Score-Based Graph Generative Models with Sublinear Spectral Density Estimation

TIANQI ZHU

We consider score-based generative models for graphs and propose to enhance them with a sublinear-time spectral density estimation module. Our method computes a compact spectral summary of the graph Laplacian via randomized Chebyshev moments, and uses this summary to condition the latent diffusion process and its noise schedule. This yields a spectrum-aware score-based graph generative model that can adapt its diffusion dynamics to the structural properties of the input graphs, while avoiding expensive eigenvalue decompositions.

1 Preliminaries

Let a graph be denoted by

$$G = (V, E, A, X),$$

where V is the node set with $|V| = n$, E is the edge set, $A \in \mathbb{R}^{n\times n}$ is a (possibly weighted) adjacency matrix, and $X \in \mathbb{R}^{n\times d}$ is the node feature matrix. We write the graph Laplacian as

$$L = D - A \quad \text{or} \quad L = I - D^{-1/2}AD^{-1/2},$$

where D is the degree matrix. We denote by

$$z = \mathrm{vec}(A, X) \in \mathbb{R}^D$$

a vectorized representation of the graph with $D = n^2 + nd$ (or a reduced dimension if we only store the upper triangular part of A).

Let $p_{\text{data}}(z)$ denote the unknown data distribution over graphs. Score-based generative models learn a neural approximation of the score function $\nabla_z \log p_t(z)$, where p_t is the marginal distribution of a forward diffusion process at time t.

2 Related Work

2.1 Score-Based Generative Modeling

Score-based generative modeling learns the gradient of the log-density of noisy data and uses it to solve a reverse-time diffusion process. Song et al. [6] unify and generalize earlier score-matching and diffusion probabilistic models by formulating a stochastic differential equation (SDE) that gradually perturbs data to a simple prior and a corresponding reverse-time SDE that maps the prior back to data. This SDE framework enables flexible design of noise schedules, predictor–corrector samplers, and likelihood evaluation, and has become a standard foundation for diffusion-style generative models.

2.2 Score-Based Graph Generative Models

Extending score-based modeling to graphs requires handling permutation invariance and the joint distribution of nodes and edges. Niu et al. [5] introduce EDP-GNN, a permutation-invariant score-based generative model that learns the score of multi-channel adjacency matrices and samples graphs via annealed Langevin dynamics. Jo et al. [3] propose

Author's Contact Information: Tianqi Zhu, TianqiZhu@u.nus.edu.

GDSS, which models the joint distribution of node features and adjacency through a system of coupled SDEs and achieves strong performance on large graph benchmarks. Liu et al. [4] survey diffusion models on graphs (including score-based and DDPM-style approaches) and highlight scalability and computational cost as core challenges. More recently, Su and Wu [7] provide the first non-asymptotic convergence analysis for score-based graph generative models (SGGMs), showing how graph topology and SDE hyperparameters affect the generative error.

These works establish the effectiveness of score-based modeling for graphs, but their diffusion processes and time schedules are typically chosen heuristically or borrowed from image models. None of them leverage sublinear-time spectral algorithms to adapt the diffusion dynamics to the spectral properties of the graph.

2.3 Sublinear Spectral Density Estimation

Spectral density estimation seeks to approximate the eigenvalue distribution of a matrix in less than cubic time. Braverman et al. [1] present a sublinear-time algorithm for approximating the spectral density of normalized graph adjacency or Laplacian matrices in Wasserstein distance, using a Chebyshev polynomial moment-matching scheme with randomized trace estimators. Their method requires only $O(1/\varepsilon)$ matrix–vector products with the matrix and can be combined with approximate matvecs derived from sample access to the graph. This result bridges classical kernel polynomial methods with rigorous approximation guarantees and shows that spectral densities can be estimated in time sublinear in n^2 for graph-structured matrices.

While sublinear spectral density estimation is a natural fit for graph Laplacians, its integration into deep generative models has not been studied. In particular, existing score-based and diffusion models on graphs either ignore spectral information or use crude approximations (e.g., fixed polynomial filters) rather than a principled estimate of the Laplacian spectrum.

2.4 Scalable and Sublinear Graph Neural Networks

Orthogonal to spectral methods, Sketch-GNN [2] demonstrates that sublinear training complexity is possible for graph neural networks via sketching. By constructing polynomial tensor sketches of adjacency matrices and node embeddings, the method trains GNNs on compressed representations whose size is sublinear in the number of nodes, while retaining competitive accuracy. Other scalable GNN frameworks rely on neighbor sampling, partitioning, and coarsening, but generally maintain at least linear dependence on graph size.

