GeneMamba: Efficient and Effective Foundation Model on Single Cell Data

Cong Qi, Hanzhang Fang, Siqi Jiang, Xun Song, Tianxing Hu, Zhi Wei
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ, USA
zhi.wei@njit.edu

***Abstract*—Transformer-based models have achieved strong performance in single-cell transcriptomics, yet scaling them to genome-wide gene sequences remains challenging due to the quadratic cost of self-attention. Recent state space models (SSMs) offer linear-time sequence modeling, but unidirectional designs may introduce ordering biases and overlook biological structure, potentially limiting robustness across datasets. To address these challenges, we propose GeneMamba, a bidirectional state space model tailored for single-cell RNA sequencing. GeneMamba integrates three key components: (1) Efficiency: BiMamba blocks enable bidirectional processing with O(L) complexity, providing substantial speedup over transformer baselines while maintaining linear scaling; (2) Biological integration: a pathway-aware contrastive objective incorporates curated pathway information to guide representation learning; (3) Scalability: large-scale pretraining on approximately 30 million curated cells (from ˜50 million raw CELLXGENE profiles) supports transfer across multiple downstream tasks without task-specific architectures. Across benchmarks including multi-batch integration, cell type annotation, and gene rank reconstruction, GeneMamba demonstrates competitive performance relative to strong baselines such as scGPT and Geneformer, with improvements observed in several settings. Ablation studies further suggest that bidirectional modeling and pathway-aware pretraining contribute to these gains. Overall, GeneMamba provides an efficient and biologically informed framework for foundation modeling in single-cell transcriptomics.**

***Index Terms*—Single-cell RNA-seq, State space models, Foundation models, Gene expression analysis, Computational biology**

I. Introduction

Single-cell RNA sequencing (scRNA-seq) has fundamentally transformed our ability to study cellular heterogeneity by enabling high-resolution profiling of gene expression at the individual cell level Korsunsky et al. [2019], Theodoris et al. [2023]. This technology allows researchers to dissect complex cellular compositions, trace differentiation trajectories, and identify rare cell populations, catalyzing significant advancements in developmental biology, disease modeling, and drug discovery Shen et al. [2023], Yang et al. [2024].

To address the computational challenges posed by high-dimensional, sparse, and noisy scRNA-seq data Du et al. [2019], Zhao et al. [2023], transformer-based architectures have emerged as the mainstream approach. Models such as scBERT Yang et al. [2022] and scGPT Cui et al. [2024] have demonstrated strong performance in tasks such as cell type classification and representation learning for downstream analysis Bian et al. [2024], Wen et al. [2023], establishing themselves as state-of-the-art solutions for capturing biologically meaningful patterns.

However, transformers face critical scalability limitations. The quadratic complexity of their self-attention mechanism—requiring $O(n^2)$ memory and computation for n genes—becomes prohibitive when processing transcriptomes with tens of thousands of genes Vaswani [2017], Dao et al. [2022]. Linear-complexity state space models (SSMs) address computational efficiency through $O(n)$ architectures, but unidirectional designs process genes in a fixed order, imposing a sequential inductive bias that conflicts with the largely order-agnostic nature of gene co-expression.

This structural mismatch stems from a fundamental representation challenge. Gene–gene relationships are graph-structured, where co-expression patterns exhibit permutation invariance—the mutual information between genes is independent of their ordering. Yet practical constraints necessitate linearizing genes into sequences. Transformers achieve permutation equivariance but at quadratic cost; unidirectional SSMs achieve linear complexity but encode sequential dependencies through recurrent state transitions. Neither approach fully reconciles biological structure with computational scalability.

Another challenge lies in how prior biological knowledge is incorporated during pretraining. Existing approaches, including pathway-based objectives, often rely on curated annotations that define functional relationships at a relatively coarse or binary level. While such supervision can guide representation learning, it may also limit flexibility in capturing more nuanced or hierarchical biological organization. Recent work in other domains has explored multi-granular learning paradigms that model relationships across different levels of abstraction Li et al. [2025], suggesting a promising direction for integrating structured knowledge without enforcing rigid constraints Wang et al. [2025], Qi et al. [2026].

To address these challenges, we introduce **GeneMamba**, a bidirectional state space model that reconciles computational efficiency with biological structure. GeneMamba processes gene sequences in both forward and backward directions with symmetric aggregation, preserving $O(n)$ complexity while approximating the permutation equivariance needed for co-expression modeling. Additionally, we incorporate a pathway-

aware contrastive learning objective that leverages curated pathway relationships to guide representation learning beyond purely data-driven patterns, while retaining flexibility for downstream adaptation.

Importantly, we decouple *bidirectional representations* from *causal supervision*: next-gene prediction is trained using prefix-only forward representations (to prevent future-token leakage), while the reverse stream is used to enrich embeddings for downstream tasks and pathway-aware contrastive regularization.

We validate GeneMamba across multi-batch integration, cell type annotation, and gene rank reconstruction, demonstrating strong computational efficiency and competitive predictive performance compared to transformer-based models and unidirectional SSMs Hao et al. [2024]. Systematic ablation studies suggest that both bidirectional modeling and pathway-aware pretraining contribute to these improvements. Our contributions are threefold:

1. We propose a bidirectional SSM architecture with symmetric aggregation that mitigates sequential ordering bias, achieving approximate permutation equivariance for gene co-expression modeling.

2. We incorporate a pathway-based contrastive learning objective that encourages similarity among functionally related genes, enabling the model to leverage curated biological knowledge during pretraining.

