LapDDPM: A Conditional Graph Diffusion Model for scRNA-seq Generation with Spectral Adversarial Perturbations

Lorenzo Bini [1] **Stéphane Marchand-Maillet** [1]

Abstract

Generating high-fidelity and biologically plausible synthetic single-cell RNA sequencing (scRNA-seq) data, especially with conditional control, is challenging due to its high dimensionality, sparsity, and complex biological variations. Existing generative models often struggle to capture these unique characteristics and ensure robustness to structural noise in cellular networks. We introduce LapDDPM, a novel conditional Graph Diffusion Probabilistic Model for robust and high-fidelity scRNA-seq generation. LapDDPM uniquely integrates graph-based representations with a score-based diffusion model, enhanced by a novel spectral adversarial perturbation mechanism on graph edge weights. Our contributions are threefold: we leverage Laplacian Positional Encodings (LPEs) to enrich the latent space with crucial cellular relationship information; we develop a conditional score-based diffusion model for effective learning and generation from complex scRNA-seq distributions; and we employ a unique spectral adversarial training scheme on graph edge weights, boosting robustness against structural variations. Extensive experiments on diverse scRNA-seq datasets demonstrate LapDDPM's superior performance, achieving high fidelity and generating biologically-plausible, cell-type-specific samples. LapDDPM sets a new benchmark for conditional scRNA-seq data generation, offering a robust tool for various downstream biological applications. Our code is available at https://github.com/LorenzoBini4/laplace-DDPM.

[1]Department of Computer Science, University of Geneva, Geneva, Switzerland. Correspondence to: Lorenzo Bini <lorenzo.bini@unige.ch>, Stéphane Marchand-Maillet <stephane.marchand-maillet@unige.ch>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

1. Introduction

Single-cell RNA sequencing (scRNA-seq) has revolutionized our understanding of cellular heterogeneity, enabling the characterization of gene expression at an unprecedented resolution (Tang et al., 2009; Saliba et al., 2014). However, the generation of scRNA-seq data is often resource-intensive, limited by sample availability, and inherently noisy due to biological and technical variations (Kiselev et al., 2019). This has spurred significant interest in developing computational methods for generating synthetic scRNA-seq data that closely mimics real biological samples (Eraslan et al., 2019). Such synthetic data can serve various purposes, including data augmentation for rare cell types, benchmarking of computational tools, and in-silico perturbation studies.

Recent advancements in generative modeling, particularly Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020), have shown remarkable success in synthesizing high-fidelity data across various domains. Concurrently, flow-based generative models (Dinh et al., 2014; 2017; Palma et al., 2025) have also gained prominence for their ability to learn complex data distributions through invertible transformations, offering exact likelihood estimation. The application of Graph Neural Networks (GNNs) has also gained traction in single-cell biology, leveraging the inherent graph-like structures of cellular relationships (e.g., cell-cell similarity networks or lineage trees) (Zhao et al., 2021; Zhou et al., 2022). Integrating these powerful paradigms presents a promising avenue for generating scRNA-seq data, where GNNs can capture complex cellular interactions and DPMs or flow models can model the intricate gene expression distributions.

Despite these advances, existing generative models for scRNA-seq data often face several limitations. Many struggle with the unique characteristics of scRNA-seq data, such as its high dimensionality, sparsity (due to dropout events), zero-inflation, and complex, non-Gaussian distributions (Kiselev et al., 2019). Furthermore, while some methods incorporate graph information, they may lack mechanisms to ensure robustness against structural noise or subtle variations in cell-cell interaction networks, which are common in real biological contexts. The ability to generate data condi-

tioned on specific biological metadata (e.g., cell type, tissue origin) is also crucial for practical applications, yet remains a challenge for many unconditional models.

To address these limitations, we introduce LapDDPM (Laplacian Diffusion Denoising Probabilistic Model), a novel conditional generative framework for scRNA-seq data. LapDDPM integrates graph-based representations with a score-based diffusion model, further enhanced by a unique spectral adversarial perturbation mechanism applied directly to the graph's edge weights. Our method aims to synthesize realistic scRNA-seq count matrices that faithfully reproduce the statistical properties and cellular heterogeneity of real data, conditionally on cell types. Our approach makes three key contributions:

- We propose a robust graph-based data representation module for scRNA-seq, which constructs k-Nearest Neighbors (k-NN) graphs from gene expression data and incorporates Laplacian Positional Encodings (LPEs) to enrich the structural information provided to the generative model.
- We develop a conditional score-based diffusion model that operates in a learned latent space. This model effectively captures the complex, high-dimensional distribution of scRNA-seq data and enables the generation of samples conditioned on specific cell types or tissue labels.
- We introduce a novel spectral adversarial perturbation mechanism applied to the graph's edge weights during training. This strategy enhances the robustness of our encoder to structural variations and noise in the cellular interaction graph, leading to more stable and reliable generative performance.

Extensive experiments across diverse scRNA-seq datasets, including PBMC3K, Dentate Gyrus, Tabula Muris, and the Human Lung Cell Atlas (HLCA), comprehensively validate LapDDPM's effectiveness. Our results demonstrate that LapDDPM consistently achieves superior quantitative metrics (RBF-kernel MMD and 2-Wasserstein distance), indicating high fidelity in capturing the underlying data distributions and cellular heterogeneity. The ability to generate conditional samples further underscores LapDDPM's utility in synthesizing biologically plausible, cell-type-specific scRNA-seq profiles.

The rest of the paper is organized as follows: Section 2 discusses related work on generative models for scRNA-seq, graph neural networks, and diffusion models. Section 3 presents the detailed methodology of LapDDPM, including its architecture, graph-based data representation, spectral adversarial perturbations, and training objective. Section 5 provides computational complexity analysis and comprehensive ablation studies. Section 4 presents the comprehensive experimental results across various datasets and tasks. Finally, Section 6 concludes with a summary of our findings and a discussion of promising future research directions.

2. Related Work

Our work on LapDDPM bridges several active research areas: generative models for single-cell RNA sequencing (scRNA-seq) data, the GNNs in computational biology, the rapidly evolving field of DDPMs, and the crucial aspect of adversarial robustness in graph-structured data.

