scHyperLink: Revealing Cell-Type-Specific Gene Regulation with Hypergraph Neural Networks

Emre Kulkul, Tolga Çukur, *Senior Member, IEEE*, and Aykut Koç, *Senior Member, IEEE*

***Abstract*— Single-cell RNA sequencing (scRNA-seq) allows gene expression to be measured at single-cell resolution, offering new opportunities to investigate Gene Regulatory Networks (GRNs), which represent the regulatory interactions between transcription factors (TFs) and their target genes. Given their relational structure, GRNs are formulated as graphs, enabling gene interaction inference to be framed as a link prediction task among graph nodes (i.e., genes). Prior work adopts Graph Neural Networks (GNNs) to this end, employing their unique ability to model inter-node relationships. However, since GNNs are inherently limited to pair-wise node interactions, they struggle to capture the higher-order dependencies characteristic of GRNs. Gene expression is regulated through multi-way feedback loops involving multiple TFs and targets, and disregarding these higher-order dependencies can lower accuracy in gene interaction inference. To overcome this limitation, we introduce scHyperLink, a hypergraph-based framework for GRN reconstruction. scHyperLink models gene interactions using Hypergraph Neural Networks (HGNNs), where hyperedges allow the simultaneous representation of multi-gene regulatory relationships. scHyperLink integrates experimentally derived interaction graphs with dynamically learned hyperedges to better reflect the underlying regulatory structure. We demonstrate that scHyperLink achieves higher accuracy than state-of-the-art on cell-type-specific benchmark datasets, particularly in sparse regimes with few known interactions. Moreover, we validate the biological relevance of scHyperLink via interpretability analyses on inferred hypergraphs and showcase its scalability to tissue-level analyses. We share the analyzed datasets and source codes for reproducibility.[1]**

***Index Terms*— Hypergraph neural networks, graph neural networks, scRNA, Gene Regulatory Networks**

I. Introduction

Gene regulatory networks (GRNs) describe the architecture of regulatory interactions between target genes and their transcription factors (TFs), i.e., encoded proteins that modulate gene expression through processes such as transcriptional activation or repression [1]. As GRNs provide a foundation for understanding the molecular logic underlying cell identity, developmental programs, disease mechanisms, and therapeutic targets [2]–[4], systematic identification of regulatory relationships through GRN reconstruction is a central task in systems biology. Traditional approaches to GRN reconstruction have focused on aggregating experimentally validated TF–gene interactions into curated databases [5], [6], or inferring regulatory links from gene co-expression patterns derived from bulk RNA-seq data [7], [8]. However, curated databases often suffer from limited coverage across cell types and environmental conditions [9], [10], while correlations in bulk RNA-seq data may reflect confounding effects or shared upstream regulators rather than direct regulation [11].

The recent advent of single-cell RNA sequencing (scRNA-seq) has enabled fine-grained analysis of gene expression at single-cell resolution, offering improved insight into genomic regulation [12]–[15]. Unlike bulk RNA-seq, which averages across heterogeneous cell populations and obscures cell-specific signals, scRNA-seq captures expression heterogeneity across diverse cell types [16]. As in other areas of bioinformatics, where machine learning have been applied to analyze diverse molecular data such as circRNA, lncRNA, and DNA sequences [17]–[20], the shift from bulk RNA-seq to scRNA-seq has attracted substantial attention for tasks including imputation, clustering, and cell-type annotation [21]–[23], owing to its ability to capture cell-specific expression patterns at high resolution. This development has renewed interest in GRN reconstruction, where single-cell–level variability enables more context-specific inference of transcriptional regulation [24]. Various learning-based methods have been proposed to infer GRNs from scRNA-seq data, including formulations based on ordinary differential equations [25], [26] that can offer suboptimal capture of nonlinear dynamics, and convolutional neural networks (CNNs) or transformers applied to co-expression matrices [27], [28] that struggle to scale efficiently to large single-cell datasets due to memory-intensive computation of co-expression matrices.

Graph neural networks (GNNs) have emerged as a powerful alternative for GRN reconstruction, given their native ability to model dependencies in structured data [29], [30], [31]. GNNs have been applied to a range of transcriptomics tasks, including cell-type classification [32], [33], [34], cell-cell communication prediction [35], [36], and imputation [37]. In the context of GRNs, a graph formulation represents genes as nodes and regulatory interactions as edges, enabling graph-based models to naturally identify patterns of genomic regulation. For instance, GNNLink [38] uses graph convolutional networks, scSGL [39] employs kernelized graphs, GENELink [40] uses graph attention networks, while GRACE [41] further integrates structural causal modeling. Despite the natural formulation of GRNs as graphs, existing GNN-based approaches fail to capture the higher-order dependencies characteristic of gene regulation, as they are inherently limited to modeling pair-wise relationships between genes. A well-known example

E. Kulkul, T. Çukur, and A. Koç are with the Dept. of Electrical-Electronics Engineering, and National Magnetic Resonance Research Center (UMRAM), Bilkent University, 06800 Ankara, Türkiye. Corresponding Author: Aykut Koç (e-mail: aykut.koc@bilkent.edu.tr).

