhao0630/FD-MVGCL`

1 Introduction

Contrastive learning is a self-supervised paradigm that learns representations by pulling semantically similar pairs together and pushing dissimilar pairs apart in the embedding space. Without labels, it has achieved 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, with *the novelty often lying in how they are constructed*. 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] adapts classical polynomial filters from graph signal processing to construct low-pass and high-pass spectral views, opening a promising direction of repurposing principled tools from adjacent fields for GCL view generation. Such designs increase view diversity but remain rigid and often require manual tuning. This motivates two questions:

Q1 **How can we parameterize view generation in GCL so that view diversity is continuous, learnable, and theoretically distinguishable across scales?**

Q2 **Can such a framework reduce both dimension/view collapse and dependence on heuristic augmentations?**

[*]Correspondence to: Feng Ji <jifeng@ntu.edu.sg>.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To address these questions, we propose a *multi-view, augmentation-free* framework based on fractional-order continuous dynamics governed by fractional differential equations (FDEs). We use the fractional derivative order as a principled mechanism for view construction, thereby yielding a learnable **continuum** of semantically distinct graph representations. In particular, the operator $\mathrm{d}^\alpha / \mathrm{d}t^\alpha$, with $\alpha \in (0, 1]$, induces a spectrum of diffusion behaviours: smaller values of α favour more localized, memory-attenuated propagation, whereas values approaching 1 produce broader, more global aggregation. By comparison, existing continuous graph models based on ordinary differential equations (ODEs) [6–12] are confined to the single regime $\alpha = 1$, and therefore yield only one graph view. FDEs generalize this setting by replacing the integer-order derivative with a fractional operator, introducing a continuous control parameter that modulates both propagation scale and memory effects [13–17]. However, prior work has largely treated α as a fixed hyperparameter within a single propagation scheme, leaving the multi-scale structure induced by varying fractional orders essentially unexplored for contrastive view generation.

Multi-view GCL arises naturally in our framework by treating α as a learnable continuous parameter, thereby generating diverse and complementary views without handcrafted filters (e.g., PolyGCL) or task-specific perturbations (e.g., BGRL). This formulation enables the model to adapt diffusion scales automatically to dataset heterogeneity, making view construction both principled and data-driven. Our main contributions are as follows:

- We propose a multi-view, augmentation-free GCL framework based on neural diffusion encoders governed by *fractional-order dynamics*. The framework is conceptually simple, computationally efficient, and flexible: each view is parameterized by a continuous order $\alpha \in (0, 1]$, inducing a continuum of diffusion scales. We learn these orders directly from data, enabling identification of informative scales without manual tuning or exhaustive grid search. To control complexity, we introduce a view-learning strategy that prunes redundant scales while preserving diversity and complementarity.
- We motivate our approach through a theoretical analysis showing that embeddings produced by FDE encoders with different fractional orders in a *linear* diffusion model are provably distinct, and that their separation increases with the gap between orders. To the best of our knowledge, this is the **first formal characterization** of multi-scale distinguishability in FDE-based contrastive learning, despite prior empirical use of FDEs in GNNs.
- We address two core challenges in GCL: (i) dimension collapse, mitigated by incorporating encoders with small fractional orders that produce less energy-concentrated, higher-rank embeddings; and (ii) view collapse, alleviated via a regularized contrastive objective that penalizes alignment of dominant directions across views, thereby promoting diverse and complementary representations without requiring negative samples.
- We present a rigorous stability analysis with explicit perturbation bounds quantifying the effects of input, parameter, and topological disturbances on encoder outputs. Empirically, our method consistently outperforms state-of-the-art GCL baselines across homophilic and heterophilic benchmarks, as well as under black-box and white-box attack scenarios, validating the effectiveness of the proposed multi-scale view generation mechanism.

2 Related Works

GCL methods are broadly divided into augmentation-based methods, which construct views via graph perturbations (e.g., edge deletion or feature masking) [18–23], and augmentation-free methods, which use distinct encoders or filters on the same graph [24, 25]. However, many augmentation-free methods still rely on negative samples [26–28, 5, 29, 30]. SDMG [31] adopts diffusion-based self-supervision with two low-frequency encoders for global signals. Negative-free methods such as BGRL [4] and CCA-SSG [32] still use augmentations, while AFGRL [33] removes both but has limited view diversity and fixed propagation depth, which can hurt heterophilic performance.

Table 1: Ours vs. SOTA GCL methods.

Method	>2 views	Aug-Free	Neg-Free	Hetero.	Adapt.
CCA-SSG ('21)	✗	✗	✓	✓	✗
BGRL ('22)	✗	✗	✓	✓	✓
AFGRL ('22)	✗	✓	✓	✗	✗
GraphACL ('23)	✗	✓	✗	✓	✗
GraphECL ('24)	✗	✓	✗	✓	✗
PolyGCL ('24)	✗	✓	✗	✓	✗
LOHA ('25)	✗	✓	✗	✓	✗
AFECL ('25)	✗	✓	✗	✗	✗
EPAGCL ('25)	✗	✗	✗	✗	✓
MHGCL-HA ('26)	✓	✗	✗	✗	✗
FD-MVGCL (ours)	✓	✓	✓	✓	✓

In contrast, our method is both augmentation-free and negative-sample-free, performing well on both homophilic and heterophilic graphs (see Table 1). We generate multi-view representations via

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.

3 Preliminaries

Problem formulation Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node features $\mathbf{X} \in \mathbb{R}^{N\times d}$ and weighted adjacency $\mathbf{A} \in \mathbb{R}^{N\times N}$, let $\mathcal{X} = (\mathbf{A}, \mathbf{X})$. An encoder f_θ maps each node v_i to an embedding $\mathbf{y}_i = f_\theta(\mathcal{X}, v_i) \in \mathbb{R}^F$, yielding $\mathbf{Y} \in \mathbb{R}^{N\times F}$. In multi-view GCL, we use multiple encoders $\{f_{\theta_k}\}_{k=1}^K$ and learn consistent yet non-collapsed node representations across views.

Fractional diffusion encoders We build each view using a fractional-order diffusion encoder [14]:

$$D_t^\alpha \mathbf{Y}(t) = \mathcal{F}(\mathbf{W}, \mathbf{Y}(t)). \tag{1}$$

In this work, our theoretical analysis focuses on the F-GRAND-l model [14] given by

$$D_t^\alpha \mathbf{Y}(t) = -\overline{\mathbf{L}}\mathbf{Y}(t), \quad \overline{\mathbf{L}} = \mathbf{I} - \overline{\mathbf{A}}, \tag{2}$$

where $\alpha \in (0, 1]$ is a learnable view parameter: different orders induce different diffusion scales. The case $\alpha = 1$ recovers the standard first-order dynamics, while smaller α yields stronger history dependence. We therefore treat α as the key mechanism for augmentation-free multi-view generation. Additional background on ODE and FDE diffusion, random-walk interpretation, and derivative definitions is deferred to Appendices A and B.

Graph signal processing For analysis, we use the eigendecomposition $\overline{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\intercal$, where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ with $\lambda_1 \le \cdots \le \lambda_N$, $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$, and $\mathbf{u}_i$ is the eigenvector associated with λ_i. The eigenvalues act as graph frequencies: lower frequencies correspond to smoother components, and higher frequencies to more localized variations. The Fourier coefficients of a graph signal $\mathbf{x}$ are given by $\{c_i = \langle \mathbf{x}, \mathbf{u}_i \rangle : i = 1, \ldots, N\}$. This spectral view is used throughout our theoretical results; detailed Fourier-coefficient definitions are given in Appendix A.

4 Theoretical Motivations

To explain how fractional-order derivatives induce multi-scale views, we rigorously analyze the distinct representations produced by FDE encoders with different orders α under the F-GRAND-l model [14] in (2). Our formal analysis is restricted to the linear F-GRAND-l encoder and is intended to justify the view-generation mechanism induced by varying α, rather than to characterize the full nonlinear FD-MVGCL pipeline. In practice, this model is discretized into m diffusion steps, each corresponding to a diffusion time τ. The outputs at these discretized diffusion times are aggregated through skip connections to form the final representation, consistent with the skip-connection interpretation in FROND [14]. Thus, the total diffusion time is $T = m\tau$.