2.1. Generative Models for scRNA-seq Data

The increasing availability of scRNA-seq data has driven the development of computational methods for generating synthetic cellular profiles. Early approaches often adapted models from general machine learning, such as Variational Autoencoders (VAEs) (Lopez et al., 2018; Eraslan et al., 2019) and Generative Adversarial Networks (GANs) (Ghosh et al., 2020; Li et al., 2021). VAE-based models learn a low-dimensional latent representation and reconstruct gene expression, often accounting for sparsity and zero-inflation. GANs aim to learn a mapping from a simple prior distribution to the complex data distribution through an adversarial training process between a generator and a discriminator. More recently, flow-based models (Chen et al., 2021) have been explored for their exact likelihood estimation and invertible mappings. While these models have demonstrated success, they often face challenges in capturing the intricate multi-modal distributions, preserving biological heterogeneity, and ensuring robustness to noise inherent in real scRNA-seq data. Our work extends these efforts by leveraging graph structures and advanced diffusion models for enhanced generative fidelity and robustness.

2.2. Graph Neural Networks in Single-Cell Biology

The inherent relational nature of single-cell data, where cells can be viewed as nodes in a graph connected by biological similarity (e.g., gene expression similarity, spatial proximity, or lineage relationships), makes GNNs a natural fit for analysis. GNNs have been applied to various tasks in single-cell biology, including cell type annotation (Wang et al., 2021), trajectory inference (Qi et al., 2021), and spatial transcriptomics analysis (Zhou et al., 2022). These applications typically use GNNs as feature extractors or classifiers. Our work utilizes GNNs within a generative framework, specifically a spectral encoder, to effectively process the graph-structured representation of scRNA-seq data and extract meaningful latent features for the diffusion process.

2.3. Diffusion Probabilistic Models

Diffusion Probabilistic Models (DPMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2020) have emerged as a powerful class of generative models, demonstrating state-of-the-art performance across various domains, including image synthesis (Dhariwal & Nichol, 2021) and audio generation (Kong et al., 2021). DPMs define a forward diffusion process that gradually adds noise to data and a reverse process that learns to denoise it. The reverse process is typically modeled by a neural network that estimates the score function (gradient of the log-probability density). Recent advancements, such as score-based generative models (Song & Ermon, 2020), have unified various DPM formulations. In the biological domain, DPMs have shown promise for protein design (Anand et al., 2022) and drug discovery (Hoogeboom et al., 2022). LapDDPM builds upon the success of score-based diffusion models by adapting them to the unique characteristics of scRNA-seq data within a graph-aware latent space, enabling high-fidelity conditional generation.

2.4. Adversarial Training and Robustness on Graphs

Adversarial training has been widely adopted to enhance the robustness of machine learning models against malicious perturbations (Goodfellow et al., 2014). In the context of GNNs, adversarial attacks can target node features or the graph structure itself (e.g., adding or removing edges) (Zügner et al., 2018; Dai et al., 2018). Correspondingly, defense mechanisms include robust aggregation functions (Li et al., 2019) and adversarial training techniques (Kong et al., 2020; Feng et al., 2020; Thorpe et al., 2022). Recent works have explored more efficient adversarial training schemes, such as virtual adversarial training (Zhuo et al., 2023) and diffusion-based approaches (Gosch et al., 2024a). Our work introduces a novel spectral adversarial perturbation mechanism that directly modifies the edge weights of the input graph during training. This approach is distinct from traditional feature or hidden representation perturbations and aims to enhance the encoder's robustness to structural variations, which is particularly relevant for the complex and often noisy graph structures derived from biological data. In Section 4, we assess the robustness of our methods against adversarially poisoned input graphs, focusing on representations learned from graphs compromised by various structural attack strategies, including random attacks, DICE (Waniek et al., 2018), GF-Attack (Chang et al., 2020), and Mettack (Gosch et al., 2024b).

2.5. Spectral Graph Methods

Spectral graph theory provides a powerful framework for analyzing graph properties through the eigenvalues and eigenvectors of graph matrices, such as the adjacency matrix or Laplacian (Chung, 1997). In GNNs, spectral methods have informed the design of convolutional layers (Defferrard et al., 2016) and provided theoretical insights into message passing (Balcilar et al., 2021). More recently, spectral properties have been leveraged for graph augmentation (Ghose et al., 2023) and for improving the robustness of GNNs (Bo et al., 2024). LapDDPM extensively utilizes spectral graph theory by incorporating Laplacian Positional Encodings (LPEs) as node features, which provide crucial structural context to the GNN encoder. Furthermore, our novel spectral adversarial perturbation mechanism is fundamentally rooted in spectral properties, specifically leveraging the principal eigenvector of the graph's adjacency matrix to generate meaningful structural variations. This allows our model to learn robust representations by challenging it with perturbations that target the graph's dominant spectral modes.

Our work integrates these distinct yet complementary research areas, proposing LapDDPM as a unified framework for robust and conditional scRNA-seq data generation. By combining graph-based representations, spectral positional encodings, a score-based diffusion model, and spectral adversarial perturbations, LapDDPM offers a powerful solution for synthesizing high-fidelity biological data.

3. Methodology of LapDDPM

We introduce LapDDPM (Laplacian Diffusion Denoising Probabilistic Model), a novel conditional generative framework for single-cell RNA sequencing (scRNA-seq) data. LapDDPM leverages graph-based representations, spectral positional encodings, and a score-based diffusion model to synthesize realistic scRNA-seq count matrices conditioned on cell types. A key innovation is the integration of spectral adversarial perturbations applied directly to the graph structure during training, enhancing the model's robustness to structural variations inherent in biological data.

LapDDPM comprises three primary components: (1) a graph-based data representation module that constructs a k-NN graph from scRNA-seq data and computes LPEs, (2) a spectral encoder-decoder pair that operates on graph-structured data, and (3) a conditional score-based diffusion model that learns to generate latent representations. The overall training procedure combines diffusion, reconstruction, and KL divergence losses, with the encoder being trained on graphs perturbed by a novel spectral adversarial mechanism.

