SPREAD: SPECTRAL RESPONSE GRAPH DIFFUSION FOR REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have emerged as a promising paradigm for graph representation learning, which employ progressive perturbation and denoising to learn discriminative representations. However, existing methods rely on uniform or predefined diffusion mechanisms that overlook how heterogeneous spectral distributions influence the diffusion dynamics and recovery capabilities of different frequency components. To address this limitation, we propose SPREAD, a spectral response graph diffusion framework that jointly adapts forward perturbation and reverse reconstruction in accordance with graph spectral properties. In the forward process, we introduce a learnable spectral response function parameterized by Bernstein polynomials to regulate noise intensity across frequency components. In the reverse process, a frequency-aware spectral reconstruction objective weights spectral bands according to the learned spectral response profiles, guiding the recovery of informative frequency components. Theoretical analysis establishes that the proposed model can uniformly approximate arbitrary continuous spectral distribution as the polynomial degree increases. Experiments on node classification benchmarks demonstrate that SPREAD achieves competitive performance compared to baseline methods, with ablation studies further supporting the effectiveness of both proposed mechanisms in SPREAD.

1 INTRODUCTION

Graph representation learning has been widely applied in social network analysis (Yang et al., 2024), recommender systems (Wang et al., 2024), and protein–ligand docking (Pei et al., 2023). Graph neural network (GNN)-based methods learn node representations via iterative neighborhood aggregation, and have achieved significant progress on the tasks of node classification, link prediction, and graph-level prediction (Kipf & Welling, 2017). However, stacking deeper GNN layers inevitably triggers two critical flaws: over-smoothing and degradation of discriminative local signals, which largely limit the representation capability and generalization (Li et al., 2018; Oono & Suzuki, 2020).

Diffusion models have achieved remarkable success across natural language processing and computer vision (Li et al., 2022; Hudson et al., 2024), and have recently emerged as a promising paradigm for graph representation learning. Graph diffusion models leverage a progressive forward noising and reverse denoising pipeline to fit the underlying distribution of graph signals, thereby alleviating the aforementioned issues such as over-smoothing and the degradation of discriminative local signals (Bodnar et al., 2022). Existing studies have explored the design of perturbation mechanisms and denoising objectives. For example, DDM (Yang et al., 2023) incorporates graph directional information from graphs to regulate noise propagation, while SDMG (Zhu et al., 2025) introduces a multi-scale smoothing loss to guide representation recovery.

Despite these advances, existing graph diffusion models adopt rigid perturbation mechanisms that apply identical noise across all diffusion timesteps. From the perspective of graph signal processing, node representations can be viewed as discrete graph signals decomposed into orthogonal frequency components (Defferrard et al., 2016). Low-frequency components primarily capture globally smooth structures and homophilic connections, whereas high-frequency components reflect fine-grained local patterns, heterophilic connections, and class decision boundaries (Bo et al., 2021; Chanpuriya & Musco, 2022). Crucially, different frequency components exhibit distinct diffusion dynamics and recovery capabilities against stochastic perturbations. A globally unified diffusion process fails to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

accommodate frequency-wise differences in information preservation. This leads to either over-smoothing of informative low-frequency structures or destruction of discriminative high-frequency details, ultimately degrading the performance of representation learning.

To address this fundamental limitation, we propose SPREAD, a spectral response graph diffusion framework for discriminative graph representation learning. The key idea in SPREAD is to explicitly model the spectral response of graph signals, so that the diffusion process can adapt to intrinsic graph spectral properties. In the forward diffusion process, we design a spectral response perturbation mechanism, which employs a learnable Bernstein-polynomial function to dynamically adjust noise intensity across different frequency components. This design enables sufficient stochastic perturbation for distribution fitting, while avoiding excessive corruption of spectral signals. In the reverse denoising process, we further devise a frequency-aware spectral reconstruction objective. Instead of hand-tuned weighting heuristics, our objective adaptively assigns weights to spectral bands based on the learned spectral response profiles, accounting for heterogeneous contributions of low- and high-frequency graph signals. Overall, these two core components construct a frequency-customized diffusion pipeline that yields more expressive and generalizable node representations.

Our contributions are summarized as follows: *(i)* We propose a spectral response perturbation mechanism. Equipped with a learnable spectral response function, SPREAD adaptively modulates the noise intensity across different frequency components, enabling spectrum-aware perturbation while preserving informative graph signals. *(ii)* We design a frequency-aware spectral reconstruction objective. Guided by spectral response profiles, this objective adaptively assigns weights to spectral bands, thereby improving the recovery quality of frequency components. *(iii)* Extensive experiments on multiple public graph datasets demonstrate that the proposed framework outperforms existing approaches and the efficacy of both proposed modules is validated.

2 PRELIMINARIES

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges respectively, and $N = |\mathcal{V}|$ is the number of nodes. Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ denote the node feature matrix and $\mathbf{A} \in \mathbb{R}^{N \times N}$ denote the adjacency matrix of $\mathcal{G}$. The degree matrix is defined as $\mathbf{D} = \mathrm{diag}\,(d_1, \ldots, d_N)$, where $d_i = \sum_{j \in \mathcal{V}} \mathbf{A}_{ij}$. The normalized graph Laplacian is formulated as $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$. It can be eigendecomposed as $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_N]$ denotes the eigenvector matrix and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_N)$ contains the corresponding eigenvalues.

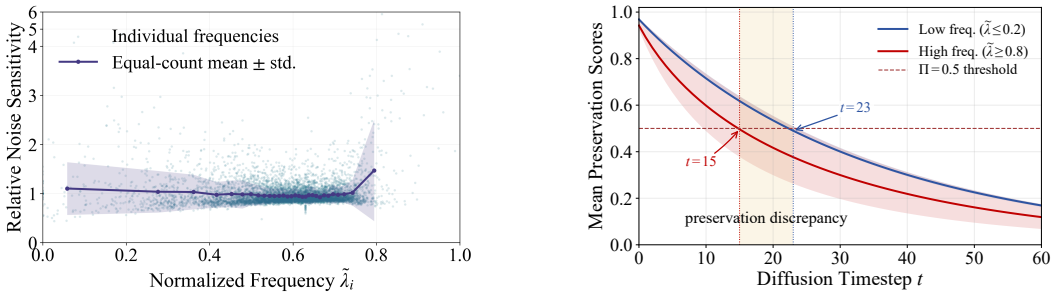
The Graph Fourier Transform (GFT) (Shuman et al., 2013) projects graph signals from the spatial domain into the spectral domain: $\widehat{\mathbf{X}} = \mathbf{U}^\top \mathbf{X}$, where $\widehat{\mathbf{X}}$ represents the spectral coefficients of graph signals. Conversely, the Inverse Graph Fourier Transform (IGFT) can reconstruct the spatial signals as $\mathbf{X} = \mathbf{U}\widehat{\mathbf{X}}$. The eigenvalues in $\mathbf{\Lambda}$ characterize graph frequencies, where smaller eigenvalues correspond to lower-frequency components with smooth variations over the graph, while larger eigenvalues represent higher-frequency components capturing rapid signal changes. Background on diffusion models is provided in Appendix C.

3 MOTIVATION: SPECTRAL HETEROGENEITY

Graph signals exhibit spectral heterogeneity, as different frequency components generally carry unequal amounts of signal energy. However, the conventional diffusion process applies a uniform perturbation mechanism to all frequencies, and thus fails to account for the disparities in spectral Signal-to-Noise Ratio (SNR).

Let $\widehat{\mathbf{X}}_0 = \mathbf{U}^\top \mathbf{X}_0$ and $\widehat{\epsilon}_{\mathcal{G}} = \mathbf{U}^\top \epsilon_{\mathcal{G}}$ denote the spectral coefficients of input signal and projected noise, respectively. Under conventional uniform diffusion, the i-th frequency component evolves as: $\widehat{\mathbf{X}}^{uni}_{t,i:} = \sqrt{\bar{\alpha}_t}\,\widehat{\mathbf{X}}_{0,i:} + \sqrt{1-\bar{\alpha}_t}\,\widehat{\epsilon}_{\mathcal{G},i:}$. For a fixed input $\mathbf{X}_0$ with d features, the spectral signal power $P_{X,i}$, expected noise power $P_{\epsilon,i}$, and relative noise sensitivity χ_i are defined as:

$$P_{X,i} = \frac{1}{d}\left\|\widehat{\mathbf{X}}_{0,i:}\right\|_2^2, \quad P_{\epsilon,i} = \mathbb{E}_{\epsilon_{\mathcal{G}}}\left[\frac{1}{d}\left\|\widehat{\epsilon}_{\mathcal{G},i:}\right\|_2^2\right], \quad \chi_i = \frac{P_{\epsilon,i}}{P_{X,i}}. \tag{1}$$



(a) Spectral heterogeneity

(b) Preservation under uniform perturbation

Figure 1: Spectral heterogeneity and its preservation on the Photo dataset. (a) Relative noise sensitivity across the full spectrum with mean and standard deviation. (b) Mean preservation scores of low- and high-frequency components under uniform perturbation.

