Gene Regulatory Network Inference with Joint Representation from Graph Neural Network and Single-Cell Foundation Model

Sindhura Kommu[1[0009−0005−3841−0202]], Yizhi Wang[2[0000−0001−9456−0743]], Yue Wang[2[0000−0002−1788−1102]], and Xuan Wang[1[0000−0002−1381−8958]]

[1] Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, USA
[2] Department of Electrical and Computer Engineering, Virginia Tech, Arlington, VA 22203, USA
{sindhura,yzwang,yuewang,xuanw}@vt.edu

Abstract. Inferring cell-type-specific gene regulatory networks (GRNs) from single-cell RNA sequencing (scRNA-seq) data is a complex task, primarily due to data sparsity, noise, and the dynamic, context-dependent nature of gene regulation across cell types and states. Recent advancements in the collection of experimentally validated data on transcription factor binding have facilitated GRN inference via supervised machine learning methods—where models learn from known TF-gene pairs to guide predictions. However, these methods still face challenges in 1) effectively representing and integrating prior knowledge, and 2) capturing regulatory mechanisms across diverse cellular contexts. To tackle the above challenges, we introduce a novel GRN inference method, scRegNet, that learns a joint representation from graph neural networks (GNNs) and pre-trained single-cell foundation models (scFMs). scRegNet combines rich contextual representations learned by large-scale, single-cell foundation models—trained on extensive unlabeled scRNA-seq datasets—with the structured knowledge embedded in experimentally validated networks through GNNs. This integration enables robust inference—the prediction of unknown gene regulatory interactions—by simultaneously accounting for gene expression patterns and established gene regulatory networks.

We evaluated our approach on seven single-cell scRNA-seq benchmark datasets from the BEELINE study [22], outperforming current state-of-the-art methods in cell-type-specific GRN inference. scRegNet demonstrates a superior ability to capture intricate regulatory interactions between genes across various cell types, providing a more in-depth understanding of cellular processes and regulatory dynamics. By harnessing the capabilities of large-scale pre-trained single-cell foundation models and GNNs, scRegNet offers a scalable and adaptable tool for advancing research in cell type-specific gene interactions and biological functions.

Code Availability: https://github.com/sindhura-cs/scRegNet

Keywords: scRNA-seq · Gene Regulatory Networks · Single-Cell Foundation Models · Graph Neural Networks · Transfer Learning · GRN Reconstruction · Supervised Machine Learning

1 Introduction

Single-cell RNA sequencing (scRNA-seq) has revolutionized the study of cellular diversity by enabling gene expression profiling at an unprecedented single-cell resolution [14]. This transformative technology has opened up new possibilities for understanding cellular identity, function, and heterogeneity [18]. One of the most critical applications of scRNA-seq is the inference of gene regulatory networks (GRNs), which represent the intricate interactions between transcription factors (TFs) and their target genes [8]. These networks govern cellular processes such as differentiation, development, and response to environmental stimuli, making GRN reconstruction crucial for deciphering the molecular mechanisms that bridge genotypes to phenotypes [3]. However, accurate GRN inference poses significant challenges. scRNA-seq data is typically sparse due to high dropout rates [16], which occurs when some gene transcripts are not captured during sequencing. Moreover, the inherent heterogeneity across different cell types and states further complicates GRN reconstruction [28].

Several computational methods have been proposed to infer GRNs from scRNA-seq data. Unsupervised approaches such as GENIE3 [13] and GRNBoost2 [20], which form the foundation of the SCENIC [1] workflow, utilize tree-based regressions to identify gene sets that are co-expressed with TFs. These methods focus on co-expression relationships but cannot often identify gene regulatory interactions. Recently, supervised GRN inference methods have gained attention by leveraging experimentally validated TF binding data from resources like ENCODE [7], ChIP-Atlas [21], and ESCAPE [30], which provide ground truth networks for specific cell types and conditions. These supervised models have the potential to learn from known TF-target gene interactions, improving inference accuracy.

One such supervised method, CNNC [33], employs convolutional neural networks (CNNs) to infer GRNs by converting gene pair co-expression profiles into image-like histograms. However, the conversion of gene expression profiles into histograms is computationally intensive, particularly for large datasets, limiting its scalability. Another supervised method, Gene Network Embedding (GNE) [15], uses multilayer perceptrons (MLPs) to encode gene expression profiles and graph topologies for GRN inference. Despite its success in some cases, GNE [15] struggles to effectively capture topological information using simple one-hot encodings of gene IDs.

Graph-based learning frameworks, such as GENELink [4] and GNNLink [19], have demonstrated promise in modeling the complex interconnections within GRNs. These methods cast GRN inference as a link prediction problem, using graph neural networks (GNNs) [29] to predict regulatory interactions between genes. While graph-based approaches offer a powerful mechanism for incorporating prior knowledge from known biological networks, they are often constrained by incomplete prior networks.

Recent advances in single-cell foundation models (scFMs), particularly those based on transformer architectures [26], have significantly advanced scRNA-seq data analysis. Models such as scFoundation [12], Geneformer [25], scBERT [31] and scGPT [9] leverage the vast amounts of unlabeled scRNA-seq data available to learn contextualized representations of gene expression, effectively capturing latent gene-gene interactions across the genome. These models, pre-trained on large-scale datasets, provide robust representations for downstream tasks like GRN inference. While the gene representations from the larger models like Geneformer and scFoundation can be retrieved in a zero-shot setting due to their scale, smaller models like scBERT often benefit from additional fine-tuning on specific cell-type datasets to enhance performance on regulatory tasks.

To address the limitations of existing methods, we propose a novel supervised GRN inference method, scRegNet, that integrates single-cell foundation models with graph neural networks. scRegNet harnesses the rich, contextual gene representations learned by pre-trained transformer models and combines them with graph embeddings derived from prior biological networks. This integration enables the model to capture both the gene expression context and regulatory network topology, essential for accurate GRN inference. Specifically, we formulate the GRN inference task as a link prediction problem, where the objective is to predict latent regulatory interactions between genes based on observed data. To enhance the model's generalizability and accuracy, we introduce a two-channel MLP-based joint learning layer. This layer processes both gene embeddings and graph features simultaneously, effectively learning from the combined representations to predict gene regulatory links. By incorporating prior knowledge, such as experimentally validated gene regulatory networks, we fine-tune the model to improve its ability to infer cell-type-specific regulatory interactions. This approach allows the model to generalize across different cell types and conditions, providing a robust framework for GRN inference.

We evaluated our framework on seven single-cell scRNA-seq benchmark datasets [22], covering a diverse range of cell types from both human and mouse sources. These datasets consist of cell-type-specific scRNA-seq data along with their corresponding experimentally validated TF-gene interactions that can

serve as ground-truth GRNs, allowing for a rigorous assessment of model performance. By employing a novel joint representation that integrates multiple data sources—such as gene expression profiles and prior regulatory networks—our method offers enhanced insights into gene regulatory mechanisms and sets a new standard in GRN inference. scRegNet consistently surpasses existing methods, demonstrating substantial improvements in both the Area Under the Receiver Operating Characteristic Curve (AUROC) and the Area Under the Precision-Recall Curve (AUPRC) across these benchmarks.

2 Methodology

Figure 1 illustrates the scRegNet framework, where pooled contextual gene representations from single-cell foundation models are integrated with graph-based gene representations from GNNs, capturing both the cell-type-specific gene expression context and structured information from existing GRNs.

2.1 Graph Structure Learning with GNNs

We leverage the intrinsic network structure of GRNs to capture complex gene regulatory relationships. The GRN inference task is formulated as a link prediction problem within a graph-based framework, where gene interactions are represented as edges and individual genes as nodes. To inform the learning process, we incorporate experimentally validated, cell-type-specific TF-target gene interactions as prior GRNs, with additional details provided in 3.1. These known interactions are partitioned into training and test sets. The training dataset is used as input to the graph encoder, providing a foundational understanding of gene interactions and regulatory interactions, thus anchoring the inference in biologically relevant contexts. The test dataset, on the other hand, serves as the benchmark for evaluating the predictive performance of the trained model.