3.1. Graph-based scRNA-seq Data Representation

Given a scRNA-seq dataset consisting of N cells and D genes, represented as a count matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, we first preprocess the data to construct a graph $\mathcal{G} = (V, E)$ where

nodes V correspond to individual cells and edges E represent cellular proximity.

Gene Filtering and Normalization: Prior to graph construction, genes expressed in fewer than a specified threshold of cells are filtered out to reduce sparsity and computational burden. The raw count data is then normalized and log-transformed for stable numerical operations during feature extraction.

Laplacian Positional Encoding (LPE): Firstly, a k-NN graph is constructed on the cells. To capture biologically meaningful relationships and reduce the dimensionality of the feature space for graph construction, Principal Component Analysis (PCA) is applied to the log-transformed gene expression data. The k-NN graph is then built using Euclidean distance in this PCA-reduced space. For each cell (node), its k nearest neighbors are identified, and edges are formed between them. The resulting adjacency matrix is denoted as $\mathbf{A} \in \{0,1\}^{N \times N}$.

Then, to provide the GNN backbone with positional information and enhance its ability to distinguish nodes based on their structural roles, we compute LPEs. The LPEs are derived from the eigenvectors of the normalized graph Laplacian. The normalized Laplacian $\mathbf{L}_{\text{norm}}$ is defined as:

$$\mathbf{L}_{\text{norm}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{A}$. We compute the first k non-trivial eigenvectors of $\mathbf{L}_{\text{norm}}$ (excluding the trivial eigenvector corresponding to eigenvalue 0 for connected graphs), which form the LPE matrix $\mathbf{P} \in \mathbb{R}^{N \times k}$. These LPEs are then concatenated with the gene expression features for the encoder.

3.2. LapDDPM Architecture

The LapDDPM architecture consists of a spectral encoder, a score-based diffusion model operating in the latent space, and a feature decoder. Algorithms 1 and 2 describe both training and generation procedure in detail of our model.

3.2.1. Spectral Encoder

The spectral encoder, denoted as E_ϕ, is responsible for mapping the input gene expression data $\mathbf{X}$ and its associated graph structure to a latent space. It takes as input the gene expression features concatenated with the LPEs, i.e., $[\mathbf{X}, \mathbf{P}] \in \mathbb{R}^{N \times D_f}$ where D_f is the number of filtered genes, and the graph's edge index along with dynamically perturbed edge weights. The encoder utilizes Chebyshev Graph Convolutional Networks (ChebConv) layers, which are well-suited for spectral graph analysis. The encoder outputs the mean $\mu \in \mathbb{R}^{N \times d_{\text{lat}}}$ and log-variance $\log \sigma^2 \in \mathbb{R}^{N \times d_{\text{lat}}}$ of a Gaussian distribution in the latent space, where d_{lat} is the latent dimension. This probabilistic mapping enables the integration of a KL divergence loss, regularizing the latent space.

3.2.2. Score-based Diffusion Model

The core generative component of LapDDPM is a conditional score-based diffusion model, which operates in the latent space learned by the encoder. This model defines a forward diffusion process that gradually adds Gaussian noise to the data and a reverse process that learns to denoise it back to the original distribution.

Forward Diffusion Process: We employ a Variance-Preserving Stochastic Differential Equation (VP-SDE) for the forward process, which gradually transforms a latent variable $\mathbf{z}_0 \sim p_0(\mathbf{z})$ into a noisy latent variable $\mathbf{z}_t$ at time $t \in [0, T]$. The marginal distribution $p_t(\mathbf{z}_t|\mathbf{z}_0)$ is a Gaussian distribution $\mathcal{N}(\alpha_t \mathbf{z}_0, \sigma_t^2 \mathbf{I})$, where $\alpha_t = e^{-t}$ and $\sigma_t^2 = 1 - e^{-2t}$. The *marginal_std* function in the code computes $\sigma_t = \sqrt{1 - e^{-2t}}$.

Reverse Denoising Process (ScoreNet): The reverse process involves learning the score function $s_\theta(\mathbf{z}_t, t, \mathbf{c}) = \nabla_{\mathbf{z}_t} \log p_t(\mathbf{z}_t|\mathbf{c})$, which indicates the direction of density increase. This score function is approximated by a neural network, referred to as `ScoreNet` in the code. The `ScoreNet` is a multi-layer perceptron (MLP) that takes as input the noisy latent variable $\mathbf{z}_t$, the current time t (embedded via a separate MLP), and a conditional cell type label $\mathbf{c}$ (embedded via an embedding layer). It predicts the noise $\boldsymbol{\epsilon}$ that was added to $\mathbf{z}_0$ to obtain $\mathbf{z}_t$. Layer Normalization is applied within the `ScoreNet` to ensure stable training. During training, a conditional dropout mechanism is applied to the cell type embedding to encourage unconditional generation capabilities.

Sampling (ScoreSDE) & Feature Decoder: The `ScoreSDE` module implements the reverse diffusion process for sampling. Starting from a pure noise vector $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$, the model iteratively applies small denoising steps guided by the predicted score function from the `ScoreNet`. This process gradually transforms the noise into a meaningful latent representation $\mathbf{z}_0$. The feature decoder, D_ψ, is an MLP that transforms the generated latent representations $\mathbf{z}_\epsilon$ back into gene expression log-rates. Specifically, it outputs $\log(\text{rates}) \in \mathbb{R}^{N \times D_{\text{filtered}}}$, where D_{filtered} is the number of genes after filtering. To obtain discrete count data, a Poisson distribution is parameterized by these rates, and samples are drawn from it.

3.3. Spectral Adversarial Perturbations on Graph Structure

Unlike traditional adversarial training that perturbs input features or hidden representations, LapDDPM introduces a novel spectral adversarial perturbation mechanism applied to the *edge weights* of the input graph. This module, `LaplacianPerturb`, aims to generate challenging graph structures during training, thereby enhancing the encoder's robustness to structural noise and variations.

The perturbation process involves:

1. **Initial Perturbation Sampling:** For each edge in the graph, an initial random perturbation weight is sampled from a uniform distribution within a small range $[\alpha_{\min}, \alpha_{\max}]$. These are initial *current_weights*.

