INDUCING FUNCTIONAL LOCALITY IN LINEAR ATTENTION FOR WHOLE-TRANSCRIPTOME GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-cell RNA-seq profiles are long, high-dimensional and sparse, imposing severe computational bottlenecks in generative modeling at the whole-transcriptome scale. Existing methods predominantly rely on Performer for linear approximation but neglect biological structural priors. We introduce PI2L, a generative framework operating across $19{,}295$ protein-coding genes that explicitly leverages the geometry of the PPI network as a soft inductive bias. PI2L embeds the interaction confidence as a Gaussian distance decay on a unit sphere, which is reformulated into an exponential kernel of inner products. The concatenation adopts a norm-preserving mixture form, where content and prior terms are allocated one shared attention logit budget conditioned on the corruption level. Empirical results demonstrate that cells synthesized by PI2L outperform those from classical single-cell generative models in terms of gene expression correlation and distributional fidelity. Overall, PI2L provides a scalable optimization paradigm for single-cell generative modeling, exhibiting broad utility for in silico cell generation and biological interpretability. Our code is anonymously available at `https://anonymous.4open.science/r/pi2l-submission/`.

1 INTRODUCTION

The development of single-cell RNA sequencing (scRNA-seq) technologies provides opportunities to model cellular states across diverse tissues, species, and conditions (Kolodziejczyk et al., 2015; Regev et al., 2017; Svensson et al., 2018; Jovic et al., 2022). Due to technical bias, high sequencing costs, and difficulty in acquiring biological samples, obtaining enough high-quality scRNA-seq data remains challenging (Suvà & Tirosh, 2019; Jiang et al., 2022; Ke et al., 2022). Deep generative models offer a powerful approach to address these challenges by synthesizing biologically meaningful scRNA-seq data, thereby reducing experimental costs and accelerating biological discovery.

Single-cell foundation models, such as scGPT (Cui et al., 2024) and Geneformer (Theodoris et al., 2023), treat genes as discrete tokens and represent cells as sequences, relying on masked modeling to capture intricate expression patterns. However, these foundation models are primarily designed for learning static representations rather than performing generative simulations. Variational autoencoders (VAEs) constitute the foundational generative framework for scRNA-seq data, and scVI (Lopez et al., 2018) has been widely adopted in practice. Meanwhile, scGAN (Marouf et al., 2020) and scDiffusion (Luo et al., 2024) synthesize cellular profiles using generative adversarial networks (GANs) (Goodfellow et al., 2014) and diffusion-based models (Rombach et al., 2022). Nevertheless, constrained by the assumption of continuous data, these generative models inherently struggle to capture the discrete nature of single-cell transcriptomic data.

To adjust to the discrete and non-sequential nature of genes, some studies use masked discrete diffusion models (MDDMs) (Austin et al., 2021; Chang et al., 2022; Shi et al., 2024; Sahoo et al., 2024; Nie et al., 2026) for the generative modeling of cellular states. Although ScDiVa (Wang et al., 2026) has demonstrated that MDDMs can perform non-autoregressive, bidirectional refinement of transcriptome profiles, it cannot model the whole transcriptome. To capture long-range gene interactions, existing single-cell models (Yang et al., 2022; Gong et al., 2023; Hao et al., 2024; Bai et al., 2026) employ Performers (Choromanski et al., 2020), which alleviate computational bottlenecks by approximating the softmax operation using kernel functions. Although computationally attractive,

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.

global aggregation of random features tends to conflate weak biological signals with noise, leading to a progressive accumulation of errors.

We investigate whether, when modeling the whole transcriptome, MDDMs with linear attention can use biological topology to modulate attention according to the corruption level. We draw inspiration from kernel-based linear transformers (Wang et al., 2020; Katharopoulos et al., 2020; Peng et al., 2021; Qin et al., 2022a;b) in natural language processing, which exploit sequential locality biases to concentrate attention on adjacent tokens due to their semantic relevance. However, unlike natural language sequences, genes do not have a canonical one-dimensional ordering. Studies on structured masking (Liutkus et al., 2021; Luo et al., 2021; Choromanski et al., 2022; Reid et al., 2025) have shown that graph priors can modulate the attention kernel without materializing the attention matrix. Following this idea, we induce functional locality between protein-coding genes from their interactions in the Protein-Protein Interaction (PPI) network (Szklarczyk et al., 2025), based on the correspondence between genes and their respective protein products.

In this work, we introduce PI2L (Protein–Protein Interaction network Informed Linear attention), a framework that leverages biological priors for high-quality scRNA-seq data generation. To make the prior compatible with linear attention, we use a spherical PPI embedding algorithm to represent the dominant interaction structure of the network on a low-dimensional unit sphere. It reformulates the Gaussian distance decay across the spherical manifold into an inner product that the exponential kernel absorbs by concatenation, while unit norm bounds the estimator variance of the kernel approximation. Therefore, PI2L can modulate the inter-gene attention and adjust the prior intensity on a per-head basis through a learnable corruption gating mechanism. Our main contributions are summarized as follows:

- We propose PI2L, a generative model grounded in masked discrete diffusion, which leverages the PPI network to capture gene dependencies within functional neighborhoods at the attention layers, while employing conditional parameterization at the feed-forward layers.
- To accommodate the sparsity of scRNA-seq data, we equip the MDDM framework with a hurdle decoder at the output stage, alongside a correspondingly tailored training objective.
- We demonstrate that PI2L outperforms classical single-cell generative models in whole-transcriptome generation across multiple tissue-specific atlases.

2 Related Work

Generative Modeling for Single-cell Data. VAE-based scVI (Lopez et al., 2018) encodes scRNA-seq data into low-dimensional latent variables and generates posterior expression estimates parameterized by a zero-inflated negative binomial distribution. scGAN (Marouf et al., 2020) and scDiffusion (Luo et al., 2024) employ GANs (Goodfellow et al., 2014) and latent diffusion models (Rombach et al., 2022), which generate in continuous spaces. Flow matching (Albergo & Vanden-Eijnden, 2022; Liu et al., 2022; Lipman et al., 2022) has emerged as an efficient continuous normalizing flow technique that is widely utilized for conditional generation (Palma et al., 2025; Li et al., 2026). To further boost generative capabilities, recent models have explored more expressive architectural paradigms. Masked discrete diffusion models (MDDMs) (Austin et al., 2021) inherently align with the discrete nature of scRNA-seq data. To mitigate high-dimensional sparsity, ScDiVa (Wang et al., 2026) prioritizes the extraction of discriminative features over reconstructing the whole transcriptome. Although Lingshu-Cell (Zhang et al., 2026a) successfully models whole-transcriptome expression, it relies on sequence compression within the embedding space. We argue that randomly partitioning unordered genes and projecting them into shorter internal sequences may disrupt the underlying biological topology, which motivates our exploration of linear attention mechanisms to circumvent computational bottlenecks.

Linear Attention in Single-cell Learning. Drawing inspiration from efficient long text processing frameworks in natural language processing, scBERT (Yang et al., 2022) utilizes the Performer (Choromanski et al., 2020) to approximate attention computations. xTrimoGene (Gong et al., 2023) and scFoundation (Hao et al., 2024) adopt an asymmetric encoder-decoder architecture, where the encoder processes only the non-zero expression values, while the lightweight decoder employs the Performer for comprehensive feature extraction. scLong (Bai et al., 2026) employs two Performers of different capacities to separately process highly expressed genes and the low-expressed or un-

expressed genes, subsequently aggregating these features via a full-length Performer. While these single-cell foundation models typically rely on a fixed, low masking ratio, we highlight that under varying masking ratios, Performers tend to confound genuine biological signals with background noise. To address this, we propose the integration of biological structures as a soft inductive bias.

Biological Structure in Single-cell Modeling. scNET (Sheinin et al., 2025) leverages a dual-view architecture to integrate scRNA-seq data with the PPI network, thereby jointly learning both gene and cell representations. scKGBERT (Li et al., 2025) treats the PPI network as a modality, employing a dual-encoder architecture to process gene expression data and biological knowledge graphs simultaneously. GREmLN (Zhang et al., 2026b) incorporates molecular interaction graphs through attention based on a graph diffusion kernel. RegFormer (Hu et al., 2026) performs topological sorting using gene regulatory networks (GRNs) and combines it with Mamba-based state-space modeling. In contrast, our approach introduces functional locality from the PPI network, dynamically modulating inter-gene attention weights in accordance with the diffusion schedule.

3 METHODOLOGY

3.1 PRELIMINARIES AND DIFFUSION MODELING

We first establish a shared gene vocabulary $\mathcal{V}$ (Appendix H.1), mapping the gene symbols to unique gene indices. During the data preprocessing phase, all datasets, the Protein-Protein Interaction (PPI) network, and the pretrained static embeddings from Geneformer are strictly aligned with the gene vocabulary $\mathcal{V}$ (Appendix H.2). For a given cell, let $\mathbf{x}_0 = [\mathrm{x}^1, \ldots, \mathrm{x}^G]^\top$ denote a discrete gene sequence of fixed length G, where $\mathrm{x}^i \geq 0$ represents the expression value of the gene with index i.

We adopt a masked discrete diffusion paradigm to model the joint distribution $p(\mathbf{x}_0)$, which employs an absorbing-state-based global stochastic corruption mechanism: once a gene expression value is masked, it cannot revert to the visible state at any subsequent diffusion time. The forward process is formulated as a continuous-time Markov process that progressively corrupts information by gradually and independently masking the gene expression values of $\mathbf{x}_0$. During training, we sample a continuous diffusion time $\mathrm{t} \sim \mathcal{U}((0, 1))$. Given diffusion time t, we define a stochastic mask vector $\mathbf{m}_t = [\mathrm{m}_t^1, \ldots, \mathrm{m}_t^G]^\top$ by independently sampling a Boolean mask for each gene. The state at diffusion time t is determined by $\mathbf{m}_t$: if $\mathrm{m}_t^i = 1$, the gene transitions to the absorbing state with $\mathrm{x}_t^i = \texttt{MASK}$; otherwise, if $\mathrm{m}_t^i = 0$, it remains unchanged as $\mathrm{x}_t^i = \mathrm{x}_0^i$. For the gene i, its forward marginal distribution is given by:

$$q(\mathrm{x}_t^i \mid \mathrm{x}_0^i) = (1 - t)\delta_{\mathrm{x}_0^i}(\mathrm{x}_t^i) + t\delta_{\texttt{MASK}}(\mathrm{x}_t^i). \tag{1}$$

At any intermediate timestep t, each token x_t^i in $\mathbf{x}_t$ retains its original state x_0^i with a survival probability $1 - t$, or transitions to the absorbing state with probability t. The reverse denoising process aims to recover the true data distribution by iteratively predicting the masked gene expression values as t transitions from 1 back to 0, which learns the conditional distribution $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_t)$. The core of this process is a parameterized mask predictor neural network $p_\theta(\cdot \mid \mathbf{x}_t)$, which takes a partially masked sequence as input and simultaneously predicts the original tokens for all masked positions.

To avoid the ordering bias introduced by autoregressive sequence modeling (Brown et al., 2020; Floridi & Chiriatti, 2020; Cui et al., 2024), PI2L incorporates biologically meaningful structures to reconstruct gene expression values bidirectionally and non-causally (Devlin et al., 2019) at masked positions based on the global context. In the following sections, we elaborate on each component in detail. The complete workflow is shown in Figure 1.

3.2 INITIAL EMBEDDINGS

Gene Identity Encoder. An scRNA expression matrix comprises two types of information: gene identities and gene expression values. Each sample shares an identical gene identity sequence $\boldsymbol{g} = [1, \ldots, G] \in \mathbb{R}^G$. The gene identity sequence is mapped to continuous embeddings by the Gene Identity Encoder: $\mathbf{H}^{\mathrm{id}} = \mathcal{F}_{\mathrm{GIE}}(\boldsymbol{g}) = \boldsymbol{E}_{\mathrm{GF}}\boldsymbol{W}_{\mathrm{proj}}$, where $\mathbf{H}^{\mathrm{id}} \in \mathbb{R}^{G \times d}$ represents the gene identity embeddings, $\boldsymbol{E}_{\mathrm{GF}}$ is a static embedding layer initialized by Geneformer (Theodoris et al., 2023), and $\boldsymbol{W}_{\mathrm{proj}}$ is a shared, bias-free linear projection layer with deterministic orthogonal initialization.

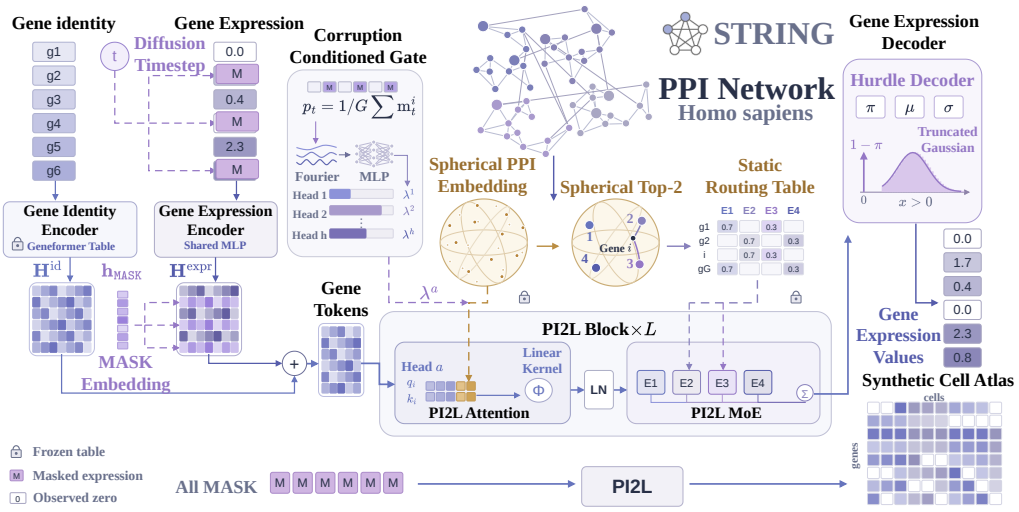

Figure 1: **Overview of PI2L Architecture.** The spherical embeddings provided by the PPI network are leveraged by two modules: they modulate the attention weights within the attention layer and facilitate efficient conditional parameterization in the feed-forward layer. The corruption conditional gate regulates the prior strength for each attention head. The three-channel outputs of the hurdle decoder jointly model a point mass at zero and a Gaussian distribution truncated to positive values.

