

Supplementary Information

I. PARAMETER SETTINGS

We trained spaNetReg using PyTorch [1]. Prior to training, we performed data partitioning and preprocessing. Specifically, 5% of the dataset was held out as a validation set. Given the class imbalance in the initial adjacency matrix (with more negative links than positive ones), we constructed the validation set by randomly selecting 5% of the positive samples and pairing them with an equal number of negative samples. The remaining samples were used for training.

The training of spaNetReg was conducted in two stages. In the first stage, we pre-trained the model using a VGAE to provide a stable initialization of the graph embeddings and accelerate convergence of the full model. The VGAE encoder consisted of two Graph Convolutional Network (GCN) layers with dimensions (64, 32), with a Layer Normalization layer inserted between them to improve stability. The model was optimized using the Adam optimizer with a learning rate of 0.001, early stopping (patience = 1000) and a maximum of 5000 epochs.

In the second stage, we trained the full model by incorporating the VAE component. The VAE encoder used fully connected layers with batch normalization: (128, 64, 32). The final embedding dimension was 32, including 5 dimensions for GP embeddings and 27 for standard Gaussian embeddings. Each GP latent dimension can be interpreted as representing a distinct spatial trend. The combination of GP and standard Gaussian embeddings allows the model to capture both spatially correlated and non-spatial variations. Increasing the number of GP dimensions (L) enhances the model's ability to represent complex spatial structures but also increases computational cost; for datasets exhibiting strong spatial correlations, L can be increased to 8–10 dimensions.

For the GP component, we adopted the MultiMatern kernel as the covariance function, following the design of spaVAE. The MultiMatern kernel captures multi-scale spatial dependencies by combining multiple Matern kernels with different smoothness parameters $\nu \in \{0.5, 1.5, 2.5\}$. Each component kernel $k_\nu(r)$ is defined as:

$$k_\nu(r) = \begin{cases} \exp(-\sqrt{2\nu} r/\rho), & \text{if } \nu = 0.5, \\ (1 + \sqrt{3} r/\rho) \exp(-\sqrt{3} r/\rho), & \text{if } \nu = 1.5, \\ \left(1 + \sqrt{5} r/\rho + \frac{5r^2}{3\rho^2}\right) \exp(-\sqrt{5} r/\rho), & \text{if } \nu = 2.5, \end{cases} \quad (1)$$

where $r = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ denotes the Euclidean distance, and ρ is the learnable length-scale parameter controlling the spatial range. In implementation, each GP latent dimension adopts an

independent scale parameter ρ_l , allowing multi-scale spatial representation across latent channels. In all experiments, the smoothness parameter was set to $\nu = 1.5$.

We applied sparse GP regression using inducing points, where the inducing points were selected using a 2D grid method with grid step n , yielding $(n + 1)^2$ inducing points. By default, we set the inducing point steps of 10 and the mini-batch size of 256. During training, the KL divergence term in the VAE was weighted by $\beta = 10$, and the VGAE loss was weighted by 10^3 . The learning rate for this stage was 10^{-5} , with early stopping (patience = 150) and a maximum of 5000 epochs.

To build a binary adjacency matrix, we performed 100 random permutations of the learned graph embeddings to construct a null distribution. The 80th percentile of this null distribution was used as a significance threshold, and only regulatory links with predicted probabilities exceeding this threshold were retained as valid regulatory links.

II. BENCHMARK METHODS FOR TRN INFERENCE

A. Data Design and Simulation for Benchmarking

We randomly constructed 20 TRNs, each consisting of interactions among 100 transcription factors. Based on these ground truth networks, we used the scMultiSim [2] tool to simulate synthetic spatial multi-omics datasets, which simultaneously simulate spatial transcriptomics (ST) and spatial ATAC-seq data. Each dataset includes the following components: spatial coordinates of 2,000 cells, organized into five spatial clusters. A gene expression matrix with 200 genes. A chromatin accessibility matrix with 600 peaks, mapped to genes at a 1:3 ratio. To systematically evaluate model performance, we further generated datasets under varying conditions, including different cell counts (500–2,000 cells), noise levels (from 0.1 to 1.0), and dropout rates (from 0.6 to 0.9), allowing us to comprehensively assess the robustness of GRN inference across diverse scenarios.