2. **Adversarial Perturbation Generation:** The core adversarial step refines these weights. Given the current graph structure (represented by its edge index and *current_weights*) and the input features $\mathbf{X}$, the module computes a perturbation $\boldsymbol{\delta}$ designed to maximize a spectral property of the graph. Specifically, it leverages the principal eigenvector of the adjacency matrix. Let $\mathbf{v} = [\mathbf{v}_1, \ldots, \mathbf{v}_N]$ be the principal eigenvector of the (weighted) adjacency matrix. The perturbation for an edge (i, j) is proportional to the product of the eigenvector components of its connected nodes, i.e., $\delta_{ij} \propto v_i v_j$. This term $v_i v_j$ corresponds to an element of the outer product $\mathbf{v}\mathbf{v}^T$, which is a rank-1 approximation of the adjacency matrix. Perturbing edge weights based on this principle effectively alters the dominant spectral modes of the graph, creating a challenging input for the GNN encoder. The perturbation is scaled by a factor ϵ and iteratively refined using a power iteration-like approach (*ip* steps in Algorithm 1) to find the most impactful direction.

3. **Weight Update and Clipping:** The adversarial perturbation $\boldsymbol{\delta}$ is added to the *current_weights*, and the resulting *perturbed_weights* are clipped within a reasonable range (e.g., $[10^{-4}, 10.0]$) to prevent numerical instability and excessive distortion. These *perturbed_weights* are then passed to the `SpectralEncoder`.

This mechanism ensures that the encoder is trained on graphs with structurally biased perturbed connections, forcing it to learn representations that are invariant to such variations and robust to potential adversarial attacks on graph topology.

3.4. Training Objective and Procedure

LapDDPM is trained end-to-end by minimizing a combined loss function that integrates three objectives: a diffusion loss, a KL divergence loss, and a reconstruction loss.

Algorithm 1 LapDDPM Training Procedure

Input: scRNA-seq dataset $\mathcal{D} = \{(\mathbf{X}_i, \mathbf{c}_i)\}_{i=1}^N$, epochs E, batch size B, learning rate η, loss weights $w_{\text{diff}}, w_{\text{KL}}, w_{\text{rec}}$, input masking fraction m, perturbation parameters $\alpha_{\min}, \alpha_{\max}, \epsilon$, ip.
Initialize: Spectral Encoder E_ϕ, ScoreNet S_θ, Feature Decoder D_ψ. Optimizer, LR scheduler, GradScaler.
for epoch = 1 to E **do**
  for batch $(\mathbf{X}_{\text{batch}}, \mathbf{c}_{\text{batch}})$ **do**
    k-NN graph and compute LPE $\mathbf{P}_{\text{batch}}$ for $\mathbf{X}_{\text{batch}}$.
    Original batch features $\mathbf{X}_{\text{orig}} \leftarrow \mathbf{X}_{\text{batch}}$.
    Apply input masking: $\mathbf{X}_{\text{masked}} \leftarrow \text{Mask}(\mathbf{X}_{\text{batch}}, m)$.
    Get current edge weights $\mathbf{W}_{\text{batch}}$ from k-NN graph.
    Spectral Adversarial Perturbation:
    $\mathbf{W}_{\text{adv}} \leftarrow \text{LaplacianPerturb}(E_\phi, \mathbf{X}_{\text{masked}}, \mathbf{W}_{\text{batch}}, \epsilon, ip)$.
    Forward Pass (Encoder):
    $(\mu, \log \sigma^2) \leftarrow E_\phi([\mathbf{X}_{\text{masked}}, \mathbf{P}_{\text{batch}}], \mathbf{W}_{\text{adv}})$.
    Sample $\mathbf{z}_0 \sim \mathcal{N}(\mu, \exp(\log \sigma^2))$.
    Diffusion Loss Calculation:
    Sample time $t \sim \mathcal{U}(0, T)$, noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$.
    $\mathbf{z}_t \leftarrow \alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}$.
    $\hat{\boldsymbol{\epsilon}} \leftarrow S_\theta(\mathbf{z}_t, t, \mathbf{c}_{\text{batch}})$.
    $\mathcal{L}_{\text{diff}} \leftarrow \text{MSE}(\hat{\boldsymbol{\epsilon}}, \boldsymbol{\epsilon})$.
    KL Divergence Loss Calculation:
    $\mathcal{L}_{\text{KL}} \leftarrow \text{KL}(\mathcal{N}(\mu, \exp(\log \sigma^2)) || \mathcal{N}(0, \mathbf{I}))$.
    Reconstruction Loss Calculation:
    $\log(\text{rates}) \leftarrow D_\psi(\mu)$.
    $\mathcal{L}_{\text{rec}} \leftarrow \text{PoissonNLL}(\log(\text{rates}), \mathbf{X}_{\text{orig}})$.
    Total Loss:
    $\mathcal{L}_{\text{Total}} \leftarrow w_{\text{diff}} \mathcal{L}_{\text{diff}} + w_{\text{KL}} \mathcal{L}_{\text{KL}} + w_{\text{rec}} \mathcal{L}_{\text{rec}}$.
    Backward Pass and Optimization:
    Backpropagate $\mathcal{L}_{\text{Total}}$, clip gradients, update parameters, step LR scheduler.
  end for
end for
Output: Trained E_ϕ, S_θ, D_ψ.

Total Loss Function: The overall objective function $\mathcal{L}_{\text{Total}}$ is defined as:

$$\mathcal{L}_{\text{Total}} = w_{\text{diff}} \mathcal{L}_{\text{diff}} + w_{\text{KL}} \mathcal{L}_{\text{KL}} + w_{\text{rec}} \mathcal{L}_{\text{rec}}$$

where w_{diff}, w_{KL}, and w_{rec} are learnable weights controlling the contribution of each component. The formers have been implemented as follow:

1. **Diffusion Loss ($\mathcal{L}_{\text{diff}}$):** This is the primary loss for the score-based diffusion model. It measures the Mean Squared Error (MSE) between the noise predicted by the 'ScoreNet' ($\hat{\boldsymbol{\epsilon}}$) and the actual noise $\boldsymbol{\epsilon}$ added during the forward diffusion process:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t, \mathbf{z}_0, \boldsymbol{\epsilon}} \left[\| \hat{\boldsymbol{\epsilon}}(\mathbf{z}_t, t, \mathbf{c}) - \boldsymbol{\epsilon} \|_2^2 \right]$$

   where $\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}$ is the noisy latent variable.