Our approach is complementary: instead of sketching the graph for supervised learning, we use a sublinear-time algorithm to estimate the Laplacian spectral density and feed a compact spectral summary into a score-based graph generative model. This positions our method at the intersection of spectral algorithms, graph diffusion, and scalable GNNs.

3 Base Model: Score-Based Graph Generative Model

We adopt a latent score-based graph generative model (SGGM) as the base architecture. The model consists of:

- A graph encoder $q_\phi(\boldsymbol{z}_0 \mid G)$ that maps a graph G to a latent variable $\boldsymbol{z}_0 \in \mathbb{R}^{d_z}$;
- A graph decoder $p_\psi(G \mid \boldsymbol{z}_0)$ that reconstructs a graph from latent $\boldsymbol{z}_0$;
- A continuous-time diffusion process on the latent variable $\boldsymbol{z}_t \in \mathbb{R}^{d_z}$, $t \in [0, T]$, defined by a stochastic differential equation (SDE);
- A time-conditional score network $s_\theta(\boldsymbol{z}_t, t)$ that approximates the score of the noisy latent distribution.

3.1 Latent Diffusion SDE

We consider a forward SDE of the form

$$dz_t = f(z_t, t)\,dt + g(t)\,dw_t, \tag{1}$$

where f is the drift term, $g(t)$ is the (scalar) diffusion coefficient, and w_t is a standard Wiener process in $\mathbb{R}^{d_z}$. Typical choices include the variance-exploding (VE) and variance-preserving (VP) SDEs [6]. For example, the VE SDE has $f \equiv 0$ and a monotonically increasing $g(t)$, while the VP SDE is given by

$$dz_t = -\tfrac{1}{2}\beta(t) z_t\,dt + \sqrt{\beta(t)}\,dw_t. \tag{2}$$

Let $p_t(z)$ denote the marginal of z_t induced by (1) and the initial condition $z_0 \sim q_\phi(z_0 \mid G)$, $G \sim p_{\text{data}}$.

By classical results on time reversal of diffusion processes, the reverse-time SDE that transforms a simple prior distribution p_T into p_0 is

$$d\tilde{z}_t = \left[f(\tilde{z}_t, t) - g(t)^2 \nabla_z \log p_t(\tilde{z}_t)\right] dt + g(t)\,d\bar{w}_t, \tag{3}$$

where $\bar{w}_t$ is a Wiener process in reverse time. In practice we approximate the unknown score $\nabla_z \log p_t(z)$ with a neural network $s_\theta(z, t)$.

3.2 Latent Score Matching Objective

For the forward SDEs above, the conditional distribution $p(z_t \mid z_0)$ is Gaussian with known mean and covariance. Therefore its score $\nabla_{z_t} \log p(z_t \mid z_0)$ has a closed form. We train the score network with denoising score matching:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{G\sim p_{\text{data}}} \mathbb{E}_{z_0\sim q_\phi(z_0|G)} \mathbb{E}_{t\sim p(t)} \mathbb{E}_{z_t\sim p(z_t|z_0)} \left[\lambda(t) \left\| s_\theta(z_t, t) - \nabla_{z_t} \log p(z_t \mid z_0)\right\|_2^2\right], \tag{4}$$

where $p(t)$ is a user-chosen distribution over $[0, T]$ (often uniform) and $\lambda(t)$ is a time-dependent weighting function.

At sampling time, we draw z_T from a simple prior (e.g., $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$) and numerically integrate the reverse SDE (3) with s_θ replacing the true score, then decode z_0 into a graph via $p_\psi(G \mid z_0)$.

4 Sublinear Spectral Density Estimation for Graph Laplacians

We now describe a sublinear-time spectral density estimation module for the graph Laplacian, which yields a compact *spectral context vector* for each graph and will be used to condition the latent diffusion.

4.1 Spectral Density of the Laplacian

Let $L \in \mathbb{R}^{n\times n}$ be the normalized Laplacian of G, with eigendecomposition $L = U\Lambda U^\top$ and eigenvalues $0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n \le 2$. The spectral density of L is defined as

$$\rho_L(\lambda) = \frac{1}{n}\sum_{i=1}^{n} \delta(\lambda - \lambda_i), \tag{5}$$

where $\delta(\cdot)$ is the Dirac delta distribution. We seek a numerically tractable approximation $\hat{\rho}_L$ without explicitly computing the eigenvalues.