We use the raw count matrix of gene expression as the initial features for nodes representing individual genes. To capture the underlying structure of GRNs, we employ three distinct GNN architectures: Graph Convolutional Networks (GCN) [17], GraphSAGE (SAmple and aggreGatE) [11], and Graph Attention Networks (GAT) [27]. Each of these architectures is tested independently to evaluate their effectiveness in learning enriched gene representations and identifying transcription factor (TF)-target gene interactions.

Let the prior GRN be represented as $G = \{V, E\}$, where V is the set of nodes (genes) and E is the set of edges (regulatory interactions). The goal is to infer structural relationships in this G by learning effective node representations through message passing, which embeds into each node information about its multi-hop neighbors. In each layer of the GNN, the current representation of the node embeddings $\{v_1^{l-1}, ..., v_T^{l-1}\}$ is fed into the layer to perform a round of information propagation among nodes to yield pre-fused updated embeddings for each node:

$$\{{v_1}^l, ..., {v_T}^l\} = GNN(\{v_1^{l-1}, ..., v_T^{l-1}\}), \;\; l = 1, ..., L \tag{1}$$

Where T is the number of nodes in G, and L is the number of layers in the GNN, and ${v_t}^l \in \mathbb{R}^{d_l}$ is the d_l-dimensional embedding vector of node t in the l-th layer. Specifically, for each layer l, the updated representation ${v_t}^l$ of each node is given by:

$${v_t}^l = \mathcal{M}(\{{v_s}^{l-1} : s \in \eta_t\}, {v_t}^{l-1}; \theta^l) \tag{2}$$

where η_t represents the set of neighboring nodes for an arbitrary node t, and $\mathcal{M}(\cdot)$, parameterized by θ_l in the l-th layer, is the message passing function for neighborhood aggregation. The neighborhood aggregation varies depending on the type of GNN employed, between GAT, GCN, or GraphSAGE.

GCN: Capturing Local Neighborhood Structure. [17] GCNs are designed to capture the local structure of the graph by aggregating features from a node's immediate neighbors. In the context of GRN inference, GCN helps model direct TF-target interactions by focusing on the immediate regulatory environment around each gene. We employ GCN-based architecture to determine its ability to capture local regulatory patterns in different cell types.

GraphSAGE: Extending to Larger Neighborhoods. [11] GraphSAGE builds upon GCN by enabling the aggregation of information from sampled neighborhoods, rather than requiring all neighbors to be included. This allows the model to generalize better to larger, more sparse networks, where sampling can provide a broader view of the regulatory landscape. We evaluate GraphSAGE to assess its effectiveness in capturing regulatory interactions by sampling neighbors dynamically.

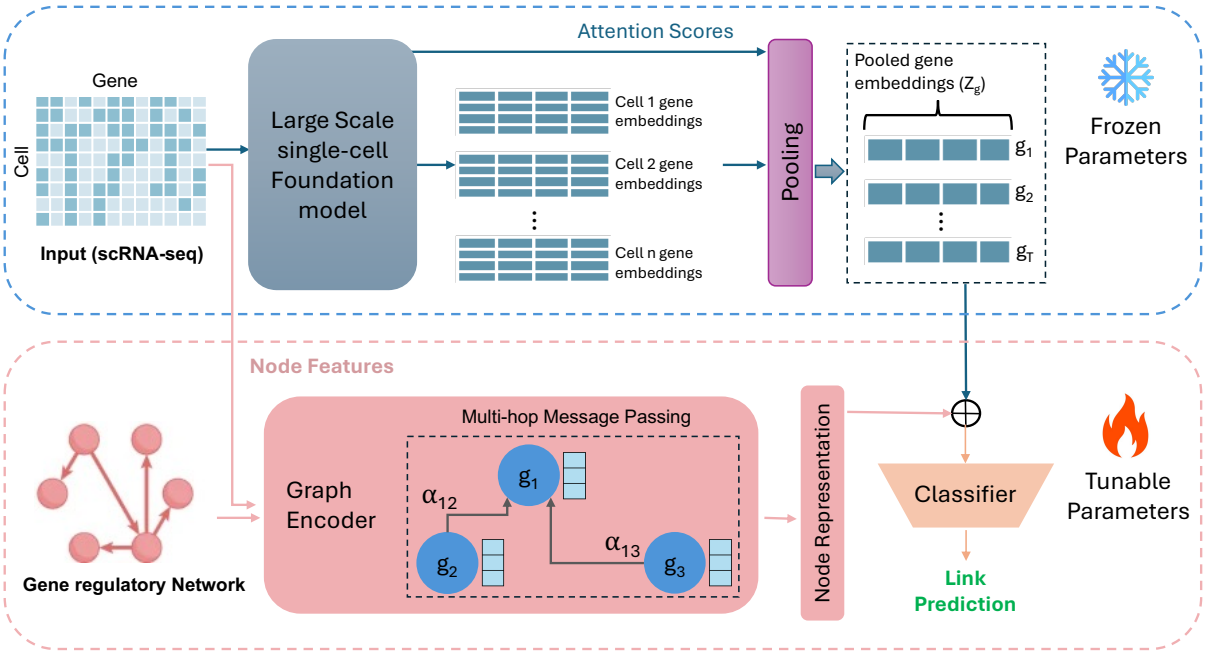

Fig. 1: **Overview of the scRegNet framework for GRN inference**. scRegNet utilizes a pre-trained single-cell foundation model (top; Section 2.2) to generate gene embeddings from scRNA-seq input which are integrated with outputs from a Graph Encoder (bottom left; Section 2.1). The combined representations are fed into a classifier (bottom right; Section 2.3) for link prediction, enabling the identification of regulatory interdependencies among genes. The architecture incorporates both frozen parameters for leveraging pre-trained knowledge and tunable parameters for domain-specific learning, facilitating the seamless integration of biological context with learned embeddings.

GAT: Enhancing Representation with Attention Mechanisms. [27] GAT introduces attention mechanisms that weigh the importance of different neighbors during the message-passing process. This enables the model to focus on the most relevant regulatory interactions for each gene, potentially capturing subtle or context-specific TF-target dynamics. By testing GAT, we aim to explore whether this added flexibility in weighting interactions can enhance GRN inference performance.

The evaluation of scRegNet with these three GNN architectures on multiple scRNA-seq datasets is detailed in Supplementary Table 6. Interestingly, after extensive hyperparameter tuning, all GNN variants show similar performance, with only slight differences observed among them. Consequently, we chose GCN for further experiments due to its computational efficiency and simplicity, making it an ideal baseline model. However, it is worth noting that as more biological knowledge is integrated, the choice of GNN architecture may have a greater impact on performance.

2.2 Pre-trained Single-Cell Foundation Models

Recent studies [12,25,31,9], have demonstrated that pre-trained transformer models possess a strong capacity to model gene-gene interactions across cells. In this study, we incorporate three large-scale pre-trained single-cell foundation models—scBERT [31], Geneformer [25], and scFoundation [12]—to capture the cell-type specific context of the scRNA-seq data in scRegNet. These models are specifically designed to capture long-distance dependencies between genes and have proven effective in processing single-cell data. All of them rely on transformer-based architectures [26] for processing embedded representations of the input gene expression data, but differ in how they represent input data, their model architectures, and training procedures. A summary of these models, detailing their architectures and key features, is provided in Supplementary Table 4. All three models employ masked language modeling (MLM) as a self-supervised pre-training strategy, allowing them to learn complex dependencies from millions of single-cell datasets. We leverage gene-level representations for each cell derived from the final layers of these scFMs and adapt them to our specific task, taking advantage of the insights encoded in their embeddings. Notably, the input design and pre-processing steps vary for each model as detailed below.

