FlowGRN: Scalable and Dropout-Robust Gene Regulatory Network Inference via Flow Matching-Based Trajectory Reconstruction

Tsz Pan TONG
University of Luxembourg
Esch-sur-Alzette, Luxembourg
tszpan.tong@uni.lu

Jun PANG
University of Luxembourg
Esch-sur-Alzette, Luxembourg
jun.pang@uni.lu

Abstract

Inferring gene regulatory networks (GRNs) from single-cell RNA sequencing (scRNA-seq) data offers insights into cellular behavior, but is complicated by the lack of temporal information and the prevalence of dropout noise. To address these challenges, we present FlowGRN, a method that integrates conditional flow matching and score matching for robust cell-level trajectory reconstruction with dynGENIE3 for scalable GRN inference. FlowGRN incorporates a novel cell similarity measure that is resilient to dropout effects in high-dimensional scRNA-seq data. Evaluation on the BEELINE benchmark demonstrates that FlowGRN achieves state-of-the-art performance on both synthetic and experimental datasets. Ablation studies validate the importance of both the dropout-robust similarity measure and the trajectory reconstruction step, highlighting FlowGRN's ability to accurately model regulatory relationships.

CCS Concepts

• **Applied computing** → **Recognition of genes and regulatory elements**; *Bioinformatics*; • **Computing methodologies** → **Neural networks**.

Keywords

Gene regulatory networks, trajectory reconstruction, conditional flow matching, neural ODE

ACM Reference Format:
Tsz Pan TONG and Jun PANG. 2025. FlowGRN: Scalable and Dropout-Robust Gene Regulatory Network Inference via Flow Matching-Based Trajectory Reconstruction. In *Proceedings of the 16th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '25), October 11–15, 2025, Philadelphia, PA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3765612.3767196

1 Introduction

Gene regulatory networks (GRNs), which map the intricate interplay between genes and their regulators, are fundamental to our understanding of cellular life. Accurate deciphering of these networks is essential for elucidating cellular processes and for designing therapeutic strategies against complex diseases [45]. Recent advances in single-cell RNA sequencing (scRNA-seq) technology have enabled the profiling of gene expression at the single-cell level, providing a wealth of data and valuable insights into cellular dynamics. However, accurately inferring GRNs from scRNA-seq data remains challenging, primarily because sequencing data are snapshots of cellular states, lacking explicit dynamic information, and dealing with extremely high-dimensional gene spaces.

Current scRNA-seq technology requires stopping the cell evolution process to capture snapshots of cell states. To balance sequencing depth and resolution, the sampling time points are often sparse, and thus key dynamic changes such as cell transitions are unobserved. Thus, cell developmental trajectories are lost during sequencing, and the low temporal resolution further hinders our ability to restore cellular dynamics. In addition, GRN inference involves tens of thousands of genes. At this scale, the Euclidean distance becomes less informative due to the curse of dimensionality, and the scalability of inference methods becomes a critical issue.

Facing the challenges of loss of cell dynamics and high dimensionality, researchers have taken various approaches to infer GRNs from scRNA-seq data. Given that gene regulation involves interactions between macromolecules governed by the laws of physics and chemistry, ordinary differential equations (ODEs) have become a natural choice for modeling dynamics. Different methods based on ODEs, both statistical [3, 24] and deep learning-based [37, 42], have been proposed to model cellular dynamics. However, these methods either oversimplify the system or are computationally intractable because of the high dimensionality. Additionally, restoring dynamics relies on accurate pseudotime estimations for each cell. However, the performance of different pseudotime estimation algorithms varies on the same dataset [34], and many GRN inference methods [29, 32] are highly sensitive to pseudotime accuracy [31].

Another research track maximizes scalability by discarding temporal information and treating the scRNA-seq data as a static snapshot. GRN inference methods such as GENIE3 [16], GRNBOOST2 [26] and PIDC [5] are widely used and show promising results on various benchmarks [7, 31]. They infer regulatory relationships by quantifying the information gain of each gene in predicting other genes, or the unique information decomposed from the mutual information between genes. Although these methods are powerful, scalable, and supported by statistical theories, they discard temporal information and make the direction of regulations unidentifiable [39]. The loss of dynamic information and the high-dimensionality problem constitute a dilemma in GRN inference, and the community is urging the development of a precise, scalable, and identifiable GRN model that utilizes temporal information and reconstructs cell dynamics in the high-dimensional gene space.

BCB '25, Philadelphia, PA, USA

ACM ISBN 979-8-4007-2200-4/2025/10
https://doi.org/10.1145/3765612.3767196

Recently, conditional flow matching (CFM) [2] has been proposed to use a neural network to model the vector field that flows high-dimensional data from one distribution to another. Compared to traditional neural ODEs [37, 42], CFM can learn the flow in >10,000 dimensions efficiently by making a specific assumption about the distribution of pairings between the source and target distributions. This motivated us to utilize CFM to reconstruct cellular trajectories and infer the GRN from scRNA-seq data.

In practice, we noticed that the heavy dropout noise in scRNA-seq data introduces additional uncertainty to the measurement of the distance between cells, which is a cornerstone in dynamic modeling. Dropouts are a common phenomenon in scRNA-seq data, where a gene is not detected in a cell even though it is expressed [17]. Some imputation methods [8, 14, 46] have been proposed to restore missing gene expression from its neighbors and latent data geometry. Unfortunately, they bring additional false positive edges to the inferred GRN, thus the imputed data are not suitable for GRN inference [22]. To address dropout noise, we propose a new similarity measure that is robust to dropout in high-dimensional spaces.

In this work, we utilize advances in conditional flow matching and score matching [43] to reconstruct cellular trajectories. We then employ dynGENIE3 [15], a dynamic version of GENIE3, to infer an identifiable GRN from the reconstructed cellular trajectories with thousands of genes. We propose a new similarity measure to address dropout noise, which experimentally shows robustness to dropout. By combining advances in CFM for dynamics modeling, reliable dynGENIE3 for GRN inference, and a dropout-robust cell similarity measure, our scalable GRN inference solution, FlowGRN, outperforms state-of-the-art methods for directed GRN inference in terms of precision. Our contribution is summarized as follows:

- We propose a new cell similarity measure that shows robustness to dropout and the curse of dimensionality.
- We apply this new measure on conditional flow matching and show that it can reconstruct cellular trajectories with non-linear dynamics, scalable up to 1,783 genes.
- We show that the gradient of the ODEs cannot accurately infer GRNs because of indirect regulations, thus dynGENIE3 is used to infer the GRN from the learned trajectory.
- Our method, FlowGRN, has the highest average precision rank in the BEELINE benchmark [31] on experimental datasets.

Our method is available at https://github.com/1250326/FlowGRN.