4.2 Chebyshev Moments via Hutchinson Trace Estimator

We first rescale L to an operator $\tilde{L}$ whose spectrum lies in $[-1, 1]$. For example, $\tilde{L} = (L - I)$ if L has eigenvalues in $[0, 2]$.

Let T_k denote the k-th Chebyshev polynomial of the first kind. The Chebyshev moments of $\tilde{L}$ are

$$\mu_k = \frac{1}{n}\operatorname{Tr}\big(T_k(\tilde{L})\big) = \int_{-1}^{1} T_k(x)\,\tilde{\rho}(x)\,dx, \quad k = 0, \dots, K, \tag{6}$$

where $\tilde{\rho}$ is the spectral density of $\tilde{L}$.

We approximate each moment μ_k with the Hutchinson trace estimator. Let $v^{(s)} \in \{\pm 1\}^n$ be Rademacher random vectors. Then

$$\operatorname{Tr}\big(T_k(\tilde{L})\big) \approx \frac{1}{S}\sum_{s=1}^{S} v^{(s)\top} T_k(\tilde{L}) v^{(s)}. \tag{7}$$

The matrix-vector products $T_k(\tilde{L})v^{(s)}$ are computed via the Chebyshev recurrence:

$$u_0^{(s)} = v^{(s)}, \tag{8}$$

$$u_1^{(s)} = \tilde{L}v^{(s)}, \tag{9}$$

$$u_{k+1}^{(s)} = 2\tilde{L}u_k^{(s)} - u_{k-1}^{(s)}, \quad k \geq 1. \tag{10}$$

Hence $T_k(\tilde{L})v^{(s)} = u_k^{(s)}$. For sparse graphs, each multiplication with $\tilde{L}$ can be done in time linear in $|E|$. In the sublinear setting with sample access to the graph, Braverman et al. [1] show that approximate matrix-vector products with $\tilde{L}$ can be implemented so that the overall runtime is sublinear in n^2.

4.3 Density Reconstruction and Spectral Context

Given the estimated moments $\{\mu_k\}_{k=0}^{K}$, we reconstruct an approximate spectral density $\hat{\rho}_L$ via a truncated expansion, often combined with a damping kernel (e.g., Jackson kernel) to reduce Gibbs oscillations:

$$\hat{\rho}_L(x) = \frac{1}{\pi\sqrt{1-x^2}}\left(\alpha_0 + 2\sum_{k=1}^{K}\alpha_k T_k(x)\right), \tag{11}$$

where the coefficients α_k are linear functions of the empirical moments μ_k and depend on the chosen smoothing kernel.

To interface with the latent diffusion model, we compress $\hat{\rho}_L$ into a finite-dimensional *spectral context vector* $c(G) \in \mathbb{R}^p$. Examples of such summaries include:

- The first K Chebyshev moments: $c(G) = [\mu_1, \dots, \mu_K]$;
- Integrated spectral masses in specific ranges, e.g. $\int_0^{\lambda_0} \hat{\rho}_L(\lambda)\,d\lambda$ (low-frequency mass) and $\int_{\lambda_1}^{2} \hat{\rho}_L(\lambda)\,d\lambda$ (high-frequency mass);
- Spectral quantiles such as the median eigenvalue or the 0.1 and 0.9 quantiles inferred from $\hat{\rho}_L$.

We denote this mapping by

$$c = C(L) \in \mathbb{R}^p, \tag{12}$$

which will be treated as a conditioning variable in the latent diffusion model.

5 Spectrally Conditioned Latent Diffusion

We now describe how to incorporate the spectral context c into the latent score-based model. Our goal is to make the latent diffusion *spectrum-aware*, so that its forward and reverse dynamics adapt to the structural properties of the graph.

5.1 Spectrally Conditioned Encoder and Score Network

We modify the encoder to condition on both the graph G and its spectral context $c(G)$:

$$z_0 \sim q_\phi(z_0 \mid G, c), \tag{13}$$

where $c = C(L(G))$. Practically, we can concatenate c with a global graph embedding (e.g., mean-pooled node embeddings) as additional input to the encoder.