Gene Expression Encoder. Another crucial component of $\mathbf{x}_0$ is the sequence of gene expression values, denoted as $\mathbf{e} = [\mathrm{e}^1, \ldots, \mathrm{e}^G]^\top \in \mathbb{R}^{G\times 1}$. Log-normalized gene expression values are mapped into continuous embeddings by the Gene Expression Encoder: $\mathbf{H}^{\text{expr}} = \mathcal{F}_{\text{GEE}}(\mathbf{e})$, where $\mathbf{H}^{\text{expr}} \in \mathbb{R}^{G\times d}$ represents the gene expression embeddings. Specifically, $\mathcal{F}_{\text{GEE}}$ is parameterized by a lightweight, shared two-layer MLP equipped with element-wise SiLU activations.

Gene Token Embedding. Let $\mathbf{h}_{\text{MASK}} \in \mathbb{R}^d$ denote the learnable mask embedding shared by all genes. For visible positions, the gene embeddings are obtained by adding the gene identity embeddings and the gene expression embeddings; for masked positions, they are computed by adding the shared $\mathbf{h}_{\text{MASK}}$ to the gene identity embeddings. The final gene embeddings are formulated as:

$$\mathbf{H}_t^{\text{in}} = \mathbf{H}^{\text{id}} + (\boldsymbol{I}_G - \text{Diag}(\mathbf{m}_t))\mathbf{H}^{\text{expr}} + \mathbf{m}_t\mathbf{h}_{\text{MASK}}^\top \in \mathbb{R}^{G\times d}. \tag{2}$$

3.3 PI2L

Spherical PPI Embedding. Following the matrix-factorization view of network embedding (Levy & Goldberg, 2014; Qiu et al., 2018), we construct spherical PPI embeddings to provide a functional locality geometry for attention modulation and expert routing. For combined scores filtered by a confidence threshold τ, we preserve continuous weights to distinguish varying interaction strengths. Symmetric degree normalization is applied to moderate connectivity-related scale differences while preserving a symmetric propagation operator. Aggregating its powers with decaying weights incorporates higher-order structural information beyond direct interactions while emphasizing short paths. By removing the diagonal of the propagation matrix, we obtain the cross-gene affinity matrix $\boldsymbol{M}$, ensuring that subsequent calibration depends on associations between distinct genes. Propagated affinities can still reflect differences in overall association strengths. To extract gene-pair associations that are enriched relative to the background, we compare each affinity with a factorized baseline determined by the marginal strengths of its endpoints in $\boldsymbol{M}$, constructing a node-strength-corrected log-enrichment matrix $\boldsymbol{P}$. The logarithm compresses the dynamic range of affinity ratios, while the enrichment threshold b controls the enrichment required for an association to be retained.

To obtain a compact representation suitable for inner product computation, we retain the r largest positive eigenmodes of $\boldsymbol{P}$, yielding the raw embedding $\boldsymbol{Z}^{\text{raw}} = \boldsymbol{U}_r\Sigma_r^{1/2}$, whose Gram matrix provides a low-rank positive semidefinite approximation of $\boldsymbol{P}$. Subsequently, non-degenerate rows are centered and normalized to unit norm: the former represents each gene relative to the embedding

centroid, while the latter makes similarity dependent on direction rather than vector magnitude. On the unit sphere, the Gaussian distance decay can be expressed as an exponential inner product kernel that can be incorporated into linear attention by concatenating the corresponding embeddings. The same angular geometry also supports spherical clustering for static expert routing. The computational procedure and treatment of degenerate genes are provided in Appendix A and Appendix B.

PI2L Block. Let $\mathbf{H}_t^{(l-1)}$ and $\mathbf{H}_t^{(l)}$ denote the input and output representations of the l-th layer, respectively, and let $\boldsymbol{Z} = [\boldsymbol{z}_1^\top; \ldots; \boldsymbol{z}_G^\top] \in \mathbb{R}^{G\times r}$ be the offline spherical PPI embeddings. The spherical PPI embedding matrix $\boldsymbol{Z}$ is treated as a fixed constant shared across all samples, diffusion timesteps, blocks, and attention heads, meaning it entails no trainable parameters and receives no gradients. PI2L adopts the Pre-LN residual structure following Performer. The normalized hidden states are projected into queries, keys, and values, and are subsequently partitioned into h attention heads. For the a-th attention head at layer l, this yields $\mathbf{Q}^{(l,a)}, \mathbf{K}^{(l,a)}, \mathbf{V}^{(l,a)} \in \mathbb{R}^{G\times d_h}$, where $d_h = d/h$. In the sequel, we omit the layer and head indices for notational simplicity, denoting the representations for the i-th token as $\mathbf{q}_i^0, \mathbf{k}_i^0, \mathbf{v}_i \in \mathbb{R}^{d_h}$.

Injection into Linear Attention. Rather than relying on dedicated timestep embeddings, the gene embeddings convey diffusion temporal information through the number of masked tokens. Specifically, we adopt the realized mask ratio $p_t = 1/G \sum_{i=1}^{G} \mathrm{m}_t^i$ as the input to the PPI prior gating mechanism. Compared to the nominal diffusion time t, which serves as a nominal scheduling index, p_t reflects the actual corruption level of each individual sample, thereby reducing dependence on specific diffusion schedule parameterizations. We encode p_t using Fourier features $\boldsymbol{\gamma}(p_t) = \big[\sin(2^0\pi p_t), \cos(2^0\pi p_t), \ldots, \sin(2^{h-1}\pi p_t), \cos(2^{h-1}\pi p_t)\big] \in \mathbb{R}^{2h}$, where h denotes the number of attention heads. $\boldsymbol{\gamma}(p_t)$ is then processed by a lightweight two-layer MLP with a sigmoid activation to bound the prior weight, yielding $\boldsymbol{\lambda}(p_t) \in (0,1)^h$. This Fourier-MLP operates as a learnable corruption conditioned gate deployed independently across layers.

To prevent contaminating the prior term, we apply symmetric scaling to the content term prior to injecting the PPI components: $\mathbf{q}_i = d_h^{-1/4}\mathbf{q}_i^0$ and $\mathbf{k}_j = d_h^{-1/4}\mathbf{k}_j^0$. We then compute the intra-head root-mean-square (RMS) norms of the queries and keys as $\sigma_q = \mathrm{sg}\left[(G^{-1}\sum_i \|\mathbf{q}_i\|_2^2)^{1/2}\right]$ and $\sigma_k = \mathrm{sg}\left[(G^{-1}\sum_j \|\mathbf{k}_j\|_2^2)^{1/2}\right]$, where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator. Because σ_q and σ_k propagate this identical scale calibration to the prior components, dimension-dependent scaling must not be redundantly re-applied with respect to the augmented dimension $d_h + r$ after concatenation.

For the current attention head, we set $\lambda = \boldsymbol{\lambda}^{(a)}(p_t)$. Prior to kernelization, we concatenate the content and prior terms of the genes:

$$\widetilde{\mathbf{q}}_i = \begin{bmatrix} \sqrt{1-\lambda}\,\mathbf{q}_i \\ \sqrt{\lambda}\,\sigma_q \boldsymbol{z}_i \end{bmatrix} \in \mathbb{R}^{d_h+r}, \qquad \widetilde{\mathbf{k}}_j = \begin{bmatrix} \sqrt{1-\lambda}\,\mathbf{k}_j \\ \sqrt{\lambda}\,\sigma_k \boldsymbol{z}_j \end{bmatrix} \in \mathbb{R}^{d_h+r}, \tag{3}$$

defining the target logit to be approximated by random features as $\widetilde{\mathbf{q}}_i^\top \widetilde{\mathbf{k}}_j = (1-\lambda)\mathbf{q}_i^\top \mathbf{k}_j + \lambda\sigma_q\sigma_k \boldsymbol{z}_i^\top \boldsymbol{z}_j$, with detailed explanation deferred to Appendix C.

We construct a positive kernel approximation using orthogonal random features (ORF). Specifically, we generate a random orthogonal projection matrix $\boldsymbol{\Omega} = [\boldsymbol{\omega}_1^\top; \ldots; \boldsymbol{\omega}_{m_{\mathrm{rf}}}^\top] \in \mathbb{R}^{m_{\mathrm{rf}}\times(d_h+r)}$ via QR decomposition, where m_{rf} denotes the number of random features and the row norms follow a χ distribution with $d_h + r$ degrees of freedom. The s-th log-feature coordinate for the query is given by $a_{Q,i,s} = \boldsymbol{\omega}_s^\top \widetilde{\mathbf{q}}_i - \frac{1}{2}\|\widetilde{\mathbf{q}}_i\|_2^2$, with an analogous formulation for the keys. Because each query only contributes to its own output computation, its stabilization term is determined by a per-token local shift; in contrast, since keys are aggregated into a shared summary across the entire sequence, all keys must share a unified global scalar:

$$c_{Q,i} = \mathrm{sg}[\max_{1\le s\le m_{\mathrm{rf}}} a_{Q,i,s}], \quad c_K = \mathrm{sg}[\max_{1\le j\le G, 1\le s\le m_{\mathrm{rf}}} a_{K,j,s}]. \tag{4}$$

The coordinate-wise stabilized positive features are defined as

$$\phi_{Q,i,s} = m_{\mathrm{rf}}^{-1/2}\exp(a_{Q,i,s} - c_{Q,i}), \quad \phi_{K,j,s} = m_{\mathrm{rf}}^{-1/2}\exp(a_{K,j,s} - c_K). \tag{5}$$

These entries are assembled into the feature vectors $\boldsymbol{\phi}_Q(\widetilde{\mathbf{q}}_i) = [\phi_{Q,i,1}, \ldots, \phi_{Q,i,m_{\mathrm{rf}}}]^\top$ and $\boldsymbol{\phi}_K(\widetilde{\mathbf{k}}_j) = [\phi_{K,j,1}, \ldots, \phi_{K,j,m_{\mathrm{rf}}}]^\top$ across the fixed coordinates. Notably, we do not inject a

coordinate-wise stabilization constant ε directly into the feature mapping, as doing so would introduce a uniform background term into the approximated kernel that is independent of both content and PPI. Instead, the numerical stabilization constant is applied exclusively to the denominator during the final normalization stage. By aggregating the keys and values to accumulate two core statistics, we obtain the single-head representation for the i-th token:

$$\mathbf{o}_i^\top = \frac{\boldsymbol{\phi}_Q(\widetilde{\mathbf{q}}_i)^\top \sum_{j=1}^{G} \boldsymbol{\phi}_K(\widetilde{\mathbf{k}}_j)\mathbf{v}_j^\top}{\boldsymbol{\phi}_Q(\widetilde{\mathbf{q}}_i)^\top \sum_{j=1}^{G} \boldsymbol{\phi}_K(\widetilde{\mathbf{k}}_j) + \varepsilon} \in \mathbb{R}^{d_h}, \qquad \varepsilon > 0. \tag{6}$$

Stacking the representations across all tokens and concatenating over the h attention heads yields $\mathbf{O}_{\mathrm{cat}} \in \mathbb{R}^{G\times d}$, followed by a linear projection with bias.

Conditional Parameter Sharing. Beyond modulating inter-gene attention, we further leverage PPI-derived functional locality to organize parameter sharing in the feed-forward layers. This encourages functionally related genes to reuse expert transformations while allowing different genes to adopt distinct combinations of experts. This sharing pattern is determined offline from the fixed PPI geometry, ensuring consistency across diverse cellular contexts and masking states.

Specifically, we apply a spherical k-means, followed by top-k_{route} assignment, to the non-degenerate spherical PPI embeddings to obtain E_{exp} unit-norm prototypes $\{\boldsymbol{\mu}_e\}_{e=1}^{E_{\mathrm{exp}}}$, where k_{route} denotes the number of experts selected per gene. Each gene i independently selects the k_{route} prototypes exhibiting the highest cosine similarities to form its expert support set $\mathcal{T}_i$. Its mixture coefficients w_{ie} are subsequently computed using a softmax with routing temperature $T_{\mathrm{route}} > 0$. Degenerate genes are excluded from the prototype fitting and assigned afterward through equal-weight load padding. The resulting offline routing table $\mathcal{R}_{\mathrm{PPI}} = \{(\mathcal{T}_i, \boldsymbol{w}_i)\}_{i=1}^{G}$ remains fixed throughout training and sampling. Each expert is a standard trainable two-layer FFN. Let $\mathbf{h}'_i$ denote the layer-normalized post-attention input and $\mathbf{h}_{i,e} = \mathrm{FFN}_e(\mathbf{h}'_i)$ denote the output of expert e. Consequently, the routing coefficients for non-degenerate genes and the final MoE output for each gene are formulated as follows:

$$w_{ie} = \frac{\exp(\boldsymbol{z}_i^\top \boldsymbol{\mu}_e / T_{\mathrm{route}})}{\sum_{e'\in\mathcal{T}_i} \exp(\boldsymbol{z}_i^\top \boldsymbol{\mu}_{e'} / T_{\mathrm{route}})}, \qquad \mathbf{y}_i = \sum_{e\in\mathcal{T}_i} w_{ie}\mathbf{h}_{i,e} \in \mathbb{R}^d. \tag{7}$$

The details of the model implementation are provided in Appendix E.

3.4 Training Objectives and Sampling

Gene Expression Decoder. Let $\mathbf{H}_t^{\mathrm{out}} \in \mathbb{R}^{G\times d}$ denote the hidden states output by the PI2L backbone. The Gene Expression Decoder employs a shared gene-wise linear readout layer to produce three channels: $[\boldsymbol{\eta}_t \quad \boldsymbol{\mu}_t \quad \boldsymbol{\rho}_t] = \mathcal{F}_{\mathrm{GED}}(\mathbf{H}_t^{\mathrm{out}})$, where $[\boldsymbol{\eta}_t \quad \boldsymbol{\mu}_t \quad \boldsymbol{\rho}_t] \in \mathbb{R}^{G\times 3}$ represent the logits for strictly positive expression events, the location parameter of the untruncated Gaussian, and the unconstrained raw scale. Subsequently, $\boldsymbol{\eta}_t$ and $\boldsymbol{\rho}_t$ are converted into the positive expression probability with $\boldsymbol{\pi}_t = \mathrm{sigmoid}(\boldsymbol{\eta}_t)$ and the positive scale parameter with $\boldsymbol{\sigma}_t = \boldsymbol{\sigma}_{\min} + \mathrm{softplus}(\boldsymbol{\rho}_t)$.