Theorem 1. *Suppose $\mathcal{G}$ is a connected graph and $0 < \alpha_l < \alpha_g$ and $\frac{1}{2} < \alpha_g < 1$ are the order parameters. For subscript $s \in \{l, g\}$ with $0 < \alpha_s < 1$, let $n_s = \lceil 1/\alpha_s \rceil - 1$. Consider a graph signal $\mathbf{x}$ with Fourier coefficients $\{c_i, i = 1, \ldots, N\}$. Let $\mathbf{z}_{\alpha_s}(T)$ be the output features of the encoder with order parameter α_s and its Fourier coefficients be $\{c_{\alpha_s,i}(t), i = 1, \ldots, N\}$. For τ sufficiently large, the following holds:*

(a) The output Fourier coefficient takes the form

$$c_{\alpha_s,i}(T) = \Big(P_{s,i}(\tau) + O(\tau^{-1})\Big)c_i = \Big(\sum_{j=0}^{n_s} b_{\alpha_s,i,j}\tau^{-j\alpha_s} + O(\tau^{-1})\Big)c_i.$$

The main term $P_{s,i}(\tau) > 0$.

(b) For fixed s, the main term $P_{s,i}(\tau)$ is decreasing w.r.t. i.

(c) For fixed i, the main term satisfies $P_{\alpha_l,i}(\tau) > P_{\alpha_g,i}(\tau)$.

Discussion. The signal $\mathbf{x}$ in Theorem 1 is a column of the feature matrix $\mathbf{X}$ and its corresponding output from the encoder with $\alpha = \alpha_s$ is $\mathbf{z}_{\alpha_s}(T)$. By Theorem 1, for $s \in \{l, g\}$, the Fourier coefficients $c_{\alpha_s,i}(t)$ of $\mathbf{Y}_{\alpha_s}(T)$ decay as the frequency index i increases. This decay is much slower for $\mathbf{Y}_{\alpha_l}(T)$

than for $\mathbf{Y}_{\alpha_g}(T)$.Consequently, the Fourier coefficients of $\mathbf{Y}_{\alpha_l}(T)$ are more spread out than those of $\mathbf{Y}_{\alpha_g}(T)$. We defer the proof of Theorem 1 to Appendix C. We also provide a non-Markovian random-walk interpretation for further insights.

Augmentation-free GCL is effective when different encoders produce 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 *local* details as $\alpha_l \to 0$. By Theorem 1, smooth components associated with small graph frequencies λ_i dominate $\mathbf{Y}_{\alpha_g}(t)$. Each such component is spanned by a vector $\mathbf{u}_i$, which represents a signal with minimal variation across edges, thereby encoding global structure. 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. Thus, we establish the correspondence: $\alpha_l \longleftrightarrow$ "local" and $\alpha_g \longleftrightarrow$ "global."

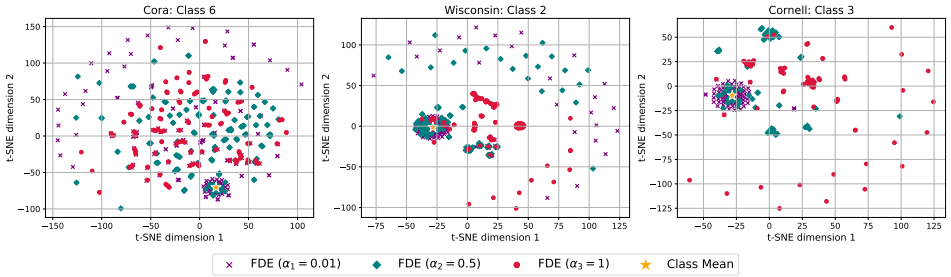


Figure 1: t-SNE of single-class node embeddings from encoders with varying FDE orders, with class means aligned for fair comparison. On Cora and Wisconsin/Cornell, smaller α yields compact, core-concentrated clusters, while larger α produces more dispersed distributions. Additional visualizations are given in Appendix E.

By selecting distinct α values, we obtain a set of FDEs with different propagation scales, which naturally generate diverse multi-scale graph views $\mathbf{Y}_{\alpha}(t)$. Numerical results 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. Small α values (e.g., 0.01) produce tightly clustered features, large values (e.g., 1) yield more dispersed distributions, and intermediate values (e.g., 0.5) balance these extremes. This demonstrates that each FDE provides a semantically distinct and complementary view, enhancing the effectiveness of contrastive learning.

Dimension collapse Dimension collapse, where features concentrate in a low-dimensional subspace within the full embedding space [34], is a key challenge in contrastive learning. We argue that $\mathbf{Y}_{\alpha_l}(t)$, generated with small α_l, helps alleviate this issue. By Theorem 1, $\mathbf{Y}_{\alpha_l}(t)$ exhibits lower energy concentration than $\mathbf{Y}_{\alpha_g}(t)$ when $\alpha_l < \alpha_g$. This means that the columns of $\mathbf{Y}_{\alpha_l}(t)$ can be expressed as $\sum_{i=1}^{N} c_i \mathbf{u}_i$, with a relatively larger number of significant $|c_i|$ values. Thus, the features are distributed across a higher-dimensional space, avoiding collapse into a low-dimensional subspace. The principal component analysis (PCA) of $\mathbf{Y}_{\alpha_l}(t)$ and $\mathbf{Y}_{\alpha_g}(t)$ for Cora and Wisconsin support this claim, shown in Fig. 2.

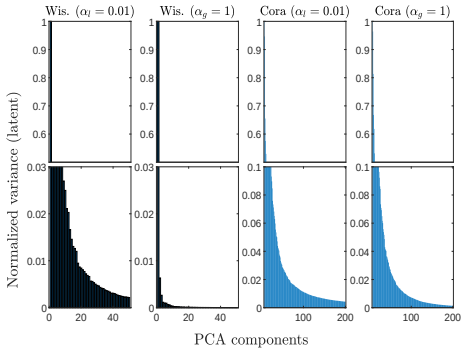


Figure 2: PCA components of features for different datasets and FDE orders. For small α_l, the bar chart is more spread out, preventing dimension collapse.

Encoder quality Model performance improves when encoders produce well-clustered node representations. Building on the strong supervised performance of FDE-based encoders [6, 14], we evaluate their unsupervised clustering ability using, for each class c, the average intra-class distance d_c^{intra}, the average inter-class distance d_c^{inter}, and their ratio $r_c = d_c^{\text{inter}}/d_c^{\text{intra}}$, where larger values indicate better clustering. As shown in Fig. 3, 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 used for comparison. This shows that FDE-based encoders learn more discriminative and better-clustered embeddings on both homophilic and heterophilic graphs.

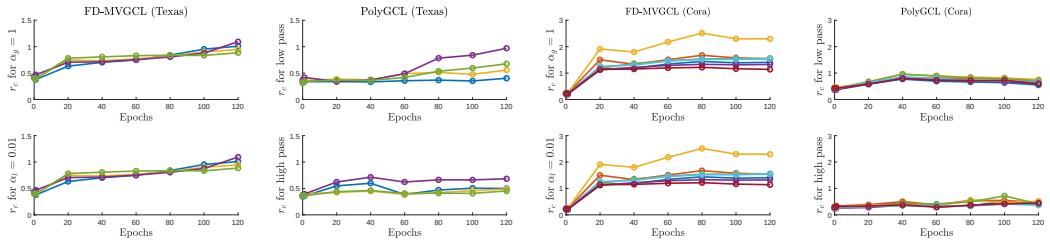


Figure 3: The variation of r_c during training, with each line corresponding to a label class and input features shown at epoch 0. The ratio generally increases initially then stabilizes. FD-MVGCL produces higher r_c values than the strong benchmark PolyGCL [5], suggesting better clustered embeddings.