This work was supported by TUBITAK 1001 Grant (124E179). The work of T. Çukur is supported by the BAGEP 2017 and GEBIP 2015 Awards. The work of A. Koç is supported by the BAGEP 2023 Award.

[1]Code and data are available at https://github.com/koc-lab/scHyperLink.

of such higher-order structure is the interferon-beta (IFN-β) enhanceosome—a multiprotein complex in which TFs such as ATF-2/c-Jun, IRF-3/IRF-7, and NF-κB assemble into a functional unit to cooperatively activate the IFN-β gene [42], [43]. Modeling such combinatorial regulation using only pair-wise interactions neglects the collective logic underlying transcriptional control.

A promising approach to overcome this limitation is hypergraphs that generalize conventional graphs by allowing hyperedges to connect any number of nodes simultaneously, enabling direct representation of higher-order relationships. By extending message passing algorithms, hypergraph neural networks (HGNNs) then allow multi-way dependencies to be learned directly from data [44]. HGNNs have shown promise in other applications such as spatial transcriptomics [45], cancer gene identification [46], drug interaction prediction [47], [48], and therapeutic target identification in traditional Chinese medicine [49]. Recent work such as scHyper [50] further explores HGNNs for cell-cell communication by constructing hypergraphs via modeling triplets of sending-cell expression, receiving-cell expression, and known ligand–receptor interactions. Note, however, that scHyper is designed to decode cell–cell signaling rather than intra-cellular regulation, differing fundamentally from GRN-focused methods. Hence, the use of HGNNs in GRN reconstruction remains limited. One exception within the GRN domain is MHHGRN [51] that employs static hypergraphs to analyze gene co-expression patterns. Notably, MHHGRN initiates hypergraphs using correlation-based adjacency measures, which may constrain its expressive potential for higher-order regulatory interactions. Thus, the potential of HGNNs for GRN reconstruction remains underexplored, particularly in a less constrained formulation that permits dynamic hypergraph structures.

Here we introduce *scHyperLink*, an HGNN framework for GRN reconstruction that addresses the key limitations of conventional GNNs in capturing higher-order regulatory dependencies. scHyperLink integrates a prior-informed hypergraph—incorporating experimentally validated interactions—with a learnable, expression-driven dynamic hypergraph to jointly encode both known and context-specific regulatory structures. Unlike HGNN-based approaches designed for non-GRN tasks, scHyperLink derives its hypergraphs directly from gene-expression profiles and experimentally validated TF–target relationships, allowing it to capture higher-order regulatory dependencies characteristic of GRNs. By propagating signals over this hybrid hypergraph, the model learns latent gene representations optimized for downstream interaction prediction. Through experiments on 14 cell-type-specific datasets, we show that scHyperLink outperforms state-of-the-art methods in GRN reconstruction. We further demonstrate the biological relevance of the learned hypergraphs through interpretability analysis [52], and validate the scalability of our approach via tissue-level regulatory inference. Our main contributions are:

- We propose scHyperLink to capture the higher-order dynamics in GRNs by learning gene representations.
- scHyperLink is able to leverage two hypergraph constructions: a static hypergraph containing prior interaction information, and a dynamic hypergraph constructing hyperedges adaptively from gene expression data.
- We demonstrate that scHyperLink outperforms state-of-the-art methods on common benchmark datasets.
- We show that scHyperLink captures biologically plausible regulatory information via the dynamic hypergraphs and showcase its scalability to tissue-level analyses.
- We show that scHyperLink enables cross-cell-type generalization, transferring regulatory information learned from well-characterized cell types to improve inference in sparsely annotated ones.

II. Related Works

A. Learning-based Approaches

Learning-based methods have gained momentum for GRN inference due to their ability to capture complex patterns from data. One of the earliest, GENIE3 [7], employs ensemble tree-based models to predict gene expression from candidate regulatory inputs. However, training a separate model for each gene renders it computationally expensive on large-scale datasets. GRNBoost2 [53] builds on this with gradient boosting machines (GBMs) to improve scalability, yet it still analyzes genes independently, limiting its ability to capture dynamic dependencies and global structure. To improve sensitivity to gene expression dynamics, SCODE [25] introduces a linear ODE model over pseudo-time trajectories, while [26] extends this framework to complex-valued ODEs. Although efficient, ODE-based approaches depend heavily on accurate pseudo-time estimation and cannot model the nonlinear dynamics or feedback loops intrinsic to GRNs.

More recent methods aim to incorporate uncertainty and richer data representations. DeepSEM [54] uses a variational autoencoder (VAE) to learn low-dimensional gene embeddings while accounting for scRNA-seq noise, and PMF-GRN [55] applies probabilistic matrix factorization with variational inference to model uncertainty. CNNC [27] transforms co-expression patterns into image histograms for gene-pair classification using CNNs, and DeepDRIM [28] further incorporates neighboring gene expression to reduce false positives. While CNN-based models achieve competitive performance, their locality bias restricts sensitivity to short-range dependencies. STGRNS [56] instead employs transformers to model long-range dependencies in gene expression. Still, both CNN and transformer-based approaches often incur substantial computational costs when constructing gene co-expression matrices, limiting scalability on large scRNA-seq datasets.