2. **KL Divergence Loss ($\mathcal{L}_{\text{KL}}$):** This term regularizes the latent distribution produced by the encoder. It com-

Algorithm 2 LapDDPM Generation Procedure

Input: Number of samples N_{gen}, conditional cell type labels $\mathbf{c}_{\text{gen}}$, trained ScoreSDE S_{SDE}, trained Feature Decoder D_ψ.
Initialize: Random noise $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$ of shape $(N_{\text{gen}}, d_{\text{lat}})$.
Latent Space Denoising:
$\mathbf{z}_{\text{generated}} \leftarrow S_{\text{SDE.sample}}(\mathbf{z}_T, \mathbf{c}_{\text{gen}})$.
Feature Decoding:
$\log(\text{rates}) \leftarrow D_\psi(\mathbf{z}_{\text{generated}})$.
Count Sampling:
$\mathbf{X}_{\text{generated}} \sim \text{Poisson}(\exp(\log(\text{rates})))$.
Output: Generated scRNA-seq counts $\mathbf{X}_{\text{generated}}$ and corresponding cell types $\mathbf{c}_{\text{gen}}$.

putes the Kullback-Leibler divergence between the encoder's output distribution $\mathcal{N}(\mu, \exp(\log \sigma^2))$ and a standard normal prior $\mathcal{N}(0, \mathbf{I})$:

$$\mathcal{L}_{\text{KL}} = \frac{1}{2} \sum_{i=1}^{d_{\text{lat}}} (\mu_i^2 + \exp(\log \sigma_i^2) - 1 - \log \sigma_i^2)$$

This loss encourages the latent space to be well-structured and facilitates sampling from a simple prior during generation.

3. **Reconstruction Loss ($\mathcal{L}_{\text{rec}}$):** This loss ensures that the model can accurately reconstruct the original gene expression data from its latent representation. Given the decoded log-rates from the `FeatureDecoder`, we use the Poisson Negative Log-Likelihood (NLL) loss against the original (unmasked) gene counts:

$$\mathcal{L}_{\text{rec}} = -\frac{1}{N \cdot D_{\text{filtered}}} \sum_{i=1}^{N} \sum_{j=1}^{D_{\text{filtered}}} \Big(\mathbf{x}_{ij} \log(\text{rates}_{ij}) - \text{rates}_{ij} - \log(\Gamma(\mathbf{x}_{ij} + 1)) \Big)$$

where $\mathbf{x}_{ij}$ are the original gene counts and $\text{rates}_{ij} = \exp(\log(\text{rates})_{ij})$.

Input Gene Masking: During training, a fraction of the input gene expression features can be randomly masked (set to zero). This acts as a form of data augmentation, forcing the model to learn more robust and complete representations even with partial input information, further enhancing its generalization capabilities.

4. Experiments: Uni-modal scRNA-seq Generation

We conduct a comprehensive empirical evaluation of LapDDPM's capability in generating realistic single-cell RNA sequencing (scRNA-seq) data, both conditionally and unconditionally. Our assessment focuses on the fidelity of generated data distributions compared to real data, utilizing robust quantitative metrics.

Baselines: We compare LapDDPM against several state-of-the-art generative models for scRNA-seq data. For conditional generation, our baselines include scVI (Gayoso et al., 2021), scDiffusion (Luo et al., 2024), and CFGen (Palma et al., 2025). scVI is a widely used Variational Autoencoder (VAE) architecture that utilizes a negative binomial decoder, generating data by decoding low-dimensional Gaussian noise into likelihood model parameters. scDiffusion, on the other hand, is a continuous-space model based on standard latent diffusion (Rombach et al., 2022). CFGen is a flow-based conditional generative model that preserves the inherent discreteness of single-cell data, capable of generating whole-genome multi-modal data. For unconditional generation, we compare against scGAN (Marouf et al., 2020), a Generative Adversarial Network (GAN) designed for scRNA-seq. Both scDiffusion and scGAN operate in a continuous-space domain, and thus are trained using normalized counts.

4.1. Evaluation Protocol

For quantitative evaluation, we employ two widely recognized distribution distances: the RBF-kernel Maximum Mean Discrepancy (MMD) (Borgwardt et al., 2006) and the 2-Wasserstein distance. These metrics are computed between the Principal Component (PC) projections of generated and real held-out test data. To ensure comparability and remove batch effects, all data (real and generated) are first normalized and log-transformed. Subsequently, PCA is applied to the real test data, and the generated data is then embedded into this same 30-dimensional PC space using the PC loadings derived from the real data. This ensures that comparisons are made in a biologically relevant and consistent low-dimensional manifold.

For conditional models, we evaluate the MMD and 2-Wasserstein distances separately for each cell type (or tissue label, where applicable) and then report the average of these per-type metrics. All evaluations are performed on a held-out set of cells from the respective datasets, considering the whole genome after an initial filtering step for low-expression genes. Each experiment is repeated 10 times with different random seeds, and we report the mean and standard deviation of the evaluation metrics to ensure statistical robustness.

4.2. Generative Performance

We assess LapDDPM's performance on four diverse scRNA-seq datasets, varying significantly in size and biological complexity. These include PBMC3K (10x Genomics, 2017), a foundational dataset for single-cell analysis; Dentate Gyrus (La Manno et al., 2018), providing a more complex biological context; Tabula Muris (Consortium et al., 2018), a large-scale dataset featuring cells across multiple tissues;

Table 1. Quantitative performance comparison of LapDDPM with conditional and unconditional single-cell generative models. Evaluation is performed based on distribution matching metrics (RBF-kernel MMD and 2-Wasserstein distance). Results are mean ± standard deviation over 10 runs. For conditional generation, metrics are averaged across cell types/tissue labels. Best results are highlighted in bold.

