GRNFomer: Accurate Gene Regulatory Network Inference Using Graph Transformer

Akshata Hegde[1,2] and Jianlin Cheng[1,2,*]

[1]Department of Electronics and Communication, University of Missouri, Columbia, MO, USA 65201 [2]Roy Blunt NextGen Precision Health, University of Missouri, Columbia, MO, USA 65201

*To whom correspondence should be addressed.

Contact: chengji@missouri.edu

Abstract

Motivation: Gene Regulatory Networks (GRNs) are crucial for understanding cellular processes, but accurately inferring them from gene expression data remains challenging due to the complex, non-linear interactions between genes and the high dimensionality of the data. We introduce GRNFormer, an advanced graph transformer model designed to accurately infer regulatory relationships between transcription factors and target genes from single-cell RNA-seq transcriptomics data, while supporting generalization across species and cell types.

Results: GRNFormer surpasses existing methods in both accuracy and scalability, achieving an AUROC of 90% and an AUPRC of 86% on test datasets. Our case study on human embryonic stem cells highlights its ability to identify biologically meaningful regulatory interactions and pathways. The freely accessible GRNFormer tool streamlines GRN inference, presenting significant potential to drive advancements in omics data analysis and systems biology.

Availability: https://github.com/BioinfoMachineLearning/GRNformer.git

1 Introduction

Gene expression regulation is pivotal for biological processes such as cell differentiation, signal transduction, and environmental responses. Proper regulation ensures genes are expressed at appropriate levels and times, supporting cellular adaptation and specialized functions. Disruptions in gene regulation can lead to diseases like cancer, neurodegenerative disorders, and developmental abnormalities. Understanding regulatory mechanisms provides valuable insights into biological pathways, such as metabolism and immune responses, and facilitates the identification of therapeutic targets through the mapping of gene regulatory networks (GRNs)[1].

However, inferring GRNs from high-throughput data like RNA-Seq transcriptomics data presents significant challenges. Traditional methods, such as correlation-based approaches, fail to capture non-linear, context-specific interactions like feedback loops and combinatorial effects. Advanced techniques, including ARACNE, CLR, and Bayesian networks, improve non-linear dependency detection but face scalability issues [2,3]. Dynamic models, such as Boolean networks and ordinal differential equations (ODEs), provide temporal insights but require detailed time-series data, which is often unavailable. RNA-Seq data's high dimensionality and noise, combined with small sample sizes, exacerbate overfitting and reduce reliability of the existing methods[4]. Recent benchmarking studies, such as BEELINE[5], underscore the limitations of conventional statistical and shallow machine learning methods for GRN inference.

To address these challenges, newer approaches, particularly those based on deep learning, have been developed. These methods are capable of learning complex, non-linear patterns from high-dimensional

data, making them better suited for inferring gene regulatory networks from RNA-Seq data [6,7,8,9]. Latest deep learning models, such as transformers, can integrate diverse types of biological information, to provide more accurate and robust predictions of gene regulatory interactions [10,11]. Transformers, the foundation of advanced large language models like ChatGPT, have revolutionized the analysis of input information through the attention mechanism [12]. It captures both global and local information while emphasizing relevant features through automatic weighting. There have been a few transformer-based GRN inference methods developed such as STGRN [13], ScTRansNet[14].

In parallel, Graph Neural Networks (GNNs) also have emerged as a useful tool for inferring GRNs, complementing transformer-based methodologies. GNNs are designed to operate on graph-structured data where nodes represent entities and edges symbolize their relationships[15]. GNNs excel in modeling the complex configurations of gene interactions, characterized by regulatory connections between transcription factors and target genes. By propagating information through these graphs, GNNs can identify the topological features of GRNs, revealing key regulators and elucidating complex hierarchical regulatory interactions vital for gene expression. Furthermore, GNNs effectively capture both local interactions and global structures, establishing them as robust instruments for clarifying biological pathways and functional modules within GRNs[16].

The combination of GNNs and transformers represents a promising frontier in GRN inference, as both models provide distinct advantages: transformers excel at handling sequence-based data and global interaction, while GNNs specialize in capturing the inherent network structure of gene interactions. In this spirit, we propose GRNFormer: a novel graph transformer model for improved GRN inference by effectively modeling both the dynamics of single-cell gene expression data and the underlying network topology of gene regulation.

The overview of the GRNFormer is shown in Figure 1. GRNFormer consists of three main novel designs: (1)TFWalker: a de-novo Transcription Factor (TF) centered subgraph sampling method to extract local or neighborhood co-expression of a transcription factor (TF) to facilitate GRN inference; (2) End-to-End Learning: (a) Gen-Transcoder: a transformer encoder representation module for encoding single-cell RNA-seq (scRNA-seq) gene expression data across different species and cell types and (b) a graph transformer model with a GRNFormer Encoder and a variational GRNFormer decoder coupled with GRN inference module for the reconstruction of GRNs; and (3) a novel inference strategy, to incorporate both node features and edge features to infer GRNs for given gene expression data of any given length.

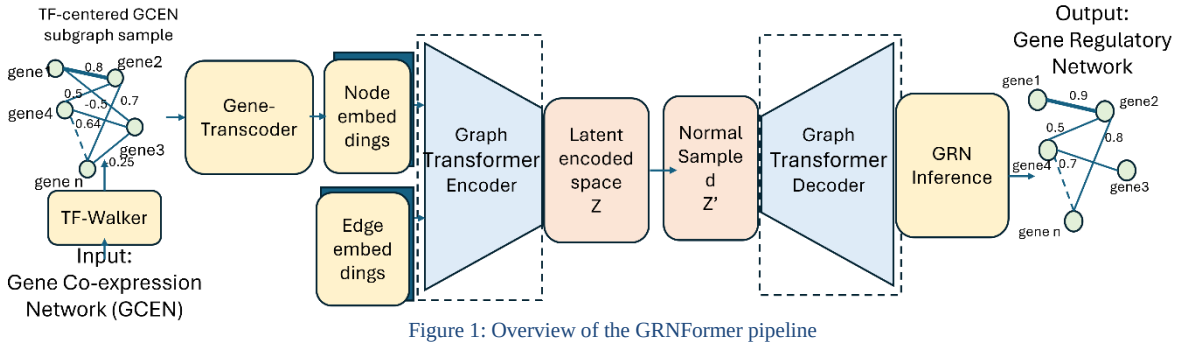

Figure 1: Overview of the GRNFormer pipeline

2 Methods

As shown in Figure 1, given a scRNA-seq dataset, a gene co-expression network is first constructed, from which a set of subgraphs are sampled by TF-Walker. The subgraphs are processed by GeneTranscoder to generate node and edge embeddings, which are fed to the variational graph transformer auto-encoder to learn a GRN representation. The representation is used to infer a gene regulatory sub-network for each subgraph. The subnetworks are aggregated to construct a full GRN. The entire GRN inference process is detailed in the subsections below.

2.1 Construction of gene co-expression network (GCEN)

The raw scRNA-seq gene expression data is transformed using the ArcSinh transformation, a technique designed to normalize and stabilize high-dimensional, highly variable data[17]. After normalization, pairwise Pearson correlation analysis is conducted on the transformed expression data to calculate correlation coefficients between every two genes across a set of single cells. The Pearson correlation quantifies the linear relationship between genes, with coefficients ranging from -1 to 1: positive values indicate positive co-expression, while negative values reflect negative co-expression. Next, a threshold is applied to filter out weak co-expression relationships, retaining gene pairs with absolute correlation values greater than 0.1 to construct a Gene Co-Expression Network (GCEN). This step is crucial as it ensures only meaningful connections, likely representing functional gene relationships, are included. The resulting output is a network graph, where nodes represent genes and edges indicate co-expression relationships.

2.2 Feature extraction

In GRNFormer, node and edge features are crucial for learning regulatory relationships among genes. The vector of gene expression values across a set of cells for each gene are utilized as node features. An extra binary node feature is used to indicate if a gene is a transcription factor (TF). The co-expression values are assigned as edge features. Nevertheless, scRNA-seq datasets often exhibit an imbalance between the number of genes and cells, leading to high-dimensional co-expression networks (GCENs) derived from a single sample. This disparity creates computational challenges, as the model must navigate a sparse and extensive network, hindering the extraction of biologically relevant insights. To address this issue, we developed an advanced subgraph sampling technique that focuses on biologically meaningful subgraphs, optimizing computational efficiency while preserving essential co-expression information for GRN inference.