The input scRNA-seq datasets are initially processed into a cell-by-gene matrix, $\mathbf{X} \in \mathbb{R}^{N \times T}$, where each element represents the RNA abundance for gene t in cell n. This matrix, referred to as the raw count matrix, is normalized using a log transformation and feature scaling to ensure compatibility with transformer architectures. To create a sequence suitable for transformer inputs, we define a sequence of gene tokens as $\{g_1, ..., g_T\}$, where T is the total number of genes in a dataset.

scBERT. [31] scBERT utilizes a combination of two embeddings for each gene: (1) a gene identity embedding using the gene2vec approach and (2) an expression level embedding representing the gene's normalized, binned log-scale expression level. These embeddings allow the model to differentiate genes in the otherwise orderless data and handle over 16,000 genes efficiently using the Performer [6] architecture for scalable attention. For each gene token g_t, the input representation is constructed as:

$$h_t^0 = emb_{gene2vec}(g_t) + emb_{expr}(g_t) \tag{3}$$

where $\text{emb}_{\text{gene2vec}}(\cdot)$ denotes the gene identity embedding and $\text{emb}_{\text{expr}}(\cdot)$ represents the expression level embedding. These input representations are processed through L successive transformer encoder layers:

$$h_t^{(l)} = Transformer(h_t^{(l-1)}), \;\; l = 1, 2, ..., L, \tag{4}$$

scBERT employs a matrix decomposition variant of the Transformer, known as Performer, to handle longer sequence lengths efficiently. With $L = 6$ layers, scBERT outputs an embedding size of 200 for each gene. The final hidden states $\{h_t^L\}_{t=1}^T$ serve as the gene-level embeddings, capturing cell-specific regulatory information suitable for downstream tasks.

scFoundation. [12] scFoundation utilizes an asymmetric encoder-decoder architecture [10] that employs attention mechanisms to optimize gene dependency extraction in sparse single-cell data. We use scFoundation in a zero-shot setting, relying on its comprehensive pre-training to extract embeddings for downstream analysis without additional fine-tuning. For the encoder, only the non-zero elements from the input matrix $\mathbf{X}$ are considered. The encoder input is formed as:

$$I = Autobin(X \odot M_{nonzero}) + Lookup(genes) \tag{5}$$

where $\odot$ denotes the element-wise product, M_{nonzero} is a mask identifying non-zero elements, and Autobin converts expression values into discretized tokens. The encoder generates gene-level representations using multi-head self-attention:

$$I_{encoder} = Transformer(f_Q(I), f_K(I), f_V(I)) \tag{6}$$

where f_Q, f_K, and f_V are linear projections for query, key, and value. The decoder utilizes this encoded information along with zero and masked expression values to reconstruct the complete data:

$$I_{full} = W_p(I_{encoder} \oplus I_{zero} \oplus I_{masked}) + b_p \tag{7}$$

where $\oplus$ represents concatenation, and W_p, b_p are learned parameters that project the decoder's embedding size. The final gene-level embeddings are then produced by the decoder through:

$$I_{decoder} = Transformer(f_Q(I_{full}), f_K(I_{full}), f_V(I_{full})) \tag{8}$$

These embeddings capture detailed gene dependencies, making them suitable for downstream GRN inference.

Geneformer. [25] Geneformer, the largest of the three models, employs a rank value encoding strategy to represent input data, prioritizing genes based on their expression levels within a cell. Given its extensive pre-training on approximately 95 million single-cell transcriptomes, we leverage its pre-trained gene representations directly without additional fine-tuning. This allows us to extract high-quality embeddings directly for tasks like GRN inference.

To prepare the input data for Geneformer, we utilize the token dictionary (TokenDict) and the gene median file (GeneMedian) provided in the model's repository. These resources ensure that the input is

accurately tokenized based on the rank value encoding strategy, maintaining consistency with the pre-trained model. Each gene's expression is normalized relative to a median reference and then converted into ranked tokens. The input matrix **R** is constructed as follows:

$$R = RankValue(X; TokenDict; GeneMedian) \tag{9}$$

The RankValue function normalizes each gene's expression using the GeneMedian values and maps them to discrete tokens via the TokenDict. For each single-cell transcriptome, Geneformer embeds each gene into an 896-dimensional space that captures the gene's contextual characteristics within the cell. These contextual embeddings are generated via multi-layer attention mechanisms similar to the equation 4, where $h_t^{(0)}$ is the initial embedding for token R_t, and $L = 20$ layers are used in the encoder. The final hidden state, $h_t^{(L)} \in \mathbb{R}^{896}$, represents the context-aware embedding for gene t. To obtain robust and generalizable gene representations, embeddings are extracted from the penultimate layer of the model, as it captures a more abstract and general feature space compared to the final layer, which tends to be task-specific.

2.3 Joint Representation Learning for GRN Inference

Once gene embeddings are extracted from the single-cell foundation model as detailed in 2.2 for each single cell in a dataset, they are aggregated at the cell type level to form a unified representation for each gene across the dataset.

Attention Pooling for scBERT. For scBERT, we utilize the attention scores across cells from the model to select the most representative cells for pooling of each gene representation. For each input gene token g_t we get the embeddings for all cells in a dataset denoted as $\{h_{t(n)}^L\}_{n=1}^N$, where N is the number of cells in a dataset. For each cell in the dataset, let $A_{i,j}^{(m,l)}$ represent the attention weight from gene i to gene j in the l-th layer and m-th attention head of the Performer. To obtain a unified attention matrix, we integrate the multi-head, multi-layer attention weights by taking an element-wise average across all layers and heads:

$$A(i,j) = \frac{1}{M \cdot L} \sum_{m=1}^{M} \sum_{l=1}^{L} A_{i,j}^{(m,l)} \tag{10}$$

where M is the number of attention heads and L is the total number of layers. In the resulting averaged attention matrix A, each element $A(i,j)$ indicates the degree to which gene i attends to gene j in a particular cell. To focus on the significance of genes within each cell n, we sum the attention scores along columns to obtain the attention-sum vector a_n:

$$a_n = \sum_{j=1}^{T} A(:,j) \tag{11}$$

where $a_n \in \mathbb{R}^T$, and T is the number of genes. These scores are normalized across cells for each gene g_t, resulting in the normalized attention weights $a_{n\,n=1}^{t\;N}$, where N is the total number of cells. Using these weights, gene embeddings are aggregated across cells. For each gene t, the aggregated embedding $Z_g[t]$ is given by:

$$Z_g[t] = \sum_{n=1}^{N} a_t^n h_{t(n)}^{(L)} \tag{12}$$

where $h_{t(n)}^{(L)}$ represents the final layer embedding of gene t in cell n. The output matrix $Z \in \mathbb{R}^{T \times d}$ consists of aggregated gene-level embeddings, where d is the dimension of the output embedding.

Mean Pooling for Geneformer and scFoundation. For Geneformer and scFoundation, we apply mean pooling instead since it is very time taking to derive attention weights. The mean-pooled embedding for gene t is calculated as:

$$Z_g[t] = \frac{1}{N} \sum_{n=1}^{N} h_{t(n)}^{(L)} \tag{13}$$

where $h_{t(n)}^{(L)}$ is the gene embedding from the final layer in cell n. This pooling strategy ensures a balanced representation that captures average gene activity across the dataset.

Integration of scFM-Based and Graph-Based Embeddings. In the final stage, we combine gene representations from both the transformer models and the Graph Neural Networks (GNNs). This integration involves concatenating the embeddings from transformers (capturing contextual gene interactions) and the graph-based embeddings (capturing structural relationships):

$$Z_{\text{joint}}[t] = Z_{\text{LLM}}[t] \oplus Z_{\text{GNN}}[t] \tag{14}$$

where $Z_{\text{trans}}[t]$ represents the transformer-derived embedding for gene t and $Z_{\text{GNN}}[t]$ represents the graph-based embedding. The concatenated embedding $Z_{\text{joint}}[t]$ serves as the unified representation for gene t.

Final Prediction through Multi-Layer Perceptrons. For each gene pair (i, j), the final embeddings are processed through two parallel channels, each consisting of multi-layer perceptrons (MLPs). These MLPs encode the joint gene representations into low-dimensional vectors:

$$e_i = MLP_1(Z_{\text{joint}}[i]), \quad e_j = MLP_2(Z_{\text{joint}}[j]) \tag{15}$$

where $\mathbf{e}_i, \mathbf{e}_j \in \mathbb{R}^d$ are the low-dimensional vectors for genes i and j, respectively. These vectors are then utilized for downstream tasks, such as measuring gene similarity or inferring gene relationships.

