S1. Configuration of hyperparameters

In LineGRN, a Variational Autoencoder (VAE) is employed to reduce the dimensionality of the initial gene expression data. Subgraph extraction and subsequent line graph transformation are then applied to the prior regulatory network. During feature extraction, a Transformer encoder and Graph Convolutional Networks (GCNs) are integrated to capture expression-level and topological features, respectively. The model is trained under the supervision of three classification loss functions, each targeting a distinct objective to improve overall performance. Model parameters are optimized using the Adam optimizer. The hyperparameter settings used during training are shown in Supplymentary Table I.

Supplymentary Table I: The setting for the LineGRN model.

Parameter	Value	Explanation
hop	2	Enclosing subgraph hop number
max_nodes_per_hop	100	Upper bound of the number of nodes per hop at subgraph extraction
batch_size	100	Batch size for training
alpha	0.1	Weight for loss component from expression embeddings
beta	0.1	Weight for loss component from local embeddings
hidden	128	Hidden layer size of GCN
learning_rate	5e-4	Learning rate in training
nhead	10	Number of attention heads
layer	2	Number of GCN layers
window_size	80	Size of the sub-vectors.

S2. Dataset splitting details

We divide the dataset preparation process into two stages: positive sample splitting and negative sample sampling. For positive sample splitting, the observed edges in the prior network of each dataset are partitioned into training, validation, and test sets in a fixed ratio of 6:2:2, and are treated as positive samples. For negative sample sampling, we adopt different strategies based on the type of ground truth network to better reflect real-world data characteristics while maintaining high-quality training. In the case of cell-type-specific ground truth networks, which typically exhibit higher edge density, we perform negative sampling based on network density. This approach ensures that the ratio of positive to negative samples in the training, validation, and test sets closely aligns with the distribution observed in the ground truth network.

$$\frac{\text{Positive}}{\text{Negative}} = \frac{\text{NetworkDensity}}{1 - \text{NetworkDensity}}. \tag{1}$$

For non-specific ChIP-seq, STRING, and LOF/GOF ground truth networks, which typically exhibit relatively low edge density, performing negative sampling strictly based on network density would result in substantial class imbalance, i.e., a significantly larger number of negative samples compared to positive ones. Such imbalance may adversely affect model training and lead to biased predictions. To address this issue, we adopt a controlled negative sampling strategy for the training set, maintaining a fixed positive-to-negative ratio of 1:6. In contrast, for the validation and test sets, we continue to use the network density-based sampling strategy, ensuring consistency with the underlying characteristics of the ground truth network.

S3. Analysis of time and memory complexity

In this section, we empirically compare the time and memory complexity of LineGRN with seven baseline methods (GNNLink [1], GENELink [2], STGRNS [3], DeepDRIM [4], CNNC [5], DeepSEM [6] and GENIE3 [7]). To evaluate the scalability of each model, we selected two datasets from the non-specific ChIP-seq network type: a large-scale dataset (hHEP) and a smaller-scale dataset (mHSC-L). All experiments were conducted on a computing platform equipped with an Intel(R) Xeon(R) Gold 6240R CPU @ 2.40 GHz and an NVIDIA GeForce RTX 3090 GPU. The results are summarized in Supplymentary Tables II and III.

Supplymentary Table II: Comparison of computational complexity on hHEP dataset.

Complexity	LineGRN	GNNLink	STGRNS	GENELink	DeepDRIM	CNNC	DeepSEM	GENIE3
Running time	41 min 29 s	4.51 s	36.08 s	23.07 s	3 h 54 min 10 s	12 min 7 s	43.33 s	45.58 s
Used memory	36 GB	820 MB	1231 MB	1220 MB	2152 MB	1326 MB	4454 MB	9.98 MB