5 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 first describe the FD-MVGCL architecture. We propose an adaptive algorithm to automatically determine the number of views and their corresponding fractional orders, and analyze the model's stability under graph perturbations to highlight its robustness.

5.1 The adaptive FD-MVGCL model

Given a graph $\mathcal{G}$ with adjacency matrix $\mathbf{A}$ and node features $\mathbf{X}$, we denote the input as $\mathcal{X} = (\mathbf{A}, \mathbf{X})$. Our goal is to learn multiple FDE encoders f_{θ_k} for $k \geq 2$ in an unsupervised manner, each capturing structural patterns at a specific scale. In adaptive FD-MVGCL, all encoders f_{θ_k} share the same architecture (see Fig. 4). For clarity, we describe 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 (1).
- **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 $\mathbf{W}_k$, fractional order α_k, and diffusion time T. Unlike prior FDE-based models such as FROND [14] and DRAGON [13], which manually tune α via grid search, we treat α values as learnable parameters, allowing the model to adaptively generate informative views from the data.

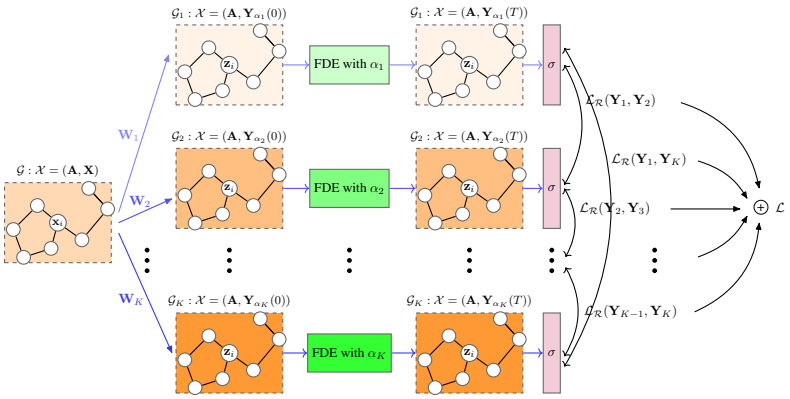


Figure 4: Overview of the proposed adaptive FD-MVGCL framework. The fractional orders are selected such that $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_K$, enabling contrastive learning across multiple distinct views.

We utilize K encoders f_{θ_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 $k' = k+1 \bmod K$. For simplicity, $\mathcal{L}_0$ is defined as the mean cosine similarity (*cosmean*; see (21) in Appendix D.3). This consecutive-pair scheme balances representational alignment and computational efficiency by enforcing consistency only between successive views. The wrap-around loss on $(\mathbf{Y}_K, \mathbf{Y}_1)$ further promotes global consis-

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 $\{\tilde{\alpha}_k\}_{k=1}^{\tilde{K}}$

repeat
  for $e = 1$ to N **do**
    Compute $\mathbf{Y}_k(T)$ via (1) for each encoder k; Compute contrastive loss $\mathcal{L}$ via (3)
    Update $\mathbf{W}_k^{(e)} \leftarrow \mathbf{W}_k^{(e-1)} - \eta \nabla_{\mathbf{W}_k} \mathcal{L}$; $\alpha_k^{(e)} \leftarrow \alpha_k^{(e-1)} - \eta_{\alpha_k} \nabla_{\alpha_k} \mathcal{L}$
    if $\alpha_k^{(e)} \notin (0, 1]$ **then** $\alpha_k^{(e)} \leftarrow \min(1, \max(\alpha_k^{(e)}, \epsilon))$ **end if**
  end for
  Cluster the learned $\alpha_k^{(N)}$ values using the log-space distance $|\log \alpha_i^{(N)} - \log \alpha_j^{(N)}|$ with threshold δ.
  In each cluster, keep the encoder whose order is closest to the cluster median; remove the rest.
until all remaining orders are separated by at least δ in log-space
return $\tilde{K}$ and $\{\tilde{\alpha}_k\}_{k=1}^{\tilde{K}}$

tency across all encoders. Thus, it reduces computational cost from all pairwise comparisons ($\mathcal{O}(K^2)$) to consecutive pairs ($\mathcal{O}(K)$), while remaining scalable and preserving representation quality.

Model/view collapse As discussed in Section 4, directly minimizing $\mathcal{L}_0$ can lead to view collapse, where two encoders produce nearly identical representations. To address this, we exploit the dominant components observed in Fig. 2. 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 [26, 5]. Its effect is further analyzed in Section 6.3. The overall contrastive loss across all encoders is given by:

$$\mathcal{L} = \sum_{k=1}^{K} \mathcal{L}_{\mathcal{R}}(\mathbf{Y}_k, \mathbf{Y}_{k'}). \tag{3}$$

Encoder reduction strategy To contain the computational cost as K increases, we adopt the encoder reduction strategy in Algorithm 1. We begin with K encoders whose fractional orders $\{\alpha_k\}_{k=1}^K$ are uniformly initialized over $(0.01, 1]$. The encoder weights $\mathbf{W}_k$ and fractional orders α_k are then jointly optimized by gradient descent on the objective in (3). After each training phase, the learned values of α_k are clustered in logarithmic order space, and a single representative encoder is retained from each cluster, thereby reducing the number of active encoders. The procedure is repeated until all remaining fractional orders are sufficiently separated. The final number of retained encoders is denoted by $\tilde{K}$. For downstream tasks, the resulting representation is constructed as a weighted combination of the learned views, $\mathbf{Y} = \sum_{k=1}^{\tilde{K}} \beta_k \mathbf{Y}_k$, where $\sum_{k=1}^{\tilde{K}} \beta_k = 1$. The representations $\mathbf{Y}_k$ are learned in an **unsupervised** manner, whereas the coefficients β_k are application specific and selected on a validation set for each downstream task.

Stability Real-world graphs often contain noisy node features or missing edges [35]. To assess the robustness of FD-MVGCL, we perform a unified stability analysis under perturbations in input features, model parameters, and graph structure. Theorem 2 establishes the stability of the FDE encoder, and Theorem 3 formalizes the stability of the full model. Proofs are provided in Appendix C.2.

Theorem 2. *Let $\mathbf{Y}(t)$ denote the solution to the original FDE encoder in* (1)*, and let $\tilde{\mathbf{Y}}(t)$ be the solution to the perturbed system:*

$$D_t^{\alpha} \tilde{\mathbf{Y}}(t) = \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \tilde{\mathbf{Y}}(t)) \text{ with } \tilde{\mathbf{Y}}(0) = \tilde{\mathbf{Y}}_0. \tag{4}$$

Define the total perturbation magnitude as

$$\varepsilon := \|\mathbf{Y}_0 - \tilde{\mathbf{Y}}_0\| + \sup_{\mathbf{Y} \in A} \|\mathcal{F}(\mathbf{W}, \mathbf{Y}) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \mathbf{Y})\|, \tag{5}$$

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:

$$\|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| = O\left(\varepsilon T^{\alpha - 1}\right) \text{ for } t \in [0, T] \tag{6}$$

Theorem 2 unifies Theorem 6.20 and Theorem 6.21 from [36] 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* $\mathbf{Y}(t) = \sum_{k=1}^{\tilde{K}} \beta_k \mathbf{Y}_k(t)$ *where each* $\mathbf{Y}_k(t)$ *is the solution to the feature dynamics in* (1)*, 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* (4)*. For sufficiently small combined perturbation* ε *in* (5)*, 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*

$$\|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| = O\left(\varepsilon T^{\alpha_{\tilde{K}}-1}\right) \text{ for } t \in [0, T]. \tag{7}$$

Theorem 3 highlights the inherent robustness of FD-MVGCL: smaller α_k values result in reduced feature discrepancies in (7), 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 6.4.

6 Experiments

6.1 Experimental setup

Datasets and splits We evaluate FD-MVGCL on both hetero. and homo. datasets. The heterophilic benchmarks include Wisconsin, Cornell, Texas, Actor, Crocodile, and the filtered Squirrel and Chameleon, where duplicate nodes are removed to avoid training–test leakage [37]. We also consider three **large-scale** heterophilic datasets: Roman-empire (Roman), Amazon-ratings (Amazon), and Arxiv-year. The homophilic datasets are Cora, Citeseer, Pubmed, Computer, Photo, and **large-scale** Ogbn-Arxiv. We use the standard public splits from [26] with more details in Appendix D.1.