3 Experimental Setup

3.1 Data

To guide the learning process within our framework, we utilize experimentally validated cell-type-specific prior GRNs from the BEELINE study [22]. BEELINE offers benchmark datasets explicitly curated for inferring GRNs from single-cell RNA sequencing (scRNA-seq) data. Our GRN reconstruction primarily focuses on transcription factor (TF)-target gene interactions, which are crucial regulators of cellular functions. These interaction networks are sourced from reputable experimentally validated databases [7,21,30]. By limiting the regulatory networks to genes that display significant variability across cells, we ensure that the inferred networks capture biologically meaningful gene interactions. For each cell type, we construct two distinct prior GRNs by selecting all TFs and the top 500 and 1000 most variable genes, respectively.

To enhance the efficiency of the training process and improve the quality of learned representations, we implement a hard negative sampling strategy. Hard negative samples are known to provide more discriminative information, accelerating training, reducing computational complexity, and leading to superior node representations [34,32]. Initially, all unobserved TF-gene interactions in the prior network are treated as potential negative samples. For each positive interaction between gene pairs (g_1, g_2), we randonly pair g_1 with a different gene g_3 (where $g_3 \neq g_2$) to generate a "hard negative" sample (g_1, g_3). This approach sharpens the model's ability to distinguish between true regulatory interactions and false associations.

Table 1: The statistics of prior networks with TFs and 500 (1000) most-varying genes

Cell Type	Species	#Cells	#TFs	#Genes	Density	Training Size	Test Size
hESC	human	759	34 (34)	815 (1260)	0.164 (0.165)	20677 (32065)	7142 (11047)
hHEP	human	426	30 (31)	874 (1331)	0.379 (0.377)	19002 (30026)	6563 (10348)
mDC	mouse	384	20 (21)	443 (684)	0.085 (0.082)	10969 (18556)	3792 (6395)
mESC	mouse	422	88 (89)	977 (1385)	0.345 (0.347)	65895 (96460)	22736 (33229)
mHSC-E	mouse	1072	29 (33)	691 (1177)	0.578 (0.566)	13632 (26565)	4718 (9164)
mHSC-GM	mouse	890	22 (23)	618 (1089)	0.543 (0.565)	9280 (17406)	3216 (6003)
mHSC-L	mouse	848	16 (16)	525 (640)	0.525 (0.507)	5976 (7392)	2076 (2560)

To ensure that the supervised model can effectively differentiate the subtle distinctions between true targets and non-target genes, we divide the positive and negative samples for each TF in a 2:1 ratio

between training and test datasets. Additionally, 10% of the TF-gene pairs from the training set are randomly selected for validation purposes. The training dataset serves as the basis for supervising graph representation learning, while the test dataset provides a benchmark to assess the accuracy of our inferred GRNs. The statistics for the prior networks, along with sizes of training and test set for each cell type are presented in Table 1. These datasets encompass multiple human and mouse cell types, offering a diverse benchmark to evaluate the performance of our approach on each cell type specific dataset and its corresponding prior GRN.

3.2 Training

All reported results are based exclusively on predictions from the held-out test set. Identical training, validation, and test splits are maintained across all supervised methods to ensure a fair comparison between models. We trained each model for 50 trials, calculating the mean and standard deviation of performance metrics across these trials to ensure robust results. The training process, which follows an end-to-end approach using frozen pre-trained foundation model layers and fine-tuning the GNN layers, is outlined in Algorithm 1 in Supplementary Section 6.3. During training, cross-entropy loss is employed to optimize the GNN layers, allowing them to adapt to specific datasets while leveraging the generalized embeddings from the frozen foundation models. Each model is trained for 300 epochs with an early stopping criterion of 10 epochs if no improvement is observed. Additionally, we apply dropout regularization and weight decay during training to ensure robust convergence.

Hyperparameter Tuning. To ensure optimal performance across multiple datasets, we developed a comprehensive hyperparameter optimization strategy. The hyperparameter search space, detailed in Supplementary Table 5, spans a wide range of parameters essential for tuning the performance of models. We leveraged Optuna [2], a powerful hyperparameter optimization framework, to systematically explore this space.

4 Results

To assess the efficacy of scRegNet for supervised GRN inference, we conducted comprehensive evaluations using datasets described in Section 3.1. The results for prior GRNs constructed with TFs+500 and TFs+1000 genes are detailed in Tables 2 and 3, respectively.

4.1 Performance on benchmark datasets

We developed three variations of scRegNet, each leveraging a distinct single-cell foundation model as the backbone: scBERT, Geneformer, and scFoundation. Details of these models, including their architecture and pre-training specifics, are provided in Table 4. These models are selected to examine the impact of different architectures, training size and pre-training regimes on GRN inference accuracy. We employ GCN as the backbone for GNN. The performance of our models is benchmarked against nine baseline methods, encompassing both traditional and graph-based methods, with detailed descriptions available in Supplementary 6.2. In our experiments, scBERT, which is pre-trained on 1 million cells, was further fine-tuned on cell-type-specific scRNA-seq data using MLM objective to refine its contextual representations for each specific cell type. In contrast, the larger models—Geneformer and scFoundation—trained on 95 million and 50 million cells, respectively, were utilized in a zero-shot setting, relying solely on pre-trained embeddings without additional task-specific fine-tuning.

Tables 2 and 3 highlight that all three variants of scRegNet outperform the baseline models in terms of both AUROC and AUPRC metrics across diverse datasets. Notably, the Geneformer-based variant, which possesses the largest number of parameters, consistently delivered superior results, achieving the highest AUROC and AUPRC scores across all evaluated cell types. Key performance improvements are as follows:

- For datasets with TFs+500, the AUROC scores of scRegNet(w Geneformer) were approximately 8.07% and 9.01% higher, on average, than those of the second-best models—GENELink [19] and GNNLink [4], respectively. AUPRC scores exhibited even more pronounced improvements, with increases of 13.18% and 8.41% over GENELink and GNNLink, respectively.
- For datasets with TFs+1000, the AUROC scores for scRegNet(w Geneformer) were, on average, 8.31% and 9.88% higher compared to GENELink and GNNLink. The AUPRC scores displayed impressive gains of 16.97% and 8.41%, respectively.

Table 2: Link prediction performance on seven scRNA-seq datasets with **500 most-variable genes**. Each dataset includes a cell-type-specific ground-truth network. The values reported are averages from 50 independent evaluations per cell type. Underlined values represent the best performance among the baseline models, while [2] and [3] denote the second and third overall best performances, respectively. scRegNet utilizing the three backbone models—scBERT, Geneformer, and scFoundation—consistently outperforms the baselines, with Geneformer achieving the highest overall performance.