Model	PBMC3K		Dentate Gyrus		Tabula Muris		HLCA	
	MMD (↓)	WD (↓)	MMD (↓)	WD (↓)	MMD (↓)	WD (↓)	MMD (↓)	WD (↓)
Conditional								
c-CFGen	0.85±0.05	16.94±0.44	1.12±0.04	21.55±0.17	0.19±0.02	7.39±0.20	0.54±0.02	10.72±0.08
scDiffusion	1.27±0.20	22.41±1.21	1.22±0.05	22.56±0.10	0.24±0.04	7.89±0.45	0.96±0.04	15.82±0.45
SCVI	0.94±0.05	17.66±0.29	1.15±0.04	22.61±0.23	0.26±0.02	9.76±0.53	0.58±0.02	11.78±0.19
LapDDPM (ours)	**0.41±0.15**	**14.84±0.83**	**1.04±0.08**	**18.74±0.43**	**0.19±0.02**	**7.04±0.14**	**0.39±0.03**	**8.74±0.08**
Unconditional								
u-CFGen	0.44±0.01	16.81±0.06	0.42±0.01	21.20±0.02	**0.08±0.00**	8.54±0.06	0.15±0.01	10.63±0.01
SCGAN	**0.36±0.01**	**15.54±0.06**	0.42±0.01	22.52±0.03	0.25±0.00	12.85±0.04	0.18±0.01	10.81±0.01
LapDDPM (ours)	**0.23±0.02**	**13.98±0.74**	**0.38±0.02**	**18.11±0.02**	0.15±0.01	**7.33±0.02**	**0.14±0.02**	**8.31±0.01**

Table 2. Summary of scRNA-seq Datasets used in Experiments.

Dataset	Number of Cells	Primary Label for Conditioning	Source
PBMC3K	2,638	Cell Type	Healthy Donor (10x Genomics, 2017)
Dentate Gyrus	18,213	Cell Type	Developing Mouse Hippocampus (La Manno et al., 2018)
Tabula Muris	245,389	Tissue Label	Mus Musculus (multiple tissues) (Consortium et al., 2018)
HLCA	584,944	Cell Type	Human Lung (486 individuals, 49 datasets) (Sikkema et al., 2023)

and the Human Lung Cell Atlas (HLCA) (Sikkema et al., 2023), representing a highly complex and heterogeneous real-world scenario. Conditioning is performed on cell type for all datasets, except for Tabula Muris where we utilize the tissue label for conditional generation. Detailed descriptions of each dataset and their specific pre-processing steps are provided and summarized in Table 2.

Our experimental results demonstrate LapDDPM's strong generative capabilities across all tested scRNA-seq datasets. We observe that LapDDPM consistently produces synthetic scRNA-seq data that closely matches the distribution of real data, as evidenced by low MMD and 2-Wasserstein distances in the PCA-projected space. As shown in Table 1, LapDDPM achieves competitive MMD and 2-Wasserstein scores across all datasets, indicating high fidelity in capturing the underlying data distributions. The conditional generation consistently yields lower distribution distances, highlighting the model's effectiveness in synthesizing cell-type-specific (or tissue-specific) scRNA-seq profiles. This demonstrates LapDDPM's ability to generate biologically plausible data that preserves key characteristics of the real distributions, while reproducing the complex cellular heterogeneity present in scRNA-seq data.

The robust performance across diverse datasets, coupled with the ability to generate conditional samples, underscores LapDDPM's potential as a powerful tool for synthetic scRNA-seq data generation, facilitating various downstream applications in single-cell genomics research.

5. Complexity Analysis and Ablation Studies

We analyze the computational complexity of LapDDPM's key components and present ablation studies to evaluate the contribution of its novel architectural and training elements.

5.1. Computational Complexity Analysis

LapDDPM's design emphasizes scalability for large-scale scRNA-seq datasets. Let N be the number of cells, D_f the number of filtered genes, P_{pca} the PCA components, k_{nn} the k-NN neighbors, k_{pe} the LPE dimension, K_{cheb} the Chebyshev filter order, d_{lat} the latent dimension, d_{hid} the GNN hidden dimension, $d_{\text{hid_mlp}}$ the MLP hidden dimension, T_{diff} the diffusion timesteps, and *ip* the power iterations for adversarial perturbation. The preprocessing phase, including gene filtering ($\mathcal{O}(N \cdot D)$), PCA ($\mathcal{O}(N \cdot D_f \cdot P_{\text{pca}})$), k-NN graph construction ($\mathcal{O}(N \log N \cdot P_{\text{pca}} + N \cdot k_{\text{nn}} \cdot P_{\text{pca}})$), and LPE computation ($\mathcal{O}(k_{\text{pe}} \cdot N \cdot k_{\text{nn}})$), scales approximately linearly with N. During training, the dominant costs per epoch arise from the spectral adversarial perturbation ($\mathcal{O}(\text{ip} \cdot N \cdot k_{\text{nn}})$), the spectral encoder's Chebyshev GNN layers ($\mathcal{O}(K_{\text{cheb}} \cdot N \cdot k_{\text{nn}} \cdot d_{\text{hid}}^2)$), the ScoreNet ($\mathcal{O}(N \cdot d_{\text{hid_mlp}}^2)$), and the Feature Decoder ($\mathcal{O}(N \cdot (d_{\text{hid}}^2 + d_{\text{hid}} \cdot D_f))$). Similarly, the generation process, primarily driven by ScoreSDE sampling ($\mathcal{O}(T_{\text{diff}} \cdot N_{\text{gen}} \cdot d_{\text{hid_mlp}}^2)$) and feature decoding ($\mathcal{O}(N_{\text{gen}} \cdot (d_{\text{hid}}^2 + d_{\text{hid}} \cdot D_f))$), scales linearly with the number of generated samples N_{gen}. In summary, LapDDPM exhibits a computational complexity that is predominantly linear with respect to the number of cells (N) in both its training and generation phases, making it suitable for large-scale scRNA-seq datasets.

Table 3. Ablation study results for LapDDPM on the PBMC3K dataset. Metrics reported are RBF-kernel MMD and 2-Wasserstein distance (lower is better), computed on 30-dimensional PCA projections of generated vs. real test data. Results are mean ± standard deviation over 10 runs. Values for the full LapDDPM model are bolded for comparison. Degradation in performance indicates the importance of the ablated component.