Baselines We compare FD-MVGCL with recent state-of-the-art unsupervised learning methods: DGI [38], GMI [24], MVGRL [39], GRACE [19], CCA-SSG [32], BGRL [4], AFGRL [33], HGRL [22], DSSL [21], SP-GCL [27], GraphACL [26], GraphECL [28], PolyGCL [5], LOHA [29], EPAGCL [23], AFECL [30], SDMG [31] and MHGCL-HA [40].

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** (see Appendix G) to demonstrate its generalization beyond node-level settings.

Setup For FD-MVGCL, we adopt the basic GRAND form of $\mathcal{F}$ in (1). Model parameters are randomly initialized and optimized with Adam. Each experiment is run with 10 random seeds (as in [26]), and we report the mean and standard deviation. All methods are tuned solely on validation accuracy. For baselines without results on certain datasets or using nonstandard splits [5, 23, 29], we reproduce results with official codes. Implementation and hyperparameter details are in Appendix D.2.

6.2 Overall performance

Node classification results on hetero./homo. datasets are reported in Tables 2 and 3, respectively. FD-MVGCL achieves state-of-the-art or competitive performance on both graph types. It ranks first or second on most hetero. benchmarks and remains competitive with leading methods (e.g., GraphACL and PolyGCL) on homo. datasets. Based on the average rank over valid per-dataset ranks only (excluding OOM and missing results), FD-MVGCL attains the best average rank on hetero. datasets and ties with GraphACL on homo. datasets, showing its strong generalization and consistency across diverse graph types. The larger gains on hetero. graphs further suggest the benefit of adaptive multi-scale diffusion scale for capturing more complex structures. 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 D.2.

Table 2: Node classification results(%) on hetero. datasets. The best and the second-best result are highlighted in **red** and **blue**, respectively.

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 3: Node classification results(%) on homophilic datasets.

Method	DGI	GMI	MVGRL	GRACE	BGRL	AFGRL	GraphACL	PolyGCL	LOHA	EPAGCL	AFECL	SDMG	MHGCL-HA	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	82.54±0.69	**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	70.94±1.42	**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	80.54±0.45	**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**	74.50±2.61	**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**	88.19±1.55	**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	OOM	- -	70.60±0.20	OOM	70.46±0.06
Avg. rank.	9.33	7.50	6.80	9.33	6.58	7.40	**2.00**	8.60	11.40	7.17	10.50	**3.17**	10.20	**2.00**

6.3 Ablation studies

Multi-view vs two-view We evaluate the performance of FD-MVGCL under various configurations of $\{\alpha_k\}_{k=1}^K$: (i) a fixed distinct two-view ($\alpha 1 = 0.01$, $\alpha_2 = 1$), (ii) a fixed three-view ($\alpha_1 = 0.01$, $\alpha_2 = 0.51$, $\alpha_3 = 1$), (iii) a fixed five-view ($\alpha_1 = 0.01$, $\alpha_2 = 0.26$, $\alpha_3 = 0.51$, $\alpha_4 = 0.75$, $\alpha_5 = 1$), and (iv) AVLA (Algorithm 1) that dynamically selects the number and values of α_k during training. These are compared on Cora and Cornell using classification accuracy (Table 4).

Feature dimension & Diffusion depths FD-MVGCL use a linear encoder to increase feature dimension and model's expressiveness. Fig. 6a validates that larger feature dimensions consistently improve performance on both homo./hetero. graphs, especially the latter. Following [14], larger diffusion depth T improves the solution accuracy of (1) and typically enhances view diversity. Table 5 shows that performance improves with T. Selected T values are listed in Appendix D.2.

Different options for $\mathcal{F}$ We further evaluate FD-MVGCL with alternative choices of $\mathcal{F}$ in (1), including CDE, GREAD and GraphCON. As reported in Table 6, FD-MVGCL remains competitive across all variants, demonstrating its generality and adaptability to diverse diffusion dynamics.

Table 4: Acc. vs. Param. Config.

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**
$\{\bar{\alpha}_k\}_{k=1}^K$	$\{0.001, 0.529, 0.736, 1\}$	$\{0.001, 0.018, 0.137, 1\}$

Table 5: Acc. vs. Diff. Depth (T).

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

Table 6: Acc. vs. different $\mathcal{F}$

Model	Cora	Cornell	Roman
GRAND [6]	84.40±0.04	72.70±0.57	72.47±0.70
CDE [41]	75.02±0.26	60.54±3.92	OOM
GREAD [42]	81.02±0.15	66.24±0.67	71.48±0.36
GraphCON [10]	80.64±0.33	67.03±0.33	68.20±0.65

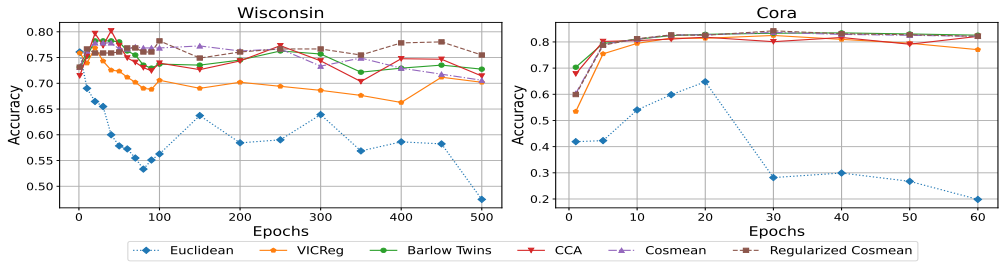
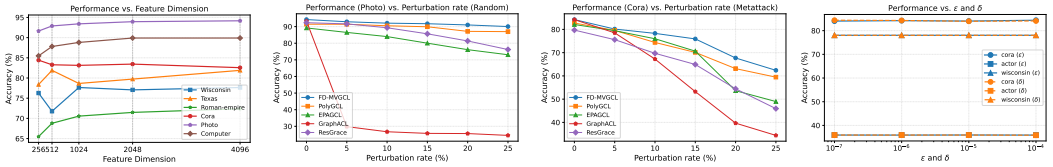


Figure 5: Accuracy vs. training epochs for various loss functions on the Wisconsin and Cora datasets.

Loss functions To evaluate FD-MVGCL across various contrastive paradigms, we compare several *contrastive loss* functions on two FDE encoder representations (i.e., the local view $\mathbf{Y}_l$ with $\alpha_l = 0.01$ and and the global view $\mathbf{Y}_g$ with $\alpha_g = 1$) and report classification accuracy on Cora (homo.) and Wisconsin (hetero.). The losses include Euclidean, Cosmean, Barlow Twins [34], VICReg [43], CCA [32], and our proposed Regularized Cosmean. As illustrated in Fig. 5, Regularized Cosmean loss consistently outperforms the others and remains stable during training, due to its ability to mitigate dimensional collapse. Definitions of these losses and additional results are in Appendix D.3.



(a) Acc. vs. Feature dim. (b) Acc. vs Rand. attack (c) Acc. vs. Metattack (d) Acc. vs. ϵ/δ sensitivity

Figure 6: Comparison across feature dimension, Metattack, Random attack and Sensitivity.

Hyperparameter sensitivity To evaluate the hyperparameter sensitivity of AVLA, we vary δ and ϵ over 10 logarithmically spaced values ranging from 10^{-7} to 10^{-4}. As shown in Fig. 6d, AVLA maintains stable performance across all settings, indicating low sensitivity to these hyperparameters.

6.4 Robustness studies

We evaluate FD-MVGCL's robustness via node classification under black- and white-box attacks on homo. (Cora, Citeseer, Photo) and hetero. (Cornell, Actor, Texas) graphs. We compare against nine baselines: robust supervised FROND [14], three robust GCL methods (GCL-Jac [44], Ariel [45], Res-GRACE [46]) and five SOTA GCL models (GraphACL, PolyGCL, LOHA, EPAGCL, SDMG).