Method	scFM Adaptation		Cell Type						
			hESC	hHEP	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
GRNBOOST2 [20]	-	AUROC	0.49	0.52	0.52	0.53	0.53	0.50	0.52
		AUPRC	0.15	0.38	0.06	0.32	0.57	0.52	0.5
GENIE3 [13]	-	AUROC	0.50	0.54	0.50	0.50	0.52	0.53	0.52
		AUPRC	0.15	0.39	0.05	0.31	0.56	0.53	0.50
PCC [23]	-	AUROC	0.47	0.49	0.54	0.51	0.49	0.54	0.55
		AUPRC	0.14	0.35	0.06	0.31	0.56	0.53	0.52
GRN-Transformer [24]	-	AUROC	0.51	0.49	0.50	0.53	0.64	0.50	0.64
		AUPRC	0.15	0.35	0.06	0.49	0.71	0.66	0.64
DeepDRIM [5]	-	AUROC	0.63	0.52	0.50	0.51	0.56	0.64	0.58
		AUPRC	0.13	0.39	0.06	0.46	0.76	0.64	0.59
CNNC [33]	-	AUROC	0.68	0.64	0.54	0.73	0.67	0.69	0.67
		AUPRC	0.25	0.46	0.06	0.48	0.74	0.68	0.64
GNE [15]	-	AUROC	0.67	0.80	0.52	0.81	0.82	0.83	0.77
		AUPRC	0.34	0.65	0.06	0.64	0.80	0.78	0.70
GENELink [4]	-	AUROC	0.82	0.84	0.71	0.88	0.87	0.89	0.83
		AUPRC	0.50	0.70	0.11	0.75	0.89	0.89	0.83
GNNLink [19]	-	AUROC	0.85	0.82	0.70	0.84	0.83	0.89	0.84
		AUPRC	0.52	0.75	**0.25**	0.76	0.88	0.89	0.85
scRegNet (w scBERT)	Fine-tuned	AUROC	$0.879^{2}\pm0.001$	$0.872^{3}\pm0.000$	$0.77^{3}\pm0.004$	$0.910^{2}\pm0.001$	$0.887^{3}\pm0.002$	$0.909^{2}\pm0.001$	$0.868^{2}\pm0.001$
		AUPRC	$0.597^{3}\pm0.001$	$0.786^{3}\pm0.001$	0.12 ± 0.004	$0.816^{3}\pm0.001$	$0.918^{3}\pm0.002$	$0.92^{2}\pm0.000$	$0.860^{2}\pm0.000$
scRegNet (w scFoundation)	Zero-shot	AUROC	$0.884^{2}\pm0.003$	$0.885^{2}\pm0.001$	**0.80**±0.003	$0.918^{2}\pm0.001$	$0.911^{2}\pm0.003$	$0.917^{2}\pm0.002$	$0.874^{2}\pm0.000$
		AUPRC	$0.608^{2}\pm0.002$	$0.804^{2}\pm0.001$	$0.14^{2}\pm0.002$	$0.828^{2}\pm0.001$	$0.930^{2}\pm0.000$	$0.922^{2}\pm0.000$	**0.872**±0.001
scRegNet (w Geneformer)	Zero-shot	AUROC	**0.893**±0.002	**0.904**±0.000	$0.799^{2}\pm0.006$	**0.932**±0.000	**0.929**±0.001	**0.938**±0.000	**0.886**±0.001
		AUPRC	**0.622**±0.001	**0.843**±0.000	$0.135^{3}\pm0.003$	**0.868**±0.000	**0.941**±0.001	**0.942**±0.001	**0.879**±0.002

Impact of scFM Backbone. The analysis underscores the impact of the pre-training architecture and training size of the scFM backbone model: 1) scRegNet(w Geneformer) consistently outperformed other models, suggesting that its encoder-only architecture, trained on a large-scale dataset using ranked normalized expression values, is particularly effective for capturing complex gene patterns at a broad scale. 2) scRegNet(w scFoundation) demonstrated robust performance, particularly in complex cell types, likely due to its broader input gene coverage (19,264 genes). 3) scRegNet(w scBERT), despite being pre-trained on a relatively smaller dataset (1 million cells), showed competitive performance after fine-tuning, underscoring the effectiveness of targeted fine-tuning in adapting smaller models to specific cell-type contexts.

Adaptation Strategies. Fine-tuning scBERT on specific cell-type datasets enabled the model to achieve performance levels similar to the zero-shot Geneformer, highlighting the viability of fine-tuning even for smaller models. In contrast, the zero-shot strategy for the larger models, Geneformer and scFoundation, allowed them to effectively generalize across multiple cell types without additional task-specific training. Although the zero-shot performance of scFoundation did not reach Geneformer's accuracy, it still surpassed most traditional baselines, demonstrating the flexibility and robustness of pre-trained models.

4.2 Ablation Study

To understand the contribution of each component within scRegNet, we performed a series of ablation studies, with the results displayed in Figure 2. The first experiment involved removing the GNN encoder, which led to a substantial decline in performance, highlighting the critical role of graph-based representation learning in refining gene embeddings. In the second ablation, we excluded the pre-trained foundation model embeddings. This omission impaired performance, demonstrating the importance of capturing diverse cellular contexts through pre-trained embeddings for accurate GRN inference. Lastly, we evaluated the impact of attentive pooling by removing this mechanism, which resulted in a noticeable performance drop. This indicates that attending to the most relevant cells for each gene is essential for enhancing the precision of GRN predictions. In these experiments we utilized scBERT as the scFM backbone and GCN as the GNN backbone for the model.

Table 3: Link prediction performance on seven scRNA-seq datasets with **1000 most-variable genes**. Each dataset includes a cell-type-specific ground-truth network. The values reported are averages from 50 independent evaluations per cell type. Underlined values represent the best performance among the baseline models, while ^{2} and ^{3} denote the second and third overall best performances, respectively. scRegNet utilizing the three backbone models—scBERT, Geneformer, and scFoundation—consistently outperforms the baselines, with Geneformer achieving the highest overall performance.

Method	scFM Adaptation		Cell Type						
			hESC	hHEP	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
GRNBOOST2 [20]	-	AUROC	0.49	0.52	0.53	0.53	0.51	0.49	0.53
		AUPRC	0.14	0.37	0.05	0.32	0.54	0.52	0.48
GENIE3 [13]	-	AUROC	0.50	0.54	0.52	0.50	0.50	0.51	0.52
		AUPRC	0.15	0.38	0.05	0.31	0.54	0.53	0.48
PCC [23]	-	AUROC	0.47	0.50	0.54	0.51	0.49	0.54	0.55
		AUPRC	0.14	0.34	0.05	0.31	0.53	0.54	0.51
GRN-Transformer [24]	-	AUROC	0.67	0.58	0.57	0.50	0.59	0.53	0.58
		AUPRC	0.16	0.53	0.05	0.51	0.69	0.61	0.52
DeepDRIM [5]	-	AUROC	0.56	0.63	0.50	0.62	0.50	0.66	0.57
		AUPRC	0.19	0.46	0.06	0.46	0.73	0.64	0.48
CNNC [33]	-	AUROC	0.72	0.66	0.56	0.73	0.72	0.69	0.62
		AUPRC	0.27	0.49	0.05	0.50	0.77	0.73	0.56
GNE [15]	-	AUROC	0.68	0.81	0.52	0.82	0.84	0.84	0.77
		AUPRC	0.34	0.66	0.05	0.65	0.81	0.81	0.68
GENELink [4]	-	AUROC	0.83	0.85	0.74	0.89	0.90	0.90	0.84
		AUPRC	0.50	0.71	0.12	0.76	0.90	0.91	0.81
GNNLink [19]	-	AUROC	0.80	0.84	0.78^3	0.84	0.87	0.92^2	0.86^2
		AUPRC	0.51	0.78^3	**0.21**	0.78	0.93	0.93	0.86
scRegNet (w scBERT)	Fine-tuned	AUROC	$0.866^2\pm0.001$	$0.871^3\pm0.002$	0.773 ± 0.002	$0.921^2\pm0.001$	$0.920^2\pm0.001$	$0.919^2\pm0.000$	$0.863^2\pm0.001$
		AUPRC	$0.588^3\pm0.001$	$0.780^3\pm0.001$	$0.14^3\pm0.004$	$0.824^2\pm0.000$	$0.943^2\pm0.001$	$0.941^2\pm0.000$	$0.936^2\pm0.003$
scRegNet (w scFoundation)	Zero-shot	AUROC	**0.882**±0.001	$0.882^2\pm0.001$	$0.802^2\pm0.002$	$0.914^3\pm0.001$	$0.923^2\pm0.001$	$0.924^2\pm0.001$	$0.864^2\pm0.001$
		AUPRC	$0.603^2\pm0.003$	$0.804^2\pm0.001$	$0.14^3\pm0.005$	$0.824^2\pm0.001$	$0.944^2\pm0.002$	$0.945^2\pm0.000$	$0.941^2\pm0.003$
scRegNet (w Geneformer)	Zero-shot	AUROC	**0.884**±0.002	**0.905**±0.000	**0.833**±0.003	**0.938**±0.001	**0.945**±0.000	**0.955**±0.001	**0.913**±0.002
		AUPRC	**0.625**±0.001	**0.85**±0.000	$0.16^2\pm0.002$	**0.869**±0.000	**0.952**±0.000	**0.964**±0.001	**0.960**±0.003