B. Graph-based Approaches

With the rise of geometric deep learning, numerous efforts have sought to leverage the graph-like structure inherent to biological regulatory systems and their abstractions as GRNs. GNE [57] employs multi-layer perceptrons (MLPs) and neighborhood information to generate gene embeddings. GNNLink [38] applies a graph convolutional network (GCN) for message passing between genes, while GENELink [40] adopts a graph attention network (GAT) to assign adaptive

weights to gene–gene interactions. GATCL [58] enhances the attention mechanism with convolutional kernels to detect local patterns, and scMGATGRN [59] introduces a multi-view attention strategy to capture up to second-order dependencies. Beyond attention-based models, GMFGRN [60] integrates matrix factorization with GCNs to model gene–cell interactions, GRACE [41] combines graph learning with structural causal modeling (SCM), and HGTCGRN [61] utilizes hierarchical graph transformers to capture local and global dependencies. GCLink [62] introduces a graph contrastive framework that generates dual views of the gene interaction graph and applies GATs to obtain more expressive gene embeddings. LineGRN [63] employs line GNNs to overcome the limitations of low-degree-node-dominated topologies and learn representations for gene–gene interactions rather than individual genes. These pioneering approaches have enabled a structured framework for modeling second-order regulatory interactions. However, they remain intrinsically limited in capturing multi-way dependencies that extend beyond pair-wise links, which are fundamental to complex gene regulatory logic.

In a recent study, MHHGRN [51] attempts to mitigate these issues by employing static hypergraphs constructed from gene expression. Yet, MHHGRN constructs hypergraphs based on various correlation methods, which are pair-wise by nature. Such construction may limit its ability to capture more complex, higher-order regulatory dependencies. Meanwhile, HyperG-VAE [64] introduces a Bayesian hypergraph generative model for scRNA-seq data, employing gene and cell encoders to simultaneously capture cellular heterogeneity and co-regulated gene modules within a shared embedding space. Nevertheless, as an unsupervised generative model, HyperG-VAE does not incorporate biological priors or link-level supervision and does not directly employ HGNN architectures for relational inference. As such, the broader capabilities of HGNNs for GRN reconstruction remain largely untapped, especially in formulations that allow for dynamic hypergraph representations. Hypergraph-based GRN reconstruction also remains underexplored in terms of its biological plausibility.

Different from existing methods, scHyperLink combines static hypergraphs built from prior biological knowledge with dynamic hypergraphs learned from expression data. This hybrid formulation enables higher-order message passing between TFs and target genes, yielding refined gene representations for more accurate and interpretable regulatory inference.

III. Preliminaries

A hypergraph is defined as $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}$ is the set of nodes and $\mathcal{E} = \{e_1, e_2, \ldots, e_{|\mathcal{E}|}\}$ is the set of hyperedges. In general, a hyperedge e_k is a non-empty subset of $\mathcal{V}$ that admits self-loops. Here we strictly adhere to each hyperedge e_k being a subset of $\mathcal{V}$ with at least two nodes. The hypergraph structure can be represented by an incidence matrix $\mathbf{H} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{E}|}$,

$$H_{i,j} = \begin{cases} 1 & \text{if } v_i \in e_j \\ 0 & \text{otherwise} \end{cases}$$

Given a node v_i, the incident hyperedges, i.e., the set of hyperedges that contain v_i, are denoted as

$$\mathcal{N}_{\mathcal{E}}(v_i) = \{e_k \in \mathcal{E} \mid v_i \in e_k\}.$$

Each node v_i has a node feature vector $\mathbf{x}_i \in \mathbb{R}^F$ (F: node feature dimension). The input node feature matrix, $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times F}$, stacks node feature vectors into rows, $\mathbf{x}_i^\top$ for $i \in \{1, 2, ..., |\mathcal{V}|\}$. Similarly, each hyperedge e_j has a hyperedge feature vector $\mathbf{z}_j \in \mathbb{R}^{F'}$ (F': hyperedge feature dimension). The hyperedge feature matrix $\mathbf{Z} \in \mathbb{R}^{|\mathcal{E}|\times F'}$ stacks these vectors into rows, $\mathbf{z}_j^\top$ for $j \in \{1, 2, ..., |\mathcal{E}|\}$.

HGNNs extend traditional GNNs by operating on hypergraphs, where hyperedges can connect more than two nodes, unlike edges in a graph. When each hyperedge connects exactly two nodes, the hypergraph reduces to a standard graph. Hyperedges allow HGNNs to capture higher-order relationships and enable more expressive message passing between nodes, surpassing the limitations of pair-wise interactions in standard GNNs. A typical HGNN layer consists of two stages: node-to-hyperedge aggregation and hyperedge-to-node aggregation. Let $\mathbf{H} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{E}|}$ be the incidence matrix of the hypergraph. The formulation of one hypergraph convolutional layer with unweighted hyperedges is proposed in [44], which updates the node features as follows:

$$\mathbf{X}^{(l+1)} = \sigma\left(\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}\mathbf{X}^{(l)}\mathbf{\Theta}\right), \quad (1)$$

where $\mathbf{\Theta}$ is a learnable weight matrix, $\sigma(\cdot)$ is a nonlinear activation function, and $\mathbf{D}_v \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ and $\mathbf{D}_e \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ are diagonal matrices representing the degrees of nodes and hyperedges, respectively:

$$\mathbf{D}_v(i,i) = \sum_{j=1}^{|\mathcal{E}|} H_{i,j}, \quad \mathbf{D}_e(j,j) = \sum_{i=1}^{|\mathcal{V}|} H_{i,j}.$$

When Eq. 1 is considered without the degree matrices for brevity, the $\mathbf{H}^\top\mathbf{X}$ operation performs node-to-hyperedge feature aggregation, and multiplying with another $\mathbf{H}$ distributes the aggregated features back to the nodes based on $\mathcal{N}_{\mathcal{E}}(v_i)$ for each node v_i. This formulation allows node features to be updated based on their connectivity through hyperedges, capturing higher-order structural information that extends beyond simple pair-wise relationships between nodes.