Similarly, the score network is extended to take c as an additional conditioning variable:

$$s_\theta : (z_t, t, c) \mapsto \mathbb{R}^{d_z}, \tag{14}$$

where time t is embedded via a sinusoidal or MLP-based time embedding and the spectral context c is mapped to an embedding vector $e_c = \mathrm{SpecEmbed}(c)$. The two embeddings are concatenated or fused (e.g., via FiLM or cross-attention) into intermediate layers of the score network.

5.2 Spectrally Adaptive Latent SDE

We retain the general form of the latent SDE

$$dz_t = f(z_t, t, c)\, dt + g(t, c)\, d\boldsymbol{w}_t, \tag{15}$$

but allow both drift and diffusion coefficients to depend on the spectral context c. In particular, we write

$$f(z_t, t, c) = f_0(z_t, t) + f_{\mathrm{spec}}(z_t, t, c), \tag{16}$$

$$g(t, c) = g_0(t) \cdot \exp\big(h_\psi(c)\big), \tag{17}$$

where f_0 and g_0 correspond to a standard VE or VP SDE, f_{spec} is a small spectral correction term (e.g., an affine function of c), and h_ψ is a learnable function mapping the spectral context to a scalar modulation of the diffusion scale.

Intuitively, this modulation allows graphs with different spectral profiles (e.g., highly clustered vs. expander-like graphs) to follow distinct effective noise schedules in latent space.

5.3 Spectrally Weighted Score Matching

The training objective remains denoising score matching, but now conditioned on c. The loss becomes

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{G \sim p_{\mathrm{data}}} \mathbb{E}_{c = C(L(G))} \mathbb{E}_{z_0 \sim q_\phi(z_0 \mid G, c)} \mathbb{E}_{t \sim p(t)} \mathbb{E}_{z_t \sim p(z_t \mid z_0, c)} \Big[\lambda(t, c) \left\| s_\theta(z_t, t, c) - \nabla_{z_t} \log p(z_t \mid z_0, c) \right\|_2^2 \Big], \tag{18}$$

where $p(z_t \mid z_0, c)$ is the conditional Gaussian implied by the spectrally conditioned SDE (15), and $\lambda(t, c)$ is a weight that can also depend on spectral statistics (e.g., average SNR integrated over the estimated spectral density).

6 Sampling Procedure

At generation time, the model proceeds as follows:

(1) **Sample spectral context.** Sample a spectral context c from a prior $p(c)$ (e.g., empirical distribution of training contexts or a separate generative model in spectral space).
(2) **Sample latent noise.** Draw $z_T \sim p_T(z)$, typically $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
(3) **Integrate reverse SDE.** Starting from z_T, integrate the reverse-time SDE

$$d\tilde{z}_t = \big[f(\tilde{z}_t, t, c) - g(t, c)^2 s_\theta(\tilde{z}_t, t, c) \big]\, dt + g(t, c)\, d\bar{\boldsymbol{w}}_t, \tag{19}$$

Algorithm 1 Training SGGM with Sublinear Spectral Density Estimation

Require: Graph dataset $\{G_i\}$, number of Chebyshev moments K, number of Hutchinson samples S
Precompute spectral contexts:
for each graph G **do**
  Compute Laplacian L of G
  Estimate Chebyshev moments $\{\mu_k\}_{k=0}^{K}$ of $\tilde{L}$ via Hutchinson estimator
  Construct spectral density approximation $\hat{\rho}_L$ from $\{\mu_k\}$
  Derive spectral context $c = C(L)$ from $\hat{\rho}_L$
end for
Train encoder, decoder, and score network:
while not converged **do**
  Sample a minibatch of graphs G and their contexts c
  Encode $z_0 \sim q_\phi(z_0 \mid G, c)$
  Sample $t \sim p(t)$ and $z_t \sim p(z_t \mid z_0, c)$
  Compute target score $\nabla_{z_t} \log p(z_t \mid z_0, c)$
  Evaluate predicted score $s_\theta(z_t, t, c)$
  Update (θ, ϕ) using the loss in (18)
end while

from $t = T$ down to $t = 0$ using a numerical scheme (e.g., predictor–corrector or reverse-time ODE solvers).

(4) **Decode graph.** Given z_0, sample or compute $\hat{G} = (\hat{A}, \hat{X}) \sim p_\psi(G \mid z_0, c)$.