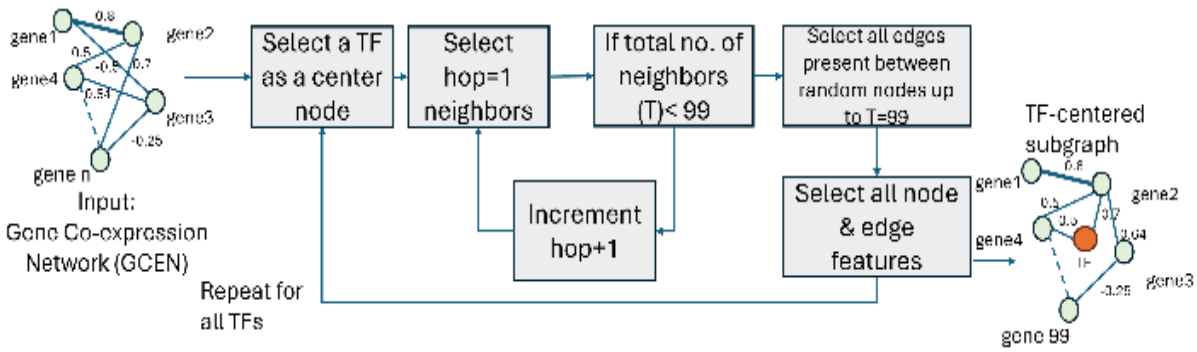

Figure 2:The workflow of TF-Walker for de novo sampling of a subgraph for each TF from GCEN

2.3 TF-Walker – de-novo subgraph sampling method

Figure 2 depicts our de novo subgraph sampling algorithm aimed at deriving significant subgraphs from the larger GCEN. This method initiates by assigning a transcription factor (TF) as the central node of the subgraph. The algorithm subsequently identifies immediate first hop neighbors (hop=1). Should the number of neighbors fall below a specified threshold (i.e., 99), the algorithm progressively increases the hop value, broadening the subgraph by integrating more distant nodes. This iterative approach persists until a biologically relevant subgraph is formed with total 100 nodes. If the number of nodes in the neighborhood exceeds the threshold (T = 100), then random nodes from the neighborhood is selected to meet the size constraint. The resultant subgraph encompasses all node and edge features to encapsulate the potential regulatory linkages within the network effectively. Once the subgraph sample is created, we apply Z score normalization across all the genes for each cell, enabling the model to understand the local context of the gene expressions present in the subgraph. By prioritizing biologically meaningful subgraphs over randomly selected ones, this methodology enhances the model's capacity to discern complex co-expression patterns while considerably reducing the computational burden associated with extensive GRN inference. Moreover, it allows us to generate many data samples from one GCEN, which are critical to train machine learning methods to infer GRNs.

2.4 Ground truth network sampling

For each sampled subgraph of a GCEN, it is essential to obtain the corresponding ground truth gene-regulatory subgraph (labels) to ensure accurate model training and validation. To achieve this, we merge three distinct regulatory information sources—cell-type specific ChIP-seq networks, non-specific ChIP-seq networks, and STRING interaction networks provided by BEELINE—to construct a comprehensive ground truth regulatory network for each cell-type applicable across human and mouse datasets. This integration enhances the model's ability to generalize across various biological contexts, capturing diverse types of regulatory interactions, co-expression relationships, and known protein-protein interactions. By leveraging multiple sources, we create a more robust and biologically relevant regulatory network, which serves as the ground truth for training and benchmarking GRN inference models.

The process of subgraph labeling works as follows. First, all genes present in a subgraph sampled by TF-Walker are identified. Next, the regulatory edges corresponding to these genes are extracted from the comprehensive ground truth regulatory network. A regulatory subgraph is then constructed by retaining only those edges that represent actual regulatory interactions between the genes in the subgraph. If the resulting regulatory subgraph lacks any edges—indicating no known regulatory relationships between the genes—the sampled subgraph is discarded. This ensures that only biologically meaningful subgraphs with verified regulatory interactions are used for model training, thereby improving the model's accuracy and ability to infer gene regulatory networks from high-dimension data.

2.5 Node features and edge features processing

Every TF-centered subgraph contains genes as nodes and the co-expressed gene pairs as edges. Each subgraph is a homogeneous network with an exact number of nodes (i.e., 100 genes).

To enhance the model's learning, the gene expression values in node features of each gene is normalized as z-scores across all the genes for each cell. This normalization helps ensure that the local context is appropriately captured, providing a more consistent representation of gene expression patterns across the cells in the dataset.

It is worth noting that, the edge feature, i.e., the co-expression value between two genes in a subgraph, is the same as the one in the entire GCEN, which reflects the global relationships between genes, while the gene expression values in node features offer a more localized perspective.

2.6 GENE-Transcoder: Transformer encoder for gene expression representation learning

Figure 3 illustrates the architecture of the GENE-Transcoder, a transformer-based encoder designed for gene expression representation learning, which is the first learning stage of GRNFormer. The primary challenge addressed here is the variability in gene expression data from scRNA-seq experiments, where the number of genes and cells can differ significantly across datasets. This variability makes it difficult for machine learning models to learn and generalize effectively. To tackle this issue, the GENE-Transcoder processes gene expression data to produce representative embeddings of fixed size that capture essential biological information regardless of samples, cell types, and species.
Given an input gene expression matrix X ∈ R^(batchsize×100×# of cells), GENE-Transcoder goes through a number of transformations through convolutional layer and transformer encoder layers before ending with a mean pooling operation.

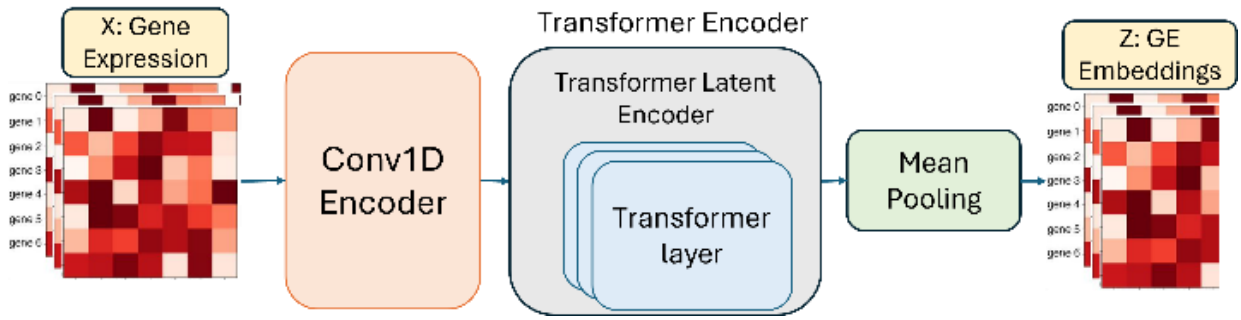

Figure 3: The architecture of Gene-Transcoder for learning the representation (embedding) of gene expression data.

First, a 1D convolutional layer processes the input matrix gene by gene (row by row). This convolution utilizes a sliding kernel along the cell axis to capture local patterns of gene expression across cells. The output from the convolutional layer is then fed into a transformer encoder, where a multi-head attention mechanism is employed. This mechanism helps the model capture both local and global interactions between genes and cells by producing embeddings that represent gene expression data in an embedding feature space. Finally, a mean pooling operation is applied to the transformer output, averaging the gene embeddings over all cells. By lowering the dimensionality and providing each gene with a consistent, meaningful representation, this enables the model to handle the inherent unpredictability in gene expression data. Detailed methodology and mathematical algorithm for the Gene-Transcoder is provided in the supplementary Algorithm S1.
The resulting gene embeddings of dimension Z ∈ R^(batchsize×100×64) are context-aware representations of gene expression, providing a robust feature set for downstream task such as gene regulatory network (GRN) inference. These embeddings are independent of dataset-specific variations, enabling the model to generalize effectively across species, cell types, and experimental conditions.

2.7 GRN representation learning

The second stage of the GRNFormer pipeline is the GRNFormer feature learning model, which is the backbone or core of the pipeline. GRNFormer model is a supervised Variational Graph Transformer Autoencoder, comprising a graph transformer encoder and a variational graph transformer decoder (Figure 4). This model processes an input subgraph consisting of the fixed gene expression representations (node embeddings) generated by the GENE-Transcoder in conjunction with co-expression edge weights as edge embeddings, subsequently generating essential high-level features (representations) for predicting the gene regulatory relationships within the subgraph. GRNFormer update both node

embeddings and edge embeddings through message passing, ensuring that the model integrates gene expression representations and co-expression weights collaboratively to predict the regulatory interactions within the specified subgraph sample. The learning process of GRNFormer inference model is described as follows.