Training Objectives. The conditional distribution $p_\theta(x_0^i \mid \mathbf{x}_t = \boldsymbol{x}_t, t)$ for each gene comprises a point mass at zero with weight $(1-\pi_t^i)$ and a Gaussian distribution $f_+(x_0^i; \mu_t^i, \sigma_t^i)$ truncated to $(0,\infty)$ with weight π_t^i. Let $\delta_i = \mathbb{I}[x_0^i > 0]$ act as an indicator of whether the target gene has a positive observed value. The gene-wise hurdle negative log-density is formulated as: $\ell_t^i = (1-\delta_i)\,\mathrm{softplus}(\eta_t^i) + \delta_i[\mathrm{softplus}(-\eta_t^i) - \log f_+(x_0^i; \mu_t^i, \sigma_t^i)]$. The training objective is defined as:

$$\mathcal{L}_{\mathrm{rec}}(\theta) = \mathbb{E}_{\mathbf{x}_0\sim p_{\mathrm{data}},\, t,\, \mathbf{m}_t}\left[\frac{1}{tG}\sum_{i=1}^{G} \mathrm{m}_t^i \ell_t^i\right]. \tag{8}$$

Sampling. For K sampling steps, we adopt a linear grid defined by $t_k = k/K$, where $k = 0, \ldots, K$. The initial state $\mathbf{x}_{t_K}$ is constructed as a fully masked sequence. During the reverse transition from t_k to t_{k-1}, each currently masked gene is unmasked with a probability of $\zeta_k = 1/k$. For the newly unmasked gene i, its expression value is sampled from the hurdle distribution:

$$\mathrm{x}_{t_{k-1}}^i \mid (\mathbf{x}_{t_k} = \boldsymbol{x}_{t_k}) \sim (1-\pi_{t_k}^i)\delta_0 + \pi_{t_k}^i\, \mathrm{TN}_{(0,\infty)}\big(\mu_{t_k}^i, (\sigma_{t_k}^i)^2\big)\,. \tag{9}$$

The coordinates that have already been unmasked remain fixed throughout the subsequent steps, and the final transition deterministically unmasks all the remaining `MASK` tokens since $\zeta_1 = 1$. The complete training and inference algorithms are provided in Algorithm 2 and Algorithm 3 (Appendix G).

4 EXPERIMENTS

In this section, we comprehensively evaluate the efficiency and quality of PI2L in imputation and unconditional generation tasks. Section 4.1 details the basic experimental setting, implementation details, and the datasets utilized. Section 4.2 investigates the learning efficiency of PI2L compared to Performer, its principal baseline. In Section 4.3, we compare PI2L against three representative baseline models across three tissue-specific atlases. Specifically, we quantitatively assess the similarity between the generated and held-out real cells using five metrics (detailed in Appendix D). In Section 4.4, we perform ablation studies to independently evaluate the contributions of the PPI network itself, as well as the attention and feed-forward layers in PI2L that operate on this network.

4.1 EXPERIMENTAL SETUP

Following bioinformatics protocols (Wolf et al., 2018), the gene expression values for each cell are normalized to a target sum of 10^4 and subsequently transformed by $\log 1\mathrm{p}$. All experiments share identical spherical PPI embeddings and static routing tables, while the specific hyperparameter configurations and related selection experiments are detailed in Appendix A. The backbone of the PI2L model is composed of 6 stacked PI2L blocks. The remaining model configuration details are provided in Table 4. We optimize the models using the AdamW optimizer. The training configurations vary slightly depending on the different tasks (detailed in Table 5). All experiments concerning the PI2L model and its variants are conducted on a single NVIDIA H200 GPU.

Datasets. We evaluate the model performance across three distinct tissue-specific datasets: (i) the PBS control subset of the PARSE 10M PBMC (Biosciences, 2024) dataset (629,701 cells); (ii) a subset of the Heart Global dataset from the Heart Cell Atlas v2 (Kanemaru et al., 2023) (704,296 cells); and (iii) a subset of the HMBA-10xMultiome-BG-Human dataset from the Allen Brain Cell (ABC) Atlas (Johansen et al., 2025) (1,683,105 cells). Comprehensive details regarding the data sources, dataset descriptions, and preprocessing procedures are provided in Appendix H.3.

4.2 IMPUTATION: EFFICIENCY ANALYSIS

We establish a 6-layer Performer (Choromanski et al., 2020) backbone (Yang et al., 2022) as an important baseline and compare its learning efficiency against PI2L through mask imputation experiments on identical held-out cells. We partition the PBMC dataset by allocating 90% of the cells for training and reserving the remaining 10% for imputation validation. Following each training epoch, we mask a fraction of gene expression values in the validation cells and predict these targets utilizing the unmasked gene expression values and all gene identities. The evaluation metric is the time-weighted hurdle negative log-likelihood at the masked positions, normalized by the total number of cells and genes, where lower values indicate better performance. All runs share the identical data split and validation masking conditions, ensuring strict comparability across models and epochs. As shown in Figure 2, PI2L achieves a lower validation imputation loss under an identical optimization update budget. The imputation capability of PI2L at the end of the first training epoch even surpasses that of the Performer after five epochs. The computational cost per block for PI2L and the Performer is detailed in Table 2.

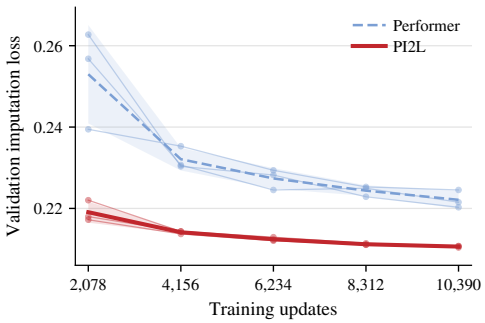

Figure 2: Validation imputation loss of PI2L and Performer on the PBMC dataset.

4.3 UNCONDITIONAL GENERATION

We employ three single-cell generative models—scVI (Lopez et al., 2018), scGAN (Marouf et al., 2020), and scDiffusion (Luo et al., 2024)—as our baselines for unconditional generation. The five models are trained across three distinct datasets, after which 10,000 cells are generated. The benchmarking results are quantified using five complementary metrics: Pearson and Spearman correlation

coefficients evaluate the global expression patterns at the gene level, while Maximum Mean Discrepancy (MMD), gene-averaged 1-Wasserstein distance (1-WD), and Coverage ($k = 5$) assess the distributional similarity and overall generative quality.

As illustrated in Table 1, all evaluated models are capable of achieving high correlations in gene expression. PI2L consistently and significantly outperforms the other models in capturing the per-gene marginal distributions. Regarding the joint distribution, PI2L performs comparably to the baselines, yet it substantially surpasses the Performer, which shares a similar linear attention architecture. Although our proposed model achieves state-of-the-art generative fidelity, this advantage comes at the expense of noticeably diminished generative diversity, manifesting as under-dispersion. This trade-off between generative fidelity and manifold coverage emerges as a systemic issue across all evaluated models. For instance, while scDiffusion attains the broadest coverage across all datasets, its performance on the 1-WD metric lags behind the other models by nearly an order of magnitude.

Figure 3 further characterizes PI2L's generative performance on the HMBA dataset. PI2L achieves the lowest MMD at every evaluated subspace dimension, from the leading principal component (0.055) to all 50 components (0.223), demonstrating a consistent advantage across the evaluated PCA subspaces (a). At these endpoints, PI2L reduces MMD by approximately 62% and 42% relative to the closest baseline, scGAN. Its per-gene distance distribution also shows a larger fraction of genes with small marginal discrepancies, supporting broad improvements in marginal fidelity (b). PI2L's local coverage exceeds that of scVI and scGAN but remains below scDiffusion, indicating a remaining limitation despite its stronger distributional fidelity (c).

Table 1: Quantitative performance comparison on the unconditional generation task. We report five metrics on three datasets across three different seeds (Best: **bolded**, Second best: underline).

Model	Pearson (↑)	Spearman (↑)	MMD (↓)	1-WD (↓)	Coverage (↑)
PARSE 10M PBMC (PBS)					
scVI	0.9964±0.0004	**0.9931**±0.0001	0.4498±0.0133	0.0172±0.0009	0.6889±0.0203
scGAN	0.9960±0.0000	0.9582±0.0013	**0.1911**±0.0074	0.0196±0.0003	0.7341±0.0034
scDiffusion	**0.9984**±0.0001	0.9917±0.0001	0.4292±0.0114	0.1555±0.0009	**0.8924**±0.0061
Performer	0.9966±0.0002	0.9815±0.0016	0.7935±0.0062	0.0127±0.0031	0.1174±0.0044
PI2L	0.9980±0.0002	0.9863±0.0006	0.4320±0.0108	**0.0096**±0.0002	0.4553±0.0348
Heart Global					
scVI	0.9828±0.0017	0.9915±0.0003	0.5439±0.0269	0.0291±0.0011	0.2412±0.0136
scGAN	0.9852±0.0004	0.9601±0.0002	**0.2927**±0.0095	0.0409±0.0011	0.2406±0.0138
scDiffusion	0.9942±0.0015	**0.9940**±0.0004	0.4476±0.0096	0.1533±0.0011	**0.5183**±0.0113
Performer	0.9818±0.0204	0.9840±0.0052	0.9261±0.0777	0.0212±0.0113	0.0115±0.0070
PI2L	**0.9962**±0.0011	0.9909±0.0009	0.2997±0.0139	**0.0116**±0.0009	0.2185±0.0136
HMBA					
scVI	0.9930±0.0004	0.9954±0.0004	0.5827±0.0039	0.0382±0.0008	0.1466±0.0050
scGAN	0.9901±0.0005	0.9685±0.0006	0.3872±0.0091	0.0652±0.0011	0.1684±0.0051
scDiffusion	0.9974±0.0008	**0.9961**±0.0002	0.4702±0.0064	0.1322±0.0016	**0.4449**±0.0051
Performer	0.9965±0.0005	0.9890±0.0003	0.7383±0.0134	0.0160±0.0010	0.0285±0.0025
PI2L	**0.9990**±0.0002	0.9952±0.0004	**0.2232**±0.0101	**0.0101**±0.0004	0.3015±0.0146

4.4 Ablation

Ablation of the PI2L Modules. As illustrated in Table 2, ablating the attention module alone leads to a severe performance degradation compared to the full PI2L model, most notably on the MMD ($+0.3490$) and Coverage (-0.3003) metrics. Conversely, removing solely the MoE module yields a marginal performance improvement. It is worth noting that under the configuration detailed in Table 4, the MoE module requires only half the FLOPs while maintaining a parameter count equivalent to that of the dense FFN in the Performer. From the perspective of the Performer baseline, utilizing the MoE module alone halves the computational cost in the feed-forward layers with virtually no impact on the final performance. Meanwhile, applying the attention module alone substantially improves the MMD (-0.3632), 1-WD (-0.0017), and Coverage ($+0.3323$) metrics. Without the attention module, the manifold coverage of the model deteriorates severely. The primary performance gains of the model stem from the attention weight modulation within the attention module, whereas

the contribution of conditional parameterization predominantly lies in enhancing computational efficiency. The synergistic cooperation between these two modules enables the model to efficiently and accurately capture deep inter-gene relationships and learn high-fidelity biological representations.

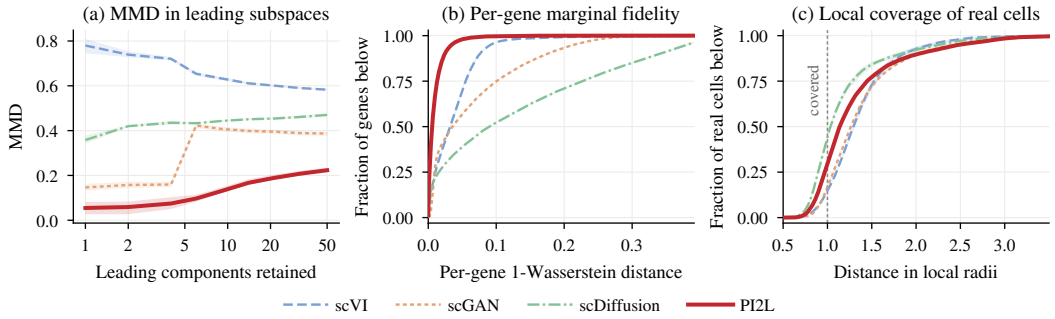

Figure 3: **Distributional fidelity and local coverage on HMBA.** (a) MMD in subspaces spanned by the leading d principal components. The PCA basis and subspace-specific kernel bandwidths are estimated from real cells and shared across models. (b) Cumulative distribution of per-gene 1-Wasserstein distances. (c) Cumulative distribution of nearest-generated-cell distances, normalized by each real cell's fifth-nearest-real-neighbor distance.

Effect of the PPI Network. The bottom two rows of Table 2 illustrate the impact of the PPI network itself on the generation. Given that the contribution of the PPI network is concentrated within the attention module, both variants adopt a standard dense FFN to completely eliminate the confounding effects of the MoE module. Rather than randomly initializing an entirely new PPI network, we permute the embedding rows of different genes. This approach ensures the preservation of the multiset of off-diagonal Gram-matrix entries, the clustering structure, and the heavy tails, leaving only the mapping between the genes and the spherical PPI embedding vectors disrupted. The variant employing the shuffled PPI network suffers a severe performance drop across three metrics: MMD ($+0.2408$), 1-WD ($+0.0227$), and Coverage (-0.2205). It exhibits the most pronounced deterioration in terms of the Pearson and Spearman correlation coefficients. Collectively, these degradations indicate that the shuffled PPI network compromises the accuracy of marginal distributions, revealing a degradation pattern distinctly different from that of ablating the attention module.

Table 2: Ablation study of $\mathrm{Attn}_{\mathrm{PI2L}}$, $\mathrm{MoE}_{\mathrm{PI2L}}$ and the PPI network on the PBMC dataset. Block FLOPs count dominant matrix operations in a single-cell forward pass through one block.