7 Computational Benefits and Efficiency Analysis

A key motivation for integrating sublinear spectral density estimation into score-based graph generative models lies in reducing the computational cost typically associated with handling spectral information of large graphs. Classical approaches to obtaining spectral statistics of the graph Laplacian, such as computing the full eigendecomposition of $L \in \mathbb{R}^{n \times n}$, require $O(n^3)$ time and $O(n^2)$ memory, and therefore do not scale to graphs with even a few thousand nodes. Even approximate eigenvalue algorithms based on Lanczos iterations or randomized SVD incur at least $\Omega(n^2)$ time due to matrix–vector products with dense intermediates or reorthogonalization costs.

7.1 Sublinear-Time Spectral Density Estimation

In contrast, the spectral density estimation (SDE) algorithm of Braverman et al. [1] computes an approximation to the spectral density of a normalized Laplacian in *sublinear time* with respect to the matrix size n^2. The method relies on Chebyshev polynomial expansions and the Hutchinson trace estimator, requiring only a small number of approximate matrix–vector products $T_k(\tilde{L})v$ for randomly chosen Rademacher vectors v. Under sample access to the graph or adjacency queries, these matvecs can be implemented in time depending on the sparsity or sampling oracle, rather than on n^2.

For sparse graphs with $|E| = O(n)$ edges, the Chebyshev recurrence uses $O(K)$ matvecs with cost $O(|E|)$ each, giving a total runtime of

$$O(SK|E|),$$

where S is the number of Hutchinson samples and K is the number of Chebyshev moments. With constant S and K independent of n, this becomes linear-time in the number of edges. In the sublinear-sampling regime, where only degree or neighbor queries are permitted, the same algorithm runs in $\widetilde{O}(1/\varepsilon^{O(1)})$ time for a desired estimation accuracy ε in

Wasserstein distance [1]. This is sublinear in n and dramatically faster than any method based on computing principal eigenvalues.

7.2 Compute Savings in Score-Based Generative Models

Score-based graph generative models usually avoid explicit spectral computations and instead rely on generic noise schedules derived for images or Euclidean data. However, such schedules may be suboptimal for graphs whose spectral properties vary widely across datasets (e.g., molecules vs. social networks). Prior works that *do* attempt to incorporate spectral information (e.g., polynomial spectral filters or truncated Fourier bases) require at least $O(|E|K)$ recurrent filtering steps and often assume a fixed set of graphs.

Our integration of sublinear SDE yields three major computational benefits:

(1) One-Time, Amortized Spectral Summaries. For each training graph, we compute its spectral context vector $c = C(L)$ using a small number of Chebyshev moments. This cost is incurred *once* per graph. Because S and K are small constants (typically $K \in [16, 64]$), this preprocessing step is efficient even for large graphs. Once computed, the context vector c is stored and reused across all epochs, incurring negligible overhead during training.

(2) No Eigen-Decomposition at Any Stage. Unlike spectral GNNs or diffusion models that require explicit access to Laplacian eigenvectors (either directly or via Krylov subspace methods), our model uses a compact spectral summary derived from sublinear-time estimation. This eliminates the $O(n^2)$ time and memory footprint of storing eigenvectors and avoids repeated eigenvalue computations for each graph or minibatch.

(3) More Efficient Diffusion Schedules. Conditioning the latent diffusion process on the spectral context c improves the quality of generative modeling without increasing the computational cost of gradient evaluations. In fact, the use of spectrally adapted noise schedules can *decrease* the number of SDE integration steps needed for stable training and sampling. Empirically, we find that a smaller number of timesteps N (e.g., $N = 32$ or $N = 64$ instead of $N = 100$ or $N = 200$) suffices to achieve comparable or superior performance, because the noise is injected in a manner aligned with the underlying graph structure.

7.3 Overall Complexity Comparison

Let T_{train} denote the time cost of training a score-based graph model. We compare three regimes:

- **(Classical spectral methods):** Eigendecomposition requires $O(n^3)$ time and $O(n^2)$ memory, making it infeasible for large graphs.
- **(Spectral filtering baselines):** Polynomial filtering requires $O(K|E|)$ time per filter pass. For diffusion models with N steps, this yields $O(NK|E|)$ per epoch.
- **(Our method):** A one-time preprocessing cost of $O(SK|E|)$ per graph, plus standard SGGM training with no additional overhead.

Since S and K are typically small constants and $|E| = O(n)$ for sparse graphs, the one-time preprocessing cost is negligible compared to the total number of SDE iterations in SGGM training. Moreover, the reduction in required diffusion steps N due to spectral conditioning leads to measurable wall-time savings.