3. We demonstrate through extensive experiments that GeneMamba achieves more than 3× computational speedup over transformer baselines while maintaining competitive performance across evaluated tasks.

The methodological novelty of GeneMamba lies in (i) applying shared-weight bidirectional scanning with gated fusion to ranked gene sequences to mitigate ordering bias at genome-scale lengths, and (ii) integrating pathway-aware contrastive regularization as a lightweight pretraining objective for single-cell foundation modeling.

The method's code is now available at https://github.com/MineSelf2016/GeneMamba.
The HuggingFace usage guide is at https://huggingface.co/mineself2016/GeneMamba.

II. Methods

A. Data Processing

Our input is not a plain token sequence: each cell contains gene identities and their expression values. We represent the data as a gene expression matrix $M \in \mathbb{R}^{c\times g}$, where c is the number of cells and g is the number of genes (unexpressed genes have zero counts).

We first preprocess the single-cell data and normalize for sequencing depth and gene-specific variation. For each cell i, we divide M_{ij} by the library size $\sum_k M_{ik}$, then scale each gene j by its non-zero median across cells (estimated efficiently via t-digest). The normalized expression is:

$$M_{ij}^{\text{norm}} = \frac{M_{ij}/\sum_{k=1}^{n} M_{ik}}{\text{t-digest}\{M_{kj} \mid M_{kj} > 0\}}$$

We then rank genes within each cell by descending M_{ij}^{norm}, which emphasizes cell-state-specific genes while down-weighting ubiquitous housekeeping genes:

$$R_i = \text{argsort}\left(-M_{ij}^{\text{norm}}\right), \quad \forall j$$

B. Bi-Mamba Architecture

Selective State Space models. State Space Models (SSMs) model sequences with a latent state updated from the previous state and current input:

$$h_t = Ah_{t-1} + Bx_t, \quad y_t = Ch_t.$$

Equivalently, the same computation can be written as a 1D convolution with kernel

$$y = x * K, \quad K = [CB,\ CAB,\ CA^2B, \dots].$$

Classic SSMs use fixed parameters A, B, C. Mamba replaces fixed dynamics with *input-dependent* (selective) dynamics by predicting parameters from the current token representation x_t (e.g., step size and input/output projections), enabling selective information propagation while preserving linear-time scaling in the sequence length.

The input to the l-th block is a length-L sequence

$$S^{(l)} = [\mathbf{x}_1^{(l)}, \mathbf{x}_2^{(l)}, \dots, \mathbf{x}_L^{(l)}], \quad \mathbf{x}_t^{(l)} \in \mathbb{R}^d.$$

We denote the forward SSM state by h_t and the reverse-direction state by $\hat{h}_t$, the fusion gate by z_t, and the combined output by o_t.

Bidirectional Mamba with shared weights. Unidirectional SSMs can introduce spurious ordering effects when genes are linearized into a sequence. To mitigate this, we construct a reversed sequence

$$\overleftarrow{S}^{(l)} = [\mathbf{x}_L^{(l)}, \mathbf{x}_{L-1}^{(l)}, \dots, \mathbf{x}_1^{(l)}],$$

where padding tokens (if used) are kept fixed to avoid alignment artifacts. We then process $S^{(l)}$ and $\overleftarrow{S}^{(l)}$ with the *same* BiMamba block (shared Conv and SSM parameters, as illustrated by the shared-weight arrows in Fig. 1). Concretely, we first apply a shared local mixing layer (Conv) and then a shared selective SSM in both directions:

$$\mathbf{u}_t^{(l)} = \text{Conv}(\mathbf{x}_t^{(l)};\, \theta_{\text{conv}}), \quad \overleftarrow{u}_t^{(l)} = \text{Conv}(\overleftarrow{x}_t^{(l)};\, \theta_{\text{conv}}),$$
$$h_t^{(l)} = \text{SSM}(\mathbf{u}_t^{(l)}, h_{t-1}^{(l)};\, \theta_{\text{ssm}}), \quad \hat{h}_t^{(l)} = \text{SSM}(\overleftarrow{u}_t^{(l)}, \hat{h}_{t-1}^{(l)};\, \theta_{\text{ssm}}).$$

After running the reverse stream, we map $\hat{h}_t^{(l)}$ back to the original index order so that both streams align at position t.

Gated fusion and output projection. We fuse forward and reverse states using a learnable gate (Fig. 1):

$$z_t^{(l)} = \sigma\big(W_z^{(l)}[h_t^{(l)};\, \hat{h}_t^{(l)}] + b_z^{(l)}\big),$$
$$o_t^{(l)} = z_t^{(l)} \odot h_t^{(l)} + (1 - z_t^{(l)}) \odot \hat{h}_t^{(l)},$$

where $[\cdot;\cdot]$ denotes concatenation, σ is the sigmoid, and $\odot$ is element-wise multiplication. Finally, we apply a nonlinearity and a linear projection to obtain the next-layer token representation:

$$\mathbf{x}_t^{(l+1)} = \mathbf{x}_t^{(l)} + W_o^{(l)}\, \phi\big(o_t^{(l)}\big).$$