2.7.1 GRNFormer Encoder: Learning Node and Edge Latent Representations

The GRNFormer Encoder (Figure 4A) is a multi-layered graph transformer encoder that processes input node embeddings and edge embeddings in a subgraph. The encoder starts by taking in the input node embeddings, denoted as $X_{node} \in \mathbb{R}^{100\times 64}$, $where\ 100\ is\ number\ of\ nodes\ in$ the subgraph and 64 is the fixed representation of the gene expression or dimension of node features from GeneTranscoder, and the edge embeddings, X_{edge}, of dimension [2,n,1], where 2 stands for source and target node, n is number of edges in the subgraph and 1 is the dimension of edge feature. Both the node embeddings and the edge features are passed through a series of GRNFormer blocks. Each GRNFormer block combines a transformer convolution layer with multi-head attention and feedforward layers to facilitate message passing between nodes and edges. This mechanism allows the model to capture both local interactions (e.g., between closely related genes) and global patterns (e.g., co-expression relationships across a subgraph).

A critical feature of the encoder is its variational nature. The last step of encoder includes two parallel streams of computation for each block: one for estimating the mean (μ) and another for the variance (σ^2) of the latent space. This is achieved through separating GRNFormer block for $\mu_{\text{GRNFormer}}$ and $\sigma_{\text{GRNFormer}}$, which compute the latent node and edge representations as $Z_{\text{node}}\ and\ Z_{\text{edge}}$. The Gaussian sampling step uses these outputs to generate sampled latent variables Z'_{node} and Z'_{edge} , ensuring that the latent representations are continuous, regularized, and biologically meaningful. The encoder encodes the input into the latent space with dimension d=16.

Biologically, the encoder ensures that the learned representations integrate both gene expression data and co-expression relationships. By doing so, it captures regulatory patterns that span cells and global co-expression networks, helping the model generalize across species and cell types.

Unlike traditional autoencoders, which establish a deterministic mapping, the variational autoencoder conceptualizes the latent variables as distributions, represented by a Gaussian distribution. The GRNFormer Encoder generates parameters (mean and variance) corresponding to this distribution for each node in the graph, from which the latent node embeddings are subsequently sampled. This probabilistic characteristic facilitates the model's ability to encapsulate uncertainty within the data, a feature that is especially critical when addressing noisy gene expression data derived from single-cell RNA sequencing.

2.7.2 GRNFormer Variational Decoder

The GRNFormer Decoder (Figure 4B) takes the sampled latent representations, Z'_{node} and Z'_{edge}, from the encoder as input and reconstructs the regulatory interactions network representations within the subgraph. The decoder consists of multiple GRNFormer blocks followed by a lightweight multi-layer perceptron (MLP) module. The decoder learns to predict node representations Z'_{node}, capturing the regulatory identity of each gene, and edge attentions Z'_{edge}, reflecting the regulatory strength between gene pairs. The last GRNFormer block of the decoder reconstructs the node representations and edge representations of original dimension of 64. Then, the light weight MLP layers map the decoded dimension of 64 to a fixed dimension d=16 to generate meaningful representation aligning with latent space dimensions, also helping in fast computations.

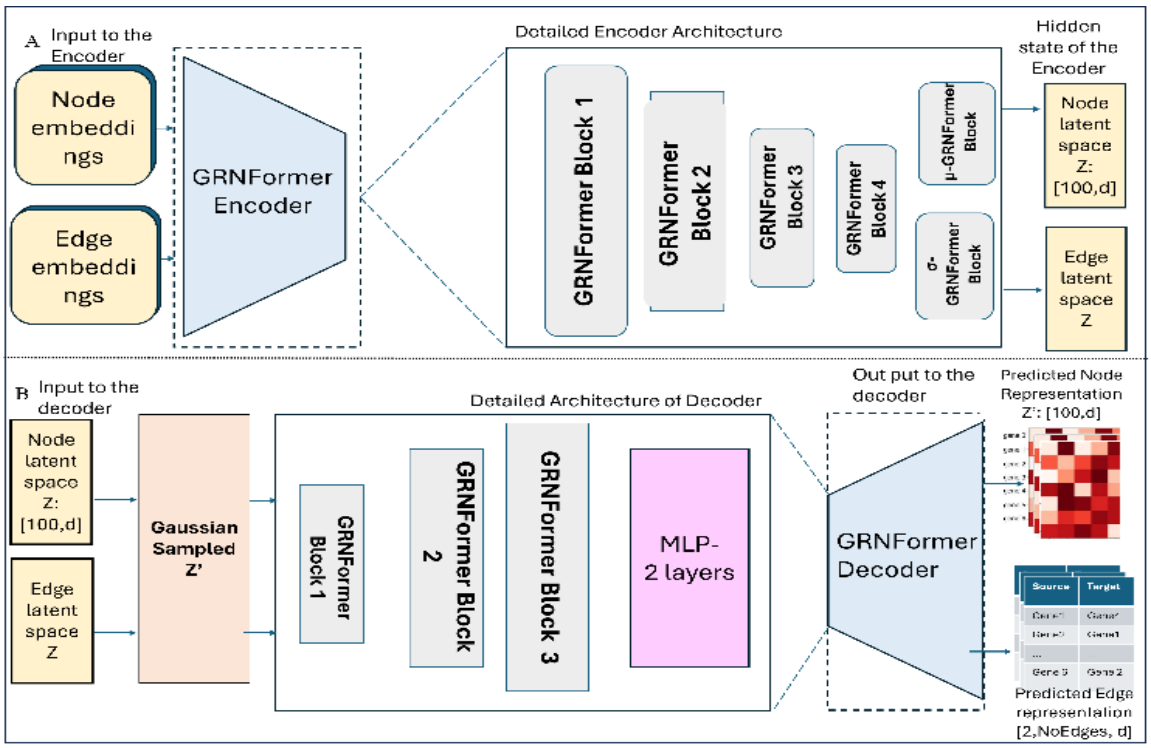

Figure 4:The architecture of GRNFormer. (A) the encoder module; (B) the decoder module.

The decoder generates two primary outputs: (1) a reconstructed node embedding matrix that represents updated regulatory roles for the input genes, and (2) edge attentions that highlight the strength of gene-to-gene regulatory interactions. These outputs serve as a proxy for the predicted gene regulatory network (GRN).

The variational decoder training approach includes a regularization term within the model's objective function, ensuring that the acquired latent space representations are smooth and continuous. This regularization process is performed by minimizing the Kullback-Leibler (KL) divergence between the approximate posterior (the derived distribution of node representations) and the prior distribution (a standard Gaussian distribution). Beyond the reconstruction loss, this regularization process enhances the model's generalization capabilities by reducing the risk of overfitting, increasing its ability to predict gene regulatory interactions across a variety of datasets.

The GRNFormer encoder and decoder work in tandem to transform gene expression and co-expression data into biologically meaningful representations, which are then used to predict GRNs. This pipeline achieves a balance between preserving local biological relevance (e.g., individual gene interactions)

and capturing global regulatory contexts (e.g., overarching network patterns), making it a powerful tool for GRN inference.
The technical details of the GRNFomer block used in the GRNFomer autoencoder and its key component - graph transformer convolution layer with pairwise attention - are described in detail below.

2.7.3 GRNFormer Block

The Graph Transformer Convolution Layer (Figure 5B) is the primary and foundational component of the GRNFormer Block shown in the block diagram Figure 5(A), functioning as the key mechanism for processing structured data. This layer employs a multi-head attention mechanism to compute the pairwise attention (explained in Section 2.7.4). Furthermore, this layer retains the merits of convolutional networks, such as local feature extraction, while augmenting their capability to model global contextual interactions through attention. This dual functionality renders the Graph Transformer Convolution Layer essential for the overall efficacy of the GRNFormer block.
Building upon the outputs generated by the Transformer Convolution Layer, the multi-layer perceptron (MLP), serves as the next critical component in the GRNFormer block. This layer further transforms the feature representations by implementing dense connections and nonlinear transformations. The MLP not only enriches the feature space but also improves the network's ability to capture higher-order interactions and abstractions inherent in the data.
To incorporate non-linearity and empower the network to learn more complex relations, the GRNFormer block incorporates the Leaky ReLU activation function following the MLP. In contrast to the conventional ReLU, which nullifies negative inputs, Leaky ReLU permits a minor gradient for these negative input features, thereby addressing potential complications such as dead neurons and enhancing gradient propagation during backpropagation. Leaky ReLU improved GRNFormer's performance drastically, compared to ReLU, as Leaky ReLU helps learn negative co-expressions that has significant impact on gene regulations. This activation function also guarantees that the network remains resilient and capable of learning meaningful features, even within deep architectures[18].
The final element of the GRNFormer block is the Batch Normalization Layer, which stabilizes and expedites the training process. By normalizing the outputs of the preceding layers across mini-batches, this layer alleviates issues such as internal covariate shift, wherein variations in layer inputs during training can impede convergence. Batch normalization ensures a more stable learning process and enhances the model's ability to generalize effectively to previously unseen data.