Attacks methods We consider four *black-box* topology attacks: Random, PRBCD [47], Nettack [48], and Metattack [49]. We also evaluate two *white-box* attacks (i.e., PGD [50] and PRBCD) that jointly perturb the graph structure and node features. All models are trained on clean graphs with adversarial perturbations applied only at inference. Details of these attacks are in Appendix F.1.

Robustness results We evaluate FD-MVGCL under perturbation rates from 0% to 25% in 5% increments, using random attacks on Photo (Fig. 6b) and Metattack on Cora (Fig. 6c). FD-MVGCL shows the smallest performance drop and consistently outperforms strong baselines (e.g., GraphACL and PolyGCL). This robustness stems from fractional-order diffusion: smaller α_k reduces feature discrepancies in (7), improving resistance to perturbations for $\alpha_k < 1$. Tables 7 and 8 further confirm its superior robustness, particularly under white-box attacks. Additional results are in Appendix F.2.

Table 7: 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**
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 8: 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**

Table 9: Comparison of training time (s), inference time (s), and GPU memory used (MiB).

Method	Training Time (s)				Inference Time (s)				GPU (MiB)			
	Cora	Wisconsin	Roman	Arxiv-year	Cora	Wisconsin	Roman	Arxiv-year	Cora	Wisconsin	Roman	Arxiv-year
GraphACL	0.06	0.08	100.32	5403.97	0.004	0.01	0.15	1.06	932	1670	14008	53960
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	6.01	1.47	1.41	0.18	0.17	1330	906	16490	14824

6.5 Complexity analysis

The training cost of FD-MVGCL comes from 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. Under the GRAND model [6], each evaluation of $\mathcal{F}(\mathbf{W}, \mathbf{Y}_j)$ costs $C = |\mathcal{E}|d$, where d is the feature dimension. Solving one FDE costs $\mathcal{O}(EC + E\log E)$, with $\log E$ due to fast convolution. For $\tilde{K}$ FDEs, the total cost is $\mathcal{O}(\tilde{K}(EC + E\log E))$, and the contrastive loss adds $\mathcal{O}(\tilde{K}N)$. Therefore, the overall training complexity of $\mathcal{O}(\tilde{K}(EC + E\log E + N))$. Training/inference time and training GPU memory are

compared with baselines in Table 9. FD-MVGCL remains computationally practical, avoids out-of-memory (OOM) issues on several large datasets where some baselines fail, and offers competitive runtime-memory tradeoffs.

7 Conclusion

We introduce FD-MVGCL, an adaptive multi-view, augmentation-free graph contrastive learning framework that leverages fractional-order graph neural diffusion to generate diverse, multi-scale representations. By adaptively varying the fractional order to the data, FD-MVGCL generates diverse, multi-scale graph views without relying on handcrafted augmentations, negative sampling or manual tuning. The proposed method achieves state-of-the-art performance, consistently surpassing existing GCL baselines in both effectiveness and robustness across a wide range of benchmark datasets.

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] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Q. Kang, X. Li, K. Zhao, W. Cui, Y. Zhao, W. Deng, and W. P. Tay, "Efficient training of neural fractional-order differential equation via adjoint backpropagation," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*. Philadelphia, USA: arXiv, Feb. 2025.

[17] 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.

[18] 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.

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

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

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

[37] 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.

[38] 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.

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

[40] M. Zou, Y. Wang, X. Meng, Z. Gan, C. Guan, and S. Leng, "Multi-head graph contrastive learning with hop augmentation for node classification," *Pattern Recognition*, p. 112055, 2025.

[41] 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.

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

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

[44] 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.

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

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

[47] 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.

[48] 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.

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

[50] 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.

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

[52] 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.

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

[54] 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.

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

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

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

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

A Additional preliminaries

This section collects background material moved from the main text for space. It includes the ODE-to-FDE diffusion viewpoint, a random-walk interpretation, and detailed GSP notation.

A.1 From discrete propagation to continuous diffusion

Traditional graph neural network (GNN) models such as GCN and GAT [51–53] perform layer-wise propagation, e.g.,

$$\mathbf{Y}^{(k)} = \overline{\mathbf{A}}\mathbf{Y}^{(k-1)}.$$

Continuous graph diffusion replaces the layer index by time. In GRAND [6],

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

with a common choice $\mathcal{F}(\mathbf{W}, \mathbf{Y}(t)) = -\overline{\mathbf{L}}\mathbf{Y}(t)$ and $\overline{\mathbf{L}} = \mathbf{I} - \overline{\mathbf{A}}$. Replacing $\frac{\mathrm{d}}{\mathrm{d}t}$ by D_t^α yields the FDE encoder in (1).

A.2 Random-walk interpretation of α

Under (2), one can view diffusion as a non-Markovian random walk [14]. For $\alpha \in (0, 1]$, the walker has heavy-tailed waiting times and transition probabilities controlled by α: smaller α leads to longer waiting times and more localized behavior, while larger α leads to more frequent transitions and more global exploration. This interpretation motivates using different α values to generate complementary local-to-global views.

A.3 Detailed GSP notation

Since $\overline{\mathbf{L}}$ is symmetric, it admits

$$\overline{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\intercal,$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ with $\lambda_1 \leq \cdots \leq \lambda_N$, and $\mathbf{u}_i$ is the eigenvector associated with λ_i. For any graph signal $\mathbf{x}$, the spectral decomposition is

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

The coefficients c_i are graph Fourier coefficients. Large values at small indices correspond to smoother components, while large values at high indices correspond to more localized variations.

B Fractional-order derivatives and GNNs

The works [14, 13] extend the ODE-based approaches by incorporating graph neural fractional-order differential equation (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.

B.1 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* [54] 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* [55] 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 [56].

B.2 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 [10] is equivalent to a system of equations:

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

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

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

A numerical solver is provided in [14].

C Theoretical discussions

C.1 Proof of Theorem 1

To prove Theorem 1, 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. For consistency, we set $e_\alpha(0, t) = 1$ the constant function independent of α. The Mittag-Leffler function is a generalization of the exponential function, and it reduces to the exponential function when $\alpha = 1$, i.e., $e_1(\lambda, t) = e^{-\lambda t}$. The Mittag-Leffler function has been widely used in the analysis of FDEs, as it serves as the fundamental solution to linear FDEs. For example, the solution to $D_t^\alpha y(t) = -\lambda y(t)$ with initial condition $y(0) = y_0$ is given by $y(t) = e_\alpha(\lambda, t) y_0$.

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 from the definition of the fractional derivative, we have

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

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 $\mathbf{c}_\alpha(t) = \mathbf{U}^\intercal \mathbf{z}_\alpha(t)$ and the i-th entry of $\mathbf{c}_\alpha(t)$ is $c_{\alpha,i}(t)$.

From [54], we have the solution $c_{\alpha,i}(t) = e_\alpha(\lambda, t)c_{\alpha,i}(0), i = 1, \ldots, N$, where $c_{\alpha,i}(0) = c_i$ depends only on the input feature. Recall that, we perform m diffusion iterations, each with diffusion time τ, and aggregate all intermediate diffusion outputs into the final representation through skip connections, consistent with the skip-connection scheme in FROND [14]. The coefficients of the spectral decomposition of the final output features are

$$c_{\alpha,i}(T) = \Big(1 + e_\alpha(\lambda,\tau) + \ldots + e_\alpha(\lambda,\tau)^m\Big)c_{\alpha,i}(0) = \left(\sum_{k=0}^{m} e_\alpha(\lambda,\tau)^k\right) c_{\alpha,i}(0). \tag{8}$$

The case $c_{\alpha,i}(0) = c_i = 0$ is trivial as we can set the main term 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.