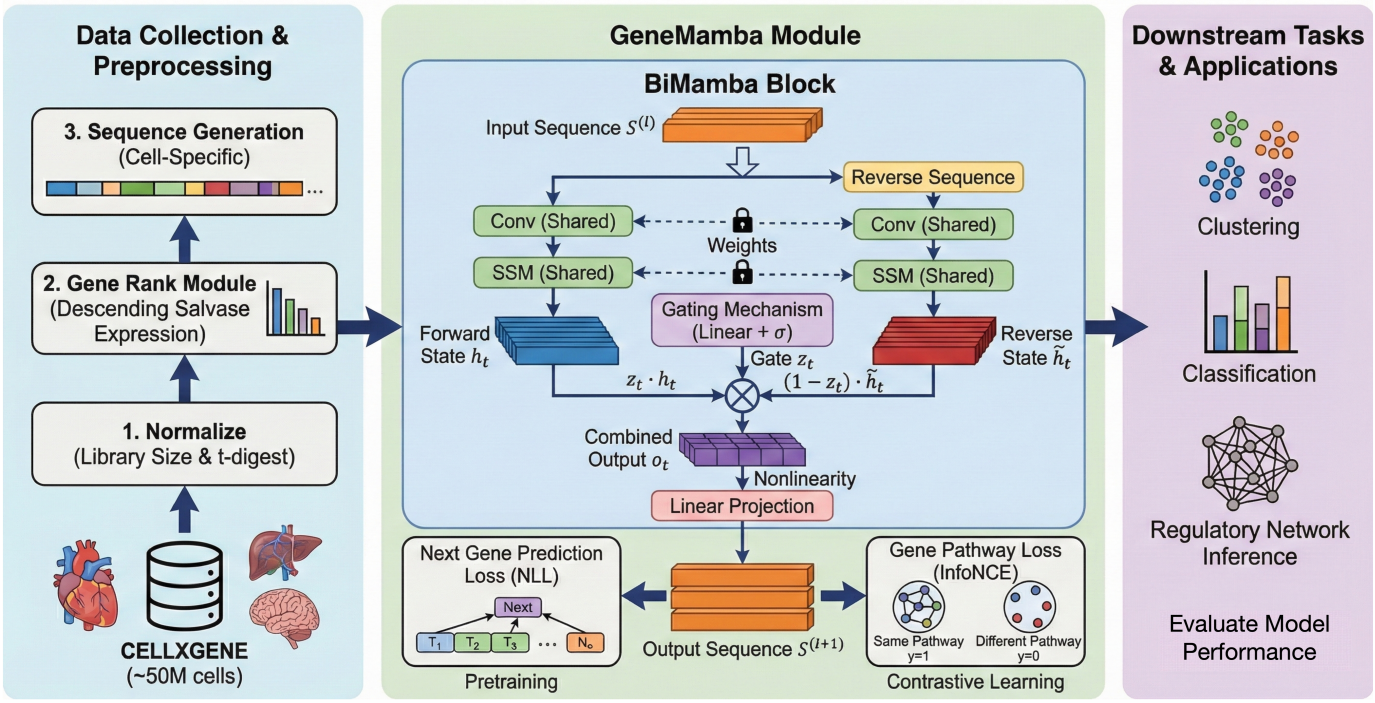

Fig. 1. **The GeneMamba architecture and its downstream task applications.** The framework begins with the collection of training data (˜50M raw cells) from CELLXGENE, encompassing a diverse array of tissues and organs. After preprocessing and quality control, the curated corpus contains ˜30M cells used for pretraining. The data is prepared through a Gene Rank Module to transform single-cell data into input sequences. The GeneMamba module then captures the contextual information within each single cell. Once pretrained, the model and its embeddings are employed for various downstream tasks to evaluate the model's performance.

Stacking N BiMamba blocks yields the recurrence

$$S^{(l+1)} = \mathrm{BiMamba}(S^{(l)}), \quad l = 0, \ldots, N-1.$$

Complexity. Because each block performs a constant amount of work per token in each direction (shared Conv + selective SSM scan), BiMamba scales linearly with sequence length L while providing bidirectional context, which is critical for modeling gene co-expression patterns that are largely order-agnostic.

C. Pretraining Objective

Next Gene Prediction Loss We use next-gene prediction as a lightweight self-supervised signal over the ranked gene sequence within each cell. A potential concern is that our encoder is bidirectional (forward + reverse + gated fusion), while next-token prediction is typically defined under a causal (prefix-only) factorization. To avoid any future-token leakage, we compute the next-gene loss *only from the forward stream* before bidirectional fusion.

Concretely, let $g_1, \ldots, g_M$ denote the ranked gene tokens of a cell. We obtain a forward representation $\mathbf{r}_t$ at position t from the forward scan (e.g., the forward block output derived from h_t) which depends only on the prefix $g_{\le t}$. We then predict the next token g_{t+1} via a linear head:

$$P(g_{t+1} \mid g_{\le t}) = \mathrm{softmax}(W_{\mathrm{lm}}\, \mathbf{r}_t).$$

The language loss, denoted as $\mathcal{L}_{\mathrm{lang}}$, is computed as the negative log-likelihood (NLL) of the true next gene token conditioned on the prefix:

$$\mathcal{L}_{\mathrm{lang}} = -\frac{1}{M-1} \sum_{t=1}^{M-1} \log P(g_{t+1} \mid g_{\le t}).$$

This training signal encourages the model to capture consistent rank-conditional dependencies across genes while preserving a causal supervision path. The reverse stream and gated bidirectional fusion are used to form richer contextual embeddings (used for downstream tasks and for the pathway contrastive objective below), but they are not used as inputs to the next-gene prediction head.