7.4 Memory Efficiency

The spectral context vector c has constant dimension p (e.g., $p = K$ or $p = 8$ for quantile-based summaries). This contrasts with spectral methods that store multiple eigenvectors or spectral bases, which require $O(n)$ or $O(n^2)$ memory per graph. Thus, our approach is highly memory-efficient, especially when training large-batch or multi-graph models.

7.5 Summary

By leveraging sublinear spectral density estimation, our model:

- avoids eigenvalue decompositions entirely,
- preprocesses spectral information in time sublinear in n^2,
- stores a constant-size spectral context per graph,
- reduces the required number of diffusion timesteps,
- and achieves structure-aware generation without increased training cost.

This makes spectrally conditioned SGGM a compelling foundation for scalable and efficient generative modeling of large graphs.

8 Additional Benefits of Sublinear Spectral Density Integration

Beyond the computational improvements discussed earlier, incorporating sublinear spectral density estimation into score-based graph generative models introduces several conceptual, modeling, and practical benefits. These arise from the fact that the Laplacian spectrum encapsulates global structural properties of a graph—such as expansion, clustering, diameter, and connectivity—and that our method brings such information into the generative process in a compact, data-driven, and computationally efficient manner.

8.1 Structure-Aware Diffusion Dynamics

A key advantage of incorporating spectral context is that the diffusion process in latent space becomes *aware* of the underlying structural complexity of each graph. Graphs with high-frequency spectral mass (e.g., expanders or locally irregular structures) require different noise injection profiles than graphs dominated by low-frequency components (e.g., community-structured or manifold-like graphs). By conditioning the drift and diffusion coefficients on the spectral context c, our generative model adapts automatically to these differences. This stands in contrast to standard diffusion models that apply a fixed, graph-agnostic noise schedule.

Practically, this reduces the mismatch between the diffusion process and the data distribution, leading to more stable training and more realistic samples.

8.2 Improved Generalization Across Graph Types

Graph datasets often contain a mixture of structurally distinct graphs. Molecular datasets, for example, include graphs with different numbers of rings, branching factors, and functional groups, while social network datasets span dense friendship graphs, sparse interaction graphs, and hub-dominated follower graphs. A single fixed diffusion schedule struggles to accommodate this diversity.

The spectral context vector serves as a light-weight global descriptor that allows the model to generalize across graph families. During training, the model implicitly clusters noise schedules according to spectral similarity, leading to improved generalization on rare or structurally unusual graphs.

8.3 Reduced Sensitivity to Hyperparameters

Diffusion models are known to be sensitive to the choice of noise schedules, number of diffusion steps, and timestep distributions. These hyperparameters largely determine the signal-to-noise ratio (SNR) profile and the difficulty of score matching at different timesteps.

Because spectral context allows the model to modulate its effective noise scale on a per-graph basis, the overall training process becomes less sensitive to hyperparameters. In practice, we observe that a wider range of timestep grids, learning rates, and weighting schedules $\lambda(t)$ produce stable training and competitive performance. This reduces the search burden typically required for diffusion models.

8.4 Enhanced Controllability and Conditional Generation

One notable advantage of having an explicit spectral representation is that it becomes possible to *steer* the generative process in the space of structural graph properties. The spectral density of the Laplacian is strongly linked to several interpretable graph characteristics:

- the number of connected components (mass at $\lambda = 0$),
- community structure (large low-frequency mass),
- edge expansion (spectral gap),
- local irregularity (large high-frequency mass),
- graph diameter (inverse relation via low eigenvalues).

By conditioning sampling on a desired spectral context vector $c^\star$, users may bias the model toward generating graphs with controlled clustering structure, expansion, or connectivity. Importantly, this does not require explicit constraints, spectral regularization, or differentiable eigen-solvers. This opens the door to a new class of *spectrally steered generative models*.

8.5 Better Robustness to Structural Noise

Real-world graphs often contain noise in the form of missing edges, spurious connections, or structural perturbations. These errors tend to manifest primarily in the high-frequency region of the Laplacian spectrum. By characterizing and conditioning on the approximate spectral density, the model can implicitly account for such perturbations and maintain robustness in its generative behavior.

Furthermore, spectral estimates aggregated across graphs in a minibatch provide a global indicator of dataset noise, allowing the diffusion process to adapt its SNR schedule accordingly.