Variant	Pearson (↑)	Spearman (↑)	MMD (↓)	1-WD (↓)	Coverage (↑)	Block FLOPs
PI2L (full)	0.9980	0.9869	0.4445	0.0096	0.4170	$\mathcal{C}_a + \mathcal{C}_\Delta + \frac{1}{2}\mathcal{C}_f$
Performer	0.9979	0.9826	0.7887	0.0104	0.1225	$\mathcal{C}_a + \mathcal{C}_f$
PI2L w/o Attn.	0.9984	0.9831	0.7935	0.0100	0.1167	$\mathcal{C}_a + \frac{1}{2}\mathcal{C}_f$
PI2L w/o MoE	0.9987	0.9870	0.4255	0.0087	0.4548	$\mathcal{C}_a + \mathcal{C}_\Delta + \mathcal{C}_f$
PI2L (shuffled)	0.9695	0.9777	0.6662	0.0314	0.2343	$\mathcal{C}_a + \mathcal{C}_\Delta + \mathcal{C}_f$

$\mathcal{C}_a = 10Gd^2 + 10Gdm_{\mathrm{rf}} + 2Ghm_{\mathrm{rf}}, \quad \mathcal{C}_\Delta = 4Gd^2 + 6Gm_{\mathrm{rf}}r, \quad \mathcal{C}_f = 16Gd^2.$

5 Conclusion

We present PI2L, a single-cell generative framework based on masked discrete diffusion. By dynamically integrating the PPI network conditioned on the corruption level, PI2L mitigates computational bottlenecks in whole-transcriptome discrete diffusion modeling while achieving accurate reconstruction of gene expression profiles. Experiments on public datasets demonstrate that PI2L outperforms existing methods in terms of biological fidelity for unconditional generation tasks, while also exhibiting superior sample efficiency. Overall, PI2L provides a computational tool for in silico single-cell simulation, and supports whole-transcriptome biological analysis.

AI USE STATEMENT

In this work, we acknowledge the use of large language models (LLMs) to assist with text editing and grammatical refinement. Additionally, AI coding agents were utilized to assist with data preprocessing, running baseline experiments, and visualizing experimental results. While the authors implemented and executed the core code, the coding agents assisted in subsequent repetitive tasks, such as running experiments across various datasets and random seeds. We have thoroughly reviewed all AI-assisted work. All conceptualization, experimental design, and result analysis were conducted by the authors, who take full responsibility for the contents of this paper.

ETHICS STATEMENT

In this paper, we propose PI2L, a single-cell generative model aimed at advancing the fields of computational biology and single-cell genomics. Our work focuses on the core features of single-cell transcriptomic data, investigating how to capture complex, high-dimensional cellular information at the whole-transcriptome scale. We acknowledge that training whole-transcriptome models requires substantial computational resources. We aim to establish PI2L as an extensible and continuously updated open-source framework—driven by the ongoing updates to the HGNC and STRING databases—to facilitate its broad application in single-cell analysis. Finally, this research does not raise any ethical concerns, nor does it involve risks of data leakage, the generation of harmful content, or any conflicts of interest.

REPRODUCIBILITY STATEMENT

We declare that all reproducibility details for the work presented in this paper are reported in the main text and the appendices. Specifically, Appendix A details the construction process of the spherical PPI embeddings and the hyperparameter selection methods, which remain applicable even with future updates to the STRING database. Appendix B supplements the handling methods for degenerate genes. Appendix D introduces the evaluation metrics for the unconditional generation task in detail. Appendix E and Appendix F provide a comprehensive description of the model components and their specific parameters. Appendix G presents the algorithms and implementation details for training and sampling with PI2L. Finally, Appendix H summarizes the public datasets and related assets utilized, alongside an introduction to our specialized data processing procedures. We have publicly released the relevant code via an anonymized link.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Ding Bai, Shentong Mo, Ruiyi Zhang, Yingtao Luo, Jiahao Gao, Jeremy Parker Yang, Qiuyang Wu, Hamidreza Rahmani, Tiffany Amariuta, Danielle Grotjahn, et al. sclong: A billion-parameter foundation model for capturing long-range gene context in single-cell transcriptomics. *Nature Communications*, 2026.

Parse Biosciences. Parse 10m pbmc 12donor 90cytokines dataset, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11315–11325, 2022.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Krzysztof Choromanski, Han Lin, Haoxian Chen, Tianyi Zhang, Arijit Sehanobish, Valerii Likhosherstov, Jack Parker-Holder, Tamas Sarlos, Adrian Weller, and Thomas Weingarten. From block-toeplitz matrices to differential equations on graphs: towards a general theory for scalable masked transformers. In *International Conference on Machine Learning*, pp. 3962–3983. PMLR, 2022.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature methods*, 21(8):1470–1480, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and machines*, 30(4):681–694, 2020.

Jing Gong, Minsheng Hao, Xingyi Cheng, Xin Zeng, Chiming Liu, Jianzhu Ma, Xuegong Zhang, Taifeng Wang, and Le Song. xtrimogene: an efficient and scalable representation learner for single-cell rna-seq data. *Advances in Neural Information Processing Systems*, 36:69391–69403, 2023.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature methods*, 21(8):1481–1491, 2024.

Luni Hu, Hua Qin, Yilin Zhang, Yi Lu, Ping Qiu, Zhihan Guo, Lei Cao, Wenjian Jiang, Yixin Shen, Qianqian Chen, et al. Regformer: a single-cell foundation model powered by gene regulatory hierarchies. *Nature Communications*, 2026.

Peng Jiang, Sanju Sinha, Kenneth Aldape, Sridhar Hannenhalli, Cenk Sahinalp, and Eytan Ruppin. Big data in basic and translational cancer research. *Nature Reviews Cancer*, 22(11):625–639, 2022.

Nelson J Johansen, Yuanyuan Fu, Matthew Schmitz, Alma Dubuc, Niklas Kempynck, Morgan Wirthlin, Aaron D Garcia, Madeleine Hewitt, Meghan A Turner, Stephanie C Seeman, et al. Cross-species consensus atlas of the primate basal ganglia. *BioRxiv*, 2025.

Dragomirka Jovic, Xue Liang, Hua Zeng, Lin Lin, Fengping Xu, and Yonglun Luo. Single-cell rna sequencing technologies and applications: a brief overview. *Clinical and translational medicine*, 12(3):e694, 2022.

Kazumasa Kanemaru, James Cranley, Daniele Muraro, Antonio MA Miranda, Siew Yen Ho, Anna Wilbrey-Clark, Jan Patrick Pett, Krzysztof Polanski, Laura Richardson, Monika Litvinukova, et al. Spatially resolved multiomics of human cardiac niches. *Nature*, 619(7971):801–810, 2023.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

May Ke, Badran Elshenawy, Helen Sheldon, Anjali Arora, and Francesca M Buffa. Single cell rna-sequencing: A powerful yet still challenging technology to study cellular heterogeneity. *BioEssays*, 44(11):2200084, 2022.

Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Valentine Svensson, John C Marioni, and Sarah A Teichmann. The technology and biology of single-cell rna sequencing. *Molecular cell*, 58(4): 610–620, 2015.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, 27, 2014.

Jiahao Li, Jiayi Dong, Peng Ye, Xiaochi Zhou, Haohai Lu, and Fei Wang. Save: A generalizable framework for multi-condition single-cell generation with gene block attention. *arXiv preprint arXiv:2604.16776*, 2026.

Yang Li, Guanyu Qiao, Hongli Du, Xin Gao, and Guohua Wang. sckgbert: a knowledge-enhanced foundation model for single-cell transcriptomics. *Genome Biology*, 26(1):402, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Antoine Liutkus, Ondřej Cıfka, Shih-Lun Wu, Umut Simsekli, Yi-Hsuan Yang, and Gael Richard. Relative positional encoding for transformers with linear complexity. In *International Conference on Machine Learning*, pp. 7067–7079. PMLR, 2021.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

Erpai Luo, Minsheng Hao, Lei Wei, and Xuegong Zhang. scdiffusion: conditional generation of high-quality single-cell data using diffusion model. *Bioinformatics*, 40(9):btae518, 2024.

Shengjie Luo, Shanda Li, Tianle Cai, Di He, Dinglan Peng, Shuxin Zheng, Guolin Ke, Liwei Wang, and Tie-Yan Liu. Stable, fast and accurate: Kernelized attention with relative positional encoding. *Advances in Neural Information Processing Systems*, 34:22795–22807, 2021.

Matteo Marouf, Pierre Machart, Vikas Bansal, Christoph Kilian, Daniel S Magruder, Christian F Krebs, and Stefan Bonn. Realistic in silico generation and augmentation of single-cell rna-seq data using generative adversarial networks. *Nature communications*, 11(1):166, 2020.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *Advances in Neural Information Processing Systems*, 38:50608–50646, 2026.

Alessandro Palma, Till Richter, Hanyi Zhang, Manuel Lubetzki, Alexander Tong, Andrea Dittadi, and Fabian Theis. Multi-modal and multi-attribute generation of single cells with cfgen. In *International Conference on Learning Representations*, volume 2025, pp. 6547–6587, 2025.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. Random feature attention. *arXiv preprint arXiv:2103.02143*, 2021.

Zhen Qin, Xiaodong Han, Weixuan Sun, Dongxu Li, Lingpeng Kong, Nick Barnes, and Yiran Zhong. The devil in linear transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 7025–7041, 2022a.

Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. *arXiv preprint arXiv:2202.08791*, 2022b.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 459–467, 2018.

Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, et al. The human cell atlas. *elife*, 6:e27041, 2017.

Isaac Reid, Kumar Dubey, Deepali Jain, William Whitney, Amr Ahmed, Joshua Ainslie, Alex Bewley, Mithun George Jacob, Aranyak Mehta, David Rendleman, et al. Linear transformer topological masking with graph random features. In *International Conference on Learning Representations*, volume 2025, pp. 68271–68295, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Ruth L Seal, Bryony Braschi, Kristian Gray, James McClay, Susan Tweedie, and Elspeth A Bruford. Genenames. org: the hgnc and pgnc resources in 2026. *Nucleic Acids Research*, 54(D1):D1098–D1107, 2026.

Ron Sheinin, Roded Sharan, and Asaf Madi. scnet: learning context-specific gene and cell embeddings by integrating single-cell gene expression data with protein–protein interactions. *Nature Methods*, 22(4):708–716, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Mario L Suvà and Itay Tirosh. Single-cell rna sequencing in cancer: lessons learned and emerging challenges. *Molecular cell*, 75(1):7–12, 2019.

Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. Exponential scaling of single-cell rna-seq in the past decade. *Nature protocols*, 13(4):599–604, 2018.

Damian Szklarczyk, Katerina Nastou, Mikaela Koutrouli, Rebecca Kirsch, Farrokh Mehryary, Radja Hachilif, Dewei Hu, Matteo E Peluso, Qingyao Huang, Tao Fang, et al. The string database in 2025: protein networks with directionality of regulation. *Nucleic acids research*, 53(D1):D730–D737, 2025.

Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Mingxuan Wang, Cheng Chen, Gaoyang Jiang, Zijia Ren, Chuangxin Zhao, Lu Shi, and Yanbiao Ma. Scdiva: Masked discrete diffusion for joint modeling of single-cell identity and expression. *arXiv preprint arXiv:2602.03477*, 2026.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):15, 2018.

Samuel L Wolock, Romain Lopez, and Allon M Klein. Scrublet: computational identification of cell doublets in single-cell transcriptomic data. *Cell systems*, 8(4):281–291, 2019.

Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature machine intelligence*, 4(10):852–866, 2022.

Han Zhang, Guo-Hua Yuan, Chaohao Yuan, Tingyang Xu, Tian Bian, Hong Cheng, Wenbing Huang, Deli Zhao, and Yu Rong. Lingshu-cell: A generative cellular world model for transcriptome modeling toward virtual cells. *arXiv preprint arXiv:2603.25240*, 2026a.

Mingxuan Zhang, Vinay Swamy, Rowan Cassius, Léo Dupire, Charilaos Kanatsoulis, Evan Paull, Mohammed AlQuraishi, Theofanis Karaletsos, and Andrea Califano. Gremln: A cellular graph structure aware transcriptomics foundation model. *bioRxiv*, pp. 2025–07, 2026b.

A SPHERICAL PPI EMBEDDING

A.1 ALGORITHM FORMULATIONS

Confidence-weighted Graph Propagation. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the undirected PPI network, where the protein products correspond one-to-one with the protein-coding genes in the gene vocabulary $\mathcal{V}$. Let c_{ij} represent the combined score between genes i and j. Instead of selecting the confidence threshold τ heuristically, we treat it as a hyperparameter to be optimized. Given the confidence threshold τ, we construct a symmetric weighted adjacency matrix $\boldsymbol{S}$ as follows:

$$S_{ij} = \begin{cases} c_{ij}/1000 & (i,j) \in \mathcal{E},\ c_{ij} \geq \tau \\ 0 & \text{otherwise} \end{cases}, \qquad S_{ii} = 0, \qquad \boldsymbol{S} = \boldsymbol{S}^\top \in \mathbb{R}^{G \times G}. \tag{10}$$

The confidence threshold controls the trade-off between network coverage and evidence reliability, whereas continuous weights preserve confidence differences among the retained edges. Let $\boldsymbol{D} = \mathrm{diag}(d_1, \ldots, d_G)$, where $d_i = \sum_j S_{ij}$. We set $D_{ii}^{-1/2} = 0$ for isolated nodes. We perform symmetric normalization on $\boldsymbol{S}$ to moderate degree-related scale differences, while preserving the operator's symmetry:

$$\hat{\boldsymbol{S}} = \boldsymbol{D}^{-1/2} \boldsymbol{S} \boldsymbol{D}^{-1/2}. \tag{11}$$

Since the spectral radius of the normalized $\hat{\boldsymbol{S}}$ satisfies $\rho(\hat{\boldsymbol{S}}) \leq 1$, its matrix powers are bounded in the spectral norm.