A larger χ_i indicates greater noise power relative to signal power. As shown in Figure 1(a), χ_i varies substantially across the spectrum of the Photo dataset, revealing heterogeneous frequency-dependent sensitivity to perturbation. The spectral SNR is defined as the ratio of the signal-term power to the expected noise-term power:

$$\mathrm{SNR}_i^{uni}(t) = \frac{\bar{\alpha}_t P_{X,i}}{(1-\bar{\alpha}_t)P_{\epsilon,i}} = \underbrace{\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}}_{\gamma_t} \frac{1}{\chi_i}, \tag{2}$$

where γ_t is a time-dependent scaling factor shared by all frequency components. Consequently, for the i-th and j-th frequency components, the relative spectral SNR is independent of the timestep: $\frac{\mathrm{SNR}_i^{uni}(t)}{\mathrm{SNR}_j^{uni}(t)} = \frac{\chi_j}{\chi_i}$. A global diffusion schedule only scales all spectral SNRs through the common factor γ_t and therefore cannot change their ratios. To characterize the implications of these disparities for information preservation, we define a bounded, power-based preservation score:

$$\Pi_i^{uni}(t) = \frac{\bar{\alpha}_t P_{X,i}}{\bar{\alpha}_t P_{X,i} + (1-\bar{\alpha}_t)P_{\epsilon,i}} = \frac{\mathrm{SNR}_i^{uni}(t)}{1+\mathrm{SNR}_i^{uni}(t)} = \frac{\gamma_t}{\gamma_t + \chi_i}. \tag{3}$$

Eq. (3) quantifies how much information the i-th frequency component preserves at timestep t under uniform diffusion, with values close to 1 indicating signal dominance and values close to 0 indicating noise dominance. As shown in Figure 1(b), the mean scores of the high- and low-frequency components reach the threshold $\Pi = 0.5$ at $t = 15$ and $t = 23$, respectively. This discrepancy shows that uniform perturbation cannot synchronize information preservation across frequencies, potentially causing over-smoothing due to low-frequency dominance or premature loss of local details. Using frequency-specific response coefficients r_i yields:

$$\mathrm{SNR}_i^{ada}(t) = \frac{\gamma_t}{r_i^2 \chi_i}, \quad \frac{\mathrm{SNR}_i^{ada}(t)}{\mathrm{SNR}_j^{ada}(t)} = \frac{r_j^2 \chi_j}{r_i^2 \chi_i}, \tag{4}$$

thereby making the relative spectral SNR adjustable. This motivates us to learn an adaptive perturbation mechanism that adjusts noise strength according to graph spectral properties.

4 Method

4.1 Overall Framework

The overall framework of SPREAD is illustrated in Figure 2. In the forward process, SPREAD transforms graph signals into the spectral domain and applies a learnable spectral response function to adaptively regulate noise intensity across different frequency components. In the reverse process, a conditional denoising network recovers node representations from a noisy distribution, guided by a frequency-aware spectral reconstruction objective that adaptively assigns weights to individual spectral bands according to their spectral response profiles.

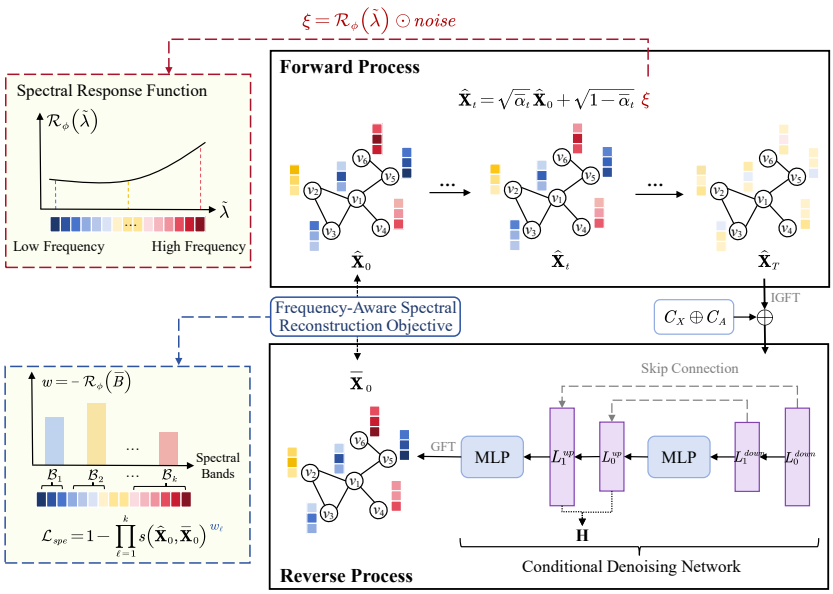


Figure 2: Overview of the SPREAD framework.

4.2 Forward Process

The forward diffusion process aims to progressively perturb node representations while preserving critical structural and semantic information. Unlike prior graph diffusion models that employ a unified noise schedule, SPREAD performs diffusion in the spectral domain.

4.2.1 Learnable Spectral Response Function

To enable frequency-dependent diffusion, SPREAD introduces a learnable spectral response function $\mathcal{R}_{\boldsymbol{\phi}}(\cdot)$ that maps each graph frequency to a corresponding response coefficient. We parameterize $\mathcal{R}_{\boldsymbol{\phi}}(\cdot)$ as a Bernstein polynomial of degree M. This parameterization supports both monotonic and non-monotonic response profiles, thereby accommodating diverse diffusion behaviors across different frequency components. The m-th Bernstein basis polynomial of degree M is defined as

$$\mathcal{S}_{m,M}(x) = \binom{M}{m} x^m (1-x)^{M-m}, \quad m = 0, \ldots, M. \tag{5}$$

Given the normalized graph frequency $\tilde{\lambda}_i \in [0, 1]$, its spectral-response coefficient is computed as

$$r_i = \mathcal{R}_{\boldsymbol{\phi}}(\tilde{\lambda}_i) = \sum_{m=0}^{M} c_m \mathcal{S}_{m,M}(\tilde{\lambda}_i), \tag{6}$$

where $\boldsymbol{\phi} = \{c_m\}_{m=0}^{M}$ denotes the set of learnable coefficients. The resulting coefficients form a spectral modulation vector $\mathbf{R} = (r_1, \ldots, r_N)^\top$, which governs the noise magnitude assigned to each frequency component.

Theorem 4.1 establishes the universal approximation capability of the proposed spectral response function and highlights its approximation advantage over constant responses.

Theorem 4.1 (Expressivity of the Learnable Spectral Response)**.** *For $M \geq 1$, the degree M Bernstein response class and constant-response class are defined as $\mathcal{F}_M = \{\sum_{m=0}^{M} c_m \mathcal{S}_{m,M} \mid c_m \in \mathbb{R}\}$ and $\mathcal{C} = \{x \mapsto c \mid c \in \mathbb{R}\}$, respectively. For any target $f \in C([0,1])$, let $\mathcal{E}_M(f) = \inf_{g \in \mathcal{F}_M} \|f - g\|_\infty$ and $\mathcal{E}_{\text{const}}(f) = \inf_{g \in \mathcal{C}} \|f - g\|_\infty$ denote the optimal uniform approximation errors of the two response classes, respectively. Then the following two properties hold:*

$$(i) \quad \lim_{M \to \infty} \mathcal{E}_M(f) = 0, \quad (ii) \quad \mathcal{E}_M(f) \leq \mathcal{E}_{\text{const}}(f), \quad \forall\, M \geq 1. \tag{7}$$

The proof of Theorem 4.1 is provided in Appendix A.1.

4.2.2 FREQUENCY-ADAPTIVE FORWARD DIFFUSION

Given the normalized graph Laplacian $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, the initial node representation $\mathbf{X}_0$ is first transformed into the spectral domain: $\widehat{\mathbf{X}}_0 = \mathbf{U}^\top\mathbf{X}_0$, where $\widehat{\mathbf{X}}_0$ represents the spectral coefficients of the graph signals. Likewise, the graph directional noise $\epsilon_{\mathcal{G}}$ (Yang et al., 2023) is projected into the spectral domain as $\hat{\epsilon}_{\mathcal{G}} = \mathbf{U}^\top\epsilon_{\mathcal{G}}$.

Based on the learnable spectral response function, SPREAD reformulates the forward process by adaptively modulating perturbation magnitude across different frequency components. Specifically, the frequency-aware noise is first constructed as: $\xi = \mathbf{R} \odot \hat{\epsilon}_{\mathcal{G}}$, where $\odot$ indicates element-wise multiplication. Given a diffusion timestep t, the forward diffusion process is formulated as: $\widehat{\mathbf{X}}_t = \sqrt{\bar{\alpha}_t}\widehat{\mathbf{X}}_0 + \sqrt{1-\bar{\alpha}_t}\xi$, where $\bar{\alpha}_t$ controls the signal preservation ratio at timestep t. Finally, the noisy representation is transformed back into the spatial domain: $\mathbf{X}_t = \mathbf{U}\widehat{\mathbf{X}}_t$.

The proposed spectral response perturbation mechanism establishes frequency-dependent diffusion trajectories while retaining the global temporal noise schedule. Moreover, the learned spectral response profiles will be reused in the reverse reconstruction objective, thereby maintaining consistency between forward perturbation and reverse recovery.

4.3 REVERSE PROCESS

4.3.1 CONDITIONAL DENOISING NETWORK

The reverse process aims to recover clean node representations from noisy states generated in the forward diffusion process. Given the perturbed representation $\mathbf{X}_t$, a conditional denoising network is employed to reconstruct the original representation:

$$\widetilde{\mathbf{X}}_0 = \rho_\theta(\mathbf{X}_t, t, \mathbf{C}_X, \mathbf{C}_A), \tag{8}$$

where $\rho_\theta(\cdot)$ denotes the denoising network, and $\mathbf{C}_X$ and $\mathbf{C}_A$ denote the conditional information derived from node features and graph structure, respectively. Details of the denoising network architecture are provided in Appendix D.

4.3.2 FREQUENCY-AWARE SPECTRAL RECONSTRUCTION OBJECTIVE

Although spatial domain reconstruction preserves node-level attribute information, it may overlook global topology and long-range dependencies. SPREAD introduces a frequency-aware spectral reconstruction objective that provides frequency-specific supervision during the reverse process.

Spectral Band Partitioning. When reconstruction losses are evaluated separately for each frequency component, noise can induce fluctuations in the corresponding loss term, impairing the stability of denoising optimization. To mitigate this effect, we group neighboring frequency components into spectral bands and reconstruct at the band level. The variance bound in Theorem 4.2 supports this design. The normalized frequencies are first sorted as $\tilde{\lambda}_{\pi_1} \leq \cdots \leq \tilde{\lambda}_{\pi_N}$, where $(\pi_1, \ldots, \pi_N)$ is a permutation of frequency indices. We then partition them into k disjoint bands. Let $q_\ell = \mathrm{round}((\ell-1)N/k)$ for $\ell \in \{1, \ldots, k+1\}$. The ℓ-th spectral band is defined as $\mathcal{B}_\ell = \{\pi_p \mid q_\ell < p \leq q_{\ell+1}\}$. Each band contains approximately N/k components, with its frequency interval determined by the graph spectrum. The representative frequency of each band is computed as $\bar{B}_\ell = \frac{1}{|\mathcal{B}_\ell|}\sum_{i\in\mathcal{B}_\ell}\tilde{\lambda}_i$.