Gene Pathway Loss To capture gene–gene relationships within biological pathways, we employ an InfoNCE-style contrastive objective that enforces similarity among genes sharing a common pathway Oord et al. [2018], Sharma and Xu [2023], Gundogdu et al. [2022]. Pathways represent functional groupings of genes that collectively contribute to specific biological processes Zien et al. [2000], García-Campos et al. [2015]. We define a binary pathway label y_{ij} indicating whether genes i and j share at least one pathway:

$$y_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ share a pathway} \\ 0, & \text{otherwise.} \end{cases}$$

Given a gene pair (i, j), we compute the cosine similarity between their embeddings:

$$\text{sim}(i, j) = \frac{\mathbf{h}_i \cdot \mathbf{h}_j}{\|\mathbf{h}_i\| \|\mathbf{h}_j\|}$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are the normalized embeddings of genes i and j. To emphasize the contrast between positive and negative pairs, we normalize the similarities using temperature scaling:

$$\tilde{\text{sim}}(i, j) = \frac{\text{sim}(i, j)}{\tau}$$

where $\tau > 0$ is a temperature hyperparameter. The InfoNCE loss for a batch of gene pairs is then formulated as:

$$\mathcal{L}_{\text{pathway}} = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \log \frac{\sum_{j \in \mathcal{B} \setminus \{i\}} y_{ij} \exp(\tilde{\text{sim}}(i, j))}{\sum_{k \in \mathcal{B} \setminus \{i\}} \exp(\tilde{\text{sim}}(i, k))}$$

Here, $\mathcal{B}$ denotes the set of genes considered in the mini-batch, and $\mathcal{I}$ is the subset of anchors that have at least one positive (i.e., $\sum_{j \in \mathcal{B} \setminus \{i\}} y_{ij} > 0$). The numerator aggregates all pathway-positive genes for anchor i (selected by y_{ij}), while the denominator contrasts against all other genes in the batch.

By leveraging pathway labels as a contrastive supervision signal, the objective pulls together genes within the same pathway while pushing apart genes from different pathways, improving the functional organization of gene embeddings.

Although BiMamba produces bidirectional embeddings, the next-gene supervision in $\mathcal{L}_{\text{lang}}$ is computed strictly from prefix-only forward representations, so future tokens do not provide direct predictive supervision for next-gene prediction.

The final pretraining loss is the weighted sum of the two losses:

$$\mathcal{L} = \mathcal{L}_{\text{lang}} + \gamma \mathcal{L}_{\text{pathway}}$$

We set the γ value to 0.1 in our experiments based on validation results from sample datasets.

III. Experiments and Results

A. Dataset and Benchmark

Our pretraining dataset was constructed from the CELLXGENE database, comprising 50,689,395 raw cells. We retained only primary data entries, eliminating 41% of samples and yielding 30,139,066 curated unique cells. We filtered for 22,053 human protein-coding or miRNA genes and excluded cells expressing fewer than 200 genes. Following normalization to fixed target values and log1p transformation, the final pretraining dataset contained 29,849,897 cells (˜30M). For each cell, genes were ranked by their normalized expression levels (computed using non-zero median values across cells), and the top 2,048 or 4,096 genes were selected as input features. For downstream evaluation, we used multiple public scRNA-seq datasets reflecting diverse tissues and experimental conditions. Cell-type annotation was evaluated on hPancreas (human pancreatic endocrine and exocrine populations) and Myeloid (myeloid-lineage immune cells including monocytes and macrophages). We constructed Myeloid_b by filtering rare cell types (proportion < 0.05) to address class imbalance. Multi-batch integration was assessed on PBMC12k, COVID-19, and Perirhinal Cortex, covering immune and brain contexts under distinct batch effects. Gene correlation analysis used Immune and BMMC datasets to evaluate preservation of biologically meaningful gene-gene relationships.

To comprehensively evaluate GeneMamba, we compare against representative methods spanning three categories: foundation models, classical baselines, and architectural variants.For foundation models, we benchmark against five transformer-based approaches: GeneFormer Theodoris et al. [2023] for transfer learning in network biology, scGPT Cui et al. [2024] with generative pretraining, scFoundation Hao et al. [2024] trained on tens of millions of cells, scBERT Yang et al. [2022] with BERT-style bidirectional pretraining, and scMulan Bian et al. [2024] for general-purpose embeddings.Classical baselines include Harmony Korsunsky et al. [2019] for batch-effect correction, HVG for highly-variable-gene selection, Seurat for comprehensive integration and annotation, scVI Wolf et al. [2018] for variational inference-based generative modeling, and we evaluate GeneMamba_U (unidirectional variant) to assess bidirectional modeling benefits. For efficiency comparisons, we include standard Transformer and FlashAttention Dao et al. [2022] as quadratic-complexity baselines.

B. Experimental Settings

Unless otherwise specified, our backbone is GeneMamba based on the Bi-Mamba architecture with 24 layers and hidden dimension 512. We use a sequence length of 2048 and a tokenizer vocabulary size of 25,426, yielding 131.48M learnable parameters, and fuse forward and backward representations with a gating mechanism. We train the model with AdamW using a learning rate of 5×10^{-5}. The per-GPU batch size is 24 with gradient accumulation of 1, resulting in a global batch size of 96 when training on 4 GPUs. Training is performed with distributed data parallelism (DDP), with unused-parameter detection enabled. Pretraining is run for five epochs on four NVIDIA A100-SXM4-80GB GPUs, taking approximately three weeks to complete.