B. Algorithm Comparison

We applied the BEELINE [3] framework to benchmark TRN results obtained from different inference methods. Specifically, we evaluated the performance of DeepWalk [4], GENIE3 [5], GRNBoost2 [6], Celloracle [7], and DeepTFni [8] using the simulated datasets. All methods were implemented using publicly available source codes, and software provided by the original authors were used to ensure consistency. spaNetReg, DeepTFni, and DeepWalk all require a noisy TRN derived from ATAC-seq data as input. Peaks in the ATAC-seq data were assigned to nearby genes based on genomic distance, thereby constructing potential TF-target

links by identifying TF binding motifs within the peaks. From scMultiSim, we obtained the peak-to-gene matrix M_{rg} (gene activity matrix), and the binary peak-by-TF matrix M_{pt} . Noise was simulated by randomly removing connections in both matrices. The noisy TRN was constructed as $M_{rg} \cdot M_{pt}$. Additionally, the peak-to-gene matrix M_{rg} was multiplied by the ATAC-seq count matrix to generate a gene activity matrix of shape $n_{cell} \times n_{gene}$, which was used as node features in the graph-based models. For methods like GENIE3, GRNBoost2, and SCENIC, we used the simulated spatial mRNA expression matrix as input.

C. Evaluation Metrics

To standardize comparisons, we converted model outputs into weighted adjacency matrices. For DeepWalk, GENIE3, and DeepTFni, the outputs were directly used. For GRNBoost2 and SCENIC, we filled missing TF-target links with zeros. To assess model performance, we adopted two widely used metrics: area under the receiver operating characteristic curve (AUROC), area under the precision-recall curve (AUPRC), accuracy (ACC). Each method was executed 20 times on each of the 20 synthetic datasets to ensure robustness and statistical significance.

III. SIMULATION EXPERIMENT

This appendix presents supplementary materials from the simulation experiments. Figure S3 demonstrates the sensitivity analysis of spaNetReg with respect to key hyperparameters. The AUPRC remains stable across different settings of inducing point steps and β , indicating that the model is highly robust to these parameters. In contrast, increasing γ , which controls the weight of the VGAE loss, initially improves performance, as stronger graph-structural constraints enhance TRN reconstruction accuracy. However, further increasing γ does not yield additional performance gains. Based on these results, we set the default hyperparameters as follows: inducing point steps = 5, $\beta = 10$, and $\gamma = 10^3$. Figure S1 illustrates the accuracy of TRNs inferred by different methods, with the dashed line indicating the accuracy of the initial skeleton TRNs constructed during preprocessing. Figure S2 shows the t-SNE visualization of simulated datasets under different noise levels, demonstrating that varying degrees of noise were effectively introduced. Table S1 summarizes the performance of various methods in inferring cell type-specific TRNs from the simulated dataset comprising five cell types.

IV. SPACE-SEQ HUMAN HIPPOCAMPUS EXPERIMENT

This appendix presents the application of the spaNetReg model to SPACE-seq data from a human hippocampus sample. Figure S4 illustrates the spatial distribution and clustering results of the dataset, the t-SNE visualization of cell populations, the correlation of cell type-specific TRNs, and the identified core TFs within the granule cell layer (GCL). The model successfully inferred cell type-specific TRNs and identified the core TFs within the GCL.

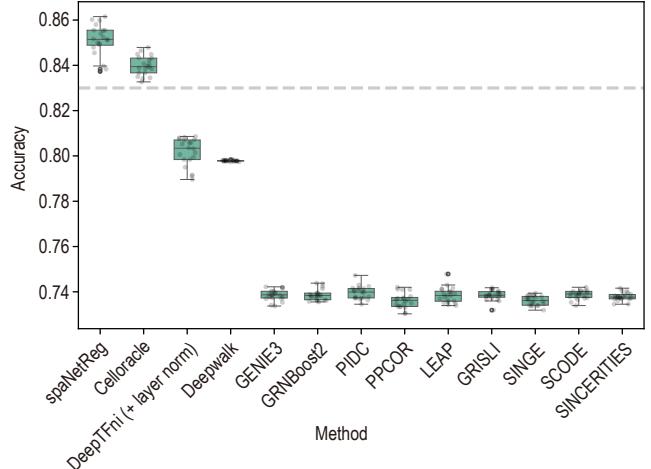


Fig. S1. Comparison of TRN inference accuracy across different models on simulated datasets. The dashed gray line indicates the accuracy of the initial skeleton TRNs constructed during preprocessing.