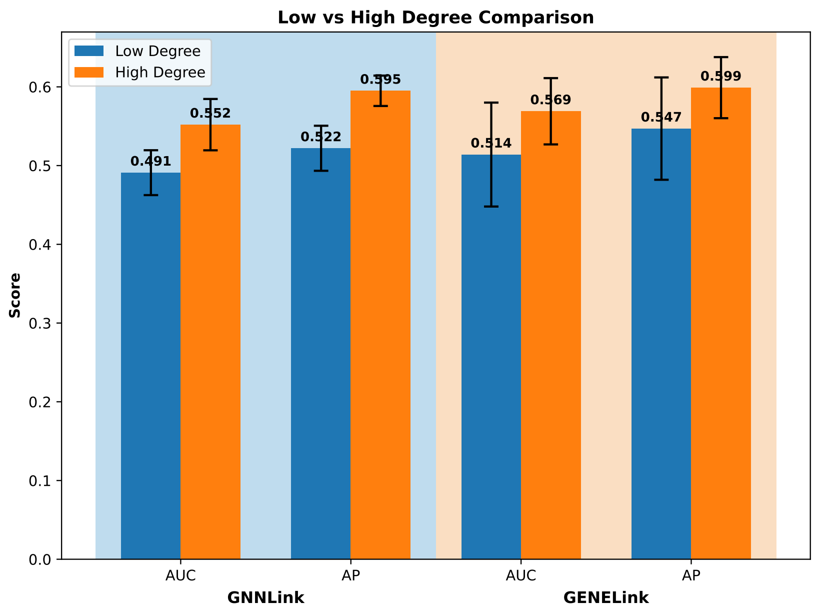
Supplymentary Table III: Comparison of computational complexity on mHSC-L dataset.

Complexity	LineGRN	GNNLink	STGRNS	GENELink	DeepDRIM	CNNC	DeepSEM	GENIE3
Running time	29.52 s	4.31 s	7.27 s	2.02 s	8 min 9 s	34.30 s	48.85 s	65.71 s
Used memory	2424 MB	779 MB	1212 MB	1213 MB	2139 MB	1320 MB	4446 MB	11.86 MB

Due to the structural complexity of our model and the need to extract subgraphs and construct line graphs for each TF-target gene pair, both time and memory consumption are relatively higher. However, by fully leveraging the rich structural information embedded in the graphs and incorporating multi-level feature fusion, the model achieves significantly improved accuracy in TF-target gene regulatory prediction, thereby fulfilling the primary objective of its design.

S4. Simulation studies

In the simulation experiment, we generated two synthetic datasets, each comprising 200 genes and 1,000 cells. Gene expression profiles were sampled from a standard normal distribution $\mathcal{N}(0, 1)$. To construct the underlying gene regulatory networks, we randomly generated positive edges between gene pairs, resulting in two distinct network structures: one dominated by low-degree nodes and the other by high-degree nodes, referred to as the low-degree graph and high-degree graph, respectively. In the low-degree graph, 90% of nodes exhibit low connectivity (degree), with the remaining 10% being high-degree hubs. The high-degree graph follows the opposite pattern. For each node, its positive edges were split into training, validation, and test sets in an 8:1:1 ratio. Negative edges were then generated by randomly sampling non-connected gene pairs, using a 1:1 positive-to-negative ratio to construct the final datasets. To evaluate the impact of graph structural differences on model performance, we applied two existing graph-based baseline methods, GNNLink [1] and GENELink [2], and conducted 10 independent runs for each graph structure. The results are summarized in Supplymentary Figure I.



Supplymentary Figure I: Performance comparison between low-degree and high-degree graph structures.

The results show that both graph-based models perform better on the high-degree graph. This confirms that a high-degree-node-dominated structure facilitates information propagation in graph neural networks, thereby effectively improving link prediction performance.