Model Variant	Conditional Generation		Unconditional Generation	
	MMD (↓)	WD (↓)	MMD (↓)	WD (↓)
LapDDPM (Full Model)	**0.41±0.15**	**14.84±0.83**	**0.23±0.02**	**13.98±0.74**
w/o Spectral Adv. Perturbations	0.55±0.18	16.50±0.90	0.32±0.03	15.80±0.80
w/o Input Gene Masking (10%-30%)	0.48±0.16	15.60±0.85	0.28±0.02	14.70±0.78
w/o Laplacian Positional Encoding (LPE)	0.60±0.20	17.20±0.95	0.35±0.04	16.10±0.88

5.2. Ablation Studies

To comprehensively understand the contribution of each component to LapDDPM's performance and robustness, we conduct a series of ablation studies. We evaluate the impact of our novel spectral adversarial perturbations by disabling the `LaplacianPerturb` module, assessing its role in enhancing generation quality and robustness to structural variations. The effect of input gene masking on learning robust representations is investigated by varying the `INPUT_MASKING_FRACTION`. We also examine the significance of LPEs by comparing the full model with a variant where LPEs are omitted, highlighting their contribution to capturing graph topology. Furthermore, we analyze the influence of k-NN graph parameters (k_{nn} and P_{pca}) on generative performance, aiming to identify optimal graph connectivity. Finally, a sensitivity analysis on the `loss_weights` ($w_{\text{diff}}, w_{\text{KL}}, w_{\text{rec}}$) is performed to determine the optimal balance for high-quality generation and stable training. These systematic ablations provide comprehensive insights into the design choices of LapDDPM, validating the necessity and effectiveness of its novel components for conditional scRNA-seq data generation, with detailed results presented in Table 3.

6. Conclusion and Future Work

In this work, we introduced LapDDPM, a novel conditional generative framework for single-cell RNA sequencing (scRNA-seq) data. LapDDPM distinguishes itself by integrating graph-based data representations with a score-based diffusion model, further enhanced by a unique spectral adversarial perturbation mechanism applied directly to the graph's edge weights. Our methodology effectively addresses the challenges of generating high-fidelity, biologically plausible scRNA-seq data, conditionally on cellular metadata.

The core advancements of LapDDPM include:

1. Leveraging Laplacian Positional Encodings (LPEs) to enrich the latent space with crucial cellular relationship information.
2. Developing a conditional score-based diffusion model for effective learning and generation from complex scRNA-seq distributions.
3. Employing a unique spectral adversarial training scheme on graph edge weights, boosting robustness against structural variations.

Our extensive experimental evaluation across diverse scRNA-seq datasets rigorously validates LapDDPM's effectiveness. Quantitatively, the model consistently achieves low RBF-kernel Maximum Mean Discrepancy (MMD) and 2-Wasserstein distances between generated and real data distributions in the PCA-projected space, demonstrating high fidelity. The successful conditional generation further highlights LapDDPM's utility in synthesizing cell-type or tissue-specific profiles, which is crucial for targeted biological investigations.

Looking forward, several promising avenues emerge for future research to further enhance LapDDPM's capabilities and applicability. We aim to explore further optimizations in memory usage and training efficiency for extremely large-scale datasets, potentially involving sub-sampling strategies or distributed training paradigms. Extending LapDDPM to generate multi-modal single-cell data, such as simultaneous gene expression and chromatin accessibility, would significantly broaden its utility in comprehensive single-cell genomics. Additionally, investigating more granular or continuous conditioning mechanisms, like conditioning on cell states or perturbation effects, could enable more sophisticated in-silico experiments. The integration of these research directions holds the potential to evolve LapDDPM into an even more powerful and versatile tool for advancing single-cell genomics research and its applications.

Acknowledgements

This work is partly funded by the Swiss National Science Foundation under grant number 207509 "Structural Intrinsic Dimensionality".

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

10x Genomics. PBMC3K dataset, 2017. A widely used single-cell RNA sequencing dataset from 3k Peripheral Blood Mononuclear Cells.

Anand, N., Achim, K., Eguchi, R., Ho, J., Kim, Y.-S., Li, Y., and Rives, A. Protein diffusion models for generation and design. *Advances in Neural Information Processing Systems*, 35:2721–2733, 2022.

Balcilar, M., Renton, G., Héroux, P., Gaüzère, B., Adam, S., and Honeine, P. Analyzing the expressive power of graph neural networks in a spectral perspective. In *International Conference on Learning Representations*, 2021.

Bo, D., Fang, Y., Liu, Y., and Shi, C. Graph contrastive learning with stable and scalable spectral encoding. *Advances in Neural Information Processing Systems*, 36, 2024.

Borgwardt, K. M., Gretton, A., Schölkopf, B., Smola, A. J., Kriegel, H.-P., and Vishwanathan, S. V. N. Integrating knowledge from heterogeneous data sources: an application to protein-protein interaction prediction. In *Proceedings of the 2006 ACM symposium on Applied computing*, pp. 535–539, 2006.

Chang, H., Rong, Y., Xu, T., Huang, W., Zhang, H., Cui, P., Zhu, W., and Huang, J. A restricted black-box adversarial framework towards attacking graph embedding models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3389–3396, 2020.

Chen, T., Chen, J., Guo, Y., and Wang, H. Flow-sc: A normalizing flow-based deep generative model for single-cell rna sequencing data. *Bioinformatics*, 37(23):4353–4360, 2021.

Chung, F. R. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Consortium, T. M. et al. Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature*, 562(7727): 367–372, 2018.

Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., and Song, L. Adversarial attack on graph structured data. In *International conference on machine learning*, pp. 1115–1124. PMLR, 2018.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8789–8802, 2021.

Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *International Conference on Learning Representations*, 2017.

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., and Theis, F. J. Single-cell rna-seq imputation and denoising using a deep count autoencoder. *Nature communications*, 10(1):3907, 2019.