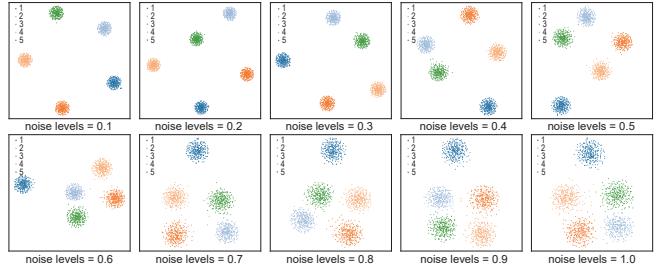


Fig. S2. t-SNE visualization of simulated datasets under different noise levels.

V. SPACE-SEQ HUMAN GLIOBLASTOMA EXPERIMENT

This appendix presents the application of SCENIC to the GBM dataset. As shown in the Figure S5, SCENIC inferred an extremely sparse TRN and failed to identify transcription factors that distinguish tumor from normal regions, highlighting its limitations for this type of analysis.

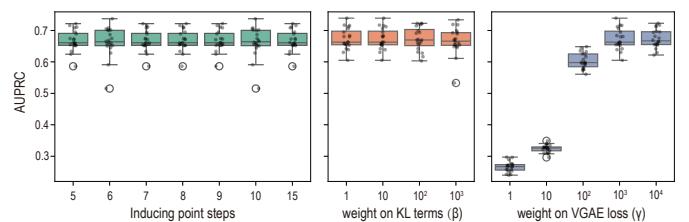


Fig. S3. Performance sensitivity of spaNetReg to key hyperparameters. Boxplots show the AUPRC values under different hyperparameter settings. The number of inducing point steps controls the granularity of the Gaussian Process approximation. The coefficient β regulates the balance of the KL divergence term (GP-VAE) in the loss function. The coefficient γ controls the weight of the VGAE loss, which governs the contribution of graph structure learning during training.

TABLE S1
PERFORMANCE COMPARISON (AUROC AND AUPRC) OF DIFFERENT MODELS ACROSS 5 CLUSTERS.

Model	Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5	
	AUROC ↑	AUPRC ↑								
Tree-based										
GENIE3	0.55	0.10	0.53	0.11	0.58	0.11	0.55	0.11	0.54	0.11
GRNBoost2	0.53	0.10	0.53	0.10	0.53	0.10	0.52	0.10	0.52	0.11
Information theory										
PIDC	0.49	0.04	0.52	0.05	0.50	0.04	0.48	0.04	0.50	0.05
Correlation										
PPCOR	0.50	0.04	0.50	0.05	0.50	0.05	0.50	0.06	0.50	0.05
LEAP	0.50	0.04	0.53	0.05	0.52	0.05	0.52	0.05	0.50	0.05
Dynamic										
GRISLI	0.40	0.03	0.45	0.04	0.44	0.04	0.41	0.04	0.43	0.04
SINGE	0.50	0.52	0.49	0.03	0.50	0.05	0.49	0.04	0.49	0.03
SCODE	0.46	0.04	0.46	0.04	0.49	0.04	0.48	0.04	0.49	0.05
SINCERITIES	0.48	0.04	0.50	0.04	0.52	0.05	0.53	0.05	0.51	0.05
Celloracle	0.77	0.27	0.77	0.28	0.76	0.31	0.77	0.28	0.79	0.36
Graph-based										
Deepwalk	0.88	0.39	0.88	0.39	0.90	0.46	0.88	0.40	0.86	0.35
DeepTFni (+layer norm)	0.85	0.37	0.85	0.38	0.83	0.36	0.82	0.34	0.85	0.37
spaNetReg	0.95	0.63	0.95	0.62	0.96	0.70	0.95	0.62	0.94	0.60