For $\lambda_i > 0$, by the asymptotic expansion of the Mittag-Leffler function $E_{\alpha,1}$ (i.e., Page 206 in [57]), we have

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

The choice of n_s ensures that $(n_s+1)\alpha_s \geq 1$, and hence, we may replace $O\big(\tau^{-(n_s+1)\alpha_s}\big)$ with $O(1/\tau)$.

Equivalently, this expansion can be written by pairing the odd and even terms as

$$e_{\alpha_s}(\lambda_i,\tau) = \sum_{r=1}^{\lfloor n_s/2 \rfloor} G_{r,s}(\lambda_i,\tau) + \mathbb{1}_{\{n_s \text{ odd}\}} \frac{\tau^{-n_s\alpha_s}}{\lambda_i^{n_s}\Gamma(1-n_s\alpha_s)} + O\left(\frac{1}{\tau}\right).$$

where

$$G_{r,s}(\lambda_i,\tau) = \frac{\tau^{-(2r-1)\alpha_s}}{\lambda_i^{2r-1}\Gamma(1-(2r-1)\alpha_s)} - \frac{\tau^{-2r\alpha_s}}{\lambda_i^{2r}\Gamma(1-2r\alpha_s)}.$$

Notice that as $n_s\alpha_S \leq 1$, $\Gamma(1-n_s\alpha_s) > 0$ and decreases w.r.t. i. Therefore, in view of (8), to establish the positivity (in (a)) and decreasing property w.r.t. i (in (b)) of the main term $P_{s,i}(\tau)$, it suffices to prove the corresponding properties of $G_{r,s}(\lambda_i,\tau)$ for each r.

For $1 \leq r \leq \lfloor n_s/2 \rfloor$, we have $2r\alpha_s \leq n_s\alpha_s < 1$. Hence, both Gamma factors in $G_{r,s}$ are positive. Moreover, $G_{r,s}$ can be written as

$$G_{r,s}(\lambda_i,\tau) = \frac{\tau^{-(2r-1)\alpha_s}}{\lambda_i^{2r-1}\Gamma(1-(2r-1)\alpha_s)}\Big[1 - \frac{\Gamma(1-(2r-1)\alpha_s)}{\lambda_i\Gamma(1-2r\alpha_s)}\tau^{-\alpha_s}\Big].$$

Assume τ large enough such that $1/(\tau^{\alpha_s}\lambda_2) \leq 1$, then $1/(\tau^{\alpha_s}\lambda_i) \leq 1$. Moreover, the Gamma function is decreasing on $(0,1)$. Therefore, the "difference factor"

$$1 - \frac{\Gamma(1-(2r-1)\alpha_s)}{\lambda_i\Gamma(1-2r\alpha_s)}\tau^{-\alpha_s} > 0,$$

and this established that $G_{r,s}(\lambda_i,\tau) > 0$.

Moreover, for any sufficiently large τ, the difference factor above can be made arbitrarily close to 1, independent of i. Therefore, $G_{r,s}(\lambda_i,\tau)/G_{r,s}(\lambda_j,\tau)$ is arbitrarily close to $\lambda_j^{2r-1}/\lambda_i^{2r-1}$. As r is within a fixed range, the above ratio decreases as j increases. This establishes (b).

For (c), as $\alpha_g > 1/2$, and hence $n_g = 1$. Therefore, $e_{\alpha_g}(\lambda_i,\tau)$ has only one non-asymptotic term:

$$e_{\alpha_g}(\lambda_i,\tau) = \frac{1}{\lambda_i\Gamma(1-\alpha_g)}\tau^{-\alpha_g} + O(\frac{1}{\tau}). \tag{9}$$

In view of (8) and the positivity of $G_{r,s}(\lambda_i, l)$, it suffices to show that

$$G_{1,l}(\lambda_i, \tau) = \frac{\tau^{-\alpha_l}}{\lambda_i \Gamma(1-\alpha_l)} - \frac{\tau^{-2\alpha_l}}{\lambda_i^2 \Gamma(1-2\alpha_l)} \geq \frac{1}{\lambda_i \Gamma(1-\alpha_g)} \tau^{-\alpha_g}.$$

As the exponent $-\alpha_l$ is strictly larger than both $-2\alpha_l$ and $-\alpha_g$, the inequality holds for sufficiently large τ. This proves (c). □

To give a concrete example applicable empirically to two view setting, we choose $\alpha_l \leq 0.1$ and $\alpha_g > 0.5$, and thus $n_g = 1$. By (8) and (9), the global-view coefficient ratio satisfies:

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

For the local view, the first paired component has leading order $G_{1,l}(\lambda_i, \tau) = \frac{1}{\lambda_i \Gamma(1-\alpha_l)} \tau^{-\alpha_l}(1 + o(1))$. Therefore, the ratio between the leading local correction and the learning global correction is

$$\frac{\lambda_i^{-1} \Gamma(1-\alpha_l)^{-1} \tau^{-\alpha_l}}{\lambda_i^{-1} \Gamma(1-\alpha_g)^{-1} \tau^{-\alpha_g}} = \frac{\Gamma(1-\alpha_g)}{\Gamma(1-\alpha_g)} \tau^{\alpha_g - \alpha_l}.$$

For example, when $\alpha_l = 1$ and $\alpha_g > 0.5$, this ratio is at least $\sqrt{\pi}/1.07 \cdot \tau^{0.4}$. This provides a quantitative explanation for the slower decay of the local-view Fourier coefficients compared with the global-view ones.

Non-Markovian random-walk interpretation We can use a non-Markovian random-walk interpretation to illustrate how varying the fractional order α induces multi-scale views. Under the FDE model (2), let ${}_i\mathbf{P}(t) \in \mathbb{R}^N$ be the probability distribution of a fractional-order random walker initialized at v_i; its j-th entry is the probability of being 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 $\Delta\tau > 0$ as the following dynamics:

1. The walker stays at its 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 interpretation 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.

C.2 Stability

C.2.1 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

$$\begin{aligned}
&\|\mathbf{Y}(t) - \tilde{\mathbf{Y}}(t)\| \\
&= \left\| \mathbf{Y}(0) - \tilde{\mathbf{Y}}(0) + I_t^\alpha(\mathcal{F}(\mathbf{W}, \mathbf{Y}(t)) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \tilde{\mathbf{Y}}(t))) \right\| \\
&\leq \|\mathbf{Y}(0) - \tilde{\mathbf{Y}}(0)\| \\
&\quad + I_t^\alpha \|\mathcal{F}(\mathbf{W}, \mathbf{Y}(t)) - \tilde{\mathcal{F}}(\tilde{\mathbf{W}}, \tilde{\mathbf{Y}}(t))\|. \qquad (10)
\end{aligned}$$

Using the triangle inequality, we bound the integrand of the second term on the right-hand side of (10) as follows:

$$\begin{aligned}\|\mathcal{F}(\mathbf{W},\mathbf{Y}(t))-\tilde{\mathcal{F}}(\tilde{\mathbf{W}},\tilde{\mathbf{Y}}(t))\| \\ \leq \|\mathcal{F}(\mathbf{W},\mathbf{Y}(t))-\mathcal{F}(\mathbf{W},\tilde{\mathbf{Y}}(t))\| \\ +\|\mathcal{F}(\mathbf{W},\tilde{\mathbf{Y}}(t))-\tilde{\mathcal{F}}(\tilde{\mathbf{W}},\tilde{\mathbf{Y}}(t))\|.\end{aligned} \tag{11}$$

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

$$\begin{aligned}\|\mathcal{F}(\mathbf{W},\mathbf{Y}(t))-\mathcal{F}(\mathbf{W},\tilde{\mathbf{Y}}(t))\| \\ \leq L\|\mathbf{Y}(t)-\tilde{\mathbf{Y}}(t)\|.\end{aligned} \tag{12}$$

By definition of the combined perturbation ε, the second term on the right-hand side of (11) is bounded as follows:

$$\begin{aligned}\|\mathcal{F}(\mathbf{W},\tilde{\mathbf{Y}}(t))-\tilde{\mathcal{F}}(\tilde{\mathbf{W}},\tilde{\mathbf{Y}}(t))\| \\ \leq \sup_{\mathbf{Y}\in A}\|\mathcal{F}(\mathbf{W},\mathbf{Y})-\tilde{\mathcal{F}}(\tilde{\mathbf{W}},\mathbf{Y})\| \leq \varepsilon.\end{aligned} \tag{13}$$