2.7.4 Graph transformer convolution layer with pairwise attention

Figure 5B depicts a graph Transformer Convolution Layer (TransConv), an architecture designed to enhance message-passing in graphs by simultaneously updating both node and edge features. This architecture leverages **pairwise attention**, which explicitly incorporates both node features X and edge features E into the computation of feature updates. Pairwise attention is a key extension of the standard attention mechanism, enabling the model to account for both node-to-node interactions and the relationships encoded in the edges. This makes it particularly powerful for tasks involving structured data, such as graph representation learning.

Pairwise attention mechanism

The pairwise attention mechanism (Figure 5B) computes the importance of a neighboring node (j) and its connecting edge (e_{ij}) relative to a target node (i). For the target node (i), a $Query\ Vector(Q)$ is computed by transforming its feature vector (X_i) using a learnable weight matrix (W_3):

$Q = W_3 X_i$.
Similarly, a $Key\,Vector(K)$ for a neighboring node (j) is calculated as: $K = W_4 X_j$. To explicitly incorporate edge information, the edge feature (E_{ij}), representing the relationship between nodes (i) and (j), is transformed using a weight matrix (W_6). The combined representation of the key vector (K) and edge feature (E_{ij}) is expressed as: $K + W_6 E_{ij}$.
The attention mechanism then computes the attention score (a_{ij}), representing the relative importance of node (j) and its connecting edge (e_{ij}) to node (i). This is done using a scaled dot-product operation:

$$a_{ij} = \text{softmax}\left(\frac{Q\cdot(K+W_6 E_{ij})}{\sqrt{d}}\right),$$

where (d) is the dimensionality of the feature space and serves as a scaling factor to stabilize the computation of dot products. The pairwise attention mechanism ensures that both node and edge features contribute meaningfully to the attention computation, allowing the model to learn the nuanced importance of edges within the graph.

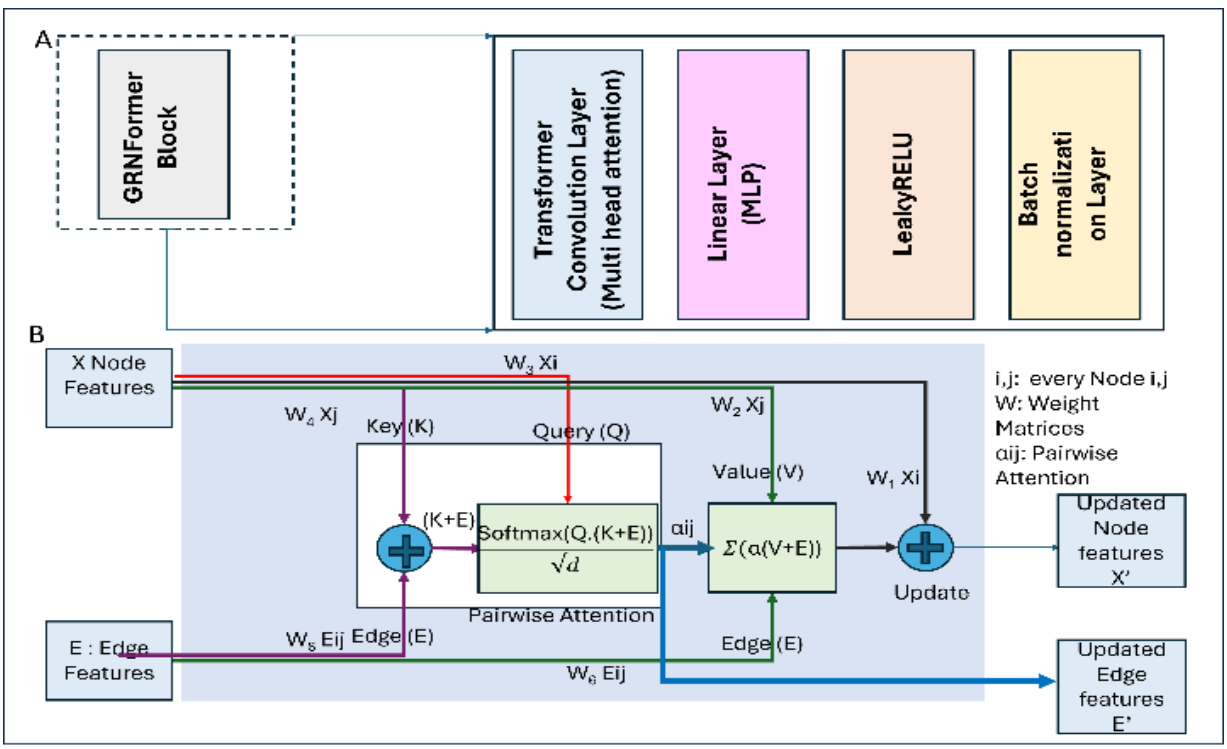

Figure 5:(A) A GRNFormer block; (B) a graph transformer convolutional layer based on the pairwise attention present in a GRNFormer block.

Message Passing and Node Feature Update

Once the attention scores (a_{ij}) are computed, they are used to aggregate information from neighboring nodes. Each neighboring node (j) sends a message to the target node (i). This message is constructed using the $Value\ Vector$ (V), which is derived by transforming (X_j) as:
$V = W_2 X_j$.

The message is further enriched by the transformed edge feature $(W_6 E_{ij})$, resulting in the final message term:

$$m_{ij} = a_{ij} \cdot \left(\boldsymbol{V} + \boldsymbol{W_6 E_{ij}}\right),$$

where the attention score (a_{ij}) adaptively scales the contribution of the message from (j) based on the pairwise importance between $(j), (i)$, and their connecting edge (e_{ij}).

The updated node feature (X_i') is obtained by combining the original feature $(W_1 X_i)$ with the aggregated messages from all its neighbors:

$$\boldsymbol{X_i'} = \boldsymbol{W_1 X_i} + \sum_{j \in \mathcal{N}(i)} m_{ij}$$

where $(\mathcal{N}(i))$ represents the set of neighbors of node (i). This formulation ensures that the updated feature of node (i) is informed not only by its own attributes but also by the weighted contributions from its local neighborhood, incorporating both node-to-node and edge-based relational information. Similarly, the edge features are updated independently as updated attention weights for edge e_{ij} :

$$\boldsymbol{E_{ij}'} = \alpha_{ij}.$$

2.8 GRN Inference

The final stage of the GRNFormer pipeline is to infer GRNs. This stage focuses on creating a probabilistic adjacency matrix that reflects the probabilistic gene regulatory network (GRN), with each element indicating the possibility of a regulatory interaction between two genes.

The GRNFormer Decoder, produces two outputs: the Predicted Regulatory Node Representation (Z') and the Predicted Edge Representation. The Predicted Node Representation (Z') is a matrix of size $[100, (d)]$, with each row encoding a learnt embedding of dimension *d* for a gene, which captures its regulatory identity. In contrast, the Predicted Edge Representation is a tensor with the shape $[2, n, (d)]$, which represents regulatory properties between gene pairs, where n denotes the number of edges, 2 the source and target nodes of an edge, and *d*=16 the dimension of the node and edge embedding.

The initial adjacency matrix is derived using an inner product operation applied to predicted node representation (Z'), specifically:

$$AdjacencyMatrix = (Z' \cdot (Z')^T)$$

This operation produces a symmetric matrix of size [100, 100], with each element quantifying the strength of the regulatory relationship between a pair of genes.

To further refine the adjacency matrix and enhance the representation of regulatory relationships among co-expressed genes, the mean-pooled Predicted Edge Representations across dimension *d=16* are added to the adjacency matrix at corresponding positions. This augmentation leverages edge-specific features captured during the decoding process and integrates them with the initial adjacency matrix derived from node representations. The addition of the mean-pooled edge representations reinforces the regulatory interaction strength, particularly for gene pairs that are co-expressed, ensuring a more robust estimation of the GRN. To transform these scores into interpretable probabilities, a sigmoid activation function is applied element-wise. The resulting matrix is the Edge Probabilities matrix or adjacency matrix, encodes the probabilistic regulatory interactions between genes, with values constrained between 0 and 1. This methodology enables the exploration of gene co-expression subgraphs, uncovering tightly regulated gene regulatory relationships. The enhancement of regulatory

relationships via the edge representation pooling ensures the detection of subtle yet biologically relevant interactions. This inference method however infers a gene regulatory sub-network for each subgraph. The gene regulatory sub-networks for all the sub-graphs associated with an input dataset can be combined to construct a full gene regulatory network (see Section 2.10 for details).