2 Related Work

2.1 Gene regulatory network inference

ScRNA-seq data are the primary source for GRN inference, providing snapshots of cell states in the form of a series of gene expression profiles. To ensure sufficient sequencing depth and resolution, sampling time points are often sparse, which limits the application of time-series models. Correlation-based methods [20, 40], information theory-based methods [5, 9, 23, 32] and regression methods [13, 29, 35] provide a statistically sound approach to infer GRNs from limited data. However, these methods are bound by the development and strong assumptions of statistical theories. ODE-based methods [3, 24] attempt to model cell dynamics with linear ODEs, which oversimplify system complexity and are computationally inefficient on large GRNs.

Taking advantage of the rapid development of machine learning, ensemble methods [15, 16, 26] offer powerful and scalable options for GRN inference. GENIE3 [16] and GRNBOOST2 [26] are the most recognized and widely used for GRN inference, and the latter is the default option in the SCENIC [1] protocol for coarse selection of regulatory pairs. Both methods train a tree ensemble model to predict the expression of each target gene from other genes, and use the importance score in each model to quantify the regulatory relationships. dynGENIE3 [15] aims to incorporate temporal information, but their reliance on continuous cellular trajectories, which are unobservable in destructive snapshot-based scRNA-seq, has largely limited their application to simulated datasets rather than real-world scenarios. Recently, researchers have tried to borrow concepts from deep learning, such as convolutional neural networks [6, 10, 33], variational autoencoder (VAE) [38], and explainable AI [19]. However, many of these emerging deep learning methods have not yet matched the ensemble methods in terms of accuracy and scalability, limiting their practical application.

GENIE3 and its variants have dominated the field of GRN inference since 2010, and researchers have sought breakthroughs from multi-omics data. Representative methods [11, 18] integrate scRNA-seq data with scATAC-seq and spatial transcriptomics data to infer GRNs. Interested readers can refer to recent reviews [4, 21]. Although multi-omics data offer new perspectives for GRN inference, revolutionizing our understanding of cell dynamics using only scRNA-seq data remains an important long-term challenge.

2.2 The curse of dimensionality

A full-genome GRN inference task contains approximately 20,000 protein-encoding genes. If we further consider protein-protein interactions, the network size would be even larger. Thus, scalability is a critical challenge in GRN inference. Some methods [24, 37] address this challenge using linear mapping and VAE to compress the high-dimensional gene space into an actionable latent space. Latent space is a powerful tool for dimensionality reduction, and it is widely used in scRNA-seq data visualization and analysis [25, 27]. However, compressing gene expression into a lower-dimensional latent space can obscure individual gene interactions. This can lead to difficulties in accurately reconstructing gene-specific dynamics and potentially introduce spurious relationships when mapping latent interactions back to the gene level.

Fortunately, GRN is sparse, and only a small fraction of genes are regulated by each other. This enables feature selection and stability selection approaches to tackle high dimensionality. In brief, feature selection methods select a subset of genes from the whole genome and use them to train models, while stability selection identifies important genes that consistently contribute to the feature-selected models in multiple runs. Contemporary methods [3, 13, 40] combine two selection methods and show promising results on different benchmarks [31, 49]. Notably, the popular GENIE3 [16] also uses a similar approach in general, because its underlying algorithm is random forest, which uses multiple subsets of genes to train decision trees, and the importance score of each gene is averaged across all trees. This suggests that the ideas of feature and stability selection are widely applicable in GRN inference.

2.3 Technical noise in scRNA-seq data

10x Genomics sequencing technology offers a high-throughput and cost-effective solution for capturing transcripts in individual cells, but it brings additional dropouts to the gene expression matrix. Jiang et al. [17] classified zero reads from sequencing into biological, technical, and sampling zeros. Biological zeros are caused by silent genes and uncaptured transcriptional bursts, technical zeros are caused by inefficient reverse transcription from mRNA to cDNA, and sampling zeros are caused by inefficient amplification and low sampling depth. Thus, only biological zeros and nonzero reads are informative for GRN inference, while technical zeros and sampling zeros are recognized as dropouts. However, most mainstream regression-based [3, 13, 15, 16, 24, 26] and correlation-based [20, 40] methods simply use Euclidean distance and do not adequately account for dropout effects. This introduces biases in modeling, which affect the GRN inference accuracy.

3 Preliminaries

3.1 Notations and problem definition

A scRNA-seq dataset contains a series of n gene expression profiles $X = \{X^{(t_0)}, X^{(t_1)}, \cdots, X^{(t_{n-1})}\}$, each profile $X^{(t_i)} \in \mathbb{R}^{g,c_i}$ contains g genes and c_i cells sampled at time t_i. A cell $x \in \mathbb{R}^g$ can be viewed as a point in the g-dimensional gene space. A GRN is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$, where $\mathcal{V}$ is the set of genes, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\omega : \mathcal{E} \rightarrow \mathbb{R}$ is the weight function (regulation strength). Positive (Negative) weights indicate activation (inhibition) regulations. An adjacency matrix $A \in \mathbb{R}^{g \times g}$ is a matrix representation of the GRN, where $A_{i,j} = \omega(i,j)$ if $(i,j) \in \mathcal{E}$ and $A_{i,j} = 0$ otherwise. We aim to infer the directed GRN $\mathcal{G}$ from the scRNA-seq dataset X without prior knowledge of the GRN structure.

3.2 Conditional flow and score matching

Cellular trajectories are crucial for inferring gene regulatory networks (GRNs). In a parallel field, structure inference, researchers have shown that trajectory data alone can accurately reconstruct the underlying dynamics of a system [48–50]. This motivates us to reconstruct the cellular trajectories to facilitate GRN inference.

Neural ODE is a powerful framework for modeling dynamical systems: it learns a vector field whose ODE solution matches observed time-series data. Recent methods, such as TrajectoryNet [42] and TIGON [37], have successfully applied neural ODEs to scRNA-seq data. However, their reliance on ODE integration makes them computationally expensive and numerically unstable, limiting their applications to small gene sets (< 50) or low-dimensional latent spaces. To overcome these limitations, we adopted the [SF]^{2}M model [43], an integration-free conditional flow matching (CFM) and score matching model. [SF]^{2}M leverages optimal transport (OT) plans between snapshots to define target vector fields, guiding the learning of the dynamics driving cellular transitions.

CFM trains the vector field referenced to the displacement between the two distributions, whereas neural ODEs train the vector field referenced to the observed trajectories, thereby incurring costly ODE integration. Unlike *lineage-level* trajectory inference methods such as Slingshot [41] and Monocle3 [44], CFM can reconstruct *cell-level* trajectories. Viewing scRNA-seq snapshots as a sequence of gene-expression distributions, CFM is a natural choice for learning the vector field driving those flows.