Adaptive Weights. SPREAD uses the same learnable spectral response function to guide spectral band weighting during reconstruction. Specifically, we evaluate the response at the representative frequency of each band to obtain its weighting score:

$$b_\ell = \mathcal{R}_{\boldsymbol{\phi}}(\bar{B}_\ell) = \sum_{m=0}^{M} c_m \mathcal{S}_{m,M}(\bar{B}_\ell). \tag{9}$$

We then apply a temperature-controlled softmax to yield normalized spectral band weights:

$$w_\ell = \mathrm{Softmax}\left(-\frac{b_\ell - \mu_b}{\tau(\sigma_b + \Delta)}\right), \tag{10}$$

where μ_b and σ_b denote the mean and standard deviation of $\{b_\ell\}_{\ell=1}^k$, respectively, τ is a temperature parameter, and Δ is a small constant added for numerical stability. The negative sign implements inverse-response weighting, assigning lower weights to strongly perturbed bands. Consequently, the reconstruction objective focuses on reliable frequency components while preventing heavily corrupted bands from dominating the optimization.

Spectral Reconstruction Loss. To supervise the recovery of spectral information, SPREAD projects both the original representation $\mathbf{X}_0$ and the reconstructed representation $\widetilde{\mathbf{X}}_0$ into the graph spectral domain: $\widehat{\mathbf{X}}_0 = \mathbf{U}^\top \mathbf{X}_0$, $\bar{\mathbf{X}}_0 = \mathbf{U}^\top \widetilde{\mathbf{X}}_0$. Then the reconstruction consistency between the original and recovered representations is measured within each spectral band:

$$s_\ell = \frac{1}{|\mathcal{B}_\ell|} \sum_{i \in \mathcal{B}_\ell} S\left(\widehat{\mathbf{X}}_0^i, \bar{\mathbf{X}}_0^i\right), \tag{11}$$

where $S(\cdot)$ denotes normalized cosine similarity, and $\widehat{\mathbf{X}}_0^i$ and $\bar{\mathbf{X}}_0^i$ denote the original and reconstructed representations of the i-th frequency component, respectively. The overall spectral reconstruction loss is formulated as a weighted geometric aggregation:

$$\mathcal{L}_{spe} = \mathbb{E}_t \left\{ \delta(t)\, \mathbb{E}_{\mathbf{X}_0} \mathbb{E}_{\mathbf{X}_t|\mathbf{X}_0} \left[1 - \prod_{\ell=1}^{k} s_\ell^{w_\ell} \right] \right\}. \tag{12}$$

Because band-wise similarities are combined through a weighted product, low similarity in any spectral band reduces the overall score, encouraging good reconstruction quality across all bands. The adaptive weights assign greater influence to more reliable spectral bands, thereby providing frequency-aware supervision for preserving informative spectral content during reverse denoising.

Spectral-band averaging reduces reconstruction similarity variance, while adaptive weights determine the tightness of band-wise reconstruction constraints, as formalized in Theorem 4.2.

Theorem 4.2 (Band-wise Stability and Constraint Propagation)**.** *Let $\zeta_i = S(\widehat{\mathbf{X}}_0^i, \bar{\mathbf{X}}_0^i)$ denote the reconstruction similarity of the i-th frequency component. Given $\mathbf{X}_0$ and t, the randomness in ζ_i arises from sampling $\mathbf{X}_t \sim q(\mathbf{X}_t \mid \mathbf{X}_0)$. For the ℓ-th spectral band, assume that $\mathrm{Var}(\zeta_i \mid \mathbf{X}_0, t) \le \sigma_\ell^2$ for $i \in \mathcal{B}_\ell$ and $\mathrm{Cov}(\zeta_i, \zeta_{i'} \mid \mathbf{X}_0, t) \le \rho_\ell \sigma_\ell^2$ for distinct $i, i' \in \mathcal{B}_\ell$, where $0 \le \rho_\ell < 1$. Then, the band-wise similarity s_ℓ satisfies*

$$\mathrm{Var}(s_\ell \mid \mathbf{X}_0, t) \le \sigma_\ell^2 \left(\rho_\ell + \frac{1 - \rho_\ell}{|\mathcal{B}_\ell|} \right). \tag{13}$$

Moreover, under an aggregate reconstruction error bound of $\eta \in [0, 1)$, the reconstruction similarity of each spectral band satisfies

$$s_\ell \ge (1 - \eta)^{1/w_\ell}, \quad \ell = 1, \ldots, k. \tag{14}$$

The proof of Theorem 4.2 is provided in Appendix A.2.

4.4 Overall Training Objectives

The proposed SPREAD jointly optimizes spectral information preservation, spatial representation recovery, and structural consistency during the reverse diffusion process. The spectral reconstruction loss $\mathcal{L}_{spe}$ introduced in Section 4.3.2 provides frequency-aware supervision by adaptively preserving informative frequency components. To jointly preserve frequency-level and node-level information, we further measure the consistency between the reconstructed and original node representations in the spatial domain:

$$\mathcal{L}_{spa} = \mathbb{E}_t \left\{ \delta(t)\, \mathbb{E}_{\mathbf{X}_0} \mathbb{E}_{\mathbf{X}_t|\mathbf{X}_0} \left[1 - S\left(\mathbf{X}_0, \widetilde{\mathbf{X}}_0\right) \right] \right\}. \tag{15}$$

To further preserve the underlying graph topology during denoising, we introduce a structural consistency loss that constrains the structural condition C_A to be consistent with the original graph Laplacian:

$$\mathcal{L}_{str} = \left\| \mathbf{L} - C_A C_A^\top \right\|_F^2. \tag{16}$$

Table 1: Comparison of accuracy (%) for node classification. Bold: best, underlined: second-best.

Category	Method	Cora	Citeseer	PubMed	Computer	Photo
Supervised	GCN	80.76±0.52	70.64±0.72	78.94±0.47	86.35±0.43	92.53±0.38
	GAT	82.56±0.51	71.68±0.44	79.00±0.23	86.58±0.85	92.64±0.61
Self-supervised	DGI	82.00±0.58	71.42±0.55	76.83±0.42	83.95±0.47	91.61±0.22
	MVGRL	83.06±0.31	72.90±0.35	79.20±0.43	87.52±0.11	91.74±0.07
	BGRL	82.70±0.60	71.10±0.80	79.60±0.50	87.98±0.11	92.82±0.31
	CCA-SSG	83.54±0.44	72.76±0.70	80.78±0.39	89.02±0.33	92.86±0.15
	SP-GCL	83.08±0.36	71.21±0.37	79.02±0.44	89.43±0.37	92.60±0.15
	GraphACL	84.10±0.29	**73.30±0.36**	<u>81.64±0.42</u>	89.36±0.48	93.26±0.14
	GraphMAE	83.18±0.37	71.90±0.30	80.12±0.43	88.60±0.20	93.60±0.20
	GraphTCM	81.58±0.39	72.63±0.25	77.16±0.36	85.41±0.46	91.81±0.20
Diffusion	DDM	82.62±0.37	72.46±0.31	79.32±0.13	89.12±0.30	93.80±0.29
	SDMG	<u>84.18±0.20</u>	72.68±0.58	79.36±0.17	<u>90.76±0.15</u>	<u>94.48±0.33</u>
Proposed	SPREAD	**84.64±0.34**	<u>73.11±0.26</u>	**81.76±0.21**	**91.81±0.32**	**95.69±0.16**

The overall training objective is formulated as:

$$\min_{\Theta} \quad \mathcal{L}(\Theta) = \gamma_1 \mathcal{L}_{spa} + \gamma_2 \mathcal{L}_{spe} + \mathcal{L}_{str}, \tag{17}$$

where Θ denotes all learnable parameters of SPREAD, and γ_1 and γ_2 are trade-off coefficients that control the contributions of the two reconstruction objectives.

5 Experiments

We evaluate the performance of SPREAD on five benchmark datasets for node classification, comparing it with representative baselines. We also conduct ablation studies to assess the effectiveness of key components, complemented by parameter sensitivity analysis and visualizations. Detailed experimental settings and the results for graph classification are provided in Appendices E and F, respectively.

5.1 Experimental Setup

Datasets. We evaluate SPREAD on both node-level and graph-level representation learning tasks. For node classification, we use five widely used datasets, including three citation networks, Cora, Citeseer, and PubMed (Sen et al., 2008), and two co-purchase networks, Computer and Photo (Shchur et al., 2018). For graph classification, we consider five benchmark datasets, including IMDB-B, IMDB-M, PROTEINS, COLLAB, and MUTAG (Yanardag & Vishwanathan, 2015). Detailed statistics and descriptions are provided in Appendix E.

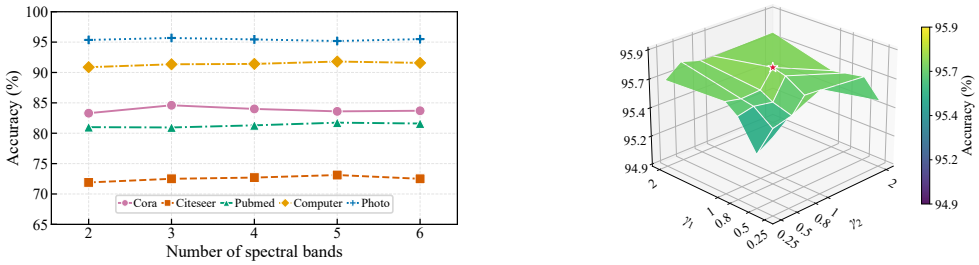
Baselines. We compare SPREAD with representative graph representation learning methods from three categories. *Supervised Methods* include GCN (Kipf & Welling, 2017) and GAT (Veličković et al., 2018). *Self-Supervised Methods* include DGI (Veličković et al., 2019), MVGRL (Hassani & Khasahmadi, 2020), BGRL (Thakoor et al., 2022), CCA-SSG (Zhang et al., 2021), SP-GCL (Wang et al., 2023), GraphACL (Xiao et al., 2023), GraphMAE (Hou et al., 2022), and GraphTCM (Fang et al., 2024), covering contrastive and generative approaches to self-supervised learning. *Diffusion-Based Methods* include DDM (Yang et al., 2023) and SDMG (Zhu et al., 2025). Further discussion of related work is provided in Appendix B.