S5. Ablation experiments for loss function

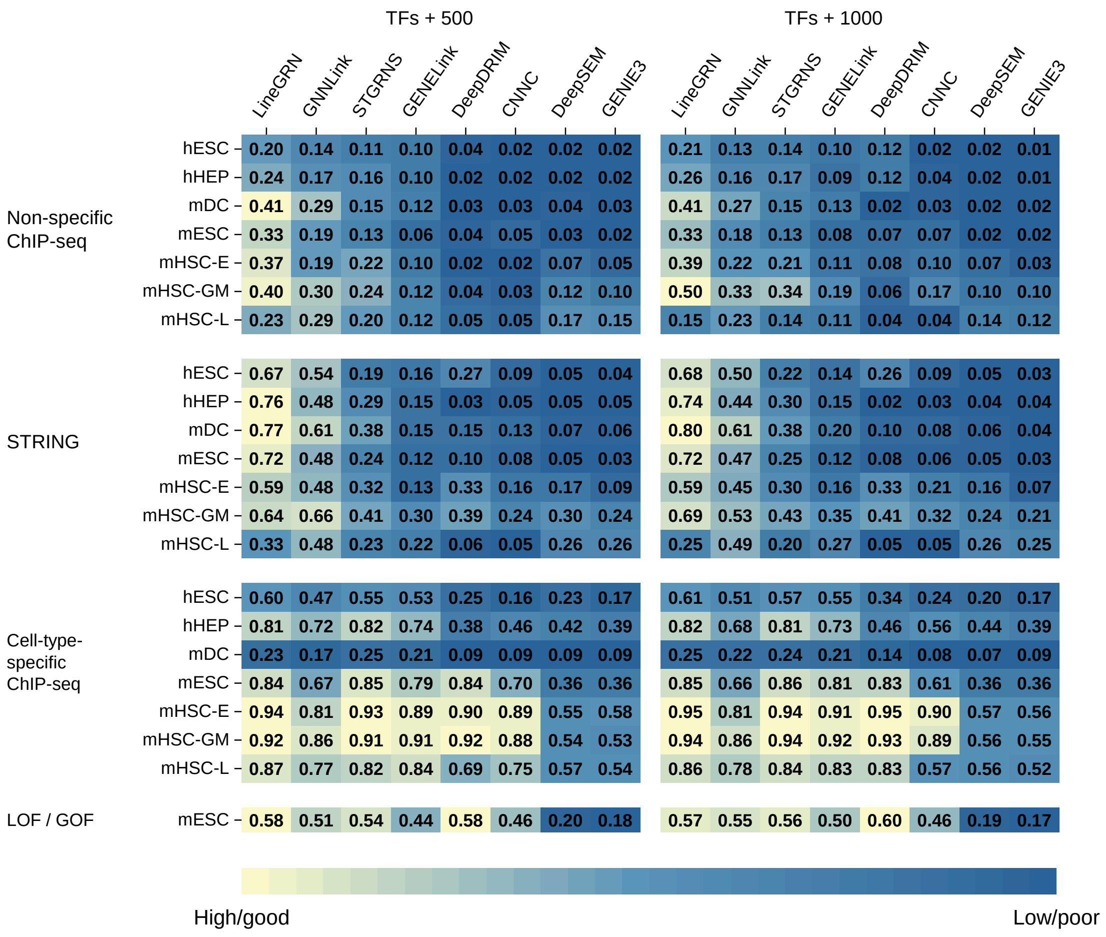
In the loss function, $\mathcal{L}_e$ and $\mathcal{L}_l$ independently constrain the classification capability of features extracted by the Expression Encoder and the Line Graph Encoder, respectively. In contrast, $\mathcal{L}_c$ is computed based on fused embeddings obtained by concatenating the expression embeddings and local graph embeddings, which are subsequently passed through multilayer perceptrons (MLPs) for classification. This component guides the model to perform classification based on the integrated representation, distinguishing it from $\mathcal{L}_e$ and $\mathcal{L}_l$, which constrain the individual encoders. To evaluate the contribution of each loss component, we conducted ablation experiments by selectively removing them. The results are summarized in Supplementary Table IV.

Supplymentary Table IV: Ablation study results.

Dataset	**w/o $\mathcal{L}_c$** AUROC	AUPRC	**w/o $\mathcal{L}_e$ and $\mathcal{L}_l$** AUROC	AUPRC	**LineGRN** AUROC	AUPRC
hESC	0.699	0.051	0.865	0.138	**0.880**	**0.197**
hHEP	0.614	0.023	0.887	0.183	**0.898**	**0.239**
mDC	0.823	0.078	0.909	0.276	**0.927**	**0.413**
mHSC-E	0.656	0.035	0.869	0.267	**0.895**	**0.373**
mHSC-GM	0.327	0.021	0.842	0.288	**0.860**	**0.397**
mHSC-L	0.542	0.055	0.806	0.159	**0.820**	**0.233**

The results show that removing any of the three loss components leads to a decline in model performance. These findings highlight the importance of all three loss functions, as each constrains distinct aspects of the model. Joint optimization of $\mathcal{L}_e$, $\mathcal{L}_l$, and $\mathcal{L}_c$ enables the model to effectively integrate complementary information from both encoders, ultimately enhancing overall performance.