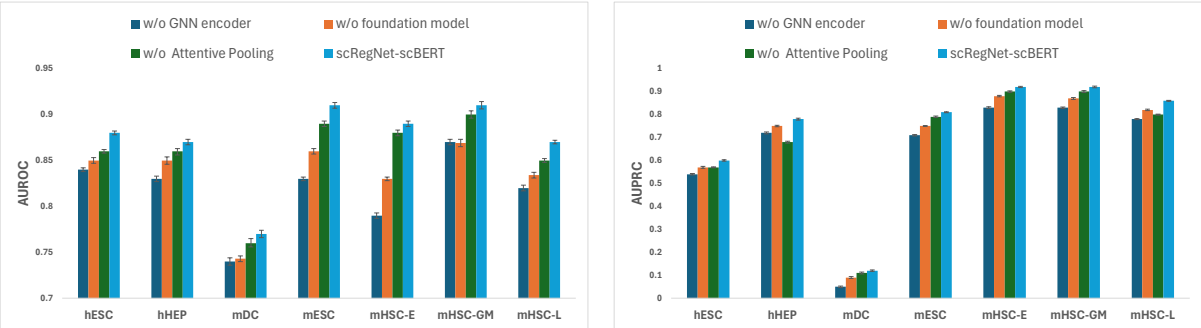

Fig. 2: Ablation study validating the contributions of the GNN encoder, scBERT encoder, and Attentive Pooling in scRegNet, using the cell-type-specific GRN with TFs + 500 genes. The plots display AUROC on the left and AUPRC on the right, highlighting the impact of each component on model performance.

4.3 Robustness Study

While our methodology leverages experimentally validated prior GRNs, in practice prior knowledge might have some noise and false positives. It is important to assess the robustness of our model against this. To this end, we evaluate the robustness of the best performing model scRegNet-Geneformer in the presence of noisy prior knowledge, we conducted experiments using noise-corrupted training data. For each noise level—1%, 2%, 3%, and 4%—we generated 10 distinct corrupted priors by introducing controlled perturbations. The performance of our model was assessed against each corrupted prior, with results visualized using box plots to depict performance variations. These results were benchmarked against the baseline method, GENELink, highlighting the stability and resilience of scRegNet under noisy priors. The findings are summarized in Figure 3.

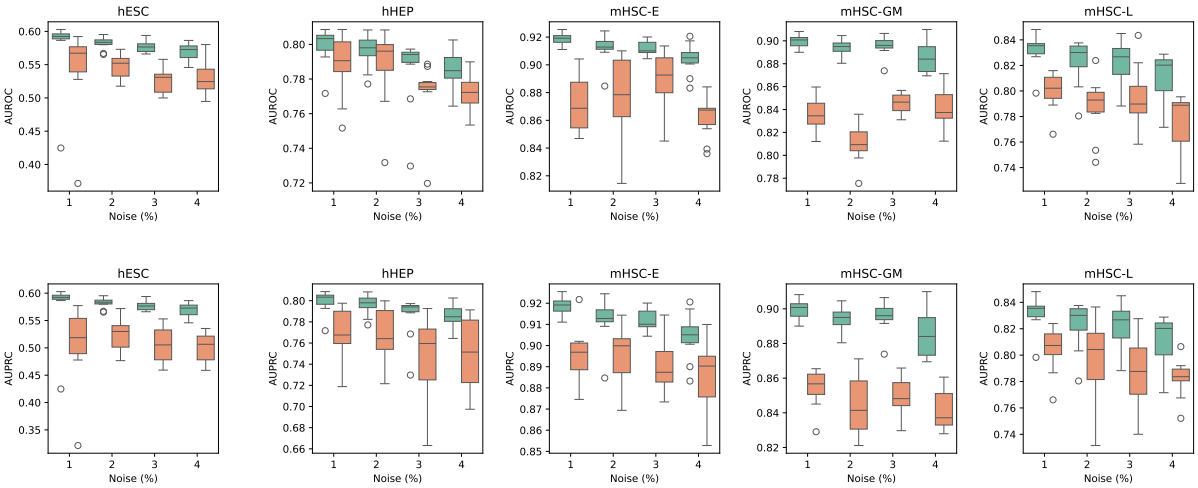

Fig. 3: Performance comparison of scRegNet-Geneformer vs. GENELink under increasing noise levels in cell-type-specific GRNs. The evaluation was conducted on networks containing TFs + 500 genes, with noise in the training dataset incrementally increased from 1% to 5%. Box plots illustrate the robustness of scRegNet in comparison to GENELink as noise levels rise, highlighting the model's stability across varying perturbations.

5 Conclusion

In this study, we introduced scRegNet, a novel framework for supervised Gene Regulatory Network (GRN) inference that seamlessly integrates single-cell foundation models with Graph Neural Networks (GNNs). Our comprehensive evaluations across seven diverse scRNA-seq datasets demonstrate that scRegNet consistently outperforms traditional and graph-only approaches, achieving superior AUROC and AUPRC scores across a range of cell types. The integration of pre-trained single-cell foundation models, such as scBERT, Geneformer, and scFoundation, has proven pivotal in improving GRN inference accuracy, with Geneformer consistently delivering the highest overall performance. This underscores the potential of leveraging large-scale, pre-trained models that encapsulate rich biological context through self-supervised learning from extensive scRNA-seq datasets. Additionally, the capability of Geneformer and scFoundation to operate in zero-shot settings highlights the adaptability of large-scale foundation models, maintaining robust performance without requiring cell-type-specific fine-tuning.

However, our current approach has limitations. The fusion of foundation models and GNNs in scRegNet is relatively shallow and non-interactive, with each modality being encoded separately and combined at the prediction stage. This restricts the capacity for dynamic information exchange between the modalities. In future work, we aim to explore deeper and more integrated fusion techniques, such as early, mid, and end fusions. These approaches will allow for simultaneous updates to both graph and foundation model parameters, facilitating more interactive and structured reasoning between the two representations. By pursuing these advanced fusion techniques, we anticipate further improvements in GRN inference accuracy and generalizability. This research direction has the potential to enhance our understanding of gene regulatory networks and their influence on cellular processes, offering valuable insights for developmental biology and personalized medicine.

Acknowledgments. Our work is sponsored by the NSF NAIRR Pilot with PSC Neocortex and NCSA Delta, Commonwealth Cyber Initiative, Children's National Hospital, Fralin Biomedical Research Institute (Virginia Tech), Sanghani Center for AI and Data Analytics (Virginia Tech), Virginia Tech Innovation Campus, and a generous gift from the Amazon + Virginia Tech Center for Efficient and Robust Machine Learning. Yizhi Wang and Yue Wang are supported by National Institutes of Health under Grants HL111362, CA271891, MH110504, and NS123719.

References

1. Aibar, S., González-Blas, C.B., Moerman, T., Huynh-Thu, V.A., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.C., Geurts, P., Aerts, J., et al.: Scenic: single-cell regulatory network inference and clustering.