Score matching also trains a vector field, but refers to the score of the distributions of the cell trajectories, restricting the divergence of the CFM model to observed regions. Score matching can be viewed as the valley in the Waddington landscape, where the vector field points toward regions of higher probability. In practice, score matching is used as regularization to the CFM model, and the cell dynamics is captured by the CFM model.

In a theoretical sense, [SF]^{2}M can be seen as solving a Schrödinger bridge problem whose marginal evolution of densities is governed by the Fokker-Planck equation. They model cellular dynamics by the stochastic differential equation (SDE) with the form $dx = u_t(x)dt + g(t)dB_t$, where $u_t(x)$ is the drift and dB_t is the Brownian motion. This SDE, together with the initial distribution p_0, induces probability paths p_t via the continuity equation $\partial_t p_t = -\nabla \cdot (u_t p_t)$. [SF]^{2}M decomposes the drift u_t into two components: probability flow drift $u_t^\circ = u_t - \frac{1}{2}g^2(t)\nabla logp_t$ and score $\frac{1}{2}g^2(t)\nabla logp_t$, which points to regions of higher probability.

Thus, the dynamics can be learned by minimizing two neural networks v_θ, s_θ with the following loss function $\mathcal{L}_{\text{unconditional}}$:

$$\mathbb{E}\left[\left\|v_\theta(x,t) - u_t^\circ(x)\right\|^2 + \lambda(t)^2 \left\|s_\theta(x,t) - \nabla logp_t(x)\right\|^2\right], \quad (1)$$

where the expectation is taken over $t \sim \mathcal{U}(0,1), x \sim p_t(x)$, and $\frac{1}{2}g(t)^2$ is absorbed into the positive weight $\lambda(t)^2$. v_θ is called the flow matching model, and s_θ is the score matching model.

Since p_t is intractable, [SF]^{2}M conditioned on the OT plan π to guide the flow. For a given pair of distributions p_0 and p_1, the OT plan π is the joint distribution of $(x_0, x_1) \sim p_0 \times p_1$, which minimizes the cost function $c(x_0, x_1)$. Sampling the OT plan π gives (x_0, x_1), which allows us to compute:

$$\begin{aligned} u_t^\circ(x|x_0,x_1) =& \frac{1-2t}{t(1-t)}(x - (tx_1 + (1-t)x_0)) + (x_1 - x_0) \\ \nabla logp_t(x|x_0,x_1) =& \frac{tx_1 + (1-t)x_0 - x}{\sigma^2 t(1-t)}, \end{aligned} \quad (2)$$

when $g(t) = \sigma$ is a constant and thus the loss function $\mathcal{L}_{\text{conditional}}$ is written as:

$$\mathbb{E}\left[\left\|v_\theta(x,t) - u_t^\circ(x|x_0,x_1)\right\|^2 + \lambda(t)^2 \left\|s_\theta(x,t) - \nabla logp_t(x|x_0,x_1)\right\|^2\right] \quad (3)$$

where the expectation is taken over $t \sim \mathcal{U}(0,1), (x_0,x_1) \sim \pi, x \sim p_t(x|x_0,x_1)$.

4 Our Method FlowGRN

To address the identified challenges in GRN inference, we propose a novel method, FlowGRN, comprising three key components.

- A dropout-robust cell similarity measure that quantifies the geodesic distance between cells in high-dimensional gene space to solve the dropout noise (detailed in Alg. 1 in the Appendix).
- A conditional flow matching and score matching model that learns the underlying vector fields and restores cell trajectories.
- The dynGENIE3 model that infers an identifiable GRN from the learned trajectories.

Figure 1 illustrates the overall pipeline of FlowGRN, and Alg. 2 in the Appendix summarizes the training procedure.

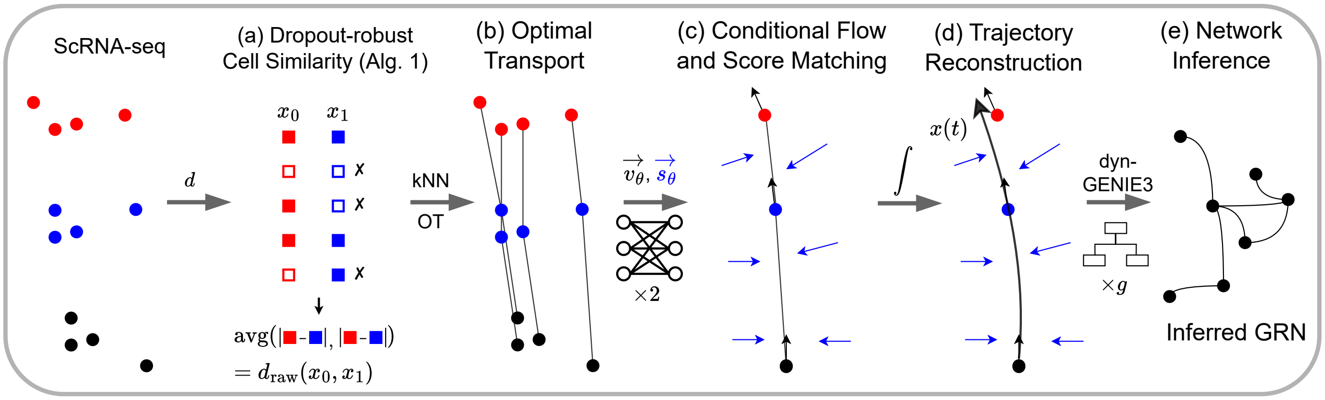
ScRNA-seq
(a) Dropout-robust Cell Similarity (Alg. 1)
(b) Optimal Transport
(c) Conditional Flow and Score Matching
(d) Trajectory Reconstruction
(e) Network Inference
d
x_0
x_1
avg(|■-■|, |■-■|)
$= d_{\text{raw}}(x_0, x_1)$
kNN
OT
$\vec{v}_\theta$, $\vec{s}_\theta$
×2
$x(t)$
$\int$
dyn-GENIE3
$\times g$
Inferred GRN

Figure 1: The overall pipeline of FlowGRN. (a) Compute d_{raw} after dropping zeros in each cell pair. (b-c) Train CFM v_θ and score matching s_θ models to learn the cell dynamics. (d) Restore cell trajectories. (e) Infer GRN with dynGENIE3 from trajectories.

4.1 Dropout-robust cell similarity measure

The scalability of [SF]^{2}M comes from the use of the optimal transport (OT) plan π to guide the flow. OT plan π is the joint distribution of $x_0 \sim p_0$ and $x_1 \sim p_1$ that minimizes the cost function $c(x_0, x_1)$. The cost function is usually defined as the Euclidean distance $\|x_0 - x_1\|_2$, but this metric is severely affected by the dropout noise in the scRNA-seq data and ignores the underlying geometry of the gene expression data manifold.