By explicitly aggregating network paths of varying lengths, we construct multi-scale associations with a controllable neighborhood range. Given the diffusion order $K_{\mathrm{diff}} \geq 1$ and the decay coefficient $0 < \alpha < 1$, we define the affinity matrix as follows:

$$\boldsymbol{M'} = \sum_{k=1}^{K_{\mathrm{diff}}} \alpha_k \, \hat{\boldsymbol{S}}^k, \qquad \alpha_k = \frac{(1-\alpha)\alpha^{k-1}}{1 - \alpha^{K_{\mathrm{diff}}}}. \tag{12}$$

The coefficients satisfy $\alpha_k > 0$ and $\sum_k \alpha_k = 1$, ensuring that shorter paths receive greater relative emphasis. The matrix power aggregates weighted walks of length k, thus allowing genes without direct connections to receive a structural affinity. Since return walks generate diagonal mass and self-cooccurrences carry no relational information, we zero out the diagonal of $\boldsymbol{M'}$ to obtain an affinity matrix that captures only inter-gene associations:

$$\boldsymbol{M} = \boldsymbol{M'} - \mathrm{diag}(\boldsymbol{M'}),$$

ensuring that the subsequent node strengths and enrichment baselines are computed based exclusively on inter-gene associations.

Strength Correction and Enrichment. To disentangle pair-specific associations from background associations induced by the overall connectivity scales of the nodes, we define the marginal quantities

$$d_i^M = \sum_j M_{ij}, \qquad \mathrm{vol} = \sum_{ij} M_{ij}. \tag{13}$$

Although symmetric normalization controls the propagation operator, the resulting marginal strength d_i^M can still vary across nodes. We employ

$$\frac{d_i^M d_j^M}{\mathrm{vol}}$$

as a factorized baseline dictated by the marginal strengths of both endpoints. By comparing each affinity M_{ij} with this factorized baseline, the enrichment transformation logarithmically compresses the dynamic range and retains only positive values satisfying

$$M_{ij} > b \, \frac{d_i^M d_j^M}{\mathrm{vol}}.$$

We thus construct the node-strength-corrected logarithmic enrichment matrix:

$$P_{ij} = \begin{cases} \left[\log \dfrac{M_{ij}\mathrm{vol}}{d_i^M d_j^M} - \log b \right]_+, & M_{ij} > 0, \\ 0, & M_{ij} = 0, \end{cases} \tag{14}$$

where $[u]_+ = \max(u, 0)$, and $b \geq 1$ denotes the enrichment threshold. This transformation preserves associations that are sufficiently enriched relative to the baseline determined by the node strengths at both ends. Increasing b imposes a more stringent threshold for retaining these associations.

Spectral Embedding and Spherical Normalization. The symmetric matrix $\boldsymbol{P}$ is not necessarily positive semidefinite, whereas the Gram matrix of real Euclidean embeddings must be positive semidefinite. We retain the top r positive eigenvalues and their eigenvectors:

$$\boldsymbol{Z}^{\mathrm{raw}} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r^{1/2}, \qquad \boldsymbol{Z}^{\mathrm{raw}} (\boldsymbol{Z}^{\mathrm{raw}})^\top = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{U}_r^\top \approx \boldsymbol{P}, \tag{15}$$

where the diagonal elements of $\boldsymbol{\Sigma}_r = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ are all strictly positive, and r does not exceed the number of positive eigenvalues of $\boldsymbol{P}$. This construction yields a low-dimensional positive semidefinite approximation suitable for inner product computations. We identify the set of degenerate genes as follows:

$$\mathcal{V}_{\mathrm{deg}} = \{ i : \|\boldsymbol{z}_i^{\mathrm{raw}}\|_2 < \epsilon \cdot \mathrm{median}_j \|\boldsymbol{z}_j^{\mathrm{raw}}\|_2 \}. \tag{16}$$

Subsequently, we compute the mean over the remaining rows:

$$\bar{\boldsymbol{z}} = \frac{\sum_{i \notin \mathcal{V}_{\mathrm{deg}}} \boldsymbol{z}_i^{\mathrm{raw}}}{|\mathcal{V} \setminus \mathcal{V}_{\mathrm{deg}}|}. \tag{17}$$

The raw embeddings of the non-degenerate genes are centered and subsequently normalized onto the unit sphere; the raw embeddings of degenerate genes are approximately zero vectors, leading to ill-conditioned normalization, whence alternative vectors are uniformly sampled from the unit sphere:

$$\boldsymbol{z}_i = \begin{cases} \dfrac{\boldsymbol{z}_i^{\mathrm{raw}} - \bar{\boldsymbol{z}}}{\|\boldsymbol{z}_i^{\mathrm{raw}} - \bar{\boldsymbol{z}}\|_2} & i \notin \mathcal{V}_{\mathrm{deg}} \\ \sim \mathrm{Uniform}(S^{r-1}) & i \in \mathcal{V}_{\mathrm{deg}}. \end{cases} \tag{18}$$

The indices of degenerate genes are preserved alongside the embeddings to distinguish these fallback representations from those possessing reliable spectral information. Centering expresses each coordinate relative to the centroid of the retained genes, and unit normalization removes embedding magnitude from similarity comparisons. Since centering and normalization modify the raw Gram matrix, the final inner products should be interpreted as derived angular similarities rather than an exact reconstruction of P_{ij}. After processing, all vectors $\boldsymbol{z}_i$ lie on the unit sphere with bounded inner products, satisfying $\|\boldsymbol{z}_i\|_2 = 1$ and $\boldsymbol{z}_i^\top \boldsymbol{z}_j \in [-1, 1]$.

Algorithm 1 Spherical PPI Embedding

Input: PPI network $\mathcal{G}$ with combined scores c_{ij}; confidence threshold τ, diffusion order K_{diff}, enrichment threshold b, PPI embedding rank r

1: $S_{ij} \leftarrow c_{ij}/1000$ for $c_{ij} \geq \tau$, else 0; symmetrize; zero the diagonal
2: $\hat{\boldsymbol{S}} \leftarrow \boldsymbol{D}^{-1/2} \boldsymbol{S} \boldsymbol{D}^{-1/2}$, with $D_{ii}^{-1/2} = 0$ for edgeless nodes
3: $\boldsymbol{M}' \leftarrow \sum_{k=1}^{K_{\mathrm{diff}}} \alpha_k \hat{\boldsymbol{S}}^k$ with $\sum_k \alpha_k = 1$; $\boldsymbol{M} \leftarrow \boldsymbol{M}' - \mathrm{diag}(\boldsymbol{M}')$
4: divide by the factorized baseline $d_i^M d_j^M / \mathrm{vol}$, take the logarithm, threshold at $\log b \to \boldsymbol{P}$
5: $\boldsymbol{U}_r, \boldsymbol{\Sigma}_r \leftarrow$ top-r positive eigenpairs of $\boldsymbol{P}$; $\boldsymbol{Z}^{\mathrm{raw}} \leftarrow \boldsymbol{U}_r \boldsymbol{\Sigma}_r^{1/2}$
6: $\mathcal{V}_{\mathrm{deg}} \leftarrow \{ i : \|\boldsymbol{z}_i^{\mathrm{raw}}\|_2 < \epsilon \cdot \mathrm{median}_j \|\boldsymbol{z}_j^{\mathrm{raw}}\|_2 \}$
7: $\bar{\boldsymbol{z}} \leftarrow |\mathcal{V} \setminus \mathcal{V}_{\mathrm{deg}}|^{-1} \sum_{i \notin \mathcal{V}_{\mathrm{deg}}} \boldsymbol{z}_i^{\mathrm{raw}}$
8: $\boldsymbol{z}_i \leftarrow (\boldsymbol{z}_i^{\mathrm{raw}} - \bar{\boldsymbol{z}}) / \|\boldsymbol{z}_i^{\mathrm{raw}} - \bar{\boldsymbol{z}}\|_2$ for $i \notin \mathcal{V}_{\mathrm{deg}}$
9: $\boldsymbol{z}_i \sim \mathrm{Uniform}(S^{r-1})$ for $i \in \mathcal{V}_{\mathrm{deg}}$

Output: $\boldsymbol{Z} \in \mathbb{R}^{G \times r}$ with unit-norm rows; degenerate gene set $\mathcal{V}_{\mathrm{deg}}$

A.2 Hyperparameter Selection

To construct the spherical PPI embeddings $\boldsymbol{Z}$, we select four hyperparameters: (1) Confidence threshold τ: Governs the reliability of the raw evidence and the resulting graph coverage. (2) Diffusion order K_{diff}: Determines the effective radius of the structural context. (3) Enrichment threshold

b: Sets the relative enrichment threshold for diffusion cooccurrences. (4) PPI embedding rank r: Controls the resolution of the resulting spherical angular geometry. Our hyperparameter selection proceeds in two stages: we first sweep over candidate combinations of $(\tau, K_{\text{diff}}, b)$ to identify the target PPI geometry that we wish to preserve, then choose a suitable rank r that stably represents that geometry.

We define a unified reference set of high-confidence edges as $\mathcal{E}_{700} = \{(i,j) \in \mathcal{E} : c_{ij} \geq 700\}$. We randomly sample 10% of the undirected edges from this set to form the validation edges $\mathcal{E}_{\text{val}} \subset \mathcal{E}_{700}$. To prevent any gene from losing its only high-confidence connection due to random sampling, we restrict the sampling process to ensure that $d_i^{\mathcal{E}_{700} \setminus \mathcal{E}_{\text{val}}} \geq 1$. These validation edges are then uniformly partitioned into graph construction validation edges and rank validation edges, yielding $\mathcal{E}_{\text{val}} = \mathcal{E}_{\text{val}}^{\theta} \dot{\cup} \mathcal{E}_{\text{val}}^{r}$, subject to the mutual exclusivity constraint $\mathcal{E}_{\text{val}}^{\theta} \cap \mathcal{E}_{\text{val}}^{r} = \varnothing$.

Stage 1: $(\tau, K_{\text{diff}}, b)$ We define the candidate search space as: $\tau \in \{400, 500, 600, 700, 800, 900\}$, $K_{\text{diff}} \in \{1, 2, 3, 4, 5, 6, 7, 8\}$, and $b \in \{1, 2, 5\}$, yielding a total of 144 candidate configurations $\Theta = \{(\tau, K_{\text{diff}}, b)\}$. The design of this candidate space is motivated by several considerations. The STRING (Szklarczyk et al., 2025) database categorizes scores of 400, 700, and 900 as medium, high, and highest confidence, noting that a threshold of 700 is conventionally adopted for conservative downstream analyses, while 400 prioritizes broader graph coverage. Furthermore, given a fixed decay factor of $\alpha = 0.5$, the incremental weights contributed by paths longer than four hops become negligible, while the corresponding matrix undergoes rapid densification. Finally, the shift values $b = 1, 2, 5$ correspond to no additional shift, mild truncation, and aggressive truncation.

In Stage 1, it is imperative to exclude both sets of validation edges simultaneously; thus, the set of edges used for graph construction is defined as $\mathcal{E}_{\text{train}}^{\theta} = \mathcal{E} \setminus (\mathcal{E}_{\text{val}}^{\theta} \cup \mathcal{E}_{\text{val}}^{r})$. For each gene i, we define its corresponding validation gene set as $\mathcal{V}_{i,\text{val}}^{\theta} = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}_{\text{val}}^{\theta}\}$, along with a fixed evaluation gene set $\mathcal{V}_{i,\text{eval}}^{\theta} = \mathcal{V} \setminus (\{i\} \cup \{j : \{i,j\} \in \mathcal{E}_{700} \setminus \mathcal{E}_{\text{val}}^{\theta}\})$. All genes incident to at least one validation edge constitute the set $\mathcal{V}_{\text{val}}^{\theta} = \{i \in \mathcal{V} : |\mathcal{V}_{i,\text{val}}^{\theta}| > 0\}$. Given a candidate configuration $\theta = (\tau, K_{\text{diff}}, b)$, we compute the complete matrix $\boldsymbol{P}_{\theta}$ and define

$$q_{\theta}(j \mid i) = \frac{\exp(P_{\theta,ij})}{\sum_{u \in \mathcal{V}_{i,\text{eval}}^{\theta}} \exp(P_{\theta,iu})}, \qquad j \in \mathcal{V}_{i,\text{eval}}^{\theta}. \tag{19}$$

The final evaluation metric is formulated as follows:

$$\mathcal{J}(\theta) = \frac{1}{|\mathcal{V}_{\text{val}}^{\theta}|} \sum_{i \in \mathcal{V}_{\text{val}}^{\theta}} \frac{1}{|\mathcal{V}_{i,\text{val}}^{\theta}|} \sum_{j \in \mathcal{V}_{i,\text{val}}^{\theta}} \log\left(|\mathcal{V}_{i,\text{eval}}^{\theta}| q_{\theta}(j \mid i)\right). \tag{20}$$

This metric measures the additional probability mass that $\boldsymbol{P}_{\theta}$ allocates to the true but masked high-confidence PPI neighbors, as opposed to a uniform random guess across the candidate genes. The metric $\mathcal{J}$ simultaneously provides a unified evaluation for all three hyperparameters:

- If τ is excessively low, low-confidence edges may diffuse probability mass to irrelevant genes; conversely, if it is overly high, the structural context necessary for recovering validation edges might be inadvertently filtered out.
- If K_{diff} is too small, the endpoints of a validation edge may fail to connect via the residual network, whereas an excessively large K_{diff} risks dispersing the probability mass across too many genes.
- A small value of b retains numerous weakly enriched relationships, while an overly large b truncates connections that are crucial for recovering the validation edges.

Consequently, maximizing $\mathcal{J}$ successfully identifies a candidate configuration that effectively reconstructs the high-confidence validation relationships. It should be noted that since this metric explicitly sets the recovery of missing high-confidence edges as its objective, it inherently introduces a structural bias towards $K_{\text{diff}} \geq 2$. Figure 4 presents a $\tau \times K_{\text{diff}}$ heatmap faceted by b, where the cell values correspond to $\mathcal{J}$. Table 3 reports the values of $\mathcal{J}$ and $\exp(\mathcal{J})$ for the optimal candidate, along with six suboptimal configurations and their relative differences Δ_{θ} compared to the optimum. We ultimately select $\theta^{\star} = (700, 4, 1)$, as it does not exhibit a significant performance gap compared to the best-performing candidate at $K_{\text{diff}} = 5$.