S6. AUPRC scores of various methods on benchmark datasets



TFs + 500

		LineGRN	GNNLink	STGRNS	GENELink	DeepDRIM	CNNC	DeepSEM	GENIE3
Non-specific ChIP-seq	hESC	0.20	0.14	0.11	0.10	0.04	0.02	0.02	0.02
	hHEP	0.24	0.17	0.16	0.10	0.02	0.02	0.02	0.02
	mDC	0.41	0.29	0.15	0.12	0.03	0.03	0.04	0.03
	mESC	0.33	0.19	0.13	0.06	0.04	0.05	0.03	0.02
	mHSC-E	0.37	0.19	0.22	0.10	0.02	0.02	0.07	0.05
	mHSC-GM	0.40	0.30	0.24	0.12	0.04	0.03	0.12	0.10
	mHSC-L	0.23	0.29	0.20	0.12	0.05	0.05	0.17	0.15
STRING	hESC	0.67	0.54	0.19	0.16	0.27	0.09	0.05	0.04
	hHEP	0.76	0.48	0.29	0.15	0.03	0.05	0.05	0.05
	mDC	0.77	0.61	0.38	0.15	0.15	0.13	0.07	0.06
	mESC	0.72	0.48	0.24	0.12	0.10	0.08	0.05	0.03
	mHSC-E	0.59	0.48	0.32	0.13	0.33	0.16	0.17	0.09
	mHSC-GM	0.64	0.66	0.41	0.30	0.39	0.24	0.30	0.24
	mHSC-L	0.33	0.48	0.23	0.22	0.06	0.05	0.26	0.26
Cell-type-specific ChIP-seq	hESC	0.60	0.47	0.55	0.53	0.25	0.16	0.23	0.17
	hHEP	0.81	0.72	0.82	0.74	0.38	0.46	0.42	0.39
	mDC	0.23	0.17	0.25	0.21	0.09	0.09	0.09	0.09
	mESC	0.84	0.67	0.85	0.79	0.84	0.70	0.36	0.36
	mHSC-E	0.94	0.81	0.93	0.89	0.90	0.89	0.55	0.58
	mHSC-GM	0.92	0.86	0.91	0.91	0.92	0.88	0.54	0.53
	mHSC-L	0.87	0.77	0.82	0.84	0.69	0.75	0.57	0.54
LOF / GOF	mESC	0.58	0.51	0.54	0.44	0.58	0.46	0.20	0.18

TFs + 1000

		LineGRN	GNNLink	STGRNS	GENELink	DeepDRIM	CNNC	DeepSEM	GENIE3
Non-specific ChIP-seq	hESC	0.21	0.13	0.14	0.10	0.12	0.02	0.02	0.01
	hHEP	0.26	0.16	0.17	0.09	0.12	0.04	0.02	0.01
	mDC	0.41	0.27	0.15	0.13	0.02	0.03	0.02	0.02
	mESC	0.33	0.18	0.13	0.08	0.07	0.07	0.02	0.02
	mHSC-E	0.39	0.22	0.21	0.11	0.08	0.10	0.07	0.03
	mHSC-GM	0.50	0.33	0.34	0.19	0.06	0.17	0.10	0.10
	mHSC-L	0.15	0.23	0.14	0.11	0.04	0.04	0.14	0.12
STRING	hESC	0.68	0.50	0.22	0.14	0.26	0.09	0.05	0.03
	hHEP	0.74	0.44	0.30	0.15	0.02	0.03	0.04	0.04
	mDC	0.80	0.61	0.38	0.20	0.10	0.08	0.06	0.04
	mESC	0.72	0.47	0.25	0.12	0.08	0.06	0.05	0.03
	mHSC-E	0.59	0.45	0.30	0.16	0.33	0.21	0.16	0.07
	mHSC-GM	0.69	0.53	0.43	0.35	0.41	0.32	0.24	0.21
	mHSC-L	0.25	0.49	0.20	0.27	0.05	0.05	0.26	0.25
Cell-type-specific ChIP-seq	hESC	0.61	0.51	0.57	0.55	0.34	0.24	0.20	0.17
	hHEP	0.82	0.68	0.81	0.73	0.46	0.56	0.44	0.39
	mDC	0.25	0.22	0.24	0.21	0.14	0.08	0.07	0.09
	mESC	0.85	0.66	0.86	0.81	0.83	0.61	0.36	0.36
	mHSC-E	0.95	0.81	0.94	0.91	0.95	0.90	0.57	0.56
	mHSC-GM	0.94	0.86	0.94	0.92	0.93	0.89	0.56	0.55
	mHSC-L	0.86	0.78	0.84	0.83	0.83	0.57	0.56	0.52
LOF / GOF	mESC	0.57	0.55	0.56	0.50	0.60	0.46	0.19	0.17