C. Cell Type Annotation

We evaluate GeneMamba on cell type annotation using three benchmark datasets: hPancreas, Myeloid, and Myeloid_b (a balanced version of Myeloid excluding rare cell types with proportion < 0.05).

GeneMamba was fine-tuned with a simple MLP classification head. As shown in Table I, GeneMamba achieves the highest accuracy (0.9713) and Macro-F1 (0.7710) on hPancreas, demonstrating robustness in capturing complex cell-type variations. On Myeloid, GeneMamba outperforms all baselines with Macro-F1 of 0.3650, while maintaining strong performance on the balanced Myeloid_b dataset (Acc: 0.9603, Macro-F1: 0.9235).

These results highlight several important trends. First, the gap between Accuracy and Macro-F1 underscores the impact

of class imbalance: high accuracy can be achieved by prioritizing dominant cell types, whereas Macro-F1 better reflects performance on minority populations. On hPancreas, GeneMamba improves Macro-F1 over strong foundation-model baselines (e.g., 0.7710 vs. 0.7632 for scGPT and 0.7450 for GeneFormer), suggesting more reliable recognition of less frequent endocrine subtypes while maintaining competitive accuracy. Second, on the challenging Myeloid dataset, absolute scores are lower for all methods, reflecting fine-grained and potentially noisy labels; GeneMamba attains the best Accuracy (0.6607) and Macro-F1 (0.3650), indicating better balanced predictions across myeloid subpopulations rather than overfitting to the most common classes. Third, on the balanced Myeloid_b dataset, most foundation models achieve strong Macro-F1; GeneMamba achieves the highest Accuracy (0.9603) but a slightly lower Macro-F1 (0.9235) than scFoundation (0.9569), suggesting remaining headroom in calibrating decision boundaries across classes even when overall separability is high. Overall, these results support that GeneMamba learns transferable cell representations that are competitive with or better than existing foundation models, particularly in settings where robustness to imbalance is critical.

D. Multi-batch Integration

In this experiment, we evaluated the multi-batch integration performance of the GeneMamba model. Multi-batch integration refers to aligning and harmonizing single-cell datasets from multiple experimental batches to minimize batch effects while preserving biologically meaningful patterns. We fine-tuned the pretrained GeneMamba model by adding a simple classification head (MLP) on the PBMC12k, COVID-19, and perirhinal cortex datasets. Using the learned embeddings from these foundation models, we performed multi-batch integration experiments.

The Avg-batch metric assesses the model's ability to correct batch effects and integrate data across multiple batches, while the Avg-bio metric evaluates the preservation of biological variation and meaningful clustering of cell types after integration. As shown in Table II, batch correction and biological preservation often present a trade-off. Harmony, as a specialized method for batch effect correction, excels in eliminating batch effects but tends to ignore biological differences. In contrast, GeneMamba demonstrates superior performance by effectively mitigating batch effects while preserving biological information. This highlights its robust capability in multi-batch integration and its biological relevance for single-cell data analysis.

E. Gene Rank Reconstruction

The GeneMamba model demonstrates exceptional capability in reconstructing ranked gene orders, a critical requirement for single-cell analysis tasks where gene expression patterns reveal cellular states and transitions. The analysis begins by randomly selecting sample cells. From these cells, input gene tokens are extracted, and output tokens are generated using the GeneMamba model. Then the gene tokens are ranked based on their predicted likelihoods, with missing tokens filled by zeros to ensure sequence consistency. This approach preserves the structural integrity of the data and facilitates direct comparison between input and output rankings.

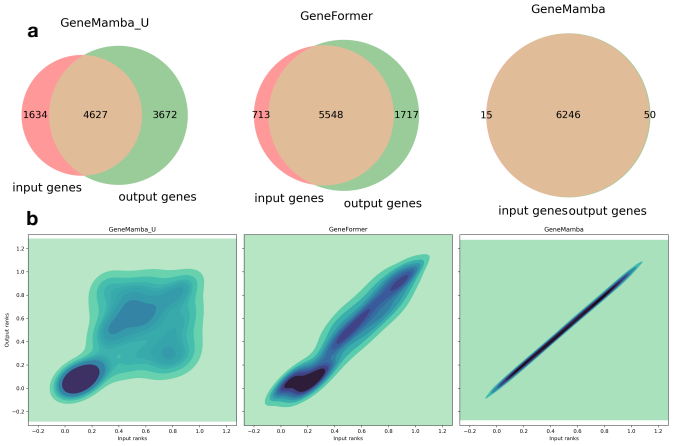

Fig. 2. **Gene rank reconstruct results on PBMC12k dataset** (a) Venn diagrams showing overlapping between input and output tokens in the PBMC12k dataset by three models: GeneMamba_U (unidirectional Mamba module as backbone), GeneFormer, GeneMamba (BiMamba module as backbone). (b) Density plots showing input and output ranking in the PBMC12k dataset by three models: GeneMamba_U, GeneFormer, GeneMamba.

The consistency of GeneMamba's predictions is validated through a Venn diagram Figure 2a, which illustrates a high degree of overlap between input and output tokens for the PBMC12k dataset. This overlap highlights the model's ability to retain critical features of the input data, an essential characteristic for single-cell studies where the integrity of gene expression ranks directly influences downstream analyses. A density plot Figure 2b provides further evidence of rank fidelity. It demonstrates that lower-ranked input genes consistently yield lower-ranked outputs, indicating a linear relationship between input and output ranks. This observation underscores the model's effectiveness in capturing patterns within the data and maintaining rank consistency.