In Section 2.3, we discuss that only biological zeros and nonzero reads are informative for GRN inference, while technical zeros and sampling zeros are recognized as dropout. Thus, the zeros are corrupted by dropout, while the non-zeros do not. This enlightens us to only use non-zeros to quantify the cell similarity.

Given two cells $x, y \in \mathbb{R}^g$, let the set of non-zero genes in x and y be $S_x = \{i | x_i \neq 0\}$ and $S_y = \{i | y_i \neq 0\}$, respectively. The size of the intersection $S_{x,y} = S_x \cap S_y$ is $|S_{x,y}|$. We define the dropout-robust cell similarity measure $d_{\text{raw}}(x, y)$ as:

$$d_{\text{raw}}(x, y) = \frac{1}{|S_{x,y}|} \sum_{i \in S_{x,y}} |x_i - y_i|. \tag{4}$$

Here, the L1 norm is used to avoid distance inflation due to the different sizes of the intersection sets $|S_{x,y}|$. If $S_{x,y}$ is an empty set, a large d_{raw} will be assigned to indicate dissimilarity. Although this measure discards the information from the biological zeros, it ensures that the distance is not affected by any dropout noise.

Addressing the geometric structure of the gene expression manifold is not a new idea and has been used in various tasks such as visualization [25, 27], imputation [46, 47], and reconstruction of developmental trajectory [36]. We first build a mesh graph G_{knn} from the gene expression matrix X using k-nearest neighbors (kNN) with our cell similarity measure $d_{\text{raw}}(x, y)$. Then, the final cell similarity d_{knn} is defined as the shortest path distance on the mesh graph G_{knn}. The shortest path distance can be efficiently calculated using Dijkstra's algorithm. This design brings geometry awareness to the cell similarity measure and robustness on dropout noise. The new measure is detailed in Alg. 1 in the Appendix, and its effectiveness is tested in Section 6.5.

4.2 Trajectory reconstruction with CFM

With the cell similarity d_{knn}, we can apply the CFM model to learn the underlying vector field $v_\theta(x, t)$ and reconstruct the trajectories. We first apply Slingshot [41] to the dataset X and obtain the cell lineages. Trajectories are reconstructed per lineage to encode the lineage information and maximize parallelism.

Since gene expression values are non-negative, we must ensure that the reconstructed trajectories are also non-negative. Constraining the boundary of the vector field or enforcing a strictly positive gradient at the boundary are not ideal solutions, as they bring additional complexity and numerical instability. We solve this issue by converting the expression value $x \in \mathbb{R}^g_{\geq 0}$ to a log domain $x' \in \mathbb{R}^g$ through the following transformation $\mathcal{T}$:

$$x' = \mathcal{T}(x) = \begin{cases} log(x + \epsilon) + 1, & \text{if } x < 1 \\ x + \epsilon, & \text{if } x \geq 1, \end{cases} \tag{5}$$

where ϵ is a small positive value to avoid undefined logarithm. This ensured that the transformation is C^1 differentiable at $x = 1$ so that training did not get stuck at the discontinuity during optimization. In addition, the transformation magnifies low expression values and avoids compressing high expression values, thereby capturing cell dynamics at different scales.

We train the CFM and score matching models following the procedure in Section 3.2 in the log domain with the loss function $\mathcal{L}_{\text{conditional}}$, conditioned on the OT plan π with the cell similarity measure d_{knn} as the cost function. The trajectories can be recomputed by solving an initial value problem (IVP) starting from $x'(t)$:

$$x'(t+\Delta t) = x'(t) + \int_t^{t+\Delta t} v_\theta(x'(s), s)\, ds, \quad t_0 \leq t + \Delta t \leq t_{n-1}. \tag{6}$$

For each cell x observed at time t_i, its transformed state $x'(t_i) = \mathcal{T}(x(t_i))$ serves as an anchor point. We then generate its trajectory segments by solving the IVP $dx'/dt = v_\theta(x', t)$ forward from t_i to t_{n-1} and backward from t_i to t_0. An intuitive workflow of this algorithm is outlined in the first half of Alg. 2 in the Appendix.

4.3 GRN inference from the learned trajectory

The learned trajectory $x(t) = \mathcal{T}^{-1}(x'(t))$ is a time series of gene expression to be fed into dynGENIE3 to infer the GRN.

dynGENIE3 [15] is an extension of GENIE3 [16] that incorporates temporal information into the GRN inference process. Same as GENIE3, dynGENIE3 uses random forest f_j to model each target gene j in a cell trajectory $x(t)$:

$$\frac{dx_j(t)}{dt} = f_j(x(t)) - \alpha_j x_j(t), \tag{7}$$

where α_j is the decay rate of gene j and $\frac{dx_j(t)}{dt}$ is approximated by finite differences:

$$\frac{dx_j(t)}{dt} = \frac{x_j(t + \Delta t) - x_j(t)}{\Delta t}. \tag{8}$$

The importance score $\mathcal{I}_j(i)$ of each random forest model f_j is used to quantify the regulatory strength of the gene i on j and to build the adjacency matrix $A[i, j] = \mathcal{I}_j(i)$. This will give a directed graph as the importance scores $\mathcal{I}_j(i)$ and $\mathcal{I}_i(j)$ are not equal in general.

To determine the type of regulation (activation or inhibition), we use the learned vector field v_θ. Specifically, for each potential edge $i \longrightarrow j$, we evaluate the sign of the (i, j)-th element of the Jacobian matrix, $(\nabla x' v_\theta(x', t))_{i,j}$, at multiple points (x', t) corresponding to the observed cellular states. The final regulation type (activation/inhibition) is assigned based on a majority vote of these signs (positive/negative). This approach allows us to assign regulation types, a feature not available from the standard dynGENIE3 output.

5 Experiments

5.1 Datasets and evaluation metrics

We tested our method on the BEELINE [31] benchmark, which includes 6 synthetic datasets, 4 curated datasets, and 7 experimental datasets. Synthetic data sets include linear (LI), linear long (LL), cyclic (CY), bifurcating (BF), bifurcating converging (BFC), and trifurcating (TF), capturing common dynamics such as long cascades and furcating branches. Curated datasets include mCAD, VSC, HSC, and GSD, which mimic the reported biological processes. Both types of datasets are simulated with BoolODE [31] to generate 10 sets of simulated scRNA-seq data with 2,000 cells.