8.6 Compatibility with Large-Scale and Streaming Graphs

Many applications involve graphs that evolve over time or arrive as streams: social networks, citation graphs, transportation networks, and dynamic molecular simulations. In such scenarios, repeatedly computing eigenvectors or Laplacian factorizations is infeasible.

Since spectral density estimation can be performed via a small number of matvecs, it is well suited for:

- **Mini-batch graph training**: Each graph independently receives its spectral summary without interactions.
- **Growing graphs**: Re-estimating moments $T_k(\tilde{L})$ is fast and does not require recomputing eigenvalues from scratch.

- **Streaming data**: Each incoming graph can be assigned a spectral signature on the fly, enabling online adaptation of diffusion dynamics.

This positions spectrally conditioned diffusion models as strong candidates for large-scale graph generation tasks where traditional spectral methods are infeasible.

8.7 Interpretability and Diagnostics

While diffusion models are often criticized for their lack of interpretability, the incorporation of spectral context provides a principled connection between the model's internal conditioning and well-understood global graph properties. This enables:

- new diagnostics for understanding generative failures (e.g., mismatch between target and generated spectra),
- interpretability of sampling trajectories in relation to structural complexity,
- tools for analyzing cluster-separation in latent diffusion space,
- post-hoc control of generated graph families by adjusting the spectral inputs.

Thus, spectral integration acts as a bridge between classical spectral graph theory and modern deep generative models.

8.8 Summary

In addition to compute savings, the use of sublinear spectral density estimation provides deeper modeling benefits: spectrum-aware diffusion dynamics, structural controllability, robustness, improved generalization, and better interpretability. These advantages highlight the broader potential of combining spectral algorithms with deep generative modeling on graphs.

9 Conclusion

We presented a spectrum-aware extension of score-based graph generative models that leverages sublinear-time spectral density estimation of the graph Laplacian. By summarizing the Laplacian spectrum into a compact context vector and conditioning the latent diffusion process on this context, the model can adapt its noise schedule and dynamics to the structural properties of each graph, without incurring the cost of explicit eigendecomposition. Our framework connects recent advances in sublinear spectral algorithms with score-based graph generation, and suggests a promising direction for scalable, structure-aware graph generative modeling.

References

[1] Vladimir Braverman, Aditya Krishnan, and Christopher Musco. 2022. Sublinear Time Spectral Density Estimation. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. ACM, New York, NY, USA, 1031–1044. arXiv:2104.03461 [cs.DS]

[2] Mucong Ding, Tahseen Rabbani, Bang An, Evan Z. Wang, and Furong Huang. 2022. Sketch-GNN: Scalable Graph Neural Networks with Sublinear Training Complexity. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 35. Curran Associates, Inc., New Orleans, LA, USA, 723–735. arXiv:2206.08763 [cs.LG]

[3] Jongmin Jo, Guankai Yang, and Noseong Park. 2022. Score-Based Generative Modeling of Graphs via the System of Stochastic Differential Equations. In *Proceedings of the 39th International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, Baltimore, Maryland, USA, 10362–10383. arXiv:2202.02514 [cs.LG]

[4] Chuxu Liu, Xiaohan Zheng, Tianxiang Yu, Meng Liu, and Aidong Zhang. 2023. Generative Diffusion Models on Graphs: Methods and Applications. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Edith Elkind (Ed.). International Joint Conferences on Artificial Intelligence Organization, Macao, SAR China, 6810–6818. arXiv:2302.02591 [cs.LG]

[5] Chengxi Niu, Yang Song, Jiaming Chen, Hanjun Dai, Bo Li, and Stefano Ermon. 2020. Permutation Invariant Graph Generation via Score-Based Generative Modeling. In *Proceedings of the 37th International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, Virtual, 7360–7369. arXiv:2003.00638 [cs.LG]

[6] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. *International Conference on Learning Representations (ICLR)* 9, 3 (2021), 1–34. arXiv:2011.13456 [cs.LG] Published in ICLR 2021.

[7] Junwei Su and Chuan Wu. 2025. A Non-Asymptotic Convergent Analysis for Score-Based Graph Generative Model via a System of Stochastic Differential Equations. In *Proceedings of the 42nd International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. Forthcoming)*, Forthcoming (Ed.). PMLR, Forthcoming, Forthcoming, Forthcoming. arXiv:2508.14351 [cs.LG] To appear.