IV. scHyperLink

A. Problem Formulation

We pose gene interaction prediction as a link prediction task on a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices representing individual genes, and $\mathcal{E}$ is the set of hyperedges, each containing at least two genes. Motivated by biological gene regulation, we model each hyperedge to reflect a TF and its neighborhood of target genes. Hence, hypergraph construction yields an incidence matrix $\mathbf{H} \in \mathbb{R}^{N\times M}$ (N: number of unique genes; M: number of TF-encoding genes). As input data, we utilize a scRNA-seq gene expression matrix $\mathbf{X} \in \mathbb{R}^{N\times C}$ (C: number of cells), and a binary gene interaction matrix $\mathbf{A}_i \in \mathbb{R}^{N\times N}$ containing experimentally validated TF-gene interactions.

To construct a supervised learning setup, we reserve some of the known interactions in $\mathbf{A}_i \in \mathbb{R}^{N\times N}$ as the test set as

positive pairs. This effectively splits the interaction matrix into a training portion $\mathbf{A}_{tr} \in \mathbb{R}^{N\times N}$ and test/validation portion $\mathbf{A}_{test} \in \mathbb{R}^{N\times N}$ such that $\mathbf{A}_i = \mathbf{A}_{tr} + \mathbf{A}_{test}$. For the splits, we first reserve 2/3 of the positive interactions randomly for the training set, and then allocate 1/10 of the training set as the validation set. The remaining 1/3 portion of the positive interactions forms the test set. Since the positive interactions are sparse, we utilize a uniform negative sampling strategy to randomly select non-interacting TF-gene pairs as negative samples for each split. To preprocess the scRNA-seq expression data, we retain only the 500 (and 1,000) most significantly varying target genes, and all TFs, based on Bonferroni-corrected p-values (thresholded at $p < 0.01$). Our data preparation and preprocessing pipeline closely follows the protocols of GENELink [40] and BEELINE [65], where batch effects were not considered since each dataset originates from a single study and is processed independently under consistent experimental conditions.

It is important to note that gene interaction matrices are not necessarily undirected or do not contain self-loops, as they contain TF-target relationships that carry directions. Hence, we utilize two different views of the interaction matrix in our framework: a symmetrized, self-loop inserted version of $\mathbf{A}_{tr} \in \mathbb{R}^{N\times N}$ denoted throughout the paper as $\mathbf{A} \in \mathbb{R}^{N\times N}$ for practical purposes, and an undirected version denoted with $\mathbf{A}_{tf} \in \mathbb{R}^{M\times N}$ that retains the M rows in $\mathbf{A}_{tr}$ belonging to TF-encoding genes, without any enforced self-loops.

With the setup described above, scHyperLink learns a mapping f_θ that takes in a scRNA-seq expression matrix $\mathbf{X} \in \mathbb{R}^{N\times C}$ and an interaction adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ to output latent gene representations $\mathbf{G} \in \mathbb{R}^{N\times D}$ (D: gene embedding dimension):

$$\mathbf{G} = f_\theta(\mathbf{X}, \mathbf{A}), \tag{2}$$

where the i-th gene had an embedding $\mathbf{g}_i \in \mathbb{R}^D$, and $\mathbf{G} \in \mathbb{R}^{N\times D}$ stacks embeddings row-wise as $\mathbf{g}_i^\top$ for $i \in \{1, 2, ..., N\}$. In this framework, we aim to learn gene embeddings such that the similarity scores between TFs and target genes reflect the likelihood of regulatory links. To this end, we view link prediction as a binary classification task, and train scHyperLink using a binary cross-entropy loss along with regularization terms. During inference, scHyperLink estimates the interaction probabilities for previously unseen TF–target pairs by computing similarity scores in the learned embedding space and uses these probabilities to classify interactions in gene pairs. Formally, a binary classifier $h(\cdot)$ receives gene embeddings $\mathbf{g}_i$ and $\mathbf{g}_j$ for each gene pair (i, j), and decodes them into interaction scores $h : (\mathbf{g}_i, \mathbf{g}_j) \to \{0, 1\}$. Hence, the overall aim of scHyperLink is to utilize an interaction adjacency matrix and an expression matrix to derive latent gene representations and decode interactions between pairs of gene embeddings as detailed in Sec. IV-B.

B. Overall Architecture

In view of the link prediction formulation, scHyperLink has three modules that are coupled in objective: a PHI (Prior Hypergraph Induction) module, a DHI (Dynamic Hypergraph Induction) module, and an HGNN module. Fig. 1 depicts the overall architecture and individual modules of scHyperLink that derive latent gene representations from an initial adjacency matrix and an expression matrix.