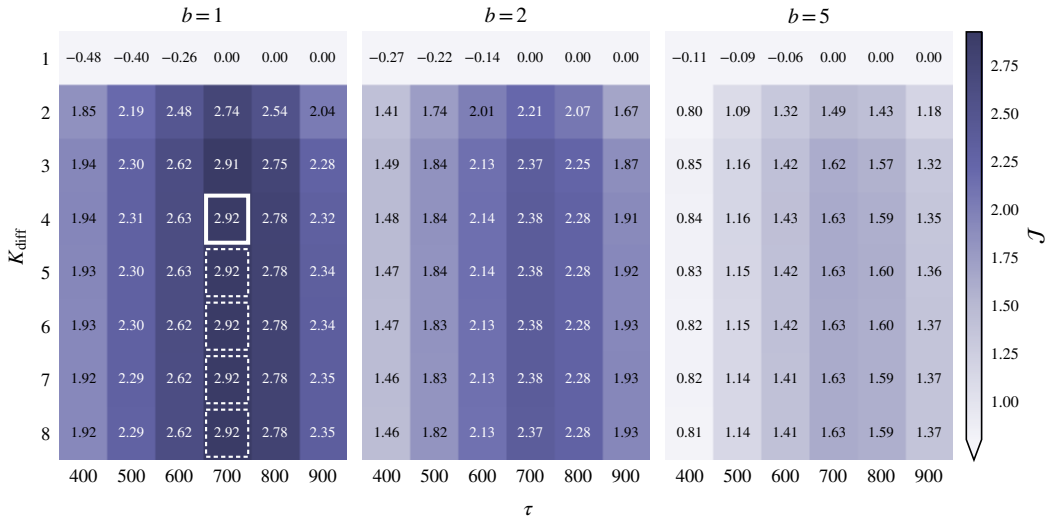

Figure 4: Grid search over τ, K_{diff}, b (144 configurations). Colors share a common scale across panels. The solid box marks the selected configuration; dashed boxes mark alternative high-scoring configurations.

Table 3: Top-7 configurations from the grid search in Figure 4. Δ_θ is measured relative to the best configuration.

τ	K_{diff}	b	$\mathcal{J}$	$\exp(\mathcal{J})$	Δ_θ
700	5	1	2.923	18.594	–
700	4	1	2.923	18.590	0.000
700	6	1	2.919	18.530	−0.003
700	7	1	2.917	18.487	−0.006
700	8	1	2.915	18.458	−0.007
700	3	1	2.909	18.346	−0.013
800	6	1	2.784	16.178	−0.139

Stage 2: Suitable r Upon fixing $\theta^\star$, we restore the graph construction validation edges $\mathcal{E}^\theta_{\text{val}}$ and subsequently remove the rank validation edges $\mathcal{E}^r_{\text{val}}$, yielding $\mathcal{E}^r_{\text{train}} = \mathcal{E} \setminus \mathcal{E}^r_{\text{val}}$. We compute the top $R_{\text{scan}} = 256$ algebraically largest eigenpairs of $\boldsymbol{P}_\theta$ for the optimal candidate and three suboptimal configurations. For each candidate, we evaluate the following quantities per eigenpair: (1) Raw eigenvalue: σ_j. (2) Normalized eigenvalue: $\widetilde{\sigma}_j = \sigma_j/\sigma_1$. (3) LOBPCG residual: $\rho_j = \|\boldsymbol{P}_\theta \boldsymbol{u}_j - \sigma_j \boldsymbol{u}_j\|_2$. (4) Sign determination tolerance: $\delta_j = \max(10\rho_j, 10^{-6}\sigma_1)$. (5) Status: classified as positive, unresolved, or negative based on the conditions $\sigma_j > \delta_j$, $|\sigma_j| \le \delta_j$, and $\sigma_j < -\delta_j$. For each candidate, we identify the terminus of the contiguous reliable positive spectrum, defined as $r_+ = \max\{r : \sigma_j > \delta_j,\ \forall j \le r\}$, as well as the index of the first reliable negative eigenvalue, given by $r_- = \min\{j : \sigma_j < -\delta_j\}$. If no reliable negative eigenvalue is detected within the scanning range, we record this as $r_- > R_{\text{scan}}$; similarly, if $r_+ = R_{\text{scan}}$, it is also treated as right-censored.

Figure 5 plots the leading normalized positive spectra for the four candidates. This figure solely demonstrates the spectral feasibility of candidate ranks and indicates the presence or absence of a natural spectral truncation, but does not dictate the final choice of r. Observations from the leading normalized positive spectra reveal that the four curves exhibit nearly identical shapes, essentially overlapping with one another. This suggests that the spectral properties in Stage 2 are robust to minor variations in the Stage 1 candidates, thereby validating the soundness of our two-stage hyperparameter selection strategy. Furthermore, we observe that the positive spectrum is substantially long; although the leading eigenvalues dominate, the subsequent eigenvalues decay at a slow rate. Such a flat spectrum is an expected behavior, indicating that the higher-order directions still encode

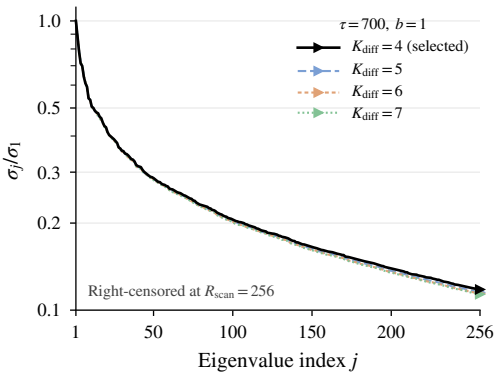

Figure 5: Leading normalized positive spectra of the four candidates, plotted on a logarithmic vertical axis. $\theta^\star$ is depicted with a thick solid line, while the remaining three configurations are shown using lightly colored dashed lines.

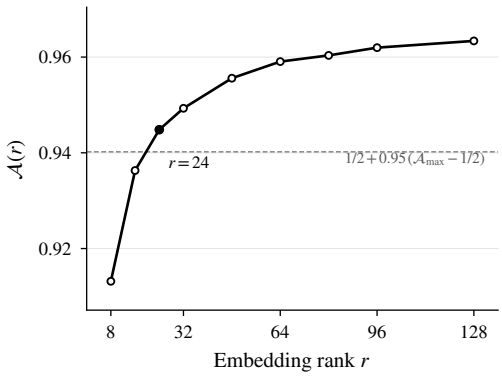

Figure 6: Grid search over r. A larger r generally yields stronger ranking capabilities. The dashed line marks 95% of the improvement over chance achieved at $r = 128$, the best-performing rank within the search range.

significant structural information, which is an inherent characteristic dictated by the high-rank nature of PPI networks. However, a flat spectrum primarily implies energy dispersion, whereas the prior only necessitates the principal ranking information. The tail modes, despite carrying energy, convey negligible signals for edge prediction. They largely encode local gene-specific structures, offering minimal discriminative power to predict whether two given genes interact. Ultimately, we must select an appropriate r based on the concentration of the predictive signal.

For the candidate $\theta^\star$, we evaluate the rank within the range $r \in \{8, 16, 24, 32, 48, 64, 80, 96, 128\}$. For each evaluated r, it is imperative to independently re-execute the deflation, de-meaning, and spherical normalization procedures; one cannot simply truncate the already normalized matrix $\boldsymbol{Z}_{R_{\text{scan}}}$. We define the spherical similarity as $s_{ij} = (\boldsymbol{z}_i)^\top \boldsymbol{z}_j$. For each gene i, its corresponding validation gene set is defined as $\mathcal{V}^r_{i,\text{val}} = \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}^r_{\text{val}}\}$, alongside a fixed evaluation gene set $\mathcal{V}^r_{i,\text{eval}} = \mathcal{V} \setminus (\{i\} \cup \{j : \{i, j\} \in \mathcal{E}_{700} \setminus \mathcal{E}^r_{\text{val}}\})$. The background comparison set is therefore formulated as $\mathcal{V}^r_{i,\text{bg}} = \mathcal{V}^r_{i,\text{eval}} \setminus \mathcal{V}^r_{i,\text{val}}$. All genes incident to at least one validation edge constitute the set $\mathcal{V}^r_{\text{val}} = \{i \in \mathcal{V} : |\mathcal{V}^r_{i,\text{val}}| > 0\}$. We employ the macro-averaged pairwise concordance to quantify the probability that a masked high-confidence interacting partner is ranked above other background targets for the same gene:

$$\mathcal{A}(r) = \frac{1}{|\mathcal{V}^r_{\text{val}}|} \sum_{i \in \mathcal{V}^r_{\text{val}}} \frac{1}{|\mathcal{V}^r_{i,\text{val}}||\mathcal{V}^r_{i,\text{bg}}|} \sum_{j \in \mathcal{V}^r_{i,\text{val}}} \sum_{u \in \mathcal{V}^r_{i,\text{bg}}} \left[\mathbb{I}(s_{ij} > s_{iu}) + \frac{1}{2} \mathbb{I}(s_{ij} = s_{iu}) \right]. \tag{21}$$

Figure 6 illustrates $\mathcal{A}(r)$ for the candidate $\theta^\star$ under various low-rank embedding configurations. At $r = 24$, the macro-averaged pairwise concordance is 94.48%, increasing to 95.91% at $r = 64$. Consequently, we conclude that the primary global interaction ranking signals are largely preserved at $r = 24$, after which performance exhibits diminishing marginal returns. Subsequent spectral directions contribute predominantly to the progressive refinement of local relationships that are hard to distinguish, a process that nearly saturates beyond $r = 64$.

Notably, under the assumption of approximate isotropy, the variances satisfy

$$\text{Var}(\mathbf{q}_i^\top \mathbf{k}_j) \approx \frac{\sigma_q^2 \sigma_k^2}{d_h}, \qquad \text{Var}(\sigma_q \sigma_k \boldsymbol{z}_i^\top \boldsymbol{z}_j) \approx \frac{\sigma_q^2 \sigma_k^2}{r}.$$

Hence, the ratio of the background standard deviation of the PPI logits to that of the content logits is approximately $\sqrt{d_h/r}$. Therefore, when r is comparable to d_h, the two geometries naturally possess approximately matched fluctuation scales prior to the λ-interpolation. Given the actual configuration of our model architecture, we ultimately select $r^\star = 64$.

B DEGENERATE GENES

During the offline construction of spherical PPI embeddings in Section 3, we identify and retain the set of degenerate genes, denoted as $\mathcal{V}_{\text{deg}}$. The remaining genes constitute the set of non-degenerate genes, defined as $\mathcal{V}_{\text{nondeg}} = \mathcal{V} \setminus \mathcal{V}_{\text{deg}}$. Empirical results in Appendix A demonstrate that the number of degenerate genes depends almost exclusively on the confidence threshold τ. This indicates that the emergence of degenerate genes is due to their inherent lack of high-confidence structural information within the PPI network, rather than an erroneous loss of information induced by our algorithm. Although we temporarily assign random unit-norm fallback vectors to these degenerate genes, such vectors would inadvertently introduce non-existent PPI structural information if left without explicit treatment.

In Equation (3), zero vectors are padded into the corresponding r-dimensional coordinate blocks for degenerate genes, ensuring that all augmented vectors reside within the same space:

$$\widetilde{\mathbf{q}}_i = \begin{bmatrix} \sqrt{1-\lambda}\,\mathbf{q}_i \\ \mathbf{0}_r \end{bmatrix} \in \mathbb{R}^{d_h+r}, \qquad \widetilde{\mathbf{k}}_j = \begin{bmatrix} \sqrt{1-\lambda}\,\mathbf{k}_j \\ \mathbf{0}_r \end{bmatrix} \in \mathbb{R}^{d_h+r}.$$

The target logit between a degenerate gene and any arbitrary gene, approximated by random features, is defined as $\widetilde{\mathbf{q}}_i^\top \widetilde{\mathbf{k}}_j = (1-\lambda)\mathbf{q}_i^\top \mathbf{k}_j$. Since the prior term for degenerate genes evaluates to zero, their actual augmented norm differs from that of non-degenerate genes. On the query side, we have

$$\|\widetilde{\mathbf{q}}_i\|_2^2 = \begin{cases} (1-\lambda)\|\mathbf{q}_i\|_2^2 + \lambda\sigma_q^2, & i \in \mathcal{V}_{\text{nondeg}}, \\ (1-\lambda)\|\mathbf{q}_i\|_2^2, & i \in \mathcal{V}_{\text{deg}}. \end{cases}$$

Symmetrically, the key side satisfies

$$\|\widetilde{\mathbf{k}}_j\|_2^2 = \begin{cases} (1-\lambda)\|\mathbf{k}_j\|_2^2 + \lambda\sigma_k^2, & j \in \mathcal{V}_{\text{nondeg}}, \\ (1-\lambda)\|\mathbf{k}_j\|_2^2, & j \in \mathcal{V}_{\text{deg}}. \end{cases}$$

Consequently, an additional term $\frac{1}{2}\lambda\sigma_q^2$ must be introduced to correct the norm of the common unit sphere template for degenerate genes, and the same principle applies to the key side. The coordinate of the s-th query log-feature is formulated as:

$$a_{Q,i,s} = \begin{cases} \boldsymbol{\omega}_s^\top \widetilde{\mathbf{q}}_i - \dfrac{1}{2}\left[(1-\lambda)\|\mathbf{q}_i\|_2^2 + \lambda\sigma_q^2\right], & i \in \mathcal{V}_{\text{nondeg}}, \\ \boldsymbol{\omega}_s^\top \widetilde{\mathbf{q}}_i - \dfrac{1}{2}\left[(1-\lambda)\|\mathbf{q}_i\|_2^2 + \lambda\sigma_q^2\right] + \dfrac{1}{2}\lambda\sigma_q^2, & i \in \mathcal{V}_{\text{deg}}, \end{cases} \qquad s = 1, \ldots, m_{\text{rf}}.$$

The coordinate of the s-th key log-feature is formulated as:

$$a_{K,j,s} = \begin{cases} \boldsymbol{\omega}_s^\top \widetilde{\mathbf{k}}_j - \dfrac{1}{2}\left[(1-\lambda)\|\mathbf{k}_j\|_2^2 + \lambda\sigma_k^2\right], & j \in \mathcal{V}_{\text{nondeg}}, \\ \boldsymbol{\omega}_s^\top \widetilde{\mathbf{k}}_j - \dfrac{1}{2}\left[(1-\lambda)\|\mathbf{k}_j\|_2^2 + \lambda\sigma_k^2\right] + \dfrac{1}{2}\lambda\sigma_k^2, & j \in \mathcal{V}_{\text{deg}}, \end{cases} \qquad s = 1, \ldots, m_{\text{rf}}.$$