Comparative analysis (Figure 2 from left to right) reveals the advantages of GeneMamba over other models: (1) The inclusion of the BiMamba module significantly enhances performance compared to the unidirectional module. This result demonstrates the necessity of bidirectional context extraction for effectively reconstructing ranked gene orders. (2) The GeneFormer model excels at predicting higher-ranked genes but struggles with lower-ranked genes, often treating them as noise due to their lower frequency. In contrast, GeneMamba captures both higher- and lower-ranked genes effectively, showcasing its robustness and sensitivity to varying gene expression levels. The metric results are summarized in Table III. By reliably reconstructing ranked gene orders, GeneMamba offers a powerful solution for understanding complex gene expression patterns. Its ability to maintain rank integrity across varying data scales positions it as an invaluable tool for single-

TABLE I
BENCHMARK ANNOTATION PERFORMANCE ACROSS DATASETS

Dataset	Metric	HVG	Seurat	scVI	scANVI	GeneFormer	scGPT	scFoundation	scBERT	scMulan	GeneMamba
hPancreas	Accuracy	0.9770	0.9668	0.9597	0.9662	0.9665	0.9710	0.9602	0.9594	0.9501	**0.9713**
	Macro-F1	0.7029	0.7346	0.6631	0.7023	0.7450	0.7632	0.7101	0.7381	0.7012	**0.7710**
Myeloid	Accuracy	0.6052	0.6571	0.6029	0.6332	0.6445	0.6341	0.6446	0.6527	0.6483	**0.6607**
	Macro-F1	0.3334	0.3627	0.3330	0.3400	0.3600	0.3562	0.3646	0.3482	0.3590	**0.3650**
Myeloid_b	Accuracy	0.8942	0.9018	0.8929	0.9231	0.9540	0.9421	0.9574	0.9451	0.9487	**0.9603**
	Macro-F1	0.8825	0.8930	0.8780	0.9147	0.9380	0.9434	**0.9569**	0.9261	0.9340	0.9235

TABLE II
BENCHMARK RESULTS OF THE MODELS ON MULTI-BATCH EXPERIMENTS WITH BATCH AND CELL METRICS

Metric	Dataset	Harmony	GeneFormer	scGPT	scFoundation	GeneMamba
Avg_batch	Immune	0.9514	0.8153	0.9194	0.8904	**0.9536**
	PBMC12k	0.9341	0.9545	**0.9755**	0.9628	0.9604
	BMMC	0.8999	0.7720	0.8431	0.7598	**0.9157**
	Perirhinal Cortex	0.9442	0.9127	0.9600	0.9560	**0.9673**
	Covid-19	**0.8781**	0.8240	0.8625	0.8346	0.8742
Avg_bio	Immune	0.6945	0.6983	0.7879	0.7337	**0.8131**
	PBMC12k	0.7990	0.7891	**0.9018**	0.8662	0.8344
	BMMC	0.6316	0.6324	0.6576	0.5250	**0.7628**
	Perirhinal Cortex	0.8595	0.8547	0.9552	**0.9606**	0.9062
	Covid-19	0.4468	0.5567	**0.6476**	0.5468	0.5537

TABLE III
GENE RANK RECONSTRUCTION ON THE PBMC12K DATASET

Model	L-Dist	BLEU	Spearman
GeneMamba_U	430	0.532	0.469
GeneFormer	23	0.968	0.703
GeneMamba	**6**	**0.987**	**0.711**

cell genomics.

F. Gene Correlation Analysis

GeneMamba exhibits exceptional capabilities in gene correlation analysis, distinguishing itself in capturing biologically meaningful patterns, maintaining raw data fidelity, and identifying unique model-specific topologies. A series of experiments evaluated GeneMamba's performance in distinguishing gene relationships, preserving correlation structures, and classifying gene types.

Differentiating Positive and Negative Gene Pairs. In the first experiment, GeneMamba's ability to differentiate between "positive" and "negative" gene pairs was assessed using Gene2Vec embeddings. Nearly 30,000 gene pairs, evenly distributed as positive (1) or negative (0), were analyzed for similarity scores using cosine similarity and Pearson correlation. GeneMamba demonstrated a significant separation between the mean similarity scores of positive and negative pairs (Figure 3 a, b). Compared to baseline models, GeneMamba achieved a clearer distinction, emphasizing its capacity to accurately represent gene relationships.

Gene-Gene Topology Analysis. The ability to capture gene-gene topologies was analyzed using adjacency matrices

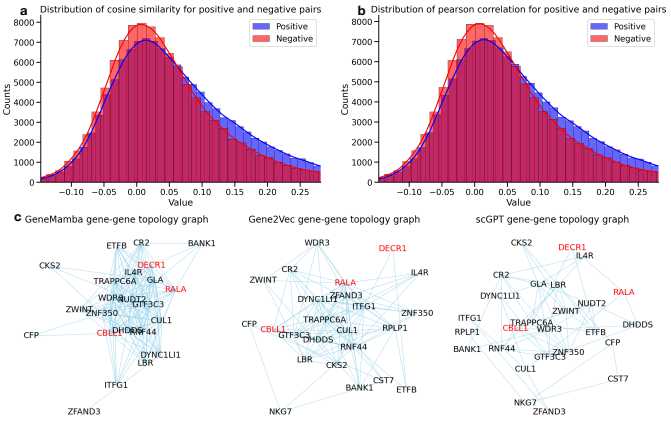

Fig. 3. **Gene Correlation Analysis.** (a) Distribution of consine similarity score for positive and negative gene pairs using Gene2Vec embeddings. (b) Distribution of pearson correlation score for positive and negative gene pairs using Gene2Vec embeddings. (c) Gene-gene topology comparison using adjacency matrices highlights shared and unique structures captured by GeneMamba, scGPT, and Gene2Vec, such as genes RALA, DECR1, and CBL1.