5.2 Node Classification

We evaluate SPREAD on node classification, with results reported in Table 1. SPREAD achieves the best performance on four datasets, demonstrating the effectiveness of the proposed framework. Notably, it yields the largest gains on the co-purchase graphs, Computer and Photo, surpassing the strongest baselines by 1.05 and 1.21 percentage points, respectively. These improvements suggest that jointly adapting perturbation and reconstruction enables SPREAD to better capture diverse

Table 2: Ablation studies. Bold: best.

Method	Cora	Citeseer	PubMed	Computer	Photo
w/o SRP	82.64±0.27	72.49±0.31	78.76±0.36	91.12±0.39	94.97±0.32
w/o Weight	83.95±0.32	72.38±0.53	81.12±0.37	91.09±0.14	95.06±0.21
w/o Spe	83.61±0.30	72.32±0.48	79.80±0.27	90.95±0.32	95.11±0.34
w/o Spa	83.78±0.29	71.96±0.30	78.98±0.25	90.15±0.38	94.62±0.26
SPREAD	**84.64±0.34**	**73.11±0.26**	**81.76±0.21**	**91.81±0.32**	**95.69±0.16**



(a) Number of spectral bands (b) Trade-off coefficients

Figure 3: Parameter sensitivity of SPREAD. (a) Number of spectral bands k across datasets for node classification. (b) Trade-off coefficients γ_1 and γ_2 of the two reconstruction objectives on the Photo dataset.

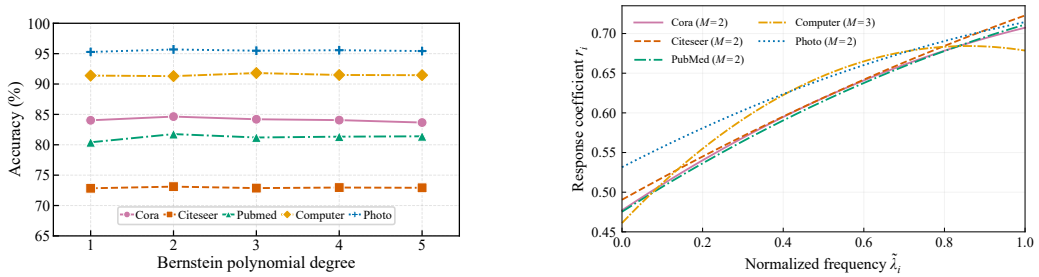
structural and interaction patterns in co-purchase networks. On citation datasets, SPREAD achieves clear gains on Cora and PubMed and remains competitive on Citeseer. Citeseer is characterized by high-dimensional and sparse node features. Its limited connectivity restricts the structural context available for frequency-adaptive reconstruction. Overall, SPREAD consistently outperforms the diffusion-based baselines across all datasets, highlighting the effectiveness of frequency-adaptive diffusion over the conventional uniform diffusion.

5.3 Ablation Study

To assess the contribution of each key component, we compare SPREAD with four ablated variants under identical settings: *w/o SRP* removes the spectral response perturbation (SRP) from the forward process; *w/o Weight* replaces adaptive spectral band weights with uniform weights; *w/o Spe* and *w/o Spa* remove $\mathcal{L}_{spe}$ and $\mathcal{L}_{spa}$, respectively. As shown in Table 2, removing any component reduces classification accuracy, supporting their contributions to representation learning. The pronounced degradation without the spectral response function highlights the importance of adapting noise intensity across frequency components. Spatial and spectral reconstruction play complementary roles by preserving node-level semantic consistency and supervising the recovery of spectral information, respectively. Replacing adaptive spectral band weights with uniform weights also reduces classification accuracy despite retaining spectral reconstruction.

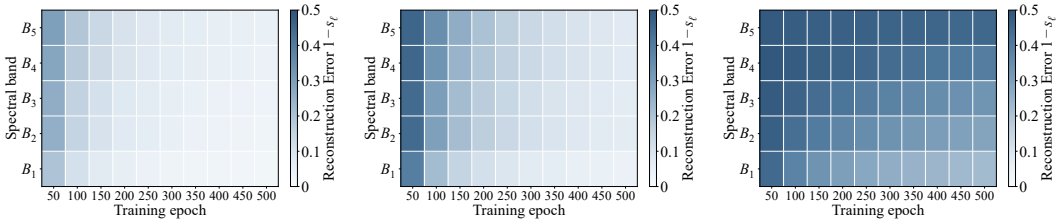
5.4 Parameter Analysis

We evaluate the sensitivity of SPREAD to the number of spectral bands k, the trade-off coefficients γ_1 and γ_2, and the Bernstein polynomial degree M. In Figure 3(a), the Cora and Photo datasets favor $k = 3$ and the remaining datasets $k = 5$, reflecting heterogeneous spectral features. Coarse partitions may obscure spectral heterogeneity, whereas overly fine partitions may reduce the reliability of band-level reconstruction signals. For the trade-off coefficients, accuracy on the Photo dataset varies by smaller than one percentage point across the tested combinations (Figure 3(b)), indicating robustness across the evaluated range. Figure 4(a) shows that SPREAD achieves its highest accuracy at $M = 2$ on the Cora, Citeseer, PubMed, and Photo datasets, and at $M = 3$ on the Computer dataset. Higher degrees yield no consistent gains, suggesting that low-degree Bernstein polynomials provide sufficient flexibility for the learnable spectral response function. Results for the remaining datasets are provided in Appendix G.



(a) Bernstein polynomial degree (b) Learned spectral response profiles

Figure 4: Analysis of the spectral response function across datasets for node classification. (a) Sensitivity to the Bernstein polynomial degree M. (b) Each profile shows the response coefficient $r_i = \mathcal{R}_{\phi}(\tilde{\lambda}_i)$ as a function of the normalized frequency $\tilde{\lambda}_i$.



(a) SPREAD (b) SDMG (c) DDM

Figure 5: Band-wise reconstruction errors $1-s_\ell$ on the PubMed dataset. All panels use the same color scale, with lighter colors indicating lower errors.

5.5 Visualization

Learnable Spectral Response Function. Figure 4(b) shows the learned spectral response profiles for five datasets, revealing how the response coefficients vary with frequency. While the response profiles for the other datasets are monotonically increasing, the profile for the Computer dataset is non-monotonic: it peaks around $\tilde{\lambda} = 0.85$ and then decreases slightly. This decline moderates noise injection at the high frequencies, potentially helping preserve discriminative high-frequency information during the forward process. This phenomenon suggests that stronger perturbation at higher frequencies is not universally preferred. Together, these observations highlight the flexibility of the learnable spectral response function to accommodate both monotonic and non-monotonic patterns, adapting noise modulation to the spectral characteristics of different datasets.

Frequency-Aware Spectral Reconstruction. Figure 5 compares the band-wise reconstruction errors $1-s_\ell$ of DDM, SDMG, and SPREAD on the PubMed dataset. DDM and SDMG show improvements primarily in the lowest-frequency band, whereas SPREAD achieves error reductions across the spectrum, particularly in $\mathcal{B}_1$. The other bands also improve despite receiving smaller weights, suggesting that response-guided weighting prioritizes low-frequency reconstruction while supporting recovery in other spectral bands. Results for the other datasets are provided in Appendix H.

6 Conclusion

We present SPREAD, a frequency-adaptive diffusion framework for graph representation learning. Its learnable spectral response function regulates noise intensity across frequencies and guides spectral band weighting during reconstruction. Experiments show that SPREAD achieves competitive classification accuracy compared with baselines, with ablation studies supporting the contributions of the spectral response perturbation and the frequency-aware spectral reconstruction. Future work will explore efficient spectral approximations for large graphs and extend SPREAD to dynamic graphs with evolving spectral properties.

7 AI USE STATEMENT

We used generative AI solely for language polishing and reviewed all revisions to preserve the intended meaning and technical content. We take full responsibility for the manuscript.

8 REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the source code of SPREAD in the supplementary material. Dataset descriptions, hyperparameter settings, evaluation protocols, and the computational environment are reported in Appendix E. The proofs of our theoretical results are provided in Appendix A.

REFERENCES

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:3950–3957, 2021.

Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Liò, and Michael Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. In *Advances in Neural Information Processing Systems*, volume 35, pp. 18527–18541, 2022.

Sudhanshu Chanpuriya and Cameron Musco. Simplified graph convolution with heterophily. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27184–27197, 2022.

Zhaoliang Chen, Zhihao Wu, Shiping Wang, and Wenzhong Guo. Dual low-rank graph autoencoder for semantic and topological networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4191–4198, 2023.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pp. 3844–3852, 2016.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Taoran Fang, Wei Chow, Yifei Sun, Kaiqiao Han, Lvbin Ma, and Yang Yang. Exploring correlations of self-supervised tasks for graphs. In *International Conference on Machine Learning*, 2024.

Yisen Gao, Xingcheng Fu, Qingyun Sun, Jianxin Li, and Xianxian Li. Toward a unified geometry understanding: Riemannian diffusion framework for graph generation and prediction. In *Advances in Neural Information Processing Systems*, volume 38, pp. 74249–74275, 2025.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, volume 119, pp. 4116–4126, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. GraphMAE: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 594–604, 2022.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. GPT-GNN: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1857–1867, 2020.

Drew A. Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K. Lampinen, Andrew Jaegle, James L. McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. SODA: Bottleneck diffusion models for representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23115–23127, 2024.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, pp. 10362–10383, 2022.

Seoyoon Kim, Hyemin Jung, and Woohyung Lim. Laplacian-guided denoising graph diffusion for graph learning with an adaptive prior. In *New Perspectives in Graph Machine Learning*, 2025.