Nature methods **14**(11), 1083–1086 (2017)
2. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. p. 2623–2631. KDD '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3292500.3330701, https://doi.org/10.1145/3292500.3330701
3. Alon, U.: Network motifs: theory and experimental approaches. Nature Reviews Genetics **8**(6), 450–461 (2007). https://doi.org/10.1038/nrg2102
4. Chen, G., Liu, Z.P.: Graph attention network for link prediction of gene regulations from single-cell rna-sequencing data. Bioinformatics **38**(19), 4522–4529 (2022)
5. Chen, J., Cheong, C., Lan, L., Zhou, X., Liu, J., Lyu, A., Cheung, W.K., Zhang, L.: Deepdrim: a deep neural network to reconstruct cell-type-specific gene regulatory network using single-cell rna-seq data. Briefings in bioinformatics **22**(6), bbab325 (2021)
6. Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., Belanger, D., Colwell, L., Weller, A.: Rethinking attention with performers (2022)
7. Consortium, T.E.P.: An integrated encyclopedia of dna elements in the human genome. Nature **489**(7414), 57–74 (2012). https://doi.org/10.1038/nature11247
8. Cramer, P.: Organization and regulation of gene transcription. Nature **573**(7772), 45–54 (2019)
9. Cui, H., Wang, C., Maan, H., Pang, K., Luo, F., Duan, N., Wang, B.: scgpt: toward building a foundation model for single-cell multi-omics using generative ai. Nature Methods (Feb 2024). https://doi.org/10.1038/s41592-024-02201-0, https://doi.org/10.1038/s41592-024-02201-0
10. Gong, J., Hao, M., Cheng, X., Zeng, X., Liu, C., Ma, J., Zhang, X., Wang, T., Song, L.: xtrimogene: an efficient and scalable representation learner for single-cell rna-seq data. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2024)
11. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS (2017)
12. Hao, M., Gong, J., Zeng, X., Liu, C., Guo, Y., Cheng, X., Wang, T., Ma, J., Zhang, X., Song, L.: Large-scale foundation model on single-cell transcriptomics. Nature Methods **21**(8), 1481–1491 (Aug 2024). https://doi.org/10.1038/s41592-024-02305-7, https://doi.org/10.1038/s41592-024-02305-7
13. Huynh-Thu, V.A., Irrthum, A., Wehenkel, L., Geurts, P.: Inferring regulatory networks from expression data using tree-based methods. PloS one **5**(9), e12776 (2010)
14. Jovic, D., Liang, X., Zeng, H., Lin, L., Xu, F., Luo, Y.: Single-cell RNA sequencing technologies and applications: A brief overview. Clin. Transl. Med. **12**(3), e694 (Mar 2022)
15. Kc, K., Li, R., Cui, F., Yu, Q., Haake, A.R.: Gne: a deep learning framework for gene network inference by aggregating biological information. BMC systems biology **13**, 1–14 (2019)
16. Kharchenko, P.V., Silberstein, L., Scadden, D.T.: Bayesian approach to single-cell differential expression analysis. Nature methods **11**(7), 740–742 (2014)
17. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2017)
18. Linnarsson, S., Teichmann, S.A.: Single-cell rna-seq: Advances and future challenges. Nature Methods **13**, 405–408 (2016). https://doi.org/10.1038/nmeth.3841
19. Mao, G., Pang, Z., Zuo, K., Wang, Q., Pei, X., Chen, X., Liu, J.: Predicting gene regulatory links from single-cell RNA-seq data using graph neural networks. Briefings in Bioinformatics **24**(6), bbad414 (11 2023). https://doi.org/10.1093/bib/bbad414, https://doi.org/10.1093/bib/bbad414
20. Moerman, T., Aibar Santos, S., Bravo González-Blas, C., Simm, J., Moreau, Y., Aerts, J., Aerts, S.: Grnboost2 and arboreto: efficient and scalable inference of gene regulatory networks. Bioinformatics **35**(12), 2159–2161 (2019)
21. Okanishi, H., Tanaka, M., Suzuki, Y.: Chip-atlas: a data-mining suite powered by full integration of public chip-seq data. EMBO reports **22**(10), e52171 (2021). https://doi.org/10.15252/embr.202152171
22. Pratapa, A., Jalihal, A.P., Law, J.N., Bharadwaj, A., Murali, T.M.: Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. Nature Methods **17**(2), 147–154 (Feb 2020). https://doi.org/10.1038/s41592-019-0690-6, https://doi.org/10.1038/s41592-019-0690-6
23. Salleh, F.H.M., Arif, S.M., Zainudin, S., Firdaus-Raih, M.: Reconstructing gene regulatory networks from knock-out data using gaussian noise model and pearson correlation coefficient. Computational biology and chemistry **59**, 3–14 (2015)
24. Shu, H., Ding, F., Zhou, J., Xue, Y., Zhao, D., Zeng, J., Ma, J.: Boosting single-cell gene regulatory network reconstruction via bulk-cell transcriptomic data. Briefings in Bioinformatics **23**(5), bbac389 (2022)
25. Theodoris, C.V., Xiao, L., Chopra, A., Chaffin, M.D., Al Sayed, Z.R., Hill, M.C., Mantineo, H., Brydon, E.M., Zeng, Z., Liu, X.S., Ellinor, P.T.: Transfer learning enables predictions in network biology. Nature **618**(7965), 616–624 (Jun 2023). https://doi.org/10.1038/s41586-023-06139-9, https://doi.org/10.1038/s41586-023-06139-9
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

27. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. International Conference on Learning Representations (2018), https://openreview.net/forum?id=rJXMpikCZ
28. Wagner, A., Regev, A., Yosef, N.: Revealing the vectors of cellular identity with single-cell genomics. Nature biotechnology **34**(11), 1145–1160 (2016)
29. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems **32**(1), 4–24 (2020)
30. Xu, J., Shao, Z., Li, X., et al.: Escape: database for integrating high-confidence human tf-target regulatory interactions. Nucleic Acids Research **43**(D1), D929–D934 (2015). https://doi.org/10.1093/nar/gku1161
31. Yang, F., Wang, W., Wang, F., Fang, Y., Tang, D., Huang, J., Lu, H., Yao, J.: scbert as a large-scale pre-trained deep language model for cell type annotation of single-cell rna-seq data. Nature Machine Intelligence **4**(10), 852–866 (Oct 2022). https://doi.org/10.1038/s42256-022-00534-z, https://doi.org/10.1038/s42256-022-00534-z
32. Yang, Z., Ding, M., Zou, X., Tang, J., Xu, B., Zhou, C., Yang, H.: Region or global? a principle for negative sampling in graph-based recommendation. IEEE Transactions on Knowledge and Data Engineering **35**(6), 6264–6277 (2023). https://doi.org/10.1109/TKDE.2022.3155155
33. Yuan, Y., Bar-Joseph, Z.: Deep learning for inferring gene relationships from single-cell expression data. Proceedings of the National Academy of Sciences **116**(52), 27151–27158 (2019)
34. Zhu, X., Jing, X.Y., Zhang, F., Zhang, X., You, X., Cui, X.: Distance learning by mining hard and easy negative samples for person re-identification. Pattern Recogn. **95**(C), 211–222 (Nov 2019). https://doi.org/10.1016/j.patcog.2019.06.007, https://doi.org/10.1016/j.patcog.2019.06.007

6 Supplementary

6.1 Comparison of Pre-trained Single-Cell Foundation Models

Here, we present a comprehensive comparison of the large-scale pre-trained single-cell foundation models utilized in our study: Geneformer, scFoundation, and scBERT. These models exhibit notable differences in training size, model architecture, design parameters, input data, and other technical features, showcasing their distinct strategies for managing single-cell transcriptomic information. It's important to note that Geneformer includes additional pre-trained models with varying training sizes and design configurations; however, we have employed the largest pre-trained model available to date for this analysis.

Table 4: Comparision of the large-scale single-cell foundation models used in our study

	Geneformer	**scFoundation**	**scBERT**
Training Size: #single-cells	95M	50M	1M
Model architecture	Encoder Only	Asymmetric Encoder-Decoder	Encoder Only
Design (Layer-Head-Dim)	Transformer: 20-14-896	Encoder Transformer: 12-12-768 Decoder Performer: 6-8-512	Performer: 6-10-200
Input value	Ranked normalized expression values	continuous normalized expression values	binned normalized expression values
Number of input genes	4096 genes with different ranks	19,264 protein-coding or mitochondrial genes	16,906 genes
Masking	Non-Zero genes only	Zero and Non-Zero	Non-Zero genes only
Pre-training Date	Apr-24	Jun-24	Dec-21

6.2 Baseline Methods

We introduce the baseline methods utilized in our study, which encompass a diverse set of approaches for inferring gene regulatory networks. These methods include traditional statistical techniques, machine learning algorithms, and deep learning models, applied to single-cell RNA-seq.