Experimental datasets are real gene expression profiles from sequencing data. There are 7 sets of gene expression profiles from human and mouse cells, namely hESC, hHep, mDC, mESC, mHSC-E, mHSC-GM, and mHSC-L. The top 500 (1,000) high-variance genes and their corresponding transcription factors are selected from each dataset, matching the "TFs + 500 (1,000) genes" in the BEELINE benchmark. Although BEELINE provides cell-type-specific and non-cell-type-specific GRNs as reference networks, these reference networks have not been updated since 2020. We retrieved the gene regulations from DoRothEA [12] and CollecTRI [28] and used them as reference networks. Their data statistics are shown in Table 2 in the Appendix.

We compare FlowGRN with the baselines on the BEELINE benchmark, which includes LEAP [40], SCODE [24], GRISLI [3], GRNVBEM [35], SINCERITIES [29], Scribe [32], GENIE3 [16], and GRNBOOST2 [26]. Among the listed baselines, only SCODE, GRNVBEM, SINCERITIES, and FlowGRN support inference of edge types (activation/inhibition). Here, we excluded SCNS [51] as it requires prior knowledge of GRN, and we excluded ppcor [20] and PIDC [5] because they cannot determine the direction of regulations.

We evaluate the performance of FlowGRN and the baselines on the BEELINE benchmark using Area under Precision-Recall Curve (AUPRC), which emphasizes the precision of the inferred GRN and minimizes the false positive rate, with higher values indicating better performance.

5.2 Experimental Setup

We first construct a series of snapshots for each dataset by clustering cells and ordering each cell cluster by their average pseudotime. Details of the preprocessing are described in Appendix A.1. During trajectory reconstruction, v_θ, s_θ in the $[\mathrm{SF}]^2\mathrm{M}$ model are implemented as MLPs with 5 hidden layers, each with 128 hidden units and SELU activations. Both neural networks are optimized by AdamW with a learning rate of 10^{-4} decayed by a factor of 2 every 100 epochs. Models are trained for 1,000 epochs with a batch size of 64 samples. Trajectories are integrated using a DOPRI5 neural ODE solver [30] with an absolute and relative tolerance of 10^{-7}. All neural network models are trained on a single NVIDIA V100 SXM2 GPU with 16GB memory, although the peak memory usage is less than 1GB and can be run on a domestic computer.

In GRN inference, dynGENIE3 is used with the default parameters: 1,000 random forest trees, each trained with $\sqrt{g}$ genes. dynGENIE3 is trained on a computing node with 128 CPUs at 2.6GHz and 256GB memory. All baselines are also run on the same platform within a given 48-hour time interval, using their default parameters wrapped in a Singularity container to ensure reproducibility. Each method, including FlowGRN, is run 10 times with different random seeds on each dataset. Model elapsed times are recorded and reported in Table 5 in the Appendix for scalability analysis.

6 Results and Discussion

6.1 FlowGRN infers identifiable GRNs in simulated data

We first evaluate FlowGRN on the BEELINE benchmark using both synthetic and curated datasets, and the average AUPRC is shown in Figures 2. FlowGRN has the highest average rank of AUPRC among all models, significantly surpassing the second-best GENIE3.

Although FlowGRN adopts dynGENIE3, a temporal extension of GENIE3, FlowGRN outperforms GENIE3, especially in the LL and CY datasets. The LL dataset features long cascades, indicating that FlowGRN effectively captures long-range gene expression dependencies. The CY dataset is characterized by cyclic dynamics, thus determining the edge direction is crucial for the GRN inference. Incorporating temporal information enables FlowGRN to capture the principal direction of cyclic dynamics, which static models like GENIE3 struggle with.

6.2 FlowGRN determines edge types in simulated data

We further test FlowGRN's ability to determine the activation and inhibition edges on top of the directed GRN inference. Figure 3 shows the average AUPRC of FlowGRN and the baselines on the BEELINE benchmark with edge types. Note that there are fewer

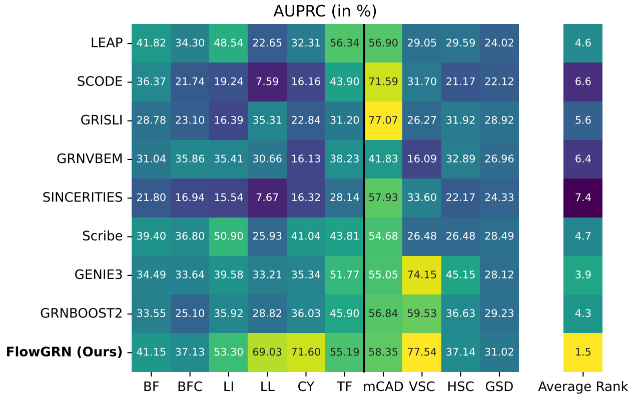

Figure 2: Average AUPRC and its average rank of directed GRN inference models on simulated datasets out of 10 runs.

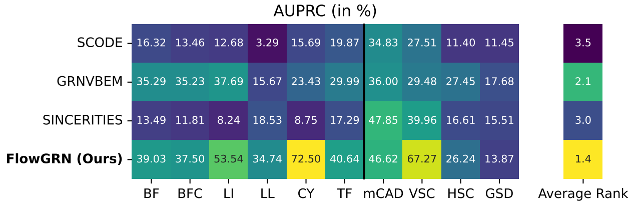

Figure 3: Average AUPRC of directed GRN inference models with edge types on simulated datasets out of 10 runs.

baselines capable of inferring edge types. FlowGRN achieves the best performance in 8 out of 10 datasets and remarkably surpasses the second-best model, GRNVBEM.

6.3 FlowGRN is a top-tier model on experimental data

We compared FlowGRN with the baselines on 7 experimental datasets, and their performances are listed in Table 1. The main difficulties of these datasets are the high dimensionality and the presence of dropout, as discussed in Sections 2.2 and 2.3. Most baselines successfully completed the tasks, except for GRNVBEM, which failed to complete half of its tasks within a 48-hour time span, and SINCERITIES, which exhibits numerical errors on the mHSC-E and mHSC-L datasets. The model run times are reported in Table 5 in the Appendix. Our model is as scalable as GENIE3 and GRNBOOST2, which can be trained in less than 10 minutes.

LEAP, GENIE3, and FlowGRN are the top-tier models on both metrics, ranking 1^{st} or 2^{nd} in over half of the tasks. LEAP, GENIE3, and FlowGRN all adopt the idea of feature selection in their methodologies (Section 2.2), suggesting that feature selection is crucial for GRN inference in high-dimension with sparse interactions.

Our model demonstrates scalability up to 1,783 genes. However, it achieves only average performance in the mDC and mESC datasets. This can be attributed to the low number of cells (383 and 421 cells) in both datasets, which is approximately half the number of cells in the other datasets. Insufficient cell samples may result in inadequate information for the model to accurately capture complex gene interactions, particularly in deep learning models. We believe that this issue can be addressed by simplifying the model's complexity or incorporating biological priors into the CFM.