2.9 Datasets and Preprocessing

2.9.1 BEELINE data

We utilized datasets provided by BEELINE to train, validate and test GRNFormer and compare it with existing methods. BEELINE provides curated gene expression datasets derived from experimental scRNA-seq studies spanning two species, seven distinct cell types, and three regulatory relationship categories. By covering multiple species, the dataset reflects evolutionary conservation and divergence in regulatory mechanisms, allowing for cross-species generalization. Similarly, the inclusion of numerous cell types ensures that the model can detect GRNs in a wide range of biological settings, from differentiated cells to stem-like states. The incorporation of different regulatory relationships strengthens the model's ability to discern between various modalities of gene regulation, increasing its interpretability and biological validity.

Specifically, we chose five datasets—three from mouse and two from humans—that represent seven diverse cell types and are critical for understanding gene expression dynamics across different biological situations. The databases include expression data from mouse hematopoietic stem cells (mHSCs) which have 3 sub cell-types (mHSC-E, mHSC-GM & mHSC-L), mouse embryonic stem cells (mESCs), mouse dendritic cells, human hepatocytes (hHEP), and human embryonic stem cells (hESCs). These datasets encompass multiple stages of cellular differentiation and responses, making them a valuable resource for understanding gene regulatory networks across species and cell types. For instance, the mHSC dataset contains expression data from 1,656 hematopoietic stem and progenitor cells, whereas the mESC dataset has data from 421 primitive endoderm cells that differentiated over time. Additionally, the hHEP dataset contains scRNA-seq data from induced pluripotent stem cells (iPSCs) differentiating into hepatocyte-like cells, and the hESC dataset tracks the differentiation of human embryonic stem cells into definitive endoderm cells.

The preprocessing of these datasets followed a standard procedure, as described in BEELINE. Raw expression values were log-transformed and filtered to remove genes with low expression across cells. Afterward, pseudotime values were computed for each dataset to reflect the progression of cellular differentiation. The preprocessed datasets from BEELINE thus provide reliable, normalized gene expression data that are ideal for training deep learning models focused on GRN inference. In addition to preparing scRNA-seq data, BEELINE provides a range of curated ground-truth networks (labels) for GRN training and evaluation. These include cell-type-specific networks constructed from ChIP-seq data for the relevant cell types, as well as non-specific and functional STRING networks that span broader regulatory interactions. We used the ground truth networks provided by the BEELINE as the gold-standard for our training, evaluation, benchmarking and case studies.

2.9.2 Training

Splitting the data into training, validation, and testing sets is crucial for ensuring adequate model training and fair evaluation. In this study, we selected five datasets—hESC, hHEP, mDC, mHSC-E, and mHSC-GM—for training and validation, while reserving the mESC and mHSC-L datasets for blind testing. This allows us to assess the model’s ability to generalize across species and cell types. These datasets feature single samples with a large number of genes and a variable number of cells. To address

this, we first construct a large gene co-expression network from the gene expression data (see Section 2.1 for details). From a large gene co-expression network, many TF-centered subgraphs are sampled by TF-Walker and shuffled to generate many inputs. We train the model to understand the context of the subgraph given. For training, we combine the ground truth regulatory interactions from three distinct sources—non-specific, cell-type-specific ChIP-seq, and STRING ground truth networks—into a unified general regulatory network as labels (see Section 2.4 for details). This comprehensive dataset is used to train the model.

We trained the GRNFormer model for 100 epochs using a batch size of 8. The architecture of GRNFormer consists of three GRNFormer layers with four attention heads, a fixed embedding dimension of 64, and a latent space dimension of 16 within the GRNFormer graph transformer autoencoder. The training process follows a supervised learning paradigm, where the task is framed as a binary classification problem to predict whether a regulatory edge exists between a pair of genes. This represents an end-to-end learning pipeline encompassing the Gene-Transcoder, the GRNFormer variational autoencoder, and the final GRN inference module.

The training objective is optimized using a composite loss function. As the task is considered a link prediction problem across all potential gene pairs within the subgraph, binary cross-entropy (BCE) loss serves as the core optimization function. To further regularize the model, a Kullback–Leibler (KL) divergence loss is added to the BCE loss, forming the final cost function. This composite loss is minimized using the Adam optimizer with adaptive learning rates. Additionally, a ReduceLROnPlateau learning rate scheduler is employed to dynamically adjust the learning rate when the loss plateaus, starting from an initial rate of 0.001[19]. To prevent overfitting, an early stopping strategy is used, and the model applies L1 regularization by default.

Handling class imbalance inherent in our dataset is another critical aspect of training, particularly in binary classification tasks. In a subgraph sample, which is highly sparse graph, containing 100 genes (corresponding to 10,000 possible regulatory edges), may have only 30 true regulatory edges, leading to extreme class imbalance. Furthermore, the class distributions can vary widely across different samples. To address this, we employed a dynamic negative sampling strategy tailored to the characteristics of our dataset. We sample a number of negative edges which are not present in the ground truth regulatory network, which is equivalent to the number of positive regulatory relationships for effective class balance in training. This approach ensures that the model learns effectively from both positive and negative examples, maintaining a balance that supports accurate edge prediction despite the sparse nature of ground-truth regulatory relationships.

The GRNFormer model is trained without incorporating explicit information about species, cell types, or specific regulatory interactions. To ensure diversity during training, subgraphs derived from the datasets are shuffled, preventing bias toward any particular species or cell type. This approach enables the model to generalize its learning, focusing solely on inferring GRNs from contextual patterns in gene expression data. As a result, GRNFormer effectively infers GRNs across diverse species and cell types, demonstrating its robustness and adaptability to various biological contexts.

2.10 GRN Prediction and Evaluation Metrics

GRNFormer is trained to infer the gene regulatory sub-network for a sub-graph extracted from a full gene co-expression network computed from a scRNA-seq dataset. In the prediction phase, a set of subgraphs, $S_!$,S_2,…, S_k, each containing T=100 nodes, is sampled from the full gene co-expression network. Each sub-graph is centered on each TF. If the number of nodes in the neighborhood exceeds the threshold (T = 100), then the neighborhood is divided into multiple subgraphs to meet the size constraint. Every TF and every node in the full gene co-expression network are covered by at least

one subgraph. The subgraphs may have overlapped nodes and edges. We apply similar transformations and normalizations as those used in the TF-Walker sampling method during the training phase. The details of sampling subgraphs are described in supplementary Algorithm S2. The gene regulatory sub-networks predicted for the subgraphs are aggregated to generate a complete predicted gene regulatory network using the strategy below.

First, the model evaluates TF-TF and TF-gene interactions predicted for each subgraph, generating a matrix of predicted probabilities for regulatory relationships as follows: (1) **Interaction Selection:** Interactions with probabilities >0.3 are chosen to capture strong and weak relationships (Selected Interactions $= \{(i,j) \mid P(\widehat{y_{ij}}) > 0.3\}$, where $P(\widehat{y_{ij}})$ represents the predicted probability of interaction between genes i and j, and $\widehat{y_{ij}}$ is the predicted edge score for the gene pair (i,j)); (2)TF Weight Assignment: Interactions are weighted as 1 if the source gene is a TF, otherwise 0; and (3) Aggregation: Weighted interactions are calculated as: Weighted $\text{Interaction}_{ij} = P(\hat{y}_{ij}) \times (\text{Counts}_{ij} + 1) \times \text{TF Weight}_{ij}$ where Counts_{ij} represents the number of times the interaction between genes i and j is sampled during the evaluation.

The final predictions are averaged across subgraphs and normalized to a [0, 1] scale by the min-max method. To address the class imbalance, where the number of negative edges vastly exceeds positive edges, we adopt a negative sampling approach in the evaluation. we perform random negative sampling from a large set of negative edges, where the number of negative edges sampled is equal to the number of positive edges. For each sampling, we calculate the evaluation metrics for both positive and negative edges. This process is repeated 100 times, effectively employing a bootstrapping method. By iterating the sampling process, we ensure that the negative samples represent the large negative edge population, reducing the skewed metrics, improving the robustness and reliability of the model's performance assessment.