C EXPLANATION

Upon concatenating the content and prior terms, the target logit approximated by random features is formulated as

$$\widetilde{\mathbf{q}}_i^\top \widetilde{\mathbf{k}}_j = (1-\lambda)\mathbf{q}_i^\top \mathbf{k}_j + \lambda\sigma_q\sigma_k \boldsymbol{z}_i^\top \boldsymbol{z}_j.$$

Since each projection row of the random features follows a standard Gaussian marginal distribution, the unbiasedness property

$$\mathbb{E}_{\boldsymbol{\Omega}}\left[\boldsymbol{\phi}_Q(\widetilde{\mathbf{q}}_i)^\top \boldsymbol{\phi}_K(\widetilde{\mathbf{k}}_j)\right] = \exp(\widetilde{\mathbf{q}}_i^\top \widetilde{\mathbf{k}}_j)$$

strictly holds for any arbitrary $\widetilde{\mathbf{q}}_i, \widetilde{\mathbf{k}}_j \in \mathbb{R}^{d_h+r}$. We denote the exponential dot-product kernel as $\kappa_{ij} = \exp(\widetilde{\mathbf{q}}_i^\top \widetilde{\mathbf{k}}_j)$. By further expanding the target logit, we obtain

$$\kappa_{ij} = \exp((1-\lambda)\mathbf{q}_i^\top \mathbf{k}_j)\exp(\lambda\sigma_q\sigma_k \boldsymbol{z}_i^\top \boldsymbol{z}_j) = (\kappa_{ij}^{\text{content}})^{1-\lambda}(\kappa_{ij}^{\text{PPI}})^{\lambda},$$

where $\kappa_{ij}^{\text{content}}$ and κ_{ij}^{PPI} represent the content kernel and the PPI spherical kernel. Given that $\boldsymbol{z}_i^\top \boldsymbol{z}_j = 1 - 1/2\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2$, it follows that

$$\kappa_{ij} \propto \exp\left((1-\lambda)\mathbf{q}_i^\top \mathbf{k}_j\right) \exp\left(-\beta_{\text{PPI}}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2\right), \tag{22}$$

where $\beta_{\text{PPI}} = \lambda\sigma_q\sigma_k/2$ serves as the effective inverse bandwidth for the PPI distance. The term $\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2$ corresponds to the chordal distance on the unit sphere, which characterizes the structural contextual discrepancy derived from offline PPI representations, rather than the shortest path or direct edge confidence in the original PPI graph. It should be noted that when $\lambda = 0$, the target exponential kernel reduces to the content kernel $\exp(\mathbf{q}_i^\top \mathbf{k}_j)$. However, this does not automatically guarantee that the output of finite orthogonal random features will exactly reproduce the original Performer element-wise, as the augmented projections still reside in the $d_h + r$ dimensional space. In summary, $\text{Attn}_{\text{PI2L}}$ interpolates between the content geometry and the PPI spherical geometry within the logit space, thereby modulating the information aggregation among genes. Furthermore, the special treatment for degenerate genes detailed in Appendix B dictates that the prior term remains uninformative when degenerate genes participate in pairing.

D Evaluation Metrics

We adopt some of the unconditional generation metrics used in Lingshu-Cell (Zhang et al., 2026a) to evaluate gene-level expression fidelity and distributional similarity between real and generated cells. Let $\boldsymbol{X}_r \in \mathbb{R}^{N_r \times G}$ and $\boldsymbol{X}_g \in \mathbb{R}^{N_g \times G}$ denote the gene expression matrices of N_r real cells and N_g generated cells. Each cell encompasses exactly the G protein-coding genes. All metrics are computed on log-normalized gene expression values.

D.1 Gene Expression Correlation

Pearson Correlation. We compute the mean expression value of each gene across all cells for both the real and generated datasets, yielding two mean gene expression vectors $\bar{\boldsymbol{x}}_r, \bar{\boldsymbol{x}}_g \in \mathbb{R}^G$. The Pearson correlation between these two vectors is then calculated as $r_{\text{Pearson}} = corr(\bar{\boldsymbol{x}}_r, \bar{\boldsymbol{x}}_g)$. By evaluating the linear consistency of mean gene expressions, this metric reflects whether the model successfully captures the global trends of genes with characteristically high or low expression. However, it is worth noting that the Pearson correlation is highly sensitive to extreme values.

Spearman Correlation. We calculate the rank-based Spearman correlation for the same two mean gene expression vectors: $r_{\text{Spearman}} = \rho_{\text{rank}}(\bar{\boldsymbol{x}}_r, \bar{\boldsymbol{x}}_g)$. This metric measures the monotonic consistency of mean expression levels. Because it relies exclusively on the relative rankings of gene expression, the Spearman correlation exhibits greater robustness to outliers, serving as a crucial complement to the Pearson correlation.

Together, the Pearson and Spearman correlations assess the model's capacity to reconstruct overall gene-level expression patterns by comparing the average expression of individual genes between the real and generated cell populations.

D.2 Distribution Fidelity

Maximum Mean Discrepancy (MMD). We first apply Principal Component Analysis (PCA) to jointly reduce the dimensionality of both the real cells $\boldsymbol{x}_r$ and the generated cells $\boldsymbol{x}_g$. The data is subsequently mapped into a Reproducing Kernel Hilbert Space (RKHS) using a multi-scale Gaussian kernel function $k(\cdot,\cdot)$. Specifically, we first calculate the mean squared pairwise distance, which serves as the base bandwidth σ_0. A bandwidth hierarchy is then constructed as

$$\sigma_s = \sigma_0 \cdot \alpha^{s-\lceil S/2 \rceil}, \quad s = 1, \ldots, 5, \quad S = 5, \quad \alpha = 2.$$

The multi-scale Gaussian kernel is thus defined as

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{s=1}^{5} \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{\sigma_s}\right).$$

The distance is computed by comparing the kernel similarities both within and across the two sets of data:

$$\mathrm{MMD}^2 = \frac{1}{N_r^2}\sum_{i,i'} k(\boldsymbol{x}_{r,i}, \boldsymbol{x}_{r,i'}) - \frac{2}{N_r N_g}\sum_{i,j} k(\boldsymbol{x}_{r,i}, \boldsymbol{x}_{g,j}) + \frac{1}{N_g^2}\sum_{j,j'} k(\boldsymbol{x}_{g,j}, \boldsymbol{x}_{g,j'}). \tag{23}$$

By comparing the mean embeddings within the RKHS, MMD effectively quantifies the distance between the real and generated cells across their global multivariate feature distributions.

Gene-averaged 1-Wasserstein Distance (1-WD). For each gene g, we independently compute the 1-Wasserstein distance between its 1D marginal distribution in the real cells, $P_r^{(g)}$, and that in the generated cells, $P_g^{(g)}$, which corresponds to the integral of the absolute difference between their respective cumulative distribution functions. These distances are then averaged across all G genes:

$$\text{1-WD} = \frac{1}{G}\sum_{g=1}^{G} W_1(P_r^{(g)}, P_g^{(g)}) \tag{24}$$

The 1-WD metric evaluates the comprehensive distributional profile at the individual gene level. In contrast to the Pearson and Spearman correlations that solely rely on mean expressions, 1-WD is acutely sensitive to the overall shape of the gene expression distributions.

Coverage. Coverage is defined as the fraction of real cells that contain at least one generated cell within their local neighborhoods:

$$\text{Coverage} = \frac{1}{N_r}\sum_{i=1}^{N_r} \mathbb{1}\Big[\ \exists j:\ \|\boldsymbol{x}_{r,i} - \boldsymbol{x}_{g,j}\| < \mathrm{NND}_k(\boldsymbol{x}_{r,i})\ \Big], \tag{25}$$

where $\mathrm{NND}_k(\boldsymbol{x}_{r,i})$ denotes the distance from the real cell i to its k-th nearest real neighbor. This metric essentially quantifies mode dropping, assessing whether the model exclusively generates the most abundant cell types while completely ignoring rare cell populations. Bounded within $[0, 1]$, the coverage score directly indicates the proportion of the true data manifold that is successfully covered.

By treating each individual cell as a discrete point in a high-dimensional space, the MMD, 1-WD, and Coverage metrics collectively evaluate whether the generated cell population faithfully recapitulates the geometric distribution of the real cells across the multidimensional space.

E Model Architecture and Implementation Details

Sampling a Boolean Mask. During training, we independently sample a diffusion time $\mathrm{t} \sim \mathcal{U}((0, 1))$ for each data sample. Given the sampled diffusion time t, we draw an independent Boolean mask $\mathrm{m}_t^i \mid \mathrm{t} = t \overset{\text{ind.}}{\sim} \mathrm{Bernoulli}(t)$ for each of the G genes within that sample. Consequently, the random mask vector for each sample is formulated as $\mathbf{m}_t = [\mathrm{m}_t^1, \ldots, \mathrm{m}_t^G]^\top$.

Gene Identity Encoder. Since we model the entire set of protein-coding genes, the same gene identity sequence is applied across all samples. This eliminates the need to explicitly feed the sequence and perform the corresponding table-lookup operations during each forward pass. Essentially, the Gene Identity Encoder functions by fine-tuning the semantic gene relationships—learned by Geneformer (Theodoris et al., 2023) from a massive human single-cell transcriptome corpus—to a specific human tissue via a projection matrix $\boldsymbol{W}_{\mathrm{proj}}$. The procedure for extracting the static embedding layer $\boldsymbol{E}_{\mathrm{GF}} \in \mathbb{R}^{|\mathcal{V}| \times d_{\mathrm{GF}}}$ from Geneformer is detailed in Appendix H.5. Here, $\boldsymbol{W}_{\mathrm{proj}} \in \mathbb{R}^{d_{\mathrm{GF}} \times d}$ denotes a bias-free linear projection layer initialized with a deterministic orthogonal weight matrix.

Gene Expression Encoder. Specifically, the two-layer MLP in the Gene Expression Encoder is formulated as follows:

$$\mathcal{F}_{\mathrm{GEE}}(\mathbf{e}) = \mathrm{SiLU}(\mathbf{e}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2,$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{1\times 32}$, $\boldsymbol{b}_1 \in \mathbb{R}^{32}$, $\boldsymbol{W}_2 \in \mathbb{R}^{32\times d}$, and $\boldsymbol{b}_2 \in \mathbb{R}^d$.

Residual Structure. Specifically, the Pre-LN residual structure of the PI2L block at the l-th layer is formulated as follows:

$$\begin{aligned} \mathbf{U} &= \mathbf{H}_t^{(l-1)} + \mathrm{Attn}_{\mathrm{PI2L}}\big(\mathrm{LN}(\mathbf{H}_t^{(l-1)})\big), \\ \mathbf{H}_t^{(l)} &= \mathbf{U} + \mathrm{MoE}_{\mathrm{PI2L}}\big(\mathrm{LN}(\mathbf{U})\big). \end{aligned} \tag{26}$$

Dropout is applied to the output of each sublayer and omitted for clarity. For the whole backbone, we have $\mathbf{H}_t^{\text{in}} = \mathbf{H}_t^{(0)}$ and $\mathbf{H}_t^{\text{out}} = \mathbf{H}_t^{(L)}$.

Corruption Conditioned Gate. Specifically, the Fourier-MLP of the Corruption Conditioned Gate is given by:

$$\boldsymbol{\lambda}(p_t) = \text{sigmoid}\left(\boldsymbol{W}_2\,\text{SiLU}(\boldsymbol{W}_1\boldsymbol{\gamma}(p_t) + \boldsymbol{b}_1) + \boldsymbol{b}_2\right) \in (0,1)^h,$$

where

$$\boldsymbol{\gamma}(p_t) = \left[\sin(2^0\pi p_t), \cos(2^0\pi p_t), \ldots, \sin(2^{h-1}\pi p_t), \cos(2^{h-1}\pi p_t)\right] \in \mathbb{R}^{2h}$$

denotes the Fourier features, with $\boldsymbol{W}_1 \in \mathbb{R}^{64\times 2h}$, $\boldsymbol{b}_1 \in \mathbb{R}^{64}$, $\boldsymbol{W}_2 \in \mathbb{R}^{h\times 64}$, and $\boldsymbol{b}_2 \in \mathbb{R}^h$.

F HYPERPARAMETERS

Table 4 and Table 5 list hyperparameters selected for efficiency.

Table 4: Model configuration. Symbols follow the notation introduced in Section 3.

Parameter	Symbol	Value
Number of genes	G	19295
Vocabulary size	$\lvert\mathcal{V}\rvert$	19295
Hidden dimension	d	512
Layers	L	6
Gene Identity Encoder		
Geneformer dimension	d_{GF}	1152
Gene Expression Encoder		
$\mathcal{F}_{\text{GEE}}$ hidden dimension	—	32
Activation	—	SiLU
PI2L		
Attention heads	h	8
Head dimension	d_h	64
Fourier-MLP dimension	—	64
PPI embedding rank	r	64
Number of random features	m_{rf}	256
Feedforward dimension	$4d$	2048
Number of experts	E_{exp}	4
Experts per token	k_{route}	2
Expert hidden dimension	d_{exp}	512
Routing temperature	T_{route}	0.03
Activation	—	GELU

Table 5: Training configuration.

Parameter	IMPUTATION	UNCONDITIONAL GENERATION
Optimizer	AdamW	
Learning rate	2×10^{-4}	1×10^{-4}
Global batch size	256	64
Weight decay	0.01	0.05
Dropout	0.0	0.1
Warmup ratio	0.05	
Epochs	5	10

G ALGORITHMS

Let $\mathcal{F}_\theta(\mathbf{x}, \mathbf{m}, p)$ denote the PI2L predictor, including the Gene Identity Encoder, Gene Expression Encoder, PI2L backbone, and Gene Expression Decoder. Here, $p = G^{-1}\sum_{i=1}^{G} \mathrm{m}^i$ controls the

layer-specific, head-wise PPI prior gates. Gene identity embeddings are computed as $\boldsymbol{E}_{\mathrm{GF}}\boldsymbol{P}$, where $\boldsymbol{E}_{\mathrm{GF}}$ is frozen and $\boldsymbol{W}_{\mathrm{proj}}$ is trainable. The spherical PPI embeddings $\boldsymbol{Z}$ and static routing table $\mathcal{R}_{\mathrm{PPI}}$ are fixed throughout training and sampling.