6.4 Trajectory reconstruction is irreplaceable

We performed an ablation study on the use of reconstructed trajectories in the GENIE3 and FlowGRN models, and the results are shown in Table 3 in the Appendix. GENIE3 initially relies on static gene expression data, but we may also use the restored trajectories as augmentations. The latter approach is labeled as "GENIE3 (w/ traj.)". In FlowGRN, the trajectories reconstructed by the CFM are fed to the dynGENIE3 model. However, we may also use the trained vector field v_θ from the CFM model to reconstruct GRN directly from its Jacobian $\nabla_{x'} v_\theta$, labeled as "FlowGRN (w/o traj.)".

This ablation study demonstrates that trajectory reconstruction is an indispensable component in the FlowGRN model, as the performance of FlowGRN without trajectories is significantly lower than that with trajectories. This is because the chain rule allows one gene to influence another indirectly via other genes, making it challenging to discern direct regulations from gene expression.

However, the effect of trajectory is almost tied in the GENIE3 model. Although augmented trajectories may be helpful in some instances, the formulation of GENIE3 is not designed to leverage these trajectories. Although both GENIE3 and dynGENIE3 train random forest models and extract the network in the same way, GENIE3 aims to capture interactions from static states, while dynGENIE3 focuses on dynamic modeling. Thus, the performance of GENIE3 with dynamic information is not significantly better than that without trajectories. Additionally, data augmentation introduces noise to GENIE3, which may also impact performance.

6.5 Dropout-robust measure boosts GRN inference

We conducted another ablation study to justify the design of our dropout-robust measure on the experimental datasets. We compared the FlowGRN model with and without the dropout-robust measure, and with and without the kNN graph geodesic distance. Both components are defined in Section 4.1, and the results are shown in Table 4 in the Appendix. Although performance differences are minor and within standard deviations, results show that the dropout-robust measure d is the primary driver of the improvement, especially on the "TFs + 1,000 genes" dataset, suggesting that the dropout-robust measure is essential in our noise-corrupted task.

7 Conclusion and Limitations

In this paper, we propose FlowGRN, a novel deep learning model for inferring GRNs from scRNA-seq data. FlowGRN is a combination of a CFM model and an ensemble tree model that reconstructs gene expression trajectories and infers the GRN from the reconstructed trajectories. To mitigate the impact of dropout, we proposed a dropout-robust cell similarity measure that captures cell-cell similarity in the presence of dropout.

We evaluated FlowGRN on the BEELINE benchmark and compared it with eight state-of-the-art GRN inference models. FlowGRN demonstrated top-tier performance in both synthetic and curated

Table 1: AUPRC ($\times 10^{-3}$) of directed GRN inference models on experimental datasets out of 10 runs. The best and second-best results are highlighted in bold and underlined, respectively. "OT" and "NE" mean "over time in a 48-hour time span" and "numerical error", respectively.

dataset	hESC	hHep	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
TFs + 500 genes							
LEAP	11.91±0.00	**12.99±0.00**	2.38±0.00	**5.36±0.00**	1.94±0.00	2.00±0.00	1.28±0.00
SCODE	10.00±0.10	10.56±0.04	2.01±0.04	0.84±0.03	1.50±0.02	1.30±0.03	0.94±0.01
GRISLI	11.05±0.00	10.57±0.00	**2.95±0.00**	0.51±0.00	1.65±0.00	1.21±0.00	1.29±0.00
GRNVBEM	OT	OT	2.34±0.00	0.55±0.00	1.62±0.00	1.39±0.00	1.14±0.00
SINCERITIES	12.47±0.00	10.60±0.00	2.69±0.00	0.51±0.00	1.49±0.00	1.37±0.00	1.18±0.00
Scribe	10.67±0.00	9.64±0.00	1.83±0.00	0.53±0.00	1.65±0.00	1.38±0.00	1.12±0.00
GENIE3	10.50±0.01	9.75±0.02	2.10±0.01	3.01±0.10	3.02±0.43	2.44±0.08	1.95±0.10
GRNBOOST2	11.97±0.02	11.42±0.02	2.32±0.01	2.45±0.62	1.78±0.04	1.93±0.11	1.63±0.37
FlowGRN (Ours)	**12.53±0.11**	12.66±0.19	2.32±0.03	1.83±0.15	**3.36±0.23**	**3.33±0.14**	**2.35±0.14**
TFs + 1,000 genes							
LEAP	8.79±0.00	**8.99±0.00**	1.38±0.00	**1.89±0.00**	1.01±0.00	1.01±0.00	0.66±0.00
SCODE	6.68±0.08	6.74±0.08	1.08±0.01	0.52±0.01	0.63±0.01	0.58±0.01	0.47±0.01
GRISLI	7.71±0.00	7.01±0.00	**1.72±0.00**	0.42±0.00	0.72±0.00	0.55±0.00	0.63±0.00
GRNVBEM	OT	OT	OT	0.42±0.00	OT	OT	0.58±0.00
SINCERITIES	8.57±0.00	6.95±0.00	1.47±0.00	0.37±0.00	NE	0.62±0.00	NE
Scribe	7.30±0.00	6.17±0.00	0.99±0.00	0.48±0.00	0.71±0.00	0.65±0.00	0.63±0.00
GENIE3	7.19±0.01	6.25±0.02	1.17±0.01	1.11±0.03	1.30±0.08	1.14±0.04	0.88±0.02
GRNBOOST2	8.29±0.01	7.48±0.02	1.33±0.01	0.88±0.07	0.82±0.02	0.88±0.05	0.70±0.05
FlowGRN (Ours)	**8.82±0.15**	8.98±0.76	1.27±0.02	0.81±0.05	**1.60±0.11**	**1.56±0.07**	**0.93±0.04**

datasets, achieving the best performance in 6 of 10 tasks. It has advantages in learning regulation types and modeling dynamics with long cascades and cycles, which are well-known challenges in GRN inference. FlowGRN is also scalable to high-dimensional experimental datasets of up to 1,783 genes with dropout, thanks to the advanced CFM model and the dropout-robust cell similarity measure. We also conducted ablation studies to validate the design of FlowGRN, demonstrating that the dropout-robust cell similarity measure and trajectory reconstruction are essential for our model.

FlowGRN has some limitations. First, FlowGRN exhibits a notable performance drop on datasets with few cells, such as mDC and mESC, which is a common pitfall of deep learning models. Second, our method cannot model the dynamics and infer network structure in a unified manner, which limits its interpretability. In the future, we will combine dynamic modeling and GRN inference in a unified framework by restricting the CFM model with biological constraints, following the approach of PINN. We will also replace the Euclidean distance in the CFM with the distance induced by the data manifold. We expect these improvements to further enhance the performance of FlowGRN on high-dimensional datasets.