The results of GRN inference are evaluated by multiple metrics, including area under ROC curve (AUROC), the area under the precision-recall curve (AUPRC), F1 score, Precision, Recall, Accuracy. AUROC and AUPRC are used as the main metrics because they are particularly useful for assessing a

Table 1: Average results of GRNFormer across 7 cell types and three kinds of regulatory networks.

Cell Types	Networks	AUROC	AUPRC	Accuracy	F1Score	Precision	Recall
hESC	Non-Specific	0.86	0.80	0.72	0.65	0.79	0.57
	Specific	0.88	0.83	0.79	0.75	0.82	0.70
	String	0.85	0.80	0.73	0.68	0.78	0.62
hHep	Non-Specific	0.89	0.84	0.85	0.85	0.84	0.86
	Specific	0.66	0.63	0.87	0.63	0.61	0.65
	String	0.87	0.84	0.81	0.81	0.80	0.83
mDC	Non-Specific	0.88	0.83	0.86	0.88	0.83	0.94
	Specific	0.66	0.65	0.86	0.64	0.59	0.69
	String	0.86	0.80	0.83	0.85	0.79	0.93
mESC	Non-Specific	0.97	0.95	0.96	0.96	0.92	1.00
	Specific	0.97	0.95	0.96	0.96	0.93	1.00
	String	0.96	0.94	0.95	0.96	0.92	1.00
mHSC-E	Non-Specific	0.94	0.92	0.84	0.81	0.91	0.75
	Specific	0.94	0.90	0.79	0.72	0.90	0.64
	String	0.93	0.91	0.81	0.77	0.90	0.70
mHSC-GM	Non-Specific	0.96	0.94	0.73	0.63	0.95	0.49
	Specific	0.95	0.91	0.63	0.40	0.91	0.27
	String	0.95	0.93	0.72	0.60	0.94	0.47
mHSC-L	Non-Specific	0.97	0.95	0.61	0.36	0.97	0.23
	Specific	0.97	0.95	0.59	0.28	0.95	0.19
	String	0.98	0.96	0.67	0.51	0.98	0.35

method's performance on imbalanced datasets, where the number of true regulatory interactions is much smaller than the non-interacting gene pairs. We benchmarked GRNFormer against four existing tools including PIDC[20], PPCOR[21], LEAP[22], and SCINCIRITES[23] that ran faster and are easily reproducible.

3 Results

3.1 The overall performance of GRNFormer on multiple datasets across cell types

We evaluated the GRNFormer across two different species, 7 different cell types (hESC, hHep, mDC, mHSC-E, mHSC-GM, mESC and mHSC-L). For each single-cell RNA-seq dataset, we created four subsets of genes using the subset creation method provided by the BEELINE that selects genes and TFs based on the corrected p-value and variance for each gene, in each cell-type dataset. The sub datasets are created based on significantly varying genes (p-value<0.01) and transcription factors (TFs). These subsets included (1) the 500 most varying genes, (2) the 1,000 most varying genes, which have p-value<0.01 as well as (3) 500 genes plus the most varying TFs, and (4) 1,000 genes plus the most varying TFs. The most varying TFs, whose variance had p-value at most 0.01 are included.
We considered three different gold standard regulatory networks namely cell-type specific Chip-seq networks, non-specific Chip-seq networks and String networks provided by BEELINE. Each ground-

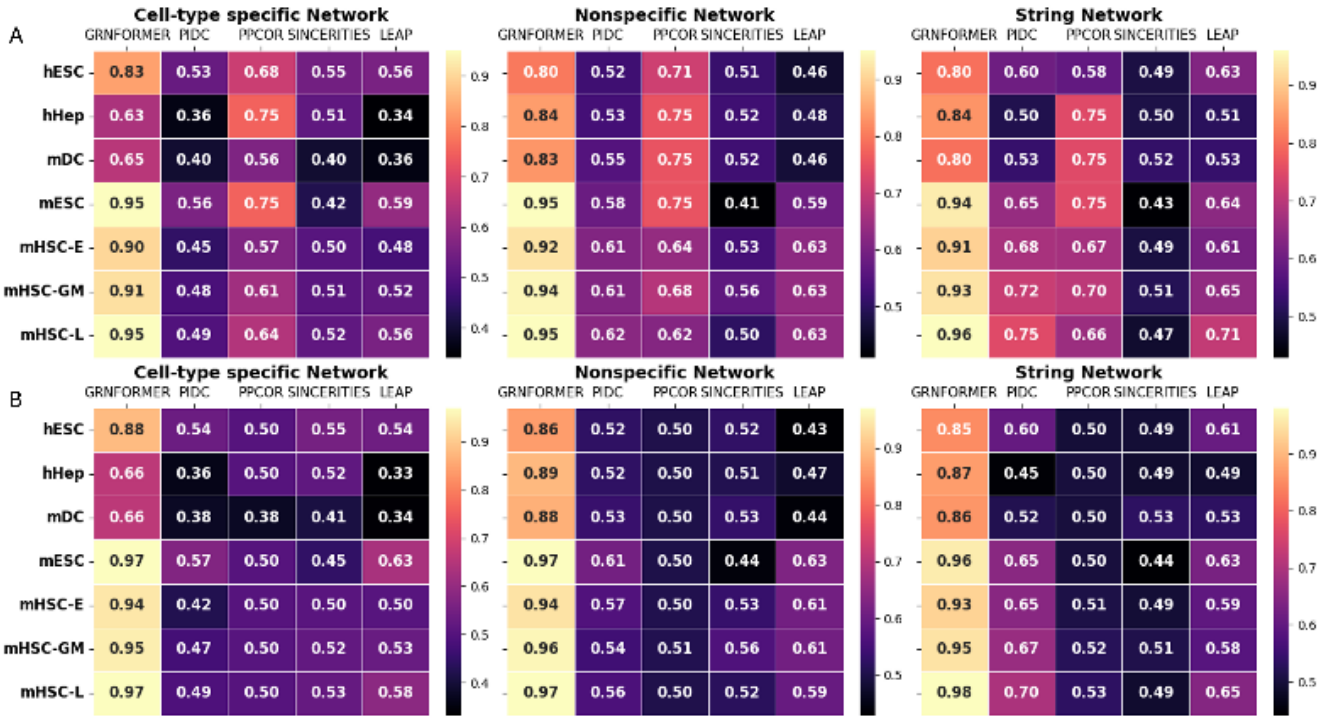

Figure 6: (A) Average AUPRC scores of the five methods (GRNFORMER, PIDC, PPCOR, SINCERITIES, and LEAP) for 7 cell types (hESC, hHep, mDC, mHSC-E, mHSC-GM, mESC and mHSC-L) and three kinds of regulatory networks. (B) average AUROC scores of the five methods. Lighter color indicates better results.

truth regulatory network was intersected with these gene subsets, resulting in 84 datasets, where we filtered the datasets that had less than 1000 significantly varying genes. After filtering we had, a total of 77 different datasets. It is worth noting that the test subgraphs created from the datasets of the five cell types (hESC,hHep,mDC ,mHSC-E,mHSC-GM) seen in the testing are unique and are different from their subgraphs used in the training because the each set of genes considered in the evaluation and the training subgraphs are different. Thus, they can test how well GRNFormer can predict gene regulatory relationships within the given set of genes for the same cell type. The evaluation on datasets for the two cell types (mESC and mHSC-L), which are completely unseen in the training phase, can

evaluate how well the model can generalize and predict gene regulatory relationships across new cell types and species.

The evaluation was carried out using various metrics, including AUROC, AUPRC, F1 score, Precision, Recall, Accuracy. Table 1 summarizes average results of Transformer across 7 cell types and three kinds of networks across 77 datasets.

In non-specific networks, the model achieved an average AUPRC of 0.89, with strong performances in datasets like hESC, hHep, and mDC. Blind datasets like mESC and mHSC-L, not included in training, also showed promising results, indicating the model's ability to generalize to unseen cell types. The model achieved average AUROC scores above 92% across all the datasets, demonstrating reliable accuracy.

In cell-type-specific networks also, the model performed, with an average AUPRC of 0.83, showing its capacity to capture context-specific regulatory patterns. For STRING protein-protein interaction networks, the average AUPRC was 0.88, indicating consistent better performance, especially for mESC (a blind test cell type), and mHSCs. Overall, the GRNFormer model achieved a mean AUPRC of 0.86 across all contexts, showcasing its versatility and ability to generalize across species and cell types.