Combining (10), (12) and (13), we have the inequality

$$\begin{aligned}\|\mathbf{Y}(t)-\tilde{\mathbf{Y}}(t)\| \\ \leq LI_t^{\alpha}\|\mathbf{Y}(t)-\tilde{\mathbf{Y}}(t)\|+(1+\frac{t^{\alpha}}{\Gamma(1+\alpha)})\varepsilon \\ \leq LI_t^{\alpha}\|\mathbf{Y}(t)-\tilde{\mathbf{Y}}(t)\|+(1+\frac{T^{\alpha}}{\Gamma(1+\alpha)})\varepsilon,\end{aligned} \tag{14}$$

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 [55, Lemma 6.19.] to (14), we obtain

$$\|\mathbf{Y}(t)-\tilde{\mathbf{Y}}(t)\| \leq \left(1+\frac{T^{\alpha}}{\Gamma(1+\alpha)}\right)\varepsilon e_{\alpha}(-L,t^{\alpha}). \tag{15}$$

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 (15) that

$$\begin{aligned}\|\mathbf{Y}(t)-\tilde{\mathbf{Y}}(t)\| &\leq \left(1+\frac{T^{\alpha}}{\Gamma(1+\alpha)}\right)\varepsilon O\left(\frac{1}{1+LT^{\alpha}}\right) \\ &= O\left(\varepsilon T^{\alpha-1}\right)\end{aligned} \tag{16}$$

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

C.2.2 Proof of Theorem 3

Subtracting the weighted average solution $\mathbf{Y}(t)$ and the perturbation one yields

$$\mathbf{Y}(t)-\tilde{\mathbf{Y}}(t)=\sum_{k=1}^{\tilde{K}}\beta_k\left(\mathbf{Y}_k(t)-\tilde{\mathbf{Y}}_k(t)\right). \tag{17}$$

Take the norm of (17) and applying the triangle inequality, we obtain

$$\|\mathbf{Y}(t)-\tilde{\mathbf{Y}}(t)\|\leq\sum_{k=1}^{\tilde{K}}\beta_k\|\mathbf{Y}_k(t)-\tilde{\mathbf{Y}}_k(t)\|. \tag{18}$$

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

$$\|\mathbf{Y}_k(t)-\tilde{\mathbf{Y}}_k(t)\|=O\left(\varepsilon T^{\alpha_k-1}\right). \tag{19}$$

Thus, the final unified perturbation bound is given by:

$$\begin{aligned}\|\mathbf{Y}(t)-\tilde{\mathbf{Y}}(t)\| &\leq \sum_{k=1}^{\tilde{K}}\beta_k O\left(\varepsilon T^{\alpha_k-1}\right) \\ &= O\left(\varepsilon T^{\alpha_{\tilde{K}}-1}\right),\end{aligned} \tag{20}$$

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

D Experimental details

D.1 Details of datasets

We refer the reader to Table 10 for detailed statistics of the datasets.

Table 10: 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%

D.2 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 $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$. After AVLA, we freeze the learned encoders and select β using validation accuracy under $\beta_k > 0$ and $\sum_k \beta_k = 1$. Starting from uniform weights, we perform a greedy coordinate-wise search: for each β_k, candidate values are evaluated with a pre-specified step $s_{\widetilde{K}} \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$, while the remaining weights are proportionally renormalized. Finer steps are usually used for smaller $\widetilde{K}$ and coarser steps usually for larger $\widetilde{K}$, keeping each sweep at $O(\widetilde{K}/s_{\widetilde{K}})$ evaluations rather than exhaustively enumerating the simplex. The search stops when no candidate improves validation accuracy. 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., $\widetilde{K}$ and corresonding $\{\alpha_k\}_{k=1}^{\widetilde{K}}$) for each dataset is provided in Table 11.

Table 11: 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	20	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	20	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	20	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	30	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	20	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	20	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	20	5	$\{0.008, 0.240, 0.523, 0.750, 1\}$	$\{0.025, 0.025, 0.075, 0.025, 0.85\}$

D.3 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 [34] and VICReg [43]. 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.

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:

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

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

$$\mathcal{L}_{\text{Cosmean}} = 1 - \frac{1}{N}\sum_{i=1}^{N}\frac{\langle \mathbf{Y}_{l,i}, \mathbf{Y}_{g,i}\rangle}{\|\mathbf{Y}_{l,i}\|_2\|\mathbf{Y}_{g,i}\|_2} \tag{21}$$

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.

Barlow Twins Loss [34] 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.

VICReg Loss (Variance-Invariance-Covariance Regularization Loss) [43] 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: 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

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

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 $\text{Cov}(\mathbf{Y}_l)_{j,k} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{Y}_{l,i,j} - \mu_{l,j}\right)\left(\mathbf{Y}_{l,i,k} - \mu_{l,k}\right)$ 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.

CCA Loss (Canonical Correlation Analysis Loss) [32] 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.

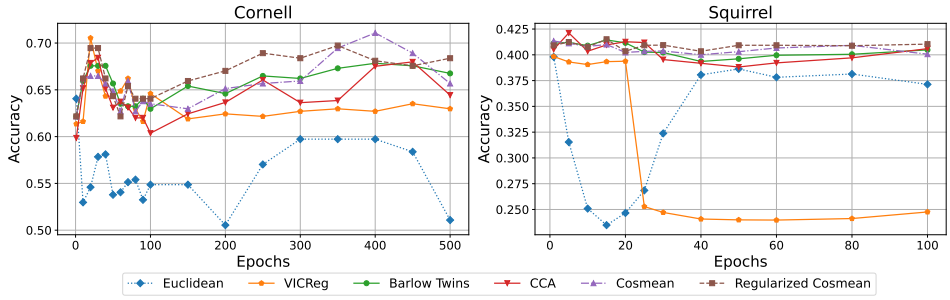


Figure 7: Accuracy vs. training epochs for various loss functions on the Cornell and Squirrel datasets.

E 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.

F Robustness studies

F.1 Attack methods

We consider four *black-box* topology attacks in the evasion setting: Random, PRBCD [47], Nettack [48], and Metattack [49]. Additionally, we evaluate two *white-box* attacks (i.e., PGD [50] and PRBCD)

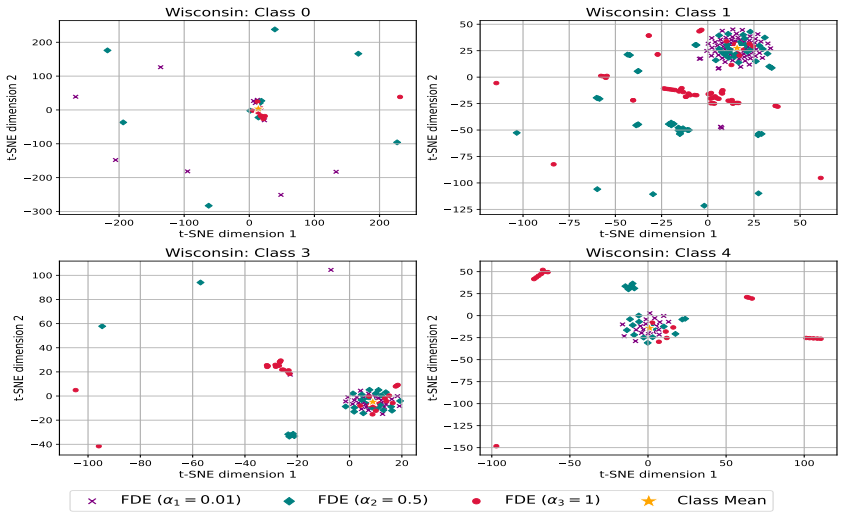


Figure 8: t-SNE visualizations of Wisconsin node features for different class labels.