Data and Code Availability

We do not generate any new data in this work. The code is available at https://github.com/1250326/FlowGRN.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

Authors Tsz Pan Tong and Jun Pang acknowledge financial support from the Institute for Advanced Studies of the University of Luxembourg through an Audacity Grant (AUDACITY-2021). This work was also supported by the Luxembourg National Research Fund (FNR) under the grant agreement INTER/NCN/24/18732364/EdgeCR.

References

[1] Sara Aibar, Carmen Bravo González-Blas, Thomas Moerman, Vân Anh Huynh-Thu, Hana Imrichova, Gert Hulselmans, Florian Rambow, Jean-Christophe Marine, Pierre Geurts, Jan Aerts, et al. 2017. SCENIC: Single-cell regulatory network inference and clustering. *Nature Methods* 14, 11 (2017), 1083–1086.

[2] Michael S Albergo, Michael Lindsey, Nicholas M Boffi, and Eric Vanden-Eijnden. 2024. Multimarginal generative modeling with stochastic interpolants. In *12th International Conference on Learning Representations, ICLR 2024*.

[3] Pierre-Cyril Aubin-Frankowski and Jean-Philippe Vert. 2020. Gene regulation inference from single-cell RNA-seq data with linear differential equations and velocity inference. *Bioinformatics* 36, 18 (2020), 4774–4780.

[4] Pau Badia-i Mompel, Lorna Wessels, Sophia Müller-Dott, Rémi Trimbour, Ricardo O Ramirez Flores, Ricard Argelaguet, and Julio Saez-Rodriguez. 2023. Gene regulatory network inference in the era of single-cell multi-omics. *Nature Reviews Genetics* 24, 11 (2023), 739–754.

[5] Thalia E Chan, Michael PH Stumpf, and Ann C Babtie. 2017. Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures. *Cell Systems* 5, 3 (2017), 251–267.

[6] Jiaxing Chen, ChinWang Cheong, Liang Lan, Xin Zhou, Jiming Liu, Aiping Lyu, William K Cheung, and Lu Zhang. 2021. DeepDRIM: a deep neural network to reconstruct cell-type-specific gene regulatory network using single-cell RNA-seq data. *Briefings in Bioinformatics* 22, 6 (2021), bbab325.

[7] Shuonan Chen and Jessica C Mar. 2018. Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data. *BMC Bioinformatics* 19, 1 (2018), 232.

[8] Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. 2019. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature Communications* 10, 1 (2019), 390.

[9] Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner. 2007. Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles. *PLOS Biology* 5, 1 (2007), e8.

[10] Yue Fan and Xiuli Ma. 2021. Gene regulatory network inference using 3D convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. AAAI Press, 99–106.

[11] Jonas Simon Fleck, Sophie Martina Johanna Jansen, Damian Wollny, Fides Zenk, Makiko Seimiya, Akanksha Jain, Ryoko Okamoto, Malgorzata Santel, Zhisong He, J Gray Camp, et al. 2023. Inferring and perturbing cell fate regulomes in human brain organoids. *Nature* 621, 7978 (2023), 365–372.

[12] Luz Garcia-Alonso, Christian H. Holland, Mahmoud M. Ibrahim, Denes Turei, and Julio Saez-Rodriguez. 2019. Benchmark and integration of resources for the estimation of human transcription factor activities. *Genome Research* 29, 8 (2019), 1363–1375.

[13] Anne-Claire Haury, Fantine Mordelet, Paola Vera-Licona, and Jean-Philippe Vert. 2012. TIGRESS: Trustful Inference of Gene REgulation using Stability Selection. *BMC Systems Biology* 6, 1 (2012), 145.

[14] Mo Huang, Jingshu Wang, Eduardo Torre, Hannah Dueck, Sydney Shaffer, Roberto Bonasio, John I Murray, Arjun Raj, Mingyao Li, and Nancy R Zhang. 2018. SAVER: gene expression recovery for single-cell RNA sequencing. *Nature Methods* 15, 7 (2018), 539–542.

[15] Vân Anh Huynh-Thu and Pierre Geurts. 2018. dynGENIE3: dynamical GENIE3 for the inference of gene networks from time series expression data. *Scientific Reports* 8, 1 (2018), 3384.

[16] Vân Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts. 2010. Inferring regulatory networks from expression data using tree-based methods. *PLOS ONE* 5, 9 (2010), e12776.

[17] Ruochen Jiang, Tianyi Sun, Dongyuan Song, and Jingyi Jessica Li. 2022. Statistics or biology: the zero-inflation controversy about scRNA-seq data. *Genome Biology* 23, 1 (2022), 31.

[18] Kenji Kamimoto, Blerta Stringa, Christy M Hoffmann, Kunal Jindal, Lilianna Solnica-Krezel, and Samantha A Morris. 2023. Dissecting cell identity via network inference and in silico gene perturbation. *Nature* 614, 7949 (2023), 742–751.

[19] Philipp Keyl, Philip Bischoff, Gabriel Dernbach, Michael Bockmayr, Rebecca Fritz, David Horst, Nils Blüthgen, Grégoire Montavon, Klaus-Robert Müller, and Frederick Klauschen. 2023. Single-cell gene regulatory network prediction by explainable AI. *Nucleic Acids Research* 51, 4 (2023), e20.

[20] Seongho Kim. 2015. ppcor: An R Package for a Fast Calculation to Semi-partial Correlation Coefficients. *Communications for Statistical Applications and Methods* 22, 6 (2015), 665–674.

[21] Jens Uwe Loers and Vanessa Vermeirssen. 2024. A single-cell multimodal view on gene regulatory network inference from transcriptomics and chromatin accessibility data. *Briefings in Bioinformatics* 25, 5 (2024), bbae382.

[22] Lam-Ha Ly and Martin Vingron. 2022. Effect of imputation on gene network reconstruction from single-cell RNA-seq data. *Patterns* 3, 2 (2022), 11 pages.

[23] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. 2006. ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics* 7, 1 (2006), S7.

[24] Hirotaka Matsumoto, Hisanori Kiryu, Chikara Furusawa, Minoru SH Ko, Shigeru BH Ko, Norio Gouda, Tetsutaro Hayashi, and Itoshi Nikaido. 2017. SCODE: an efficient regulatory network inference algorithm from single-cell RNA-Seq during differentiation. *Bioinformatics* 33, 15 (2017), 2314–2321.