derived from embeddings. For the PBMC12K dataset, distances between embeddings of randomly selected gene pairs (100 cells in our case) were calculated and binarized based on threshold value. Then we build the topology graph based on the adjacent matrix (Figure 3c) for each evaluated model. Notably, while identifying unique model-specific topologies, all three models captured biologically significant genes and their interactions, such as RALA, DECR1, and CBL1. This

suggests that the foundation models share a similar representation space.

G. Efficiency Analysis

TABLE IV
COMPUTATIONAL EFFICIENCY COMPARISON OF TRANSFORMER, FLASHATTENTION, AND GENEMAMBA BACKBONES AT DIFFERENT INPUT SEQUENCE LENGTHS.

Backbone	Params	FLOPs/sample	Seq. len	Time (s/sample)
Transformer	130M	2.85E+12	2048	0.2227
Transformer	130M	7.18E+12	4096	0.6025
Transformer	130M	2.02E+13	8192	1.8904
FlashAttention	130M	2.62E+12	2048	0.2087
FlashAttention	130M	7.03E+12	4096	0.5899
FlashAttention	130M	1.96E+13	8192	1.8342
GeneMamba	130M	1.58E+12	2048	0.1284
GeneMamba	130M	3.15E+12	4096	0.2393
GeneMamba	130M	6.31E+12	8192	0.4885

To demonstrate the computational advantages of Mamba over Transformer-based approaches, we compared training time and FLOPs across three 130M-parameter architectures: standard Transformer, FlashAttention, and GeneMamba. Sequence lengths of 2048, 4096, and 8192 genes represent typical single-cell transcriptomic scenarios. Results are in Table IV.

GeneMamba's efficiency gains scaled with sequence length. At 2048 genes, GeneMamba achieved 0.1284 seconds per sample versus 0.2227 for Transformer and 0.2087 for FlashAttention (1.73× and 1.63× speedups). At 4096 genes, GeneMamba required 0.2393 seconds versus 0.6025 and 0.5899 (2.52× and 2.47× speedups). At 8192 genes, GeneMamba processed samples in 0.4885 seconds versus 1.8904 and 1.8342 (3.87× and 3.76× speedups).

FLOPs analysis corroborates these results. GeneMamba consumed 1.58E+12 FLOPs at 2048 genes (55.4% of Transformer's 2.85E+12), with this advantage intensifying at longer sequences: 43.9% at 4096 genes and 31.2% at 8192 genes. FlashAttention reduced FLOPs by 8% and training time by 6-7% over standard Transformer but remained substantially slower than GeneMamba.

These results confirm that linear-complexity Mamba provides substantial computational benefits over quadratic attention, with efficiency gains scaling favorably as sequence length increases.

H. Ablation Study

To investigate the contributions of key components, we performed ablation experiments evaluating model depth, embedding dimensions, and pathway loss weight on hPancreas and PBMC12k datasets. We compared three configurations: lightweight (12 layers, 256-dim, $\gamma = 0.05$), baseline (24 layers, 512-dim, $\gamma = 0.1$), and deep (48 layers, 768-dim, $\gamma = 0.2$). Results are in Table V.

TABLE V
ABLATION STUDY OF MODEL CONFIGURATIONS ON HPANCREAS AND PBMC12K DATASETS.

Configuration	Accuracy	BLEU	Training Time
12-layer, 256-dim, $\gamma = 0.05$	0.961	0.932	45%
24-layer, 512-dim, $\gamma = 0.1$	0.971	0.966	100% (baseline)
48-layer, 768-dim, $\gamma = 0.2$	0.972	0.968	210%

The baseline achieved 97.1% accuracy and 0.966 BLEU score. Scaling to 48 layers yielded marginal gains (97.2% accuracy, 0.968 BLEU) while requiring 210% training time. The lightweight model showed degradation to 96.1% accuracy and 0.932 BLEU. Increasing γ from 0.05 to 0.2 improved BLEU scores (0.932 to 0.968) with minimal impact on accuracy. These findings confirm the baseline configuration balances accuracy and computational efficiency, with diminishing returns beyond 24 layers and 512 dimensions.

While complete pathway loss ablation was computationally prohibitive, we tracked its training dynamics over 16,000 steps. Pathway loss started at 9.8, dropped sharply to 7.0 within 4,000 steps, then stabilized at 6.6 after 12,000 steps. This steady decline indicates GeneMamba progressively aligned gene representations with biological pathway structures. The stabilization coupled with downstream task improvements confirms pathway regularization effectively guides learning toward biologically interpretable features without destabilizing primary objectives.

IV. DISCUSSION AND CONCLUSION

GeneMamba aims to bridge a central tension in single-cell foundation models: capturing genome-scale dependencies with practical compute while avoiding inductive biases that are misaligned with the largely order-agnostic nature of gene co-expression. The proposed bidirectional Mamba backbone provides a linearly scalable alternative to quadratic attention, and the pathway-aware contrastive objective injects structured biological priors during pretraining.