Thomas N. Kipf and Max Welling. Variational graph auto-encoders. In *NIPS Workshop on Bayesian Deep Learning*, 2016.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Joongwon Lee, Seonghwan Kim, Seokhyun Moon, Hyunwoo Kim, and Woo Youn Kim. FragFM: Hierarchical framework for efficient molecule generation via fragment-level discrete flow matching. In *International Conference on Learning Representations*, 2026.

Jingci Li, Guangquan Lu, Zhengtian Wu, and Fuqing Ling. Multi-view representation model based on graph autoencoder. *Information Sciences*, 632:439–453, 2023a.

Jintang Li, Ruofan Wu, Wangbin Sun, Liang Chen, Sheng Tian, Liang Zhu, Changhua Meng, Zibin Zheng, and Weiqiang Wang. What's behind the mask: Understanding masked graph modeling for graph autoencoders. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1268–1279, 2023b.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3538–3545, 2018.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4328–4343, 2022.

Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. SnapFusion: Text-to-image diffusion model on mobile devices within two seconds. In *Advances in Neural Information Processing Systems*, volume 36, pp. 20662–20678, 2023c.

Haifeng Luo, Huanhuan Li, Tianqiang Huang, and Liqing Huang. PMGAE: Self-supervised graph representation with proximity matrix reconstruction auto-encoders. *Neurocomputing*, 671: 132617, 2026.

Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. SPECTRE: Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators. In *International Conference on Machine Learning*, pp. 15159–15179, 2022.

Giorgia Minello, Alessandro Bicciato, Luca Rossi, Andrea Torsello, and Luca Cosmo. Generating graphs via spectral diffusion. In *International Conference on Learning Representations*, 2025.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 4474–4484, 2020.

Peizhi Niu, Yu-Hsiang Wang, Vishal Rana, Chetan Rupakheti, Abhishek Pandey, and Olgica Milenkovic. DMol: A highly efficient and chemical motif-preserving molecule generation platform. In *Advances in Neural Information Processing Systems*, volume 38, pp. 157397–157431, 2025.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020.

Qizhi Pei, Kaiyuan Gao, Lijun Wu, Jinhua Zhu, Yingce Xia, Shufang Xie, Tao Qin, Kun He, Tie-Yan Liu, and Rui Yan. FABind: Fast and accurate protein-ligand binding. In *Advances in Neural Information Processing Systems*, pp. 55963–55980, 2023.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1150–1160, 2020.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L. Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. In *International Conference on Learning Representations*, 2022.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations*, 2023.

Bohao Wang, Jiawei Chen, Changdong Li, Sheng Zhou, Qihao Shi, Yang Gao, Yan Feng, Chun Chen, and Can Wang. Distributionally robust graph-based recommendation system. In *Proceedings of the ACM Web Conference*, pp. 3777–3788, 2024.

Haonan Wang, Jieyu Zhang, Qi Zhu, Wei Huang, Kenji Kawaguchi, and Xiaokui Xiao. Single-pass contrastive learning can work for both homophilic and heterophilic graph. *Transactions on Machine Learning Research*, 2023. ISSN 2835–8856.

Daniel Wesego. Graph representation learning with diffusion generative models. In *New Perspectives in Graph Machine Learning*, 2025.

Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z. Li. SimGRACE: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference*, pp. 1070–1079, 2022.

Teng Xiao, Huaisheng Zhu, Zhengyu Chen, and Suhang Wang. Simple and asymmetric graph contrastive learning without augmentations. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16129–16152, 2023.

Yu Xie, Jie Jia, Chao Wen, Deyu Li, and Ming Li. Multi-topology contrastive graph representation learning. *Science China Information Sciences*, 69:122102, 2026.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Pinar Yanardag and S.V.N. Vishwanathan. Deep graph kernels. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015.

Run Yang, Yuling Yang, Fan Zhou, and Qiang Sun. Directional diffusion models for graph representation learning. In *Advances in Neural Information Processing Systems*, pp. 32720–32731, 2023.

Yonghui Yang, Le Wu, Zihan Wang, Zhuangzhuang He, Richang Hong, and Meng Wang. Graph bottlenecked social recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3853–3862, 2024.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5812–5823, 2020.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pp. 12121–12132, 2021.

Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S. Yu. From canonical correlation analysis to self-supervised graph neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 76–89, 2021.

Cai Zhou, Xiyuan Wang, and Muhan Zhang. Unifying generation and prediction on graphs with latent graph diffusion. In *Advances in Neural Information Processing Systems*, volume 37, pp. 61963–61999, 2024.

Junyou Zhu, Langzhou He, Chao Gao, Dongpeng Hou, Zhen Su, Philip S. Yu, Juergen Kurths, and Frank Hellmann. SDMG: Smoothing your diffusion models for powerful graph representation learning. In *International Conference on Machine Learning*, 2025.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

A Proofs of Theorems

A.1 Proof of Theorem 4.1

Theorem 4.1 For $M \geq 1$, the degree M Bernstein response class and constant-response class are defined as $\mathcal{F}_M = \{\sum_{m=0}^{M} c_m \mathcal{S}_{m,M} \mid c_m \in \mathbb{R}\}$ and $\mathcal{C} = \{x \mapsto c \mid c \in \mathbb{R}\}$, respectively. For any target $f \in C([0,1])$, let $\mathcal{E}_M(f) = \inf_{g \in \mathcal{F}_M} \|f - g\|_\infty$ and $\mathcal{E}_{\text{const}}(f) = \inf_{g \in \mathcal{C}} \|f - g\|_\infty$ denote the optimal uniform approximation errors of the two response classes, respectively. Then the following two properties hold:

$$\text{(i)} \quad \lim_{M \to \infty} \mathcal{E}_M(f) = 0, \quad \text{(ii)} \quad \mathcal{E}_M(f) \leq \mathcal{E}_{\text{const}}(f), \quad \forall\, M \geq 1. \tag{18}$$

Proof. We prove the two stated claims in three steps and derive the strict comparison for nonconstant targets in the final step.

Step 1: Basic properties of the Bernstein basis. For any numerical value $x \in [0,1]$, the Bernstein basis functions are nonnegative. Moreover, the binomial theorem gives

$$\sum_{m=0}^{M} \mathcal{S}_{m,M}(x) = \sum_{m=0}^{M} \binom{M}{m} x^m (1-x)^{M-m} = \big(x + (1-x)\big)^M = 1. \tag{19}$$

Step 2: Uniform approximation of a continuous response. Let $f \in C([0,1])$ and choose $c_m = f(m/M)$ for $m = 0, \ldots, M$. The resulting Bernstein approximation is $B_M f(x) = \sum_{m=0}^{M} f(m/M) \mathcal{S}_{m,M}(x)$. Thus, the function $B_M f$ belongs to the response class $\mathcal{F}_M$.

For a fixed $x \in [0,1]$, an auxiliary random variable $Y_M \sim \mathrm{Binomial}(M, x)$ is introduced to express the Bernstein approximation as an expectation. By the binomial probability mass function,

$$B_M f(x) = \mathbb{E}\left[f\left(\frac{Y_M}{M}\right)\right], \quad \mathrm{Var}\left(\frac{Y_M}{M}\right) = \frac{x(1-x)}{M} \leq \frac{1}{4M}. \tag{20}$$

Fix $\varepsilon > 0$. Since f is continuous on the compact interval $[0,1]$, it is uniformly continuous and bounded. Hence, there exists $\delta > 0$ such that $|f(u) - f(v)| < \varepsilon/2$ whenever $|u - v| < \delta$. Let $C_f = \|f\|_\infty$. By Chebyshev's inequality and Eq. (20), we obtain

$$\mathbb{P}\left(\left|\frac{Y_M}{M} - x\right| \geq \delta\right) \leq \frac{1}{4M\delta^2}. \tag{21}$$

Therefore,

$$|B_M f(x) - f(x)| \leq \mathbb{E}\left[\left|f\left(\frac{Y_M}{M}\right) - f(x)\right|\right] \tag{22}$$

$$\leq \frac{\varepsilon}{2} + 2C_f \mathbb{P}\left(\left|\frac{Y_M}{M} - x\right| \geq \delta\right) \tag{23}$$

$$\leq \frac{\varepsilon}{2} + \frac{C_f}{2M\delta^2}. \tag{24}$$

The bound is independent of x. Thus, for all sufficiently large M, $\|B_M f - f\|_\infty < \varepsilon$. Since $B_M f \in \mathcal{F}_M$, it follows that

$$0 \leq \mathcal{E}_M(f) \leq \|B_M f - f\|_\infty \longrightarrow 0. \tag{25}$$

Therefore, $\lim_{M\to\infty} \mathcal{E}_M(f) = 0$.