Supplymentary Figure II: Summary of the performance of LineGRN and seven comparison methods based on AUPRC metrics across 44 scenarios, including 7 cell types, 4 network types, and two gene variations (top 500 and top 1000 highly variable genes). Each cell represents the average of five repeated experiments. Darker colors indicate lower AUPRC values, reflecting poorer performance, with values less than or equal to 0.5 suggesting performance similar to random classification.

S7. Details of benchmark datasets

Supplymentary Table V: Details about 44 benchmark datasets with TFs and 500 (1000) most-varying genes

		Non-specific ChIP-seq			STRING		
Dataset	Cell	TFs	Target	Density	TFs	Target	Density
hESC	758	283(292)	753(1138)	0.016(0.014)	343(351)	511(695)	0.024(0.021)
hHEP	425	322(332)	825(1217)	0.015(0.013)	409(414)	646(874)	0.028(0.024)
mDC	383	250(254)	634(969)	0.019(0.016)	264(273)	479(664)	0.038(0.032)
mESC	421	516(522)	890(1214)	0.015(0.013)	495(499)	638(785)	0.024(0.021)
mHSC-E	1071	144(147)	442(674)	0.022(0.020)	156(161)	291(413)	0.029(0.027)
mHSC-GM	889	82(88)	297(526)	0.030(0.029)	92(100)	201(344)	0.040(0.037)
mHSC-L	847	35(37)	164(192)	0.048(0.043)	39(40)	70(81)	0.048(0.045)
		Cell-type-specific ChIP-seq			LOG/GOF		
Dataset	Cell	TFs	Target	Density	TFs	Target	Density
hESC	758	34(34)	815(1260)	0.164(0.165)	-	-	-
hHEP	425	30(31)	874(1331)	0.379(0.377)	-	-	-
mDC	383	20(21)	443(684)	0.085(0.082)	-	-	-
mESC	421	88(89)	977(1385)	0.345(0.347)	34(34)	774(1098)	0.158(0.154)
mHSC-E	1071	29(33)	691(1177)	0.578(0.566)	-	-	-
mHSC-GM	889	22(23)	618(1089)	0.543(0.565)	-	-	-
mHSC-L	847	16(16)	525(640)	0.525(0.507)	-	-	-

References

[1] G. Mao, Z. Pang, K. Zuo, et al. "Predicting gene regulatory links from single-cell RNA-seq data using graph neural networks." *Briefings in Bioinformatics*, vol. 24, no. 6, p. bbad414, 2023.

[2] G. Chen, Z. P. Liu. "Graph attention network for link prediction of gene regulations from single-cell RNA-sequencing data." *Bioinformatics*, vol. 38, no. 19, pp. 4522-4529, 2022.

[3] J. Xu, A. Zhang, F. Liu, et al. "STGRNS: An interpretable transformer-based method for inferring gene regulatory networks from single-cell transcriptomic data." *Bioinformatics*, vol. 39, no. 4, p. btad165, 2023.

[4] J. Chen, C. W. Cheong, L. Lan, et al. "DeepDRIM: A deep neural network to reconstruct cell-type-specific gene regulatory network using single-cell RNA-seq data." *Briefings in Bioinformatics*, vol. 22, no. 6, p. bbab325, 2021.

[5] Y. Yuan, Z. Bar-Joseph. "Deep learning for inferring gene relationships from single-cell expression data." *Proceedings of the National Academy of Sciences USA*, vol. 116, no. 52, pp. 27151-27158, 2019.

[6] H. Shu, J. Zhou, Q. Lian, et al. "Modeling gene regulatory networks using neural network architectures." *Nature Computational Science*, vol. 1, pp. 491-501, 2021.

[7] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, et al. "Inferring regulatory networks from expression data using tree-based methods." *PLoS One*, vol. 5, no. 9, p. e12776, 2010.