Further, we also evaluated the model on F1 score, precision, Recall, Accuracy, for a deeper understanding of the inferred regulatory networks. Not only AUPRC but also the F1 score are considered important metrics for classification problems. F1 score, is consistently high in most of the cases except the mHSC-L dataset. Table 1, shows that the model performs with high precision above 90% for most the of the datasets, whereas Accuracy and Recall reach up to the maximum of 0.96 and 1.00 respectively, For mHSC-L which is a blind test cell type, precision is high while recall is low, which shows that, the model predicts highly precise regulatory interactions despite missing some true positives.

3.2 Comparison of GRNFormer with four existing methods

We benchmarked GRNFormer against several state-of-the-art methods widely used for GRN inference: PIDC, PPCOR, SINCERITIES, and LEAP (Figure 6). These algorithms represent a range of methodologies, including correlation-based techniques, probabilistic approaches, and network

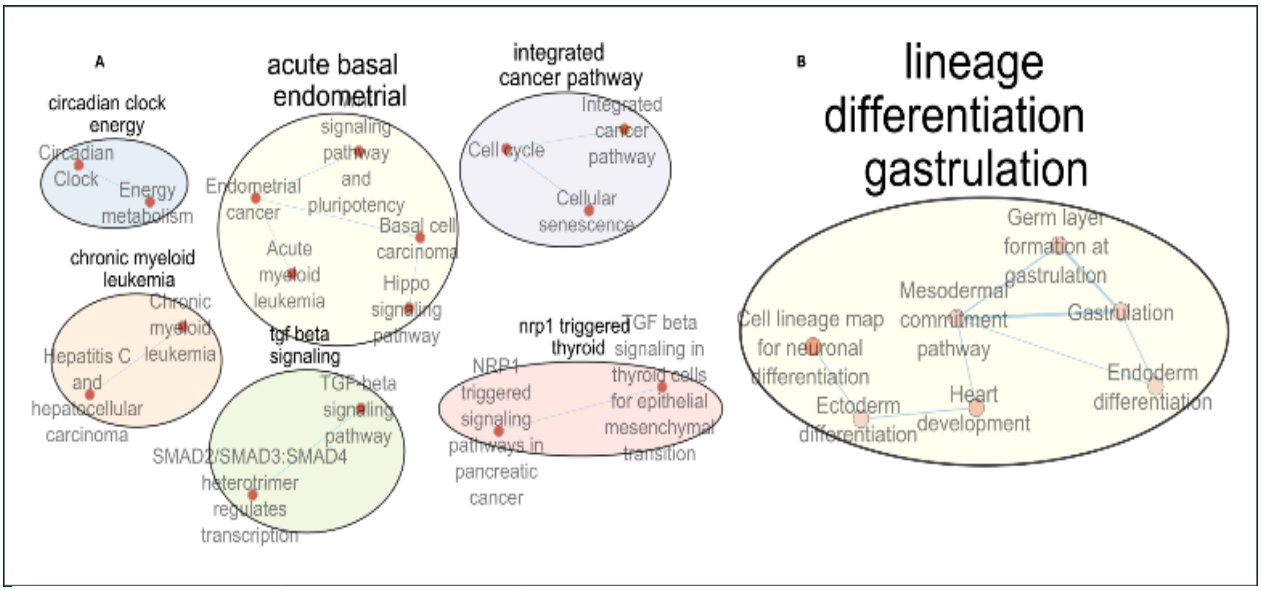

Figure 7: A: Enrichment map of predicted GRN networks of hESC cell type that matched with the ground truth networks. B: Novel pathway predicted by GRNFormer not present in ground truth network.

inference frameworks. PIDC leverages information theory for identifying relationships, while PPCOR uses partial correlation to infer edges. SINCERITIES is a statistical approach for identifying causal gene interactions, and LEAP applies a fast linear correlation-based method to detect regulatory links. The benchmarking results assess GRNFormer's performance across experiments involving over 500 and 1000 transcription factors (TFs) and genes under nonspecific, specific, and STRING regulatory network settings. GRNFormer outperforms existing methods like PIDC, PPCOR, SINCERITIES, and LEAP, demonstrating superior generalizability, robustness, and strong blind evaluation results on datasets of two unseen cell types (mESC and mHSC-L).

In nonspecific networks, GRNFormer achieves the highest AUPRC, particularly excelling in blind cell type evaluations, with average score up to 0.95 for mESC and mHSC-L. This reflects the model's ability to generalize across species and cell types, maintaining high performance across various regulatory interactions. Competing methods, including PPCOR and PIDC, show moderate performance but struggle to adapt as effectively.

In cell-type specific networks, GRNFormer continues to excel with AUPRC values as high as 0.95 for mESC and 0.90–0.95 for mHSC datasets. It consistently identifies gene regulatory interactions, demonstrating robustness in datasets with clearer regulatory signals. Other methods, like PPCOR, show inconsistent performance.

In STRING networks, which involve more complex gene relationships, GRNFormer achieves AUPRC scores up to 0.96 on mESC and mHSC-L datasets, showcasing its adaptability without dataset-specific optimizations, while PPCOR and LEAP demonstrate limited variability and lower performance in these complex networks.

Focusing on blind evaluations of mESC and mHSC-L, GRNFormer is the most consistent and robust method, achieving top-tier AUPRC scores across all settings and TF scales. It excels in handling datasets with large or variable regulatory structures, further highlighting its effectiveness in real-world applications. One notable finding is that GRNFormer consistently achieved an AUROC above 90% for mouse and human cell-types on all the three regulatory networks outperforming all other methods such as PIDC, PPCOR, SINCERITIES, and LEAP. Refer Supplementary Figure S1, for summary of AUROC and AUPRC scores for individual experiments with 500 genes and with most varying TFs and 500 genes, respectively.

2.3.A case study on human embryonic stem cells

We performed a case study on Human embryonic stem cells (hESCs) that are pluripotent cells which self-renew and differentiate into all human cell types, making them crucial for developmental biology and cancer research. Their regulatory mechanisms overlap with those dysregulated in cancer, particularly in stem cells that drive tumor growth and resistance. GRNs are essential for understanding hESC biology, regulating pluripotency and differentiation. In our case study on hESC with 500genes and most varying TFs, GRNFormer predicted key TF-gene interactions revealed by hESC ChIP-seq data, identifying central TFs like POU5F1, SOX2, MYC, and NANOG. GRNFormer also reveals how these networks are disrupted in diseases like cancer highlighting stem-like traits in cancer cells.

3.3.1 Core Regulatory Network and Cancer Implications

We first considered the subnetworks of inferred gene regulatory network, which are verified by the ground truth network as well as validated the presence of biomarkers of hESC cell type present in our inferred network (Figure S2A). The GRN inferred by GRNFormer highlights pivotal TFs like SOX2, MYC, POU5F1 (OCT4), and NANOG, central to pluripotency and cancer[24]. SOX2 maintains stemness via interactions with EGR2 and CTCF, while MYC drives proliferation through links with TP53 and STAT3[25]. Key interactions include MYC with TP53 and SMAD3 [26] (cell cycle/apoptosis), SOX2 with ZEB1 [27](metastasis), and POU5F1 with AR (hormonal signaling). Interactions like NANOG –

NODAL[28] , MYC–HIF1A (metabolism)[28] and SOX2–CTCF[30] (chromatin regulation) connect stem cell biology with tumor progression. Refer to the Supplementary Figure S2(A) for the complete predicted network that is verified by the ground truth network.

3.3.2 Insights into hESC Pathways and Cancer

We performed a Gen e Ontology (GO) enrichment analysis on the predicted sub-GRN using goProfiler Enrichment analysis tool built in the Cytoscape software suite, which takes in genes in the predicted GRN and compares them with the comprehensive databases to determine the overrepresented GO terms or pathways based on gene counts,[31] to dive deep into the pathway analysis of the network. We considered all the unique pathways with high significance in Reactome, KEGG and WikiPathways databases. The enrichment map was created based on the intersecting or shared common genes. The enrichment map shown in Figure 7 (A) highlights the TGF-beta signaling pathway, particularly the SMAD2/SMAD3:SMAD4 heterotrimer, as central to regulating pluripotency and self-renewal. Wnt signaling, critical for maintaining pluripotency, also emerges as a driver of oncogenesis when dysregulated. Other pathways, such as cell cycle regulation and cellular senescence, underline key mechanisms in hESC proliferation, differentiation, and aging-related dysfunction. The GRNFormer-inferred GRN reveals intricate links between metabolic processes and signaling pathways, such as coupling energy metabolism with the circadian clock, essential for sustaining hESC energetic demands during differentiation. Shared pathways, including Hippo signaling, connect pluripotency with tumor suppression, reflecting overlaps between hESC biology and diseases like acute and chronic myeloid leukemia. Predicted interactions highlight the dynamic interplay between pathways like Wnt signaling, energy metabolism, and cell cycle regulation. The GRN inferred by GRNFormer uncovers multilayered relationships, providing a comprehensive framework for understanding stem cell biology and its intersection with oncogenesis[32,33,34].