Step 3: Comparison with constant responses. For any constant $c \in \mathbb{R}$, set $c_0 = \cdots = c_M = c$. By Eq. (19), we obtain

$$\sum_{m=0}^{M} c_m \mathcal{S}_{m,M}(x) = c \sum_{m=0}^{M} \mathcal{S}_{m,M}(x) = c. \tag{26}$$

Hence, $\mathcal{C} \subseteq \mathcal{F}_M$, and taking the infimum over the larger response class yields

$$\mathcal{E}_M(f) \leq \mathcal{E}_{\mathrm{const}}(f), \quad M \geq 1. \tag{27}$$

For completeness, suppose in addition that f is nonconstant, and let $a = \min_{x\in[0,1]} f(x)$ and $b = \max_{x\in[0,1]} f(x)$. Continuity guarantees that both extrema are attained, while nonconstancy implies $b - a > 0$. For any $c \in \mathbb{R}$,

$$\|f - c\|_\infty \geq \max\{|a - c|, |b - c|\}, \tag{28}$$

$$\max\{|a - c|, |b - c|\} \geq \frac{b-a}{2}. \tag{29}$$

Here, Eq. (29) follows from $b - a \leq |b - c| + |a - c| \leq 2\max\{|a - c|, |b - c|\}$. Setting $c = (a+b)/2$ yields $\|f - c\|_\infty = (b-a)/2$, so this lower bound is attained. Consequently,

$$\mathcal{E}_{\mathrm{const}}(f) = \frac{b-a}{2} > 0. \tag{30}$$

Since $\mathcal{E}_M(f) \to 0$, there exists M_0 such that, for any $M \geq M_0$,

$$\mathcal{E}_M(f) < \frac{b-a}{2} = \mathcal{E}_{\mathrm{const}}(f). \tag{31}$$

Thus, for a nonconstant target, the Bernstein response achieves a strict approximation advantage over constant responses when its degree is sufficiently large. □

A.2 Proof of Theorem 4.2

Theorem 4.2. Let $\zeta_i = S(\widehat{\mathbf{X}}_0^i, \bar{\mathbf{X}}_0^i)$ denote the reconstruction similarity of the i-th frequency component. Given $\mathbf{X}_0$ and t, the randomness in ζ_i arises from sampling $\mathbf{X}_t \sim q(\mathbf{X}_t \mid \mathbf{X}_0)$. For the ℓ-th spectral band, assume that $\mathrm{Var}\,(\zeta_i \mid \mathbf{X}_0, t) \le \sigma_\ell^2$ for $i \in \mathcal{B}_\ell$ and $\mathrm{Cov}\,(\zeta_i, \zeta_{i'} \mid \mathbf{X}_0, t) \le \rho_\ell \sigma_\ell^2$ for distinct $i, i' \in \mathcal{B}_\ell$, where $0 \le \rho_\ell < 1$. Then, the band-wise similarity s_ℓ satisfies

$$\mathrm{Var}(s_\ell \mid \mathbf{X}_0, t) \le \sigma_\ell^2 \left(\rho_\ell + \frac{1-\rho_\ell}{|\mathcal{B}_\ell|} \right). \tag{32}$$

Moreover, under an aggregate reconstruction error bound of $\eta \in [0, 1)$, the reconstruction similarity of each spectral band satisfies

$$s_\ell \ge (1-\eta)^{1/w_\ell}, \quad \ell = 1, \dots, k. \tag{33}$$

Proof. From the definition of the band-wise similarity $s_\ell = \frac{1}{|\mathcal{B}_\ell|} \sum_{i \in \mathcal{B}_\ell} \zeta_i$, its conditional variance can therefore be expanded as

$$\mathrm{Var}\left(s_\ell \mid \mathbf{X}_0, t\right) = \frac{1}{|\mathcal{B}_\ell|^2} \left[\sum_{i \in \mathcal{B}_\ell} \mathrm{Var}\left(\zeta_i \mid \mathbf{X}_0, t\right) + \sum_{\substack{i, i' \in \mathcal{B}_\ell \\ i \ne i'}} \mathrm{Cov}\left(\zeta_i, \zeta_{i'} \mid \mathbf{X}_0, t\right) \right] \tag{34}$$

$$\le \frac{|\mathcal{B}_\ell| \sigma_\ell^2 + |\mathcal{B}_\ell| \big(|\mathcal{B}_\ell| - 1\big) \rho_\ell \sigma_\ell^2}{|\mathcal{B}_\ell|^2} \tag{35}$$

$$= \sigma_\ell^2 \Big(\rho_\ell + \frac{1-\rho_\ell}{|\mathcal{B}_\ell|} \Big). \tag{36}$$

For $\rho_\ell < 1$, the resulting upper bound decreases as the cardinality $|\mathcal{B}_\ell|$ increases. Thus, a larger band produces a smaller upper bound on the conditional variance of its averaged similarity.

We next derive the per-band lower bound from the aggregate reconstruction constraint. Let $G = \prod_{r=1}^k s_r^{w_r}$. The aggregate reconstruction error for a given reconstruction is $1 - G$. Under the assumed error bound, we have $1 - \prod_{r=1}^k s_r^{w_r} = 1 - G \le \eta$, and hence $G \ge 1 - \eta$. Since $0 \le s_r \le 1$ and $w_r > 0$, every factor in the product is at most 1. Therefore, for an arbitrary band ℓ, we have

$$\prod_{r \ne \ell} s_r^{w_r} \le 1, \tag{37}$$

and consequently,

$$G = s_\ell^{w_\ell} \prod_{r \ne \ell} s_r^{w_r} \le s_\ell^{w_\ell}. \tag{38}$$

Combining this inequality with $G \ge 1 - \eta$ gives $s_\ell^{w_\ell} \ge 1 - \eta$. Because $w_\ell > 0$, it follows that

$$s_\ell \ge (1-\eta)^{1/w_\ell}, \quad \ell = 1, \dots, k, \tag{39}$$

which is the required per-band lower bound. Finally, this threshold is increasing in w_ℓ: for fixed $\eta \in (0, 1)$,

$$\frac{\partial}{\partial w_\ell} \log \left[(1-\eta)^{1/w_\ell} \right] = -\frac{\log(1-\eta)}{w_\ell^2} > 0. \tag{40}$$

Hence a larger weight w_ℓ imposes a larger lower bound on s_ℓ, i.e., a tighter reconstruction requirement on the corresponding band. □

Implication for quantile-based spectral partitioning. Suppose that N discrete frequency components are divided into K bands, where $1 \le k \le N$. Under a quantile-based partition, each band contains either $\lfloor N/k \rfloor$ or $\lceil N/k \rceil$ components. Hence,

$$\mathrm{Var}(s_\ell \mid \mathbf{X}_0, t) \le \sigma_\ell^2 \left(\rho_\ell + \frac{1-\rho_\ell}{|\mathcal{B}_\ell|} \right) \tag{41}$$

$$\le \sigma_\ell^2 \left(\rho_\ell + \frac{1-\rho_\ell}{\lfloor N/k \rfloor} \right). \tag{42}$$

If k divides N, then $|\mathcal{B}_\ell| = N/k$ for each band, and the bound reduces to

$$\mathrm{Var}(s_\ell \mid \mathbf{X}_0, t) \le \sigma_\ell^2 \left(\rho_\ell + \frac{k(1-\rho_\ell)}{N} \right). \tag{43}$$

In contrast, an equal-width partition may produce highly unequal or even empty bands when the frequency components are nonuniformly distributed. Among nonempty bands, a smaller cardinality yields a weaker cardinality-dependent variance guarantee. Therefore, under comparable within-band variance and correlation parameters, quantile partitioning makes the cardinality-dependent part of the variance bound more uniform across spectral bands.

B Related Work

B.1 Graph Representation Learning

Graph representation learning aims to embed graph data into a low-dimensional continuous space, thereby capturing the latent semantic information of nodes, edges, and global structures. Existing methods in this field generally fall into two broad categories: contrastive learning and generative learning.

Contrastive graph learning methods learn discriminative representations by maximizing the consistency between positive pairs while pushing apart negative pairs. Early studies mainly focus on the construction of contrastive views. For instance, DGI (Veličković et al., 2019) maximizes mutual information between local and global representations to capture graph structures. GraphCL (You et al., 2020) and GRACE (Zhu et al., 2020) generate augmented views through structural and attribute perturbations. MVGRL (Hassani & Khasahmadi, 2020) further incorporates graph diffusion views to encode multi-hop topological information. To alleviate the dependence on negative samples, subsequent works such as BGRL (Thakoor et al., 2022) align only two augmented views of the same node via an encoder-predictor architecture, while CCA-SSG (Zhang et al., 2021) aligns cross-view representations through canonical correlation analysis. More recently, GraphACL (Xiao et al., 2023) learns node representations through asymmetric contrastive learning without graph augmentations. SP-GCL (Wang et al., 2023) and MCGRL (Xie et al., 2026) capture multi-granular structural semantics from spectral and multi-channel perspectives, respectively. Despite these advances, contrastive methods typically rely on manually designed graph augmentations, making their performance highly sensitive to the choice of view construction rules.

Generative graph learning methods learn rich latent representations by modeling graph distributions and reconstructing original information. Early generative frameworks primarily build upon graph autoencoders. For example, VGAE (Kipf & Welling, 2016) reconstructs the adjacency matrix via variational inference, while DLR-GAE (Chen et al., 2023) and MaskGAE (Li et al., 2023b) enhance structural modeling from local and global topology reconstruction perspectives, respectively. With the emergence of masked modeling, GraphMAE (Hou et al., 2022) learns robust representations by randomly masking node features and then reconstructing them. Subsequent extensions, including GPT-GNN (Hu et al., 2020), MVGAE (Li et al., 2023a), and PMGAE (Luo et al., 2026), further explore joint generative modeling of both graph structures and node attributes. Despite their effectiveness, nearly all existing generative graph learning methods are constrained by single-step reconstruction objectives and fail to explicitly model the progressive evolution of graph signals.

B.2 Graph Diffusion Models

Diffusion models, as powerful probabilistic generative models, characterize complex data distributions through a two-stage pipeline that progressively perturbs data with noise and learns the reverse denoising process. Recently, this paradigm has been extended to both graph generation and graph representation learning.

For graph generation, diffusion models aim to synthesize realistic and diverse graph samples. EDP-GNN (Niu et al., 2020) and GDSS (Jo et al., 2022) treat node features and graph structures as continuous stochastic processes. DiGress (Vignac et al., 2023) adopts a diffusion process to better accommodate the discrete nature of graphs. SPECTRE (Martinkus et al., 2022) and GGSD (Minello et al., 2025) impose spectral constraints on the diffusion process to preserve topological consistency.

To address the challenge of large-scale graph generation, DMol (Niu et al., 2025) and FragFM (Lee et al., 2026) adopt staged generation and fragment-level modeling, respectively. These studies demonstrate the potential of diffusion models for graph distribution modeling. However, their primary objective remains the generation of high-quality graph samples, rather than learning discriminative representations for downstream tasks.