The PHI module utilizes prior TF-gene interaction knowledge to form a static hypergraph with inductive bias denoted as $\mathbf{H}_0$. Since experimental interaction data are often noisy, sparse, and imbalanced across TFs, we adopt the DHI module to learn a dynamic hypergraph $\mathbf{H}_L \in \mathbb{R}^{N\times M}$ that complements the neighborhood information carried by the static hypergraph in a data-driven manner, and learns refined hyperedge features. Following the PHI and DHI modules, both constructing a binary hypergraph incidence matrix, we obtain a final hypergraph as $\mathbf{H}_F = \mathbf{H}_0 \vee \mathbf{H}_L$, where $\vee$ is the element-wise logical OR operation. The HGNN module utilizes the final hypergraph and the learned hyperedge features to propagate representations between neighboring genes, while mapping expression profiles of genes to low-dimensional embeddings. Individual modules are mathematically described in Secs. IV-C, IV-D and IV-E.

When the framework is considered sequentially, the PHI module uses the TF-filtered version of $\mathbf{A}$, denoted by $\mathbf{A}_{tf}$, to obtain the prior hypergraph $\mathbf{H}_0$. Concurrently, the input data $(\mathbf{X}, \mathbf{A})$ is processed by the DHI module to produce a dynamic hypergraph $\mathbf{H}_L$. This is combined with the prior hypergraph $\mathbf{H}_0$ to form the final hypergraph $\mathbf{H}_F$. Alongside, the DHI module also generates refined hyperedge features $\hat{\mathbf{X}}_\mathcal{E}$, which, together with the expression matrix $\mathbf{X}$ and the final hypergraph $\mathbf{H}_F$, are fed into the HGNN module. The HGNN module employs a dual-stream architecture with cross-attention between streams to produce low-dimensional gene representations. By utilizing two separate streams, the distinct functional roles of TF-encoding genes and target genes can be captured more effectively. Therefore, the whole mapping function f_θ in Eq. 2 can be decomposed as

$$\mathbf{H}_0 = f_{PHI}(\mathbf{A}) = \mathbf{A}_{tf}^\top, \tag{3}$$

$$\begin{bmatrix}\hat{\mathbf{X}}_\mathcal{E} & \mathbf{H}_L\end{bmatrix} = f_{DHI}(\mathbf{X}, \mathbf{A}), \tag{4}$$

$$\mathbf{H}_F = \mathbf{H}_0 \vee \mathbf{H}_L, \tag{5}$$

$$\mathbf{G} = f_{HGNN}(\mathbf{X}, \hat{\mathbf{X}}_\mathcal{E}, \mathbf{H}_F), \tag{6}$$

where the DHI module output in Eq. 4 comprises the hyperedge feature matrix and the learned hypergraph matrix.

The gene embedding matrix $\mathbf{G} \in \mathbb{R}^{N\times D}$ is then utilized to obtain the TF-gene interaction scores via the dot product between individual genes, whether it be TF-encoding or target genes. For gene i and gene j with embeddings $\mathbf{g}_i$ and $\mathbf{g}_j$, the normalized interaction score $p_{ij} \in [0, 1]$ is computed with

$$p_{ij} = \text{Sigmoid}\left(\mathbf{g}_i^\top \mathbf{g}_j\right). \tag{7}$$

Given the true TF-gene interactions $y_{ij} \in \{0, 1\}$ and the computed probabilities $p_{ij} \in [0, 1]$, the primary training objective is to minimize the binary cross-entropy (BCE) loss:

$$\mathcal{L}_{\text{BCE}} = -\sum_{(i,j)\in\mathcal{S}} \left[y_{ij} \log p_{ij} + (1-y_{ij}) \log(1-p_{ij})\right], \tag{8}$$

where $\mathcal{S}$ is the set of positive and negative TF-gene pairs chosen from $\mathbf{A}_{tr}$. To improve regularization and stability, we additionally incorporate two auxiliary losses: the Kullback–Leibler (KL) divergence between the learned variational distribution of the hyperedges and a unit Gaussian prior, which