[25] Leland McInnes, John Healy, and James Melville. 2018. *UMAP: Uniform manifold approximation and projection for dimension reduction.* arXiv:1802.03426 doi:10.48550/arXiv.1802.03426

[26] Thomas Moerman, Sara Aibar Santos, Carmen Bravo González-Blas, Jaak Simm, Yves Moreau, Jan Aerts, and Stein Aerts. 2019. GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics* 35, 12 (2019), 2159–2161.

[27] Kevin R Moon, David Van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, William S Chen, Kristina Yim, Antonia van den Elzen, Matthew J Hirn, Ronald R Coifman, et al. 2019. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology* 37, 12 (2019), 1482–1492.

[28] Sophia Müller-Dott, Eirini Tsirvouli, Miguel Vazquez, Ricardo O Ramirez Flores, Pau Badia-i Mompel, Robin Fallegger, Dénes Türei, Astrid Lægreid, and Julio Saez-Rodriguez. 2023. Expanding the coverage of regulons from high-confidence prior knowledge for accurate estimation of transcription factor activities. *Nucleic Acids Research* 51, 20 (2023), 10934–10949.

[29] Nan Papili Gao, SM Minhaz Ud-Dean, Olivier Gandrillon, and Rudiyanto Gunawan. 2018. SINCERITIES: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. *Bioinformatics* 34, 2 (2018), 258–266.

[30] Michael Poli, Stefano Massaroli, Atsushi Yamashita, Hajime Asama, Jinkyoo Park, and Stefano Ermon. 2021. TorchDyn: implicit models and neural numerical methods in PyTorch. In *Neural Information Processing Systems, Workshop on Physical Reasoning and Inductive Biases for the Real World*, Vol. 2.

[31] Aditya Pratapa, Amogh P Jalihal, Jeffrey N Law, Aditya Bharadwaj, and TM Murali. 2020. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature Methods* 17, 2 (2020), 147–154.

[32] Xiaojie Qiu, Arman Rahimzamani, Li Wang, Bingcheng Ren, Qi Mao, Timothy Durham, José L McFaline-Figueroa, Lauren Saunders, Cole Trapnell, and Sreeram Kannan. 2020. Inferring Causal Gene Regulatory Networks from Coupled Single-Cell Expression Dynamics Using Scribe. *Cell Systems* 10, 3 (2020), 265–274.

[33] Caleb C Reagor, Nicolas Velez-Angel, and A J Hudspeth. 2023. Depicting pseudotime-lagged causality across single-cell trajectories for accurate gene-regulatory inference. *PNAS Nexus* 2, 4 (2023), pgad113.

[34] Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. 2019. A comparison of single-cell trajectory inference methods. *Nature Biotechnology* 37, 5 (2019), 547–554.

[35] Manuel Sanchez-Castillo, David Blanco, Isabel M Tienda-Luna, MC Carrion, and Yufei Huang. 2018. A Bayesian framework for the inference of gene regulatory networks from time and pseudo-time series data. *Bioinformatics* 34, 6 (2018), 964–970.

[36] Manu Setty, Michelle D Tadmor, Shlomit Reich-Zeliger, Omer Angel, Tomer Meir Salame, Pooja Kathail, Kristy Choi, Sean Bendall, Nir Friedman, and Dana Pe'er. 2016. Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nature biotechnology* 34, 6 (2016), 637–645.

[37] Yutong Sha, Yuchi Qiu, Peijie Zhou, and Qing Nie. 2024. Reconstructing growth and dynamic trajectories from single-cell transcriptomics data. *Nature Machine Intelligence* 6, 1 (2024), 25–39.

[38] Hantao Shu, Jingtian Zhou, Qiuyu Lian, Han Li, Dan Zhao, Jianyang Zeng, and Jianzhu Ma. 2021. Modeling gene regulatory networks using neural network architectures. *Nature Computational Science* 1, 7 (2021), 491–501.

[39] David M Simcha, Laurent Younes, Martin J Aryee, and Donald Geman. 2013. Identification of direction in gene networks from expression and methylation. *BMC systems biology* 7 (2013), 1–15.

[40] Alicia T Specht and Jun Li. 2017. LEAP: constructing gene co-expression networks for single-cell RNA-sequencing data using pseudotime ordering. *Bioinformatics* 33, 5 (2017), 764–766.

[41] Kelly Street, Davide Risso, Russell B Fletcher, Diya Das, John Ngai, Nir Yosef, Elizabeth Purdom, and Sandrine Dudoit. 2018. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* 19 (2018), 1–16.

[42] Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. 2020. TrajectoryNet: A Dynamic Optimal Transport Network for Modeling Cellular Dynamics. In *37th International Conference on Machine Learning*. PMLR, 9526–9536.

[43] Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. 2024. Simulation-free Schrödinger bridges via score and flow matching. In *27th International Conference on Artificial Intelligence and Statistics*. 1279–1287.

[44] Koen Van den Berge, Hector Roux de Bézieux, Kelly Street, Wouter Saelens, Robrecht Cannoodt, Yvan Saeys, Sandrine Dudoit, and Lieven Clement. 2020. Trajectory-based differential expression analysis for single-cell sequencing data. *Nature communications* 11, 1 (2020), 1201.

[45] Monique GP Van Der Wijst, Dylan H de Vries, Harm Brugge, Harm-Jan Westra, and Lude Franke. 2018. An integrative approach for building personalized gene regulatory networks for precision medicine. *Genome Medicine* 10 (2018), 1–15.

[46] David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al. 2018. Recovering gene interactions from single-cell data using data diffusion. *Cell* 174, 3 (2018), 716–729.

[47] Florian Wagner, Yun Yan, and Itai Yanai. 2017. K-nearest neighbor smoothing for high-throughput single-cell RNA-Seq data. *BioRxiv* (2017), 217737.

[48] Aoran Wang and Jun Pang. 2022. Iterative structural inference of directed graphs. In *36th International Conference on Neural Information Processing Systems*. Curran Associates, Inc., 8717–8730.

[49] Aoran Wang, Tsz Pan Tong, Andrzej Mizera, and Jun Pang. 2024. Benchmarking Structural Inference Methods for Interacting Dynamical Systems with Synthetic Data. In *Advances in Neural Information Processing Systems*, Vol. 37. Curran Associates, Inc., 135129–135185.

[50] Aoran Wang, Tsz Pan Tong, and Jun Pang. 2023. Effective and Efficient Structural Inference with Reservoir Computing. In *40th International Conference on Machine Learning*. PMLR, 36391–36410.

[51] Steven Woodhouse, Nir Piterman, Christoph M Wintersteiger, Berthold Göttgens, and Jasmin Fisher. 2018. SCNS: a graphical tool for reconstructing executable regulatory networks from single-cell genomic data. *BMC Systems Biology* 12 (2018), 1–7.

Received 16 June 2025; accepted 28 August 2025