Beyond generation, diffusion models have recently emerged as a promising paradigm for graph representation learning, leveraging progressive noising-denoising mechanisms. DDM (Yang et al., 2023) utilizes directional information obtained from graphs to adjust noise propagation, thereby better preserving the original topology, while SDMG (Zhu et al., 2025) further introduces a low-frequency encoder and a multi-scale smoothing loss to recover low-frequency details. LapDiff (Kim et al., 2025) incorporates Laplacian priors to guide representation evolution along graph structures. DDGAE (Wesego, 2025) integrates a discrete diffusion model into an autoencoder framework for learning representations of graph-structured data. In addition, LGD (Zhou et al., 2024) and GeoMancer (Gao et al., 2025) explore unified frameworks that bridge generative and predictive tasks to improve model generalization. Despite these advances, existing methods typically rely on uniform or statically predefined noise schedules. They overlook the inherent frequency heterogeneity of graph signals and cannot dynamically adapt the diffusion process to the spectral properties of different graphs. This critical gap is exactly what our work addresses.

C Diffusion Models

Diffusion Probabilistic Models (DPMs) are a class of generative models that learn complex data distributions through a forward noising process and a reverse denoising process (Ho et al., 2020; Song et al., 2021). Given an initial sample $\mathbf{x}_0 \sim q(\mathbf{x})$, the forward diffusion process gradually perturbs data by injecting noise, which can be described by a Stochastic Differential Equation (SDE):

$$d\mathbf{x}_t = f(\mathbf{x}_t, t)dt + g(t)d\mathbf{w}_t, \tag{44}$$

where $t \in [0, T]$ denotes the diffusion time, $f(\cdot, t)$ is the drift function, $g(t)$ is the diffusion coefficient, and $\mathbf{w}_t$ represents the standard Wiener process. As t increases, the distribution of $\mathbf{x}_t$ gradually evolves toward a simple prior distribution, typically a Gaussian distribution. To recover the original data distribution, DPMs learn a reverse-time denoising process that transforms noisy samples back into clean data. According to the reverse-time SDE, the denoising process is formulated as:

$$d\mathbf{x}_t = \left[f(\mathbf{x}_t, t) - g^2(t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right] dt + g(t)d\bar{\mathbf{w}}_t, \tag{45}$$

where $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ denotes the score function, and $\bar{\mathbf{w}}_t$ is the reverse-time Wiener process. Since the score function is generally intractable, it is approximated using a neural network $s_\theta(\mathbf{x}_t, t)$, which is optimized through Denoising Score Matching (DSM):

$$\mathcal{L}_{DSM} = \mathbb{E}_t \left\{ \delta(t)\, \mathbb{E}_{\mathbf{x}_0} \mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0} \left[\| s_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0) \|^2 \right] \right\}, \tag{46}$$

where $\delta(t)$ is a weighting function controlling the contribution of different diffusion steps. In practice, instead of directly estimating the score, diffusion models often adopt alternative parameterizations, such as a data prediction network ρ_θ, i.e., $\rho_\theta(\mathbf{x}_t, t) = (\mathbf{x}_t - \epsilon_t \varepsilon_\theta(\mathbf{x}_t, t))/\beta_t$, where $\varepsilon_\theta(\mathbf{x}_t, t) = -\epsilon_t s_\theta(\mathbf{x}_t, t)$, β_t and ϵ_t control the signal and noise scales, respectively. The score matching objective can be equivalently formulated as a reconstruction loss:

$$\mathcal{L}_{DSM} = \mathbb{E}_t \left\{ (\delta(t)\beta_t^2/\epsilon_t^4) \mathbb{E}_{\mathbf{x}_0} \mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0} \left[\| \rho_\theta(\mathbf{x}_t, t) - \mathbf{x}_0 \|^2 \right] \right\}. \tag{47}$$

Recently, diffusion models have been explored in graph representation learning. Different from conventional Euclidean data, graph representations contain structural dependencies among nodes. Specifically, given node features $\mathbf{X}$, the diffusion model progressively perturbs these representations and learns to recover their semantic information. The intermediate representations extracted from the denoising process are denoted by: $\mathbf{H} = h_\theta(\mathbf{X}_t, t)$, where $\mathbf{H}$ denotes the learned graph representations, which can be further utilized for downstream tasks.

D Details of the Conditional Denoising Network

U-Net Architecture. The denoising network $\rho_\theta(\cdot)$ adopts a U-Net architecture (Dhariwal & Nichol, 2021; Li et al., 2023c) to progressively recover representations from noisy inputs. Specif-

Table 3: Statistics and hyperparameters for node classification datasets.

Dataset		Cora	Citeseer	PubMed	Computer	Photo
Statistics	# Nodes	2,708	3,327	19,717	13,752	7,650
	# Edges	5,429	4,732	44,338	245,861	119,081
	# Classes	7	6	3	10	8
Hyperparameters	num_hidden	512	512	1024	512	512
	num_layers	7	2	7	7	7
	bands	3	5	5	5	3
	learning_rate	$4e^{-4}$	$1e^{-4}$	$2e^{-4}$	$4e^{-4}$	$4e^{-4}$
	num_hop	2	2	7	8	7
	beta_schedule	Sigmoid	Sigmoid	Linear	Const	Const

ically, the encoder extracts hierarchical features through downsampling layers $\{L_z^{down}\}_{z=0}^n$, capturing high-level semantics from corrupted graph signals. The decoder then reconstructs the representations through corresponding upsampling layers $\{L_z^{up}\}_{z=0}^n$, while skip connections integrate intermediate features from the encoder and decoder stages. This multi-scale feature fusion facilitates the recovery of both global structures and local patterns. After successive upsampling operations, the decoder outputs the hidden representation $\mathbf{H} = \text{Concat}\left(\{\mathbf{H}_z^{up}\}_{z=0}^n\right)$, which serves as the final representation for subsequent optimization.

Graph-Aware Conditional Information. To provide additional guidance during the denoising process, SPREAD incorporates both feature-level and structure-level information (Zhu et al., 2025). Specifically, we obtain $\mathbf{C}_X$ using a Graph Attention Network (GAT), which aggregates neighboring node features with adaptive attention weights. $\mathbf{C}_A$ is obtained by applying a multi-layer perceptron (MLP) to random-walk positional encodings. By jointly incorporating $\mathbf{C}_X$ and $\mathbf{C}_A$ as conditional inputs, SPREAD can capture both contextual semantic information and topological dependencies throughout the denoising process.

E Experimental Setup

E.1 Datasets

We evaluate SPREAD on five datasets for node classification and five datasets for graph classification. The benchmark datasets for node classification consist of citation and co-purchase networks, where each dataset is represented as a single attributed graph. The benchmark datasets for graph classification contain collections of graphs from social-network and bioinformatics domains.

Node Classification. Cora, Citeseer, and PubMed are citation networks, where nodes represent scientific publications, edges denote citation relationships, and labels correspond to research topics. Computer and Photo are co-purchase networks, where nodes represent products, edges indicate frequently co-purchased products, and labels denote product categories.

Graph Classification. MUTAG consists of molecular graphs, where nodes and edges represent atoms and chemical bonds, respectively, and labels indicate mutagenic activity. IMDB-B and IMDB-M are social network datasets constructed from movie co-appearance relationships, with graph labels corresponding to movie genres. PROTEINS consists of protein structure graphs, with secondary structure elements as nodes and labels indicating whether a protein is an enzyme. COLLAB consists of scientific collaboration networks, where graphs represent researcher collaboration patterns.

Detailed dataset statistics and the final hyperparameter configurations for node classification and graph classification are reported in Table 3 and Table 4, respectively.

E.2 Settings

For node classification, we use classification accuracy as the evaluation metric and report the mean and standard deviation across five random seeds. For Cora, Citeseer, and PubMed, we use the standard public splits, with 20 labeled nodes per class for training, 500 nodes for validation, and 1,000

Table 4: Statistics and hyperparameters for graph classification datasets.

Dataset		MUTAG	IMDB-B	IMDB-M	PROTEINS	COLLAB
Statistics	# Graphs	188	1,000	1,500	1,113	5,000
	Avg. # Nodes	17.93	19.77	13.00	39.06	74.49
	# Classes	2	2	3	2	3
Hyperparameters	num_hidden	128	128	256	512	512
	num_head	4	4	2	4	4
	bands	5	6	6	6	6
	learning_rate	$3e^{-4}$	$2e^{-5}$	$1e^{-5}$	$1e^{-5}$	$1e^{-5}$
	num_hop	8	8	8	8	8
	attn_drop	0.1	0.1	0.2	0.2	0.4
	feat_drop	0.2	0.2	0.2	0.2	0.4
	beta_schedule	Sigmoid	Sigmoid	Linear	Sigmoid	Sigmoid

Table 5: Comparison of accuracy (%) for graph classification. Bold: best, underlined: second-best. "–" indicates that the data is not reported in the original paper.

Category	Method	IMDB-B	IMDB-M	PROTEINS	COLLAB	MUTAG
Supervised	GIN	75.18±4.42	<u>52.13±2.31</u>	**76.15±2.30**	80.26±1.57	89.20±5.24
	DiffPool	72.68±3.23	–	75.02±3.08	79.11±2.12	84.90±7.21
Self-supervised	GraphCL	71.10±0.49	48.43±0.47	74.46±0.43	71.57±1.16	86.74±1.15
	JOAO	70.12±2.07	49.26±0.64	74.62±0.41	69.47±0.39	87.14±1.25
	GCC	72.0	49.4	–	78.9	–
	MVGRL	74.12±0.51	51.28±0.43	–	–	89.40±1.05
	GraphMAE	75.58±0.63	51.57±0.40	75.25±0.35	80.26±0.37	87.96±1.21
	SimGRACE	71.38±0.61	–	75.20±0.14	71.41±0.85	89.10±1.24
Diffusion	DDM	75.08±0.19	51.23±0.20	71.86±0.38	80.96±0.23	90.37±0.36
	SDMG	<u>75.94±0.42</u>	52.02±0.32	72.98±0.18	**82.10±0.21**	<u>90.93±0.24</u>
Proposed	SPREAD	**76.50**	**53.00**	73.32	<u>81.32</u>	**91.49**

nodes for testing. For Computer and Photo, we randomly split the nodes into training, validation, and test sets with proportions of 10%, 10%, and 80%, respectively. For graph classification, we pretrain SPREAD on all graphs without using labels and evaluate the learned representations using an SVM with stratified 10-fold cross-validation. Each split uses nine folds for training and one for testing, with the SVM regularization parameter selected by five-fold cross-validation within the training set. We use one pretraining seed per dataset and report the mean accuracy across the ten test folds. All experiments are implemented in PyTorch and PyTorch Geometric and conducted on a Linux server equipped with 124 GB of RAM and two NVIDIA GeForce RTX 4090 GPUs.