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2025.3650661

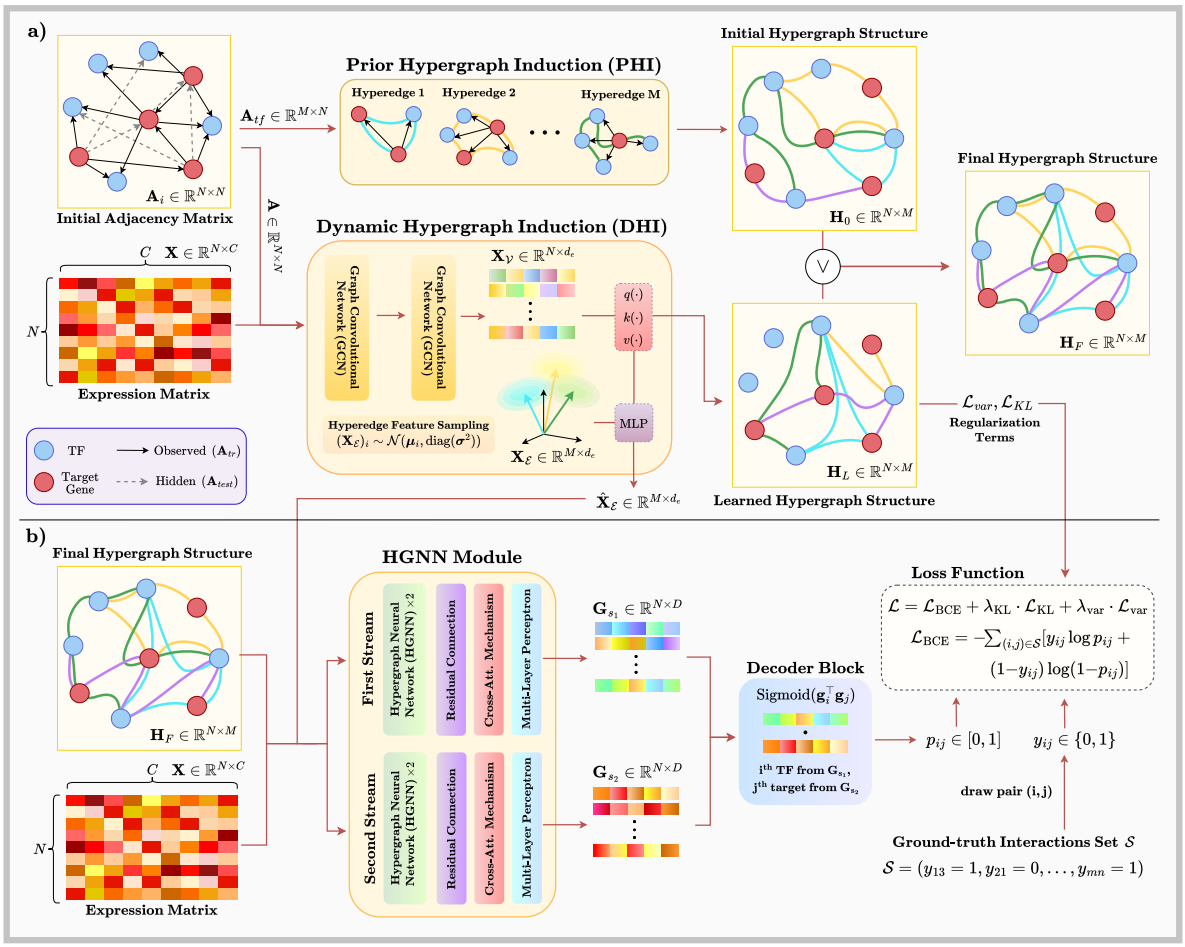

Fig. 1: The overall architecture of scHyperLink is illustrated. (a) The process begins with the PHI and DHI modules, which collectively form a hypergraph structure where each TF has its own hyperedge. Note that although each hyperedge is defined by a single TF (the regulator), other TFs may appear within the hyperedge as target genes. (b) The output hypergraph structure and the refined hyperedge features are fed along with the expression matrix to the HGNN module, which learns low-dimensional gene representations. These gene representations are then decoded to obtain interaction predictions.

encourages a structured latent space for hyperedge sampling, and a column-wise variance loss on the hypergraph $\mathbf{H}_L$, which prevents over-saturation or collapse of the hyperedge assignments by discouraging imbalanced distributions of gene memberships. The variance loss is constructed with

$$\mathcal{L}_{\text{var}} = \frac{1}{M}\sum_{j=1}^{M}\left\|\mathbf{H}_L^{(:,j)} - \mu_j \mathbf{1}\right\|_2^2, \tag{9}$$

where $\mathbf{1} \in \mathbb{R}^N$ is a vector of ones, $\mathbf{H}_L^{(:,j)} \in \mathbb{R}^N$ is the j-th column of $\mathbf{H}_L$, and μ_j is the row-wise mean of $\mathbf{H}_L^{(:,j)}$:

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N}\mathbf{H}_L^{(i,j)}. \tag{10}$$

$\mathcal{L}_{\text{var}}$ loss aims to penalize low-variance columns, thereby discouraging the degenerate case where a small number of hyperedges dominate the membership assignments while others remain sparsely connected. The final regularized loss is:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{var}}\mathcal{L}_{\text{var}}, \tag{11}$$

where λ_{KL} and λ_{var} are hyperparameters controlling the contribution of the KL and variance regularization terms.

During inference, the decoder uses a fixed threshold of 0.5 to classify whether a gene pair interacts. Interaction scores above this threshold predict the presence of a regulatory link between the gene pair, while scores below indicate its absence. Formally, the classification rule for the gene pair (i, j) is:

$$\mathbb{I}\{\text{Sigmoid}\left(\mathbf{g}_i^\top \mathbf{g}_j\right) \geq 0.5\}, \tag{12}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function that is equal to one if the condition is satisfied, and zero otherwise.