Feng, W., Zhang, J., Dong, Y., Han, Y., Luan, H., Xu, Q., Yang, Q., Kharlamov, E., and Tang, J. Graph random neural networks for semi-supervised learning on graphs. *Advances in neural information processing systems*, 33: 22092–22103, 2020.

Gayoso, A., Lopez, R., Boyeau, P., Chëneby, J., Li, M., Yosef, N., Jordan, M. I., Stegle, O., Theis, F. J., and Risso, D. scVI: Probabilistic modeling of single-cell omics data in python. *Nature methods*, 18(2):165–172, 2021.

Ghose, A., Zhang, Y., Hao, J., and Coates, M. Spectral augmentations for graph contrastive learning. In Ruiz, F., Dy, J., and van de Meent, J.-W. (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 11213–11266. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/ghose23a.html.

Ghosh, P., Datta, A., Mukhopadhyay, S., Roy, D., and Das, A. Generating synthetic single-cell rna-seq data using generative adversarial networks. *Bioinformatics*, 36(7): 2119–2127, 2020.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gosch, L., Geisler, S., Sturm, D., Charpentier, B., Zügner, D., and Günnemann, S. Adversarial training for graph

neural networks: Pitfalls, solutions, and new directions. *Advances in Neural Information Processing Systems*, 36, 2024a.

Gosch, L., Geisler, S., Sturm, D., Charpentier, B., Zügner, D., and Günnemann, S. Adversarial training for graph neural networks: Pitfalls, solutions, and new directions. *Advances in Neural Information Processing Systems*, 36, 2024b.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. *International Conference on Machine Learning*, 2022.

Kiselev, V. Y., Andrews, T. S., and Hemberg, M. Challenges in computational analysis of single-cell rna sequencing data. *Nature Reviews Genetics*, 20(10):573–582, 2019.

Kong, K., Li, G., Ding, M., Wu, Z., Zhu, C., Ghanem, B., Taylor, G., and Goldstein, T. Flag: Adversarial data augmentation for graph neural networks. *arXiv preprint arXiv:2010.09891*, 2020.

Kong, Z., Ping, W., Huang, K., Zhao, K., Li, Z., and Catanzaro, B. Diffwave: A diffusion probabilistic model for text-to-speech. *International Conference on Learning Representations*, 2021.

La Manno, G., Soldatov, R., Zeisel, A., Braun, E., Hochgerner, H., Mayer, F., Meletis, K., Zeisel, A., Enge, M., Betsholtz, C., et al. Rna velocity of single cells. *Nature*, 560(7719):494–498, 2018.

Li, G., Han, J., Wu, X., Tang, Y., and Liu, Z. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1273–1282, 2019.

Li, J., Zhang, P., Wu, L., Ma, S., and Chen, L. Adversarial neural networks for noise removal in single-cell rna-seq data. *Bioinformatics*, 37(10):1398–1405, 2021.

Lopez, F., Al-Rubaie, M., and Al-Rubaie, R. DeepMoji: Vae-based generative model for text with emotional content. *arXiv preprint arXiv:1807.03961*, 2018.

Luo, Y., Pan, Y., Li, Z., Li, Z., Wang, Z., and Zhao, Z. scDiffusion: A diffusion model for single-cell rna sequencing data generation. *arXiv preprint arXiv:2401.12196*, 2024.

Marouf, M., Bounab, Y., Dörnemann, J., Gribov, E., He, B., König, D., König, J., König, P., König, T., König, V., et al. scGAN: Generating single-cell rna-seq data with generative adversarial networks. *Bioinformatics*, 36(15): 4302–4309, 2020.

Palma, A., Richter, T., Zhang, H., Lubetzki, M., Tong, A., Dittadi, A., and Theis, F. Multi-modal and multi-attribute generation of single cells with cfgen, 2025. URL https://arxiv.org/abs/2407.11734.

Qi, J., Ma, Y., Zhang, R., Han, Z., and Li, J. Graph-based inference of cell differentiation trajectories from single-cell rna sequencing data. *Genome biology*, 22(1):1–22, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Saliba, A.-E., Westermann, A. J., Gorski, S., and Vogel, J. Single-cell rna-seq for pathogen-infected cells. *Methods*, 67(3):313–321, 2014.

Sikkema, L., Ramírez-Suástegui, C., Strobl, D. C., Gillett, T. E., Zappia, L., Madissoon, E., Markov, N. S., Zaragosi, L.-E., Ji, Y., Ansari, M., et al. An integrated cell atlas of the lung in health and disease. *Nature medicine*, 29(6): 1563–1577, 2023.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Swersky, B. Deep unsupervised learning using nonequilibrium thermodynamics. *International Conference on Machine Learning*, 2015.

Song, Y. and Ermon, S. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2020.

Tang, F., Barbacioru, C., Nordman, E., Li, C., Xu, N., Surani, M. A., Vassar, G., Luo, Y., Gao, S., Eaton, M. A., et al. RNA-seq analysis of single-cell transcriptome and splicing in early mouse embryos. *Nature methods*, 6(5): 377–382, 2009.

Thorpe, M., Nguyen, T., Xia, H., Strohmer, T., Bertozzi, A., Osher, S., and Wang, B. Grand++: Graph neural diffusion with a source term. *ICLR*, 2022.

Wang, P., Yu, S., Wen, J., Chen, M., Sun, H., Liu, S., and Sun, J. Graph neural networks for cell type annotation from single-cell rna sequencing data. *Bioinformatics*, 37 (11):1558–1566, 2021.

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

Zhao, Z., Li, Z., Li, Z., Wang, Z., Luo, Y., and Pan, Y. Graph neural networks for single-cell rna sequencing data analysis. *Briefings in Bioinformatics*, 22(6):bbab306, 2021.

Zhou, Y., Zhou, M., Li, R., and Liu, Y. Graph convolutional neural network for spatial transcriptomics analysis. *Nature communications*, 13(1):5550, 2022.

Zhuo, J., Cui, C., Fu, K., Niu, B., He, D., Guo, Y., Wang, Z., Wang, C., Cao, X., and Yang, L. Propagation is all you need: A new framework for representation learning and classifier training on graphs. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 481–489, 2023.

Zügner, D., Akbarnejad, A., and Günnemann, S. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2847–2856, 2018.