From an algorithmic perspective, the observed robustness in downstream tasks can be interpreted through the selective state-space mechanism: input-dependent gating offers a natural way to suppress batch-specific or noisy signals while preserving consistent, cell-type-related expression programs. In parallel, pathway regularization encourages embeddings to respect curated functional groupings, acting as a soft inductive bias that complements data-driven learning without hard-coding pathway membership.

Several limitations remain. Our pretraining corpus, while extensive at ˜30M curated cells (from ˜50M raw CELLXGENE profiles), may underrepresent rare states and specialized tissues. The current study focuses on transcriptomics; extending the framework to multi-modal settings (e.g., chromatin accessibility, protein, or spatial context) is an important direction. Finally, despite pathway regularization, interpretability remains challenging; developing tools to translate learned representations into testable biological hypotheses, and to evaluate zero-shot or few-shot generalization to unseen cell types, are promising avenues for future work.

In summary, GeneMamba provides an efficient foundation-model backbone for single-cell transcriptomics by combining a bidirectional, linearly scalable state-space architecture with pathway-aware pretraining. This design enables practical modeling of long gene sequences while encouraging representations that remain aligned with biological structure, supporting robust transfer across diverse downstream tasks.

REFERENCES

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

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, and Patrick T Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Hongru Shen, Jilei Liu, Jiani Hu, Xilin Shen, Chao Zhang, Dan Wu, Mengyao Feng, Meng Yang, Yang Li, Yichen Yang, Wei Wang, Qiang Zhang, Jilong Yang, Kexin Chen, and Xiangchun Li. Generative pretraining from large-scale transcriptomes for single-cell deciphering. *Iscience*, 26(5), 2023.

Xiaodong Yang, Guole Liu, Guihai Feng, Dechao Bu, Pengfei Wang, Jie Jiang, Shubai Chen, Qinmeng Yang, Hefan Miao, Yiyang Zhang, Zhenpeng Man, Zhongming Liang, Zichen Wang, Yaning Li, Zheng Li, Yana Liu, Yao Tian, Wenhao Liu, Cong Li, Ao Li, Jingxi Dong, Zhilong Hu, Chen Fang, Lina Cui, Zixu Deng, Haiping Jiang, Wentao Cui, Jiahao Zhang, Zhaohui Yang, Handong Li, Xingjian He, Liqun Zhong, Jiaheng Zhou, Zijian Wang, Qingqing Long, Ping Xu, Hongmei Wang, Zhen Meng, Xuezhi Wang, Yangang Wang, Yong Wang, Shihua Zhang, Jingtao Guo, Yi Zhao, Yuanchun Zhou, Fei Li, Jing Liu, Yiqiang Chen, Ge Yang, and Xin Li. Genecompass: deciphering universal gene regulatory mechanisms with a knowledge-informed cross-species foundation model. *Cell Research*, pages 1–16, 2024.

Jingcheng Du, Peilin Jia, Yulin Dai, Cui Tao, Zhongming Zhao, and Degui Zhi. Gene2vec: distributed representation of genes based on co-expression. *BMC genomics*, 20:7–15, 2019.

Suyuan Zhao, Jiahuan Zhang, and Zaiqing Nie. Large-scale cell representation learning via divide-and-conquer contrastive learning. *arXiv preprint arXiv:2306.04371*, 2023.

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.

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*, pages 1–11, 2024.

Haiyang Bian, Yixin Chen, Xiaomin Dong, Chen Li, Minsheng Hao, Sijie Chen, Jinyi Hu, Maosong Sun, Lei Wei, and Xuegong Zhang. scmulan: a multitask generative pre-trained language model for single-cell analysis. In *International Conference on Research in Computational Molecular Biology*, pages 479–482. Springer, 2024.

Hongzhi Wen, Wenzhuo Tang, Xinnan Dai, Jiayuan Ding, Wei Jin, Yuying Xie, and Jiliang Tang. Cellplm: pre-training of cell language model beyond single cells. *bioRxiv*, pages 2023–10, 2023.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Zihan Li, Yiqing Wang, Sina Farsiu, and Paul Kinahan. Boosting medical visual understanding from multi-granular language learning. *arXiv preprint arXiv:2511.15943*, 2025.

Wenbo Wang, Cong Qi, and Zhi Wei. Modeling tcr-pmhc binding with dual encoders and cross-attention fusion. In *2025 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 5083–5090. IEEE, 2025.

Cong Qi, Hanzhang Fang, Siqi Jiang, Tianxing Hu, and Zhi Wei. Lantern: Tcr-peptide binding prediction via large language model representations. *PeerJ*, 14:e20980, 2026.

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*, pages 1–11, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Divya Sharma and Wei Xu. Regenne: genetic pathway-based deep neural network using canonical correlation regularizer for disease prediction. *Bioinformatics*, 39(11):btad679, 2023.

Pelin Gundogdu, Carlos Loucera, Inmaculada Alamo-Alvarez, Joaquin Dopazo, and Isabel Nepomuceno. Integrating pathway knowledge with deep neural networks to reduce the dimensionality in single-cell rna-seq data. *BioData Mining*, 15:1–21, 2022.

Alexander Zien, Robert Küffner, Ralf Zimmer, and Thomas Lengauer. Analysis of gene expression data with pathway scores. In *Ismb*, volume 8, pages 407–417, 2000.

Miguel A García-Campos, Jesús Espinal-Enríquez, and Enrique Hernández-Lemus. Pathway analysis: state of the art. *Frontiers in physiology*, 6:383, 2015.

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