C. PHI Module: Initializing a Static TF-Target Hypergraph

The PHI module is designed to ensure that the hypergraph captures prior knowledge from the ground-truth TF-gene pairs as an inductive bias. We inject this prior structure into an initial static hypergraph $\mathbf{H}_0 \in \mathbb{R}^{N\times|\mathcal{E}|}$ with $\mathbf{H}_0 = \mathbf{A}_{tf}^\top$, where $\mathbf{A}_{tf} \in \mathbb{R}^{M\times N}$ is the TF-filtered version of the training adjacency matrix introduced in Sec. IV-A. This injection yields $|\mathcal{E}| = M$ number of hyperedges, where each hyperedge represents a single TF and its gene neighborhood based on the interactions present in the ground-truth data.

D. DHI Module: Learning a Dynamic TF-Target Hypergraph

The DHI module is designed to learn a dynamic hypergraph $\mathbf{H}_L \in \mathbb{R}^{N\times M}$ that complements the prior structure $\mathbf{H}_0$, where each TF is initially associated with a hyperedge based on known interactions. The DHI module adaptively updates the static hypergraph structure by introducing new gene memberships into the existing hyperedges based on the relationship

between hyperedge-gene features, which necessitates hyperedge features denoted as $\mathbf{X}_{\mathcal{E}} \in \mathbb{R}^{M \times d_e}$ to be present. Instead of directly encoding these features with expression vectors of TFs, we treat hyperedge features as latent variables sampled from a trainable distribution, which can capture task-specific representations. Since this module aims to assign new gene memberships to hyperedges, the hyperedge distributions can be viewed as centroids in high-dimensional space, and genes as queries that search for which clusters to join.

More formally, we aim to learn a distribution $\mathcal{P}(\mathbf{X}_{\mathcal{E}} \mid \mathbf{A}, \mathbf{X})$ over hyperedge features conditioned on the gene expression matrix and the interaction matrix pair. Specifically, we model each hyperedge feature $(\mathbf{X}_{\mathcal{E}})_i$ as being drawn from a Gaussian distribution with a learnable mean $\boldsymbol{\mu}_i$ and a shared learnable variance vector $\boldsymbol{\sigma}$, i.e.,

$$(\mathbf{X}_{\mathcal{E}})_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}^2)), \tag{13}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{M \times d_e}$ contains one mean vector per hyperedge, and $\boldsymbol{\sigma} \in \mathbb{R}^{1 \times d_e}$ is shared across all M hyperedges. This formulation allows each hyperedge to learn a distinct direction in space with the mean vector, and the shared variance reduces computational complexity. Since the sampling process is discrete and cannot produce gradients, we use the reparameterization method [66] to make $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ trainable, i.e., we sample a normally distributed matrix $\mathbf{Q} \in \mathbb{R}^{M \times d_e}$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and obtain the i^{th} hyperedge feature vector with

$$(\mathbf{X}_{\mathcal{E}})_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma} \odot \mathbf{Q}_i, \tag{14}$$

where $\odot$ is the Hadamard product. After sampling the hyperedge features, we process the expression matrix $\mathbf{X} \in \mathbb{R}^{N \times C}$ together with the normalized interaction matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ through a two-layered GCN block, projecting the expression vectors onto the hyperedge feature space with

$$\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}, \tag{15}$$

$$\tilde{\mathbf{X}} = \sigma\left(\mathrm{LN}(\tilde{\mathbf{A}}\mathbf{X}\mathbf{W}_1)\right), \tag{16}$$

$$\mathbf{X}_{\mathcal{V}} = \sigma\left(\mathrm{LN}(\tilde{\mathbf{A}}\tilde{\mathbf{X}}\mathbf{W}_2)\right), \tag{17}$$

where $\mathbf{D} \in \mathbb{R}^{N \times N}$ is the adjacency degree matrix of $\mathbf{A}$, $\mathbf{X}_{\mathcal{V}} \in \mathbb{R}^{N \times d_e}$ are the new node features, $\mathbf{W}_1 \in \mathbb{R}^{C \times d_e}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_e \times d_e}$ are learnable weight matrices, $\mathrm{LN}(\cdot)$ is the LayerNorm function, and $\sigma(\cdot)$ is the nonlinear activation function, which is instantiated as the LeakyReLU function throughout this manuscript. With this projection, the node features benefit from the pair-wise relationships observed in the interaction matrix $\mathbf{A}$ with propagations.

To refine the hyperedge features with the obtained gene representations, we draw inspiration from the attention mechanism in transformers [67] by viewing the hyperedges as queries that search for nearby nodes, and genes as keys and values that will be aggregated into the hyperedges. We use three mapping functions $q(\cdot)$, $k(\cdot)$, and $v(\cdot)$, each comprising a learnable weight matrix, to obtain

$$\alpha_{\mathcal{E}_i \to \mathcal{V}_j} = \frac{e^{\left(q(\mathbf{X}_{\mathcal{E}})_i \left[k(\mathbf{X}_{\mathcal{V}})_j^\top\right]\right)}}{\sum_l e^{\left(q(\mathbf{X}_{\mathcal{E}})_i \left[k(\mathbf{X}_{\mathcal{V}})_l^\top\right]\right)}}, \tag{18}$$