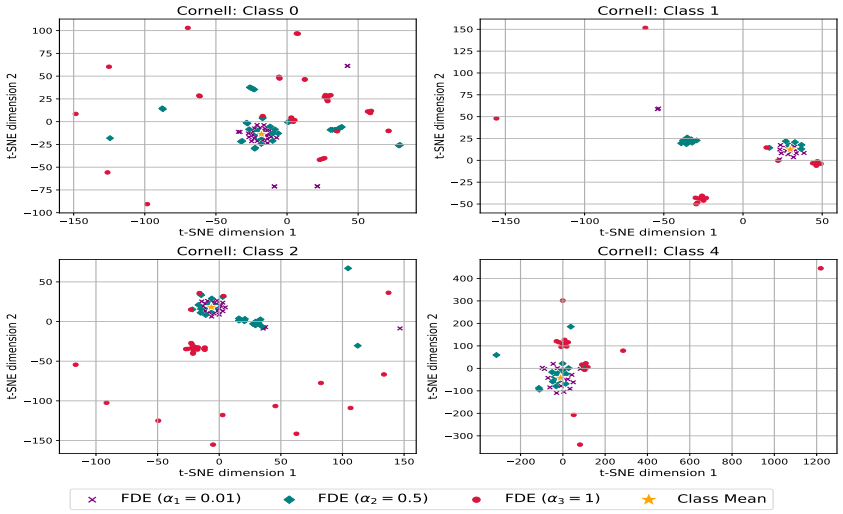


Figure 9: t-SNE visualizations of Cornell node features for different class labels.

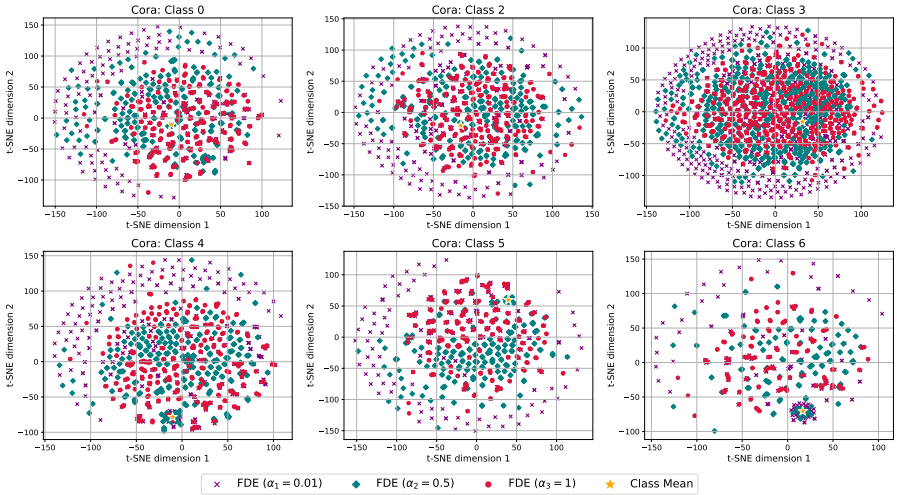


Figure 10: t-SNE visualizations of node features from selected Cora labels. Each point shows embeddings from FDE encoders with different α.

that jointly perturb both the graph structure and node features. The details of these attack methods are as follows:

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.

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.

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.

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.

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. 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.

F.2 More robustness results

To further assess the robustness of FD-MVGCL, we perform node classification under black-box attacks on additional homophilic and heterophilic datasets (e.g., Cora, Citeseer, Photo, Wisconsin, Cornell, Texas and Actor). As shown in Tables 12 and 13, the results reinforce the robustness of FD-MVGCL across a broader range of benchmarks.

Table 12: 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**
Wisconsin	*clean*	67.84±3.84	43.53±6.19	56.08±4.31	52.35±7.18	69.22±0.40	**76.08±3.33**	76.05±6.08	63.73±3.95	**78.04±0.08**
	Random	69.61±4.49	44.71±6.43	51.18±5.44	51.76±6.27	51.56±5.63	75.23±3.13	**76.47±4.12**	59.02±4.59	**80.00±4.00**
	PRBCD	67.65±5.28	44.71±6.72	55.88±4.41	51.37±6.67	52.55±5.13	74.60±3.14	**75.29±4.12**	60.39±6.61	**78.24±5.15**
	Metattack	64.51±5.98	43.53±4.09	50.98±4.64	50.59±6.06	52.15±5.08	**76.67±3.92**	74.71±4.31	60.39±4.79	**77.25±5.35**
	Nettack	70.78±6.17	44.71±5.32	55.29±5.02	50.00±5.70	53.73±5.16	77.65±3.92	75.49±3.73	59.02±2.97	**76.86±5.10**
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 13: 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**
Wisconsin	*clean*	67.84±3.84	52.35±7.18	69.22±0.40	**76.92±1.48**	74.27±1.99	63.73±3.95	52.68±1.21	**78.04±0.08**
	PGD	62.16±6.01	30.45±6.89	8.04±3.75	66.27±6.06	**66.86±7.52**	18.43±10.57	15.29±17.60	**76.86±5.67**
	PRBCD	57.06±5.71	27.56±8.36	11.37±3.56	**60.59±5.65**	57.06±7.36	33.92±11.20	18.24±12.15	**67.25±3.73**
Cornell	*clean*	**63.24±9.38**	51.08±5.19	59.33±1.48	44.11±1.12	55.35±1.32	53.78±3.72	45.59±0.67	**72.70±0.57**
	PGD	44.36±8.34	30.58±6.13	29.46±10.23	38.65±6.17	**48.11±5.10**	20.00±9.98	39.46±11.67	**48.38±6.10**
	PRBCD	43.81±8.00	28.13±5.47	37.03±8.74	36.76±7.47	**44.05±7.65**	20.54±13.20	13.24±8.24	**42.97±7.88**

G 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 14, 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.

Table 14: 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	PTC-MR
MVGRL[39]	74.02±0.30	75.20±0.40	**62.50±1.70**
GraphACL[26]	73.50±0.70	- -	- -
SDMG [31]	73.16±0.16	72.66±3.16	56.70±2.02
InfoGraph[58]	74.44±0.40	72.85±1.70	61.70±1.40
GraphCL[59]	74.39±0.45	**78.62±0.40**	- -
JOAO[60]	74.55±0.41	77.32±0.54	- -
JOAO2[60]	75.35±0.09	77.40±1.11	- -
SimGRACE[61]	75.35±0.09	77.44±1.11	- -
DRGCL[62]	75.20±0.60	78.40±0.70	- -
CI-GCL [63]	**76.50±0.10**	**79.63±0.30**	- -
FD-MVGCL	**75.42±0.29**	78.45±0.43	**62.85±6.19**

H Limitations and future work

This work focuses on static graphs, a deliberate choice to ensure fair comparison with existing methods that operate in the same setting. Our core contributions of removing complex augmentations and negative samples via fractional diffusion are orthogonal to the static/dynamic distinction. Extending it to dynamic graphs is a promising future direction, as the continuous-order nature of fractional diffusion is naturally suited to modeling temporal evolution.

I Broader impacts

This paper introduces a pioneering framework for augmentation-free contrastive learning by means of using FDE-based graph neural diffusion models, poised to significantly influence both the development of GNNs and their applications across diverse domains. By leveraging continuous dynamics governed by fractional differential equations, our approach enhances the flexibility and robustness of representation learning while eliminating the need for both complex augmentations and negative

samples. The societal impact of this work depends on a commitment to ethical standards and responsible use, ensuring that advancements in contrastive learning lead to positive outcomes without exacerbating biases, inequality, or misuse in sensitive applications.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions and scope are detailed in the abstract and introduction Section 1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations in Section H.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The theorems proposed in Section 4 are supported by detailed proofs provided in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have included the implementation details in Appendix D and the hyperparameters in Appendix D.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have provided the source code in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We have included the implementation details in Appendix D and the hyperparameters in Appendix D.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report mean and standard deviation values in our main experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the computing cost in Section 6.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed broader impacts in Section I.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original paper that produced the code package or dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.