3.3.3 GRNFormer predicts novel significant Gastrulation pathway

The predictive novelty of GRNFormer exhibits in its ability of identifying novel regulatory pathways absent in the considered ground-truth hESC-specific networks (Figure 7B)). The enrichment analysis on the inferred GRN subnetwork that do not exist in the corresponding ground truth network uncovers some new pathways supported by experimental evidence, notably the gastrulation pathway, which is essential for germ layer formation and embryonic development[35]. The enrichment map highlights pathways related to gastrulation, lineage differentiation, and germ layer specification, while the GRN reveals key gene-gene regulatory interactions involved in mesodermal and ectodermal differentiation (Supplementary Figure S2B).

Beyond development, gastrulation has implications in cancer biology. Its pathways, particularly epithelial-mesenchymal transition (EMT), are often reactivated in cancer, driving metastasis. This underscores the relevance of GRNFormer's predictions for both stem cell biology and cancer research, offering potential therapeutic insights. Overall, GRNFormer identifies novel, biologically meaningful pathways, making it a useful tool for understanding stem cell development and cancer progression.

3 Conclusion

GRNFormer is an advanced graph transformer model designed for efficient and accurate inference of gene regulatory networks from single-cell RNA-seq data. It demonstrates the exceptional generalization capability across species and cell types. It also substantially outperforms several existing methods, delivering biologically meaningful regulatory interactions, as validated by our case studies. GRNFormer offers a powerful tool to uncover key regulatory mechanisms. In the future, we aim to extend

this work to encompass a broader range of species and cell types, establishing a foundational model to drive future advancements in gene regulation studies and systems biology.

Acknowledgements

This work is supported by an NSF grant (#: CCF 2343612).

References

1. Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., & Watson, J. D. (1994). Molecular biology of the cell (Vol. 3, pp. 966-996). New York: Garland.
2. Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R. D., & Califano, A. (2006, March). ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. In *BMC bioinformatics* (Vol. 7, pp. 1-15). BioMed Central.
3. Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., Cottarel, G., ... & Gardner, T. S. (2007). Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biology*, *5*(1), e8.
4. Marbach, D., Costello, J. C., Küffner, R., Vega, N. M., Prill, R. J., Camacho, D. M., ... & Stolovitzky, G. (2012). Wisdom of crowds for robust gene network inference. *Nature methods*, *9*(8), 796-804.
5. Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A., & Murali, T. M. (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, *17*(2), 147-154.
6. Meyer, P. E., Kontos, K., Lafitte, F., & Bontempi, G. (2007). Information-theoretic inference of large transcriptional regulatory networks. *EURASIP journal on bioinformatics and systems biology*, *2007*, 1-9.
7. Hecker, M., Lambeck, S., Toepfer, S., Van Someren, E., & Guthke, R. (2009). Gene regulatory network inference: data integration in dynamic models—a review. *Biosystems*, *96*(1), 86-103.
8. Huynh-Thu, V. A., & Sanguinetti, G. (2019). Gene regulatory network inference: an introductory survey. *Gene regulatory networks: Methods and protocols*, 1-23.
9. Banf, M., & Rhee, S. Y. (2017). Computational inference of gene regulatory networks: approaches, limitations and opportunities. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, *1860*(1), 41-52.
10. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. https://doi.org/10.1038/nature14539
11. Fogelberg, C., & Palade, V. (2009). Machine learning and genetic regulatory networks: a review and a roadmap. *Foundations of Computational, Intelligence Volume 1: Learning and Approximation*, 3-34.
12. Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.
13. Xu, J., Zhang, A., Liu, F., & Zhang, X. (2023). STGRNS: an interpretable transformer-based method for inferring gene regulatory networks from single-cell transcriptomic data. *Bioinformatics*, *39*(4), btad165.
14. Kommu, S., Wang, Y., Wang, Y., & Wang, X. (2024). Gene Regulatory Network Inference from Pre-trained Single-Cell Transcriptomics Transformer with Joint Graph Learning. *arXiv preprint arXiv:2407.18181*.
15. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open*, *1*, 57-81.

16. Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
17. Johnson, K. A., & Krishnan, A. (2022). Robust normalization and transformation techniques for constructing gene coexpression networks from RNA-seq data. *Genome biology*, *23*, 1-26.
18. Xu, J., Li, Z., Du, B., Zhang, M., & Liu, J. (2020, July). Reluplex made more practical: Leaky ReLU. In *2020 IEEE Symposium on Computers and communications (ISCC)* (pp. 1-7). IEEE.
19. Mukherjee, K., Khare, A., & Verma, A. (2019). A simple dynamic learning rate tuning algorithm for automated training of DNNs. *arXiv preprint arXiv:1910.11605*.
20. Chan, T. E., Stumpf, M. P., & Babtie, A. C. (2017). Gene regulatory network inference from single-cell data using multivariate information measures. *Cell systems*, *5*(3), 251-267.
21. Kim, S. (2015). ppcor: an R package for a fast calculation to semi-partial correlation coefficients. *Communications for statistical applications and methods*, *22*(6), 665.
22. Papili Gao, N., Ud-Dean, S. M., Gandrillon, O., & Gunawan, R. (2018). SINCERITIES: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. *Bioinformatics*, *34*(2), 258-266.
23. Specht, A. T., & Li, J. (2017). LEAP: constructing gene co-expression networks for single-cell RNA-sequencing data using pseudotime ordering. *Bioinformatics*, *33*(5), 764-766.
24.
25. Yu, L., Yu, T. T., & Young, K. H. (2019). Cross-talk between Myc and p53 in B-cell lymphomas. *Chronic diseases and translational medicine*, *5*(03), 139-154.
26. Feng, X. H., Liang, Y. Y., Liang, M., Zhai, W., & Lin, X. (2002). Direct interaction of c-Myc with Smad2 and Smad3 to inhibit TGF-β-mediated induction of the CDK inhibitor p15Ink4B. *Molecular cell*, *9*(1), 133-143.
27. Ahmadi-Beni, R., Vand-Rajabpour, F., Ahmadifard, M., Daneshpazhooh, M., Noormohammadpour, P., Rahmati, J., ... & Tabrizi, M. (2020). Decreased Sox2 messenger RNA expression in basal cell carcinoma. *Indian Journal of Dermatology*, *65*(3), 178-182.
28. IV Santaliz-Ruiz, L. E., Xie, X., Old, M., Teknos, T. N., & Pan, Q. (2014). Emerging role of nanog in tumorigenesis and cancer stem cells. *International journal of cancer*, *135*(12), 2741-2748.
29. Gordan, J. D., Thompson, C. B., & Simon, M. C. (2007). HIF and c-Myc: sibling rivals for control of cancer cell metabolism and proliferation. *Cancer cell*, *12*(2), 108-113.
30. Chakraborty, S., Kopitchinski, N., Zuo, Z., Eraso, A., Awasthi, P., Chari, R., ... & Rocha, P. P. (2023). Enhancer–promoter interactions can bypass CTCF-mediated boundaries and contribute to phenotypic robustness. *Nature genetics*, *55*(2), 280-290.
31. Reimand, J., Isserlin, R., Voisin, V., Kucera, M., Tannus-Lopes, C., Rostamianfar, A., ... & Bader, G. D. (2019). Pathway enrichment analysis and visualization of omics data using g: Profiler, GSEA, Cytoscape and EnrichmentMap. *Nature protocols*, *14*(2), 482-517.
32. Mullen, A. C., & Wrana, J. L. (2017). TGF-β family signaling in embryonic and somatic stem-cell renewal and differentiation. *Cold Spring Harbor perspectives in biology*, *9*(7), a022186.
33. Gao, W., Li, R., Ye, M., Zhang, L., Zheng, J., Yang, Y., ... & Zhao, Q. (2022). The circadian clock has roles in mesenchymal stem cell fate decision. *Stem Cell Research & Therapy*, *13*(1), 200.
34. Sebio, A., & Lenz, H. J. (2015). Molecular pathways: hippo signaling, a critical tumor suppressor. *Clinical Cancer Research*, *21*(22), 5002-5007.
35. Stronati, E., Giraldez, S., Huang, L., Abraham, E., McGuire, G. R., Hsu, H. T., ... & Estarás, C. (2022). YAP1 regulates the self-organized fate patterning of hESC-derived gastruloids. *Stem cell reports*, *17*(2), 211-220.