F Graph Classification

F.1 Baselines

For graph classification, we compare SPREAD with representative graph representation learning methods from three categories. *Supervised Methods* include GIN (Xu et al., 2019) and DiffPool (Ying et al., 2018). *Self-Supervised Methods* include GraphCL (You et al., 2020), JOAO (You et al., 2021), GCC (Qiu et al., 2020), MVGRL (Hassani & Khasahmadi, 2020), GraphMAE (Hou et al., 2022) and SimGRACE (Xia et al., 2022), covering contrastive and generative approaches to self-supervised learning. *Diffusion-Based Methods* include DDM (Yang et al., 2023) and SDMG (Zhu et al., 2025).

F.2 Results Analysis

We evaluate SPREAD on five benchmark datasets for graph classification, with results reported in Table 5. SPREAD achieves the best performance on the IMDB-B, IMDB-M, and MUTAG datasets,

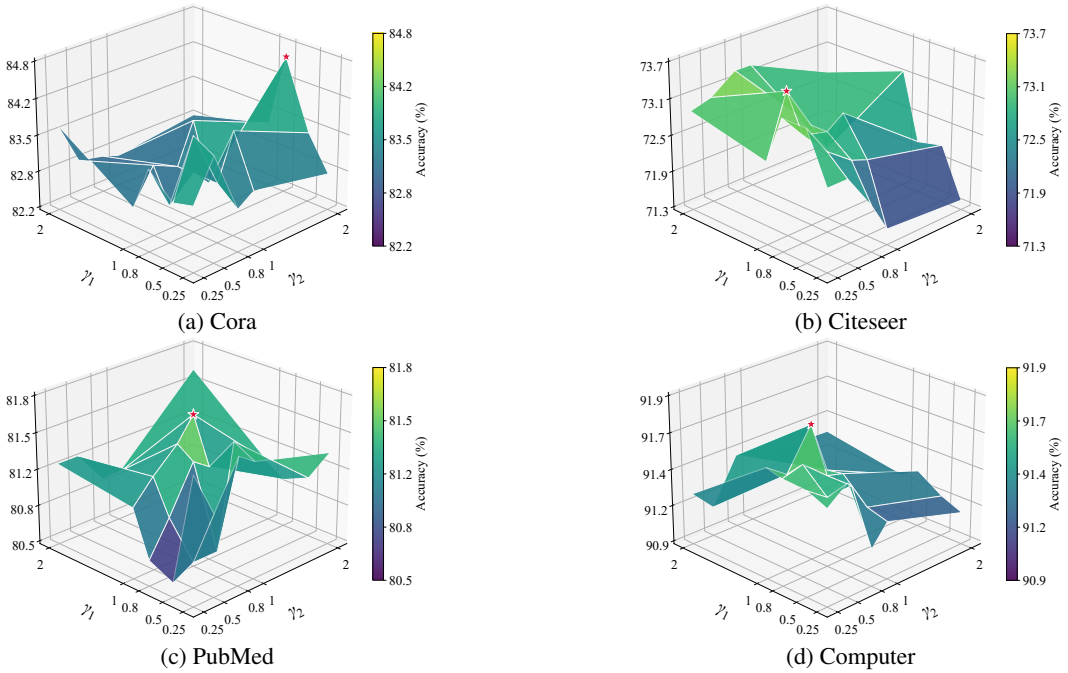
Table 6: Ablation studies. Bold: best.

Method	IMDB-B	IMDB-M	PROTEINS	COLLAB	MUTAG
w/o SRP	75.90	52.40	72.24	80.42	90.94
w/o Weight	75.80	52.27	69.81	81.18	86.20
w/o Spe	74.90	52.20	72.50	81.14	86.67
w/o Spa	74.60	51.93	72.74	79.80	90.91
SPREAD	**76.50**	**53.00**	**73.32**	**81.32**	**91.49**

exceeding the strongest baselines by 0.56, 0.87, and 0.56 percentage points, respectively. Compared with diffusion-based methods, SPREAD outperforms DDM on all five datasets and SDMG on four. These results suggest that jointly adapting forward perturbation and reverse reconstruction to spectral characteristics benefits graph-level representation learning on both social and molecular graphs. On the PROTEINS dataset, SPREAD outperforms both diffusion-based baselines but trails GIN and several self-supervised methods. PROTEINS contains protein structure graphs labeled as enzymes or non-enzymes, with substantial within-class variation in size and connectivity. When reconstruction emphasizes within-class structural variation rather than differences between classes, more accurate recovery does not necessarily translate into better classification performance. Furthermore, the reported ablation results in Table 6 show that the full model consistently outperforms its ablated variants, supporting the contribution of each component.

G Parameter Analysis

Figure 6 shows the sensitivity of SPREAD to the trade-off coefficients γ_1 and γ_2 on the Cora, Citeseer, PubMed, and Computer datasets, complementing the results on the Photo dataset in the main text. These coefficients control the contributions of the spatial reconstruction objective $\mathcal{L}_{spa}$ and the frequency-aware spectral reconstruction objective $\mathcal{L}_{spe}$, respectively. The limited variation in node classification accuracy across the tested coefficient combinations on each dataset suggests that SPREAD maintains relatively stable performance under the settings of γ_1 and γ_2.



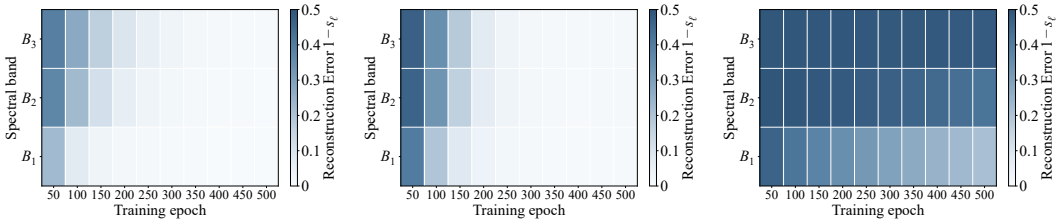
(a) Cora (b) Citeseer

(c) PubMed (d) Computer

Figure 6: Sensitivity of SPREAD to the trade-off coefficients γ_1 and γ_2 on (a) Cora, (b) Citeseer, (c) PubMed, and (d) Computer.

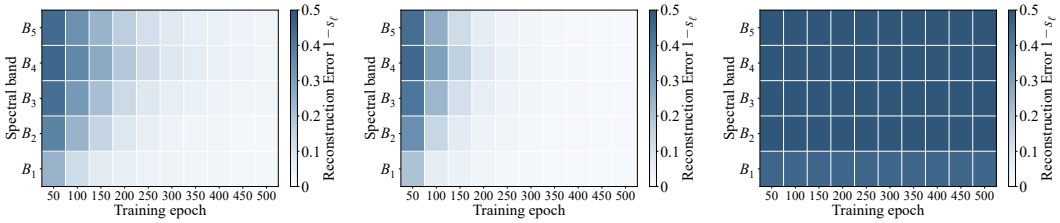
H VISUALIZATION

Figures 7–10 compare the band-wise reconstruction errors $1-s_\ell$ of DDM, SDMG, and SPREAD on the Cora, Citeseer, Computer, and Photo datasets. On the Computer and Photo datasets, SPREAD achieves broader error reductions across the spectrum, particularly in $\mathcal{B}_1$. On the Cora and Citeseer datasets, the improvements are more dataset-dependent, with SDMG remaining competitive in several bands. These results further suggest that response-guided weighting changes the frequency-specific reconstruction process.



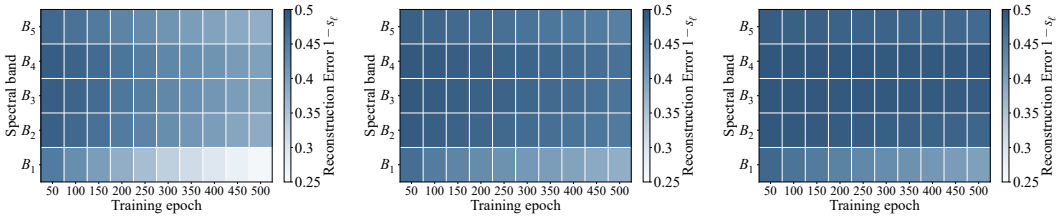
(a) SPREAD (b) SDMG (c) DDM

Figure 7: Band-wise reconstruction errors $1-s_\ell$ on the Cora dataset. All panels use the same color scale, with lighter colors indicating lower errors.



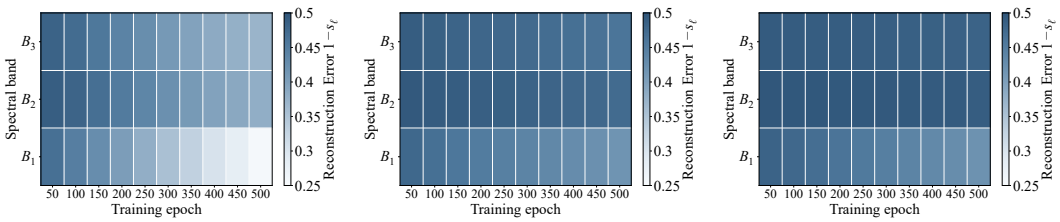
(a) SPREAD (b) SDMG (c) DDM

Figure 8: Band-wise reconstruction errors $1-s_\ell$ on the Citeseer dataset. All panels use the same color scale, with lighter colors indicating lower errors.



(a) SPREAD (b) SDMG (c) DDM

Figure 9: Band-wise reconstruction errors $1-s_\ell$ on the Computer dataset. All panels use the same color scale, with lighter colors indicating lower errors.



(a) SPREAD (b) SDMG (c) DDM

Figure 10: Band-wise reconstruction errors $1-s_\ell$ on the Photo dataset. All panels use the same color scale, with lighter colors indicating lower errors.