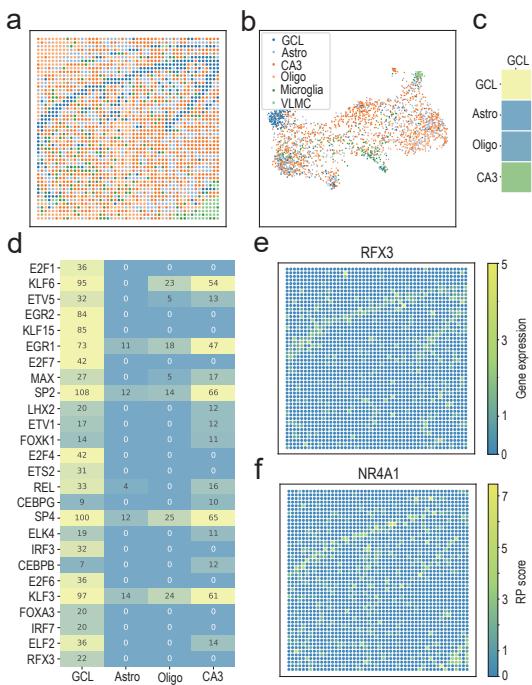


Fig. S4. spaNetReg infers cell-specific TRNs in spatial ATAC-RNA-seq human hippocampal data. a. Spatial distribution of human hippocampal tissue. b. t-SNE visualization of cell clusters in the human hippocampus. c. Correlation of TF degree vectors across different cell types in the human hippocampus. d. 26 TFs with cell-type specific regulatory networks. For each TF, its degree in four cell type are listed. e. Spatial expression pattern of RFX3 in the human hippocampus. f. The spatial distribution of the RP score for NR4A1, calculated by MAESTRO [9], reflects the regulatory activity of this transcription factor.

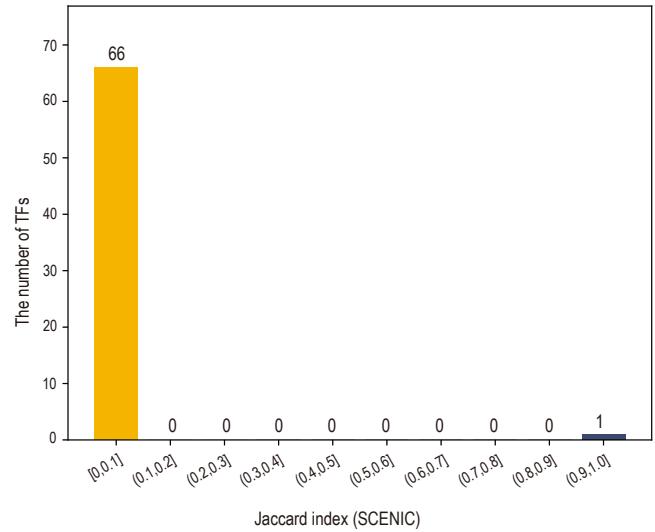


Fig. S5. The Jaccard index of core TF interactions between GBM and astrocytes, inferred by the SCENIC, orange indicates core TFs with TRN of dramatic change, navy blue indicates core TFs with GRN of moderate change.

REFERENCES

- [1] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. M. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in Neural Information Processing Systems 32 (Nips 2019)*, vol. 32, 2019.
- [2] H. Li, Z. Zhang, M. Squires, X. Chen, and X. Zhang, “scmultisim: simulation of single-cell multi-omics and spatial data guided by gene regulatory networks and cell-cell interactions,” *Nat Methods*, 2025.
- [3] A. Pratapa, A. P. Jalihal, J. N. Law, A. Bharadwaj, and T. M. Murali, “Benchmarking algorithms for gene regulatory network inference from

- single-cell transcriptomic data,” *Nat Methods*, vol. 17, no. 2, pp. 147–154, 2020.
- [4] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’14)*, pp. 701–710, 2014.
 - [5] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, “Inferring regulatory networks from expression data using tree-based methods,” *PLoS One*, vol. 5, no. 9, 2010.
 - [6] T. Moerman, S. Aibar Santos, C. Bravo Gonzalez-Blas, J. Simm, Y. Moreau, J. Aerts, and S. Aerts, “Grnboost2 and arboreto: efficient and scalable inference of gene regulatory networks,” *Bioinformatics*, vol. 35, no. 12, pp. 2159–2161, 2019.
 - [7] K. Kamimoto, B. Stringa, C. M. Hoffmann, K. Jindal, L. Solnica-Krezel, and S. A. Morris, “Dissecting cell identity via network inference and in silico gene perturbation,” *Nature*, vol. 614, no. 7949, pp. 742–751, 2023.
 - [8] H. Li, Y. Sun, H. Hong, X. Huang, H. Tao, Q. Huang, L. Wang, K. Xu, J. Gan, H. Chen, and X. Bo, “Inferring transcription factor regulatory networks from single-cell atac-seq data based on graph neural networks,” *Nature Machine Intelligence*, vol. 4, no. 4, pp. 389–400, 2022.
 - [9] C. Wang, D. Sun, X. Huang, C. Wan, Z. Li, Y. Han, Q. Qin, J. Fan, X. Qiu, Y. Xie, C. A. Meyer, M. Brown, M. Tang, H. Long, T. Liu, and X. S. Liu, “Integrative analyses of single-cell transcriptome and regulome using maestro,” *Genome Biol.*, vol. 21, no. 1, p. 198, 2020.