- GNNLink [19] is a graph neural network model that uses a GCN-based interaction graph encoder to capture gene expression patterns.
- GENELink [4] proposes a graph attention network (GAT) approach to infer potential GRNs by leveraging the graph structure of gene regulatory interactions.
- GNE (gene network embedding) [15] proposes a multilayer perceptron (MLP) approach to encode both gene expression profiles and network topology for predicting gene dependencies.
- CNNC [33] proposes inferring GRNs using deep convolutional neural networks (CNNs).
- DeepDRIM [5] is a supervised deep neural network that utilizes images representing the expression distribution of joint gene pairs as input for binary classification of regulatory relationships, considering both target TF-gene pairs and potential neighbor genes.
- GRN-transformer [24] is a weakly supervised learning method that utilizes axial transformers to infer cell type-specific GRNs from single-cell RNA-seq data and generic GRNs.
- Pearson correlation coefficient (PCC) [23] is a traditional statistical method for measuring the linear correlation between two variables, often used as a baseline for GRN inference.
- GRNBoost2 [20] is a gradient boosting-based method for GRN inference.
- GENIE3 [13] is a random forest-based machine learning method that constructs GRNs based on regression weight coefficients, and won the DREAM5 In Silico Network Challenge in 2010.

6.3 End-to-End Training Algorithm for Gene Regulatory Network (GRN) Inference

Algorithm 1 details the end-to-end training process of scRegNet for inferring GRNs using a combination of a pre-trained single-cell foundation model and a graph neural network (GNN).

Algorithm 1 End-to-End Training for Gene Regulatory Network (GRN) Inference

Input: Pre-trained Single-Cell Foundation Model F, raw count matrix $\mathbf{X}$,
dataset $\mathcal{D}$ with positive and negative gene pairs, Gene Set $\mathcal{T}$
Output: Optimized Model for GRN Reconstruction

Dataset Preparation: Split dataset $\mathcal{D}$ into training set $\mathcal{D}_{\text{train}}$, validation set $\mathcal{D}_{\text{val}}$, and test set $\mathcal{D}_{\text{test}}$
Embedding Extraction: Compute gene embeddings using F: $\mathbf{E} = F(\mathbf{X})$
Attentive/Avg Pooling: $p_t = POOLING([E_t^{cell1}, E_t^{cell2}, ..., E_t^{celln}]), \quad \forall t \in \mathcal{T}$
Model Freezing: Freeze the layers of the pre-trained foundation model F
GNN Initialization: Initialize node embeddings for the GNN: $v_t^0 \leftarrow x_t, \forall t \in \mathcal{T}$
for each training epoch **do**
  Graph Encoding: $\hat{v}_t \leftarrow \text{GNN}(v_t^0, \mathcal{D}_{\text{train}})$
  for each batch b containing (TF, target) gene pairs in $\mathcal{D}_{\text{train}}$ **do**
    Joint Embedding Construction: Concatenate transformer and graph embeddings:

$$Z_{\text{joint}}[\text{TF}_b] = \text{CONCAT}(\mathbf{p}_{\text{TF}_b}, v_{\text{TF}_b}), \quad Z_{\text{joint}}[\text{target}_b] = \text{CONCAT}(\mathbf{p}_{\text{target}_b}, v_{\text{target}_b})$$

    MLP Processing: $e_{\text{TF}} = \text{MLP}_1(Z_{\text{joint}}[\text{TF}]), \quad e_{\text{target}} = \text{MLP}_2(Z_{\text{joint}}[\text{target}])$
    Link Prediction: $h_b = \text{CONCAT}(e_{\text{TF}_b}, e_{\text{target}_b}), \quad \text{prob} = \text{softmax}(\mathbf{W}h_b + \mathbf{b})$
    Loss Calculation: Compute binary cross-entropy loss $\mathcal{L}_{\text{CE}}$
    Parameter Update: Optimize the GNN and MLP weights via backpropagation
  end for
end for
Evaluation: Assess the model on $\mathcal{D}_{\text{test}}$ and report performance metrics (AUROC, AUPRC)
return Optimized Model $\mathcal{M}^*$

6.4 Hyperparameter Search Space

For each combination of dataset and model, we conducted 50 optimization trials, ensuring a thorough investigation of potential configurations. Optuna's Tree-structured Parzen Estimator (TPE) was employed to intelligently navigate the search space, prioritizing regions with higher potential based on previous trials. We also applied early stopping criteria to save computational resources by halting underperforming trials early.

Table 5: * Applicable only for Graph Attention Network models

hyperparameter	search space	type
learning rate	[1e-5, 1e-2]	continual
weight decay	[1e-5, 1e-4]	continual
dropout	[0.1, 0.8]	continual
#GNN layers	[1, 6]	discrete
#MLP layers	[1, 6]	discrete
batch size	[32, 56]	discrete
hidden dimension	[4, 256]	discrete
#attention heads*	[1, 8]	discrete
reduction*	[concatenate, mean]	discrete
alpha*	[0.01, 0.5]	continual
optimizer	[Adam, RMSprop, SGD]	discrete

6.5 Additional Experimental Results

Comparative analysis of link prediction performance in GRNs using different GNN architectures as backbone encoders for scRegNet, with Geneformer serving as the foundation model. The experiments were conducted using two different sets of most variable genes combined with transcription factors (TFs): (a) 500 most variable genes and (b) 1000 most variable genes.

Table 6: Comparative Analysis of Link Prediction Performance in GRNs Using Popular GNN Variants as Backbone Graph-Based Encoders for scRegNet, with Geneformer Serving as the Foundation Model Backbone.

((a)) TFs + 500 Most Variable Genes

Cell Type		**GCN**	**SAGE**	**GAT**
hESC	AUROC	0.886 ± 0.002	0.884 ± 0.004	0.886 ± 0.004
	AUPRC	0.622 ± 0.004	0.623 ± 0.003	0.614 ± 0.002
hHEP	AUROC	0.902 ± 0.002	0.901 ± 0.003	0.901 ± 0.003
	AUPRC	0.841 ± 0.003	0.839 ± 0.004	0.838 ± 0.003
mDC	AUROC	0.808 ± 0.003	0.788 ± 0.001	0.808 ± 0.002
	AUPRC	0.157 ± 0.002	0.148 ± 0.002	0.157 ± 0.001
mESC	AUROC	0.926 ± 0.001	0.926 ± 0.004	0.926 ± 0.002
	AUPRC	0.860 ± 0.001	0.859 ± 0.001	0.859 ± 0.003
mHSC-E	AUROC	0.923 ± 0.001	0.923 ± 0.001	0.921 ± 0.004
	AUPRC	0.941 ± 0.002	0.941 ± 0.003	0.939 ± 0.002
mHSC-GM	AUROC	0.931 ± 0.002	0.930 ± 0.002	0.929 ± 0.003
	AUPRC	0.939 ± 0.001	0.939 ± 0.001	0.938 ± 0.003
mHSC-L	AUROC	0.879 ± 0.002	0.877 ± 0.002	0.878 ± 0.002
	AUPRC	0.879 ± 0.002	0.874 ± 0.001	0.876 ± 0.003

((b)) TFs + 1000 Most Variable Genes

Cell Type		**GCN**	**SAGE**	**GAT**
hESC	AUROC	0.883 ± 0.002	0.882 ± 0.002	0.885 ± 0.002
	AUPRC	0.624 ± 0.003	0.620 ± 0.004	0.625 ± 0.001
hHEP	AUROC	0.905 ± 0.004	0.905 ± 0.003	0.905 ± 0.002
	AUPRC	0.848 ± 0.001	0.848 ± 0.002	0.847 ± 0.002
mDC	AUROC	0.828 ± 0.002	0.831 ± 0.003	0.828 ± 0.004
	AUPRC	0.162 ± 0.002	0.146 ± 0.002	0.162 ± 0.003
mESC	AUROC	0.931 ± 0.002	0.931 ± 0.001	0.930 ± 0.003
	AUPRC	0.866 ± 0.002	0.866 ± 0.002	0.865 ± 0.004
mHSC-E	AUROC	0.941 ± 0.004	0.941 ± 0.002	0.940 ± 0.002
	AUPRC	0.956 ± 0.003	0.954 ± 0.001	0.953 ± 0.002
mHSC-GM	AUROC	0.936 ± 0.003	0.937 ± 0.004	0.935 ± 0.003
	AUPRC	0.949 ± 0.002	0.950 ± 0.001	0.948 ± 0.003
mHSC-L	AUROC	0.875 ± 0.001	0.875 ± 0.003	0.875 ± 0.002
	AUPRC	0.865 ± 0.001	0.865 ± 0.001	0.864 ± 0.004