The predictor returns $[\boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{\rho}]$, with $\boldsymbol{\pi} = \mathrm{sigmoid}(\boldsymbol{\eta})$ and $\boldsymbol{\sigma} = \boldsymbol{\sigma}_{\min} + \mathrm{softplus}(\boldsymbol{\rho})$, where $\boldsymbol{\sigma}_{\min} = 10^{-3}$. We write the gene-wise hurdle negative log-likelihood as

$$\ell(\mathbf{x}; \eta, \mu, \sigma) = \begin{cases} \mathrm{softplus}(\eta), & x = 0, \\ \mathrm{softplus}(-\eta) - \log f_+(x; \mu, \sigma), & x > 0, \end{cases} \tag{27}$$

where

$$f_+(x; \mu, \sigma) = \frac{\phi\big((x-\mu)/\sigma\big)}{\sigma\, \Phi(\mu/\sigma)}, \qquad x > 0. \tag{28}$$

ϕ and Φ denote the standard probability density function (PDF) and the cumulative distribution function (CDF) of the standard Gaussian, respectively. When employing a linear time grid for sampling, the probability of revealing a currently masked gene is given by:

$$\zeta_k = 1 - \frac{t_{k-1}}{t_k} = \frac{1}{k}.$$

Consequently, if gene i remains masked at time step t_k, its exact posterior distribution is formulated as:

$$q(\mathrm{x}^i_{t_{k-1}} \mid \mathrm{x}^i_{t_k} = \texttt{MASK}, \mathrm{x}^i_0) = \frac{k-1}{k}\delta_{\texttt{MASK}}(\mathrm{x}^i_{t_{k-1}}) + \frac{1}{k}\delta_{\mathrm{x}^i_0}(\mathrm{x}^i_{t_{k-1}}). \tag{29}$$

To adapt to unconditional generation, PI2L employs a 50% self-conditioning rate and increases the sampling frequency of heavily masked samples. On 50% of optimizer steps (ramped linearly over the first 5% of steps), we draw $t' \sim \mathcal{U}(t, 1)$ per cell, hide each visible gene with probability $(t'-t)/(1-t)$, and replace it by a hurdle sample from a gradient-free pass at t'. Targets and masked positions stay real, so only the context changes; the extra pass costs $\approx 15\%$ training time. Each cell draws $t \sim \mathcal{U}[0.9, 1)$ with probability 0.25 and $t \sim \mathcal{U}[0, 1)$ otherwise, raising that interval's share of training cells from 10% to 32.5%. Per-gene $1/t$ weights are unchanged and validation keeps uniform t, so validation NLL stays comparable across runs. For genes that are originally absent from the dataset, we pad their values with 0 in Appendix H.2. During sampling, the expression values of these genes are set deterministically to 0 rather than `MASK`. Based on our tests, we found that setting the number of sampling steps to 100 can balance the generation quality and sampling efficiency.

H Datasets and Assets

H.1 Gene Vocabulary

We utilize the HUGO Gene Nomenclature Committee (HGNC) (Seal et al., 2026) database, accessible via `https://www.genenames.org/`, as the sole authoritative source for our gene vocabulary. Given the dynamic nature of this database, we note that the version used for constructing our vocabulary was last updated on June 5, 2026. The *Gene entry status* attribute indicates the current validity of a gene entry within the HGNC database. We specifically filter for the `Approved` status to exclusively retain official gene symbols that are validated and made publicly available by HGNC. Furthermore, the *Gene locus type* characterizes the locus category of each gene entry, which primarily encompasses the following three types.

- `Protein-coding genes`: genes that encode proteins.
- `Non-coding RNA`: genes that do not encode proteins but produce functional RNA molecules.
- `Pseudogene`: DNA sequences that structurally resemble protein-coding genes but typically lack the ability to encode functional proteins.

Ultimately, our constructed unified gene vocabulary comprises $19,295$ approved protein-coding genes.

Algorithm 2 PI2L Training

Input: Data distribution p_{data}, fixed Geneformer embeddings E_{GF}, spherical PPI embeddings $\boldsymbol{Z}$, and static routing table $\mathcal{R}_{\text{PPI}}$

Initialize trainable parameters θ (AdamW optimizer), including $\boldsymbol{W}_{\text{proj}}$ and $\mathbf{h}_{\text{MASK}}$
while not converged **do**
  Sample cells $\mathbf{x}_0 \sim p_{\text{data}}$ from the dataset
  Sample diffusion time $t \sim \mathcal{U}((0,1))$
  For each gene i, independently sample mask $\mathrm{m}_t^i \sim \text{Bernoulli}(t)$ for $i = 1, \ldots, G$
  Construct the corrupted gene expression token:

$$\mathrm{x}_t^i \leftarrow \begin{cases} \texttt{MASK}, & \mathrm{m}_t^i = 1, \\ \mathrm{x}_0^i, & \mathrm{m}_t^i = 0 \end{cases} \qquad i = 1, \ldots, G$$

  $p_t \leftarrow G^{-1} \sum_{i=1}^{G} \mathrm{m}_t^i$ ▷ Realized mask fraction
  Forward pass: obtain hidden states from the PI2L backbone
  Predict hurdle distribution $(\boldsymbol{\eta}_t, \boldsymbol{\mu}_t, \boldsymbol{\rho}_t)$ for all positions
  $\boldsymbol{\pi}_t \leftarrow \text{sigmoid}(\boldsymbol{\eta}_t)$; $\boldsymbol{\sigma}_t \leftarrow \sigma_{\min} + \text{softplus}(\boldsymbol{\rho}_t)$
  for all i such that $\mathrm{m}_t^i = 1$ **do**
    $\ell_t^i \leftarrow \ell(\mathrm{x}_0^i; \eta_t^i, \mu_t^i, \sigma_t^i)$
  end for
  $\widehat{\mathcal{L}}_{\text{rec}} \leftarrow \dfrac{1}{tG} \displaystyle\sum_{i:\, \mathrm{m}_t^i = 1} \ell_t^i$ ▷ An empty sum is zero
  Update parameters θ with AdamW using $\nabla_\theta \widehat{\mathcal{L}}_{\text{rec}}$
end while

Output: Trained parameters θ

Algorithm 3 PI2L Sampling

Input: Trained PI2L predictor $\mathcal{F}_\theta$; number of sampling steps $K \geq 1$, spherical PPI embeddings $\boldsymbol{Z}$, and static routing table $\mathcal{R}_{\text{PPI}}$

Define diffusion schedule $t_k \leftarrow k/K$ for $k = 0, \ldots, K$
Initialize $\mathbf{x}_1 \leftarrow (\texttt{MASK}, \ldots, \texttt{MASK})$ (fully masked); $\mathbf{m}_1 \leftarrow \mathbf{1}$
for $k = K, K-1, \ldots, 1$ **do**
  $p_{t_k} \leftarrow G^{-1} \sum_{i=1}^{G} \mathrm{m}_{t_k}^i$
  Run mask predictor to obtain $(\boldsymbol{\eta}_{t_k}, \boldsymbol{\mu}_{t_k}, \boldsymbol{\rho}_{t_k}) \leftarrow \mathcal{F}_\theta(\mathbf{x}_{t_k}, \mathbf{m}_{t_k}, p_{t_k})$
  $\boldsymbol{\pi}_{t_k} \leftarrow \text{sigmoid}(\boldsymbol{\eta}_{t_k})$; $\boldsymbol{\sigma}_{t_k} \leftarrow \sigma_{\min} + \text{softplus}(\boldsymbol{\rho}_{t_k})$
  $\zeta_k \leftarrow 1 - t_{k-1}/t_k = 1/k$
  $\mathbf{x}_{t_{k-1}} \leftarrow \mathbf{x}_{t_k}$; $\mathbf{m}_{t_{k-1}} \leftarrow \mathbf{m}_{t_k}$ ▷ Preserve existing values
  for all i such that $\mathrm{m}_{t_k}^i = 1$, in parallel **do**
    Sample $u_i \sim \text{Bernoulli}(\zeta_k)$
    if $u_i = 1$ **then**
      Sample $b_i \sim \text{Bernoulli}(\pi_{t_k}^i)$
      if $b_i = 0$ **then**
        $\mathbf{x}_{t_{k-1}}^i \leftarrow 0$
      else
        Sample $\mathbf{x}_{t_{k-1}}^i \sim \text{TN}_{(0,\infty)}(\mu_{t_k}^i, (\sigma_{t_k}^i)^2)$
      end if
      $\mathrm{m}_{t_{k-1}}^i \leftarrow 0$
    end if
  end for
end for

Output: Synthetic expression vector $\mathbf{x}_0 \in \mathbb{R}_{\geq 0}^G$ ▷ All remaining masks are revealed at $k = 1$

H.2 Alignment

The unified gene vocabulary utilizes the gene symbol as the unique identifier for each gene, serving as the standard to which all datasets, PPI networks, and Geneformer must be aligned. The alignment

procedure follows a standardized pipeline across all these resources. Initially, we prioritize exact matches of gene symbols, isolating and separately recording the unmatched genes from both the gene vocabulary and the target datasets or assets. For the unmatched genes in our vocabulary, we query their symbol reports in the HGNC database to extract their *previous symbols*, *alias symbols*, and *Ensembl IDs*. A secondary matching phase is then conducted to align the remaining unmatched genes from the datasets or assets against these extracted *previous symbols*, *alias symbols*, and *Ensembl IDs*. Finally, any genes matched via *alias symbols* or those resulting in multiple candidate alignments are subjected to a manual review to resolve ambiguities and verify their accuracy.

H.3 DATASETS

We conducted unconditional generation experiments using three datasets: PBMC, Heart, and HMBA. These datasets were obtained from the following sources:

- PBMC: `https://www.parsebiosciences.com/datasets/10-million-human-pbmcs-in-a-single-experiment/`
- Heart: `https://www.heartcellatlas.org/`
- HMBA: `https://allen-brain-cell-atlas.s3.us-west-2.amazonaws.com/index.html#expression_matrices/HMBA-10xMultiome-BG/20250630/`

Quality Control. To filter out low-quality cells, damaged cells, suspected doublets, and empty droplets, we implement the following quality control (QC) pipeline. Initially, we retain cells expressing a minimum of 200 genes. We establish lower and upper thresholds for total UMI counts at 1,000 and 100,000, retaining only the cells that fall within this specified range. Cells with sub-threshold UMI counts are excluded as they likely represent sequencing errors or empty droplets, whereas those surpassing the upper limit are discarded as potential doublets or multiplets. Additionally, we retain cells where the proportion of mitochondrial transcripts does not exceed 20%. To further identify and discard suspected doublets, we employ Scrublet (Wolock et al., 2019) with the expected doublet rate set to 0.06, 30 principal components, and a simulated doublet ratio of 2.0.

Log-normalization. In the unprocessed scRNA count matrix, the raw counts are nonnegative integers representing the copy numbers of RNA transcripts. We normalize these raw counts using a size factor of $10,000$ per cell, followed by a $\log1\mathrm{p}$ transformation.

H.4 PROTEIN-PROTEIN INTERACTION NETWORK

We obtained the protein-protein interaction (PPI) network from STRING (version 12.0) (Szklarczyk et al., 2025), which is available at: `https://string-db.org/`. We specify the target organism as *Homo sapiens* (taxonomic identifier 9606). Instead of empirically filtering based on the *combined confidence score*, we retain all interactions and systematically sweep the confidence threshold as a hyperparameter, as detailed in Appendix A.

The rationale for utilizing the PPI network is that it captures species-level information, thereby rendering the PPI-guided PI2L framework universally applicable across all human scRNA-seq datasets. Although gene regulatory networks (GRNs) can provide more direct and robust relative gene positions, they inherently serve as an incomplete prior.

H.5 GENEFORMER

Geneformer is a foundation model pre-trained on a large-scale corpus of human single-cell transcriptomes spanning a diverse array of tissues. We initialize the embedding layer within the Gene Identity Encoder using Geneformer-V2-316M, which is publicly available at `https://huggingface.co/ctheodoris/Geneformer`. Compared to Geneformer-V1-10M, the pre-training corpus of Geneformer-V2-316M has been expanded from approximately 30 million to 104 million human single-cell transcriptomes. Furthermore, the vocabulary of Geneformer-V2-316M consists of only about $20,000$ protein-coding genes, exhibiting a high degree of overlap with our constructed gene vocabulary. The `Token_ID` of Geneformer-V2-316M is strictly utilized for initialization purposes;

subsequent training and inference consistently rely on the `Index` of our gene vocabulary. Within the weights of Geneformer-V2-316M, the source embedding table is denoted as $\boldsymbol{E}_{\mathrm{GF}}^{\mathrm{src}} \in \mathbb{R}^{20275 \times d_{\mathrm{GF}}}$, where $d_{\mathrm{GF}} = 1152$. We extract the corresponding weights from this table based on the mapping relationship $\mathbf{E}_{\mathrm{GF}}[\mathrm{Index}] = \boldsymbol{E}_{\mathrm{GF}}^{\mathrm{src}}[\mathrm{Token_ID}]$. For the 59 unmapped genes, we uniformly sample vectors with replacement from the $19{,}236$ mapped gene embeddings. Subsequently, a dimension-wise perturbation $\epsilon_j \sim \mathcal{N}\left(0, (0.01\,\sigma_j)^2\right)$ is introduced, where σ_j represents the empirical standard deviation of the j-th dimension across the mapped vectors. To ensure that our results are fully auditable and reproducible, we record the sampled source rows, the random seed, and the missing-row mask. The aforementioned initialization process is executed offline; we save the resulting weights $\boldsymbol{E}_{\mathrm{GF}} \in \mathbb{R}^{|\mathcal{V}| \times d_{\mathrm{GF}}}$ in `safetensors` format, alongside a manifest that documents the Geneformer checkpoint revision, the mapping file hash, the tensor shape, the 59 missing indices, the random seed, and the perturbation parameters. During training and inference, only this lightweight initialization asset is loaded, thereby circumventing the overhead of loading the entire Geneformer-V2-316M model.