where $\alpha_{\mathcal{E}_i \to \mathcal{V}_j}$ is the hyperedge-to-gene attention coefficient. In matrix form, we obtain $\boldsymbol{\alpha}_{\mathcal{E} \to \mathcal{V}} \in \mathbb{R}^{M \times N}$ that comprises all such coefficients. To let each hyperedge aggregate distinctively without propagating too much noise, we keep only the highest k_1 coefficients per hyperedge with $\mathrm{topk}(\boldsymbol{\alpha}_{\mathcal{E} \to \mathcal{V}}, k_1)$, where the $\mathrm{topk}(\mathbf{A}, L)$ operator sets the largest L entries in each row of a given matrix $\mathbf{A}$ to one, and sets all other entries to zero. The filtered attention matrix then aggregates information from the remaining nodes into the hyperedges with the value matrix $v(\mathbf{X}_{\mathcal{V}})$. The sampled features and the aggregated features are then concatenated and processed through a multi-layer perceptron (MLP) as follows:

$$\hat{\mathbf{X}}_{\mathcal{E}} = \mathrm{MLP}\left(\mathbf{X}_{\mathcal{E}} \parallel \mathrm{topk}(\boldsymbol{\alpha}_{\mathcal{E} \to \mathcal{V}}, k_1) v(\mathbf{X}_{\mathcal{V}})\right), \tag{19}$$

where $\parallel$ is the concatenation operator. Now, each refined hyperedge feature in $\hat{\mathbf{X}}_{\mathcal{E}} \in \mathbb{R}^{M \times d_e}$ carries information from both its distribution and the k_1 genes closest to the distribution.

With refined hyperedge features present, we now aim to assign genes to hyperedges based on some distance criterion. Viewing the hyperedges as centroids or keys, and genes as queries that search for hyperedges to join into, we utilize the same latent spaces defined with $q(\cdot)$, $k(\cdot)$, and $v(\cdot)$ to obtain similarity scores with

$$\alpha_{\mathcal{V}_i \to \mathcal{E}_j} = \frac{e^{\left(q(\mathbf{X}_{\mathcal{V}})_i \left[k(\hat{\mathbf{X}}_{\mathcal{E}})_j^\top\right]\right)}}{\sum_l e^{\left(q(\mathbf{X}_{\mathcal{V}})_i \left[k(\hat{\mathbf{X}}_{\mathcal{E}})_l^\top\right]\right)}}, \tag{20}$$

where $\alpha_{\mathcal{V}_i \to \mathcal{E}_j}$ is the gene-to-hyperedge attention coefficient. In matrix form, we obtain $\boldsymbol{\alpha}_{\mathcal{V} \to \mathcal{E}} \in \mathbb{R}^{N \times M}$, where each row corresponds to a probability distribution over all hyperedges for a given gene. To prevent noise propagation and over-smoothing, we set the k_2 highest coefficients in each row to one, while setting all other entries to zero. This binarization uses the raw attention scores for ranking and yields a clean set membership for stable hypergraph message passing. With this operation, each hyperedge can receive new gene members, effectively creating a binary-valued dynamic hypergraph with

$$\mathbf{H}_L = \mathrm{topk}(\boldsymbol{\alpha}_{\mathcal{V} \to \mathcal{E}}, k_2), \tag{21}$$

which is then added to the static hypergraph $\mathbf{H}_0$ to yield the final hypergraph $\mathbf{H}_F \in \mathbb{R}^{N \times M}$. To clarify and visualize the dynamic hypergraph construction process, we provide an illustrative toy example in Supp. Fig. 2.

E. HGNN Module: Creating Gene Representations

After a single forward pass through the DHI module, the resulting pair $(\mathbf{H}_F, \hat{\mathbf{X}}_{\mathcal{E}})$ is passed to the HGNN module, along with the gene expression matrix $\mathbf{X}$. As outlined in Sec. IV-B, the HGNN module adopts a dual-stream design to capture the differing functional roles and expression profiles of TFs and target genes, where the streams are architecturally identical but with independent parameter initializations to learn complementary representations. Each stream is structurally symmetric and begins with an arbitrary number of hypergraph neural network layers, propagating features across gene representations using the modified hypergraph structure.

While our HGNNs follow the formulation described in Sec. III, we modify the standard forward pass computation in Eq. 1 following the design introduced in TDHNN [68]. In conventional HGNNs, given an incidence matrix $\mathbf{H} \in \mathbb{R}^{N \times M}$ and node features $\mathbf{V} \in \mathbb{R}^{N \times F}$, the product $\mathbf{H}^\top\mathbf{V}$ aggregates node information into hyperedge representations, which are then projected back to the nodes via multiplication with $\mathbf{H}$.

TDHNN proposes bypassing this node-to-hyperedge aggregation when useful hyperedge features are already available, directly propagating information from hyperedges to nodes instead. Since our DHI module already produces refined hyperedge features $\hat{\mathbf{X}}_{\mathcal{E}} \in \mathbb{R}^{M \times d_e}$, the aggregation step $\mathbf{H}\mathbf{H}^\top\mathbf{V}$ becomes redundant. Instead, we directly project the available hyperedge features back to the nodes and combine them with a node-wise transformation to update $\mathbf{V}$. The TDHNN-inspired forward pass is formulated as:

$$\tilde{\mathbf{H}}_F = \mathbf{D}_v^{-1/2}\mathbf{H}_F\mathbf{D}_e^{-1/2}, \tag{22}$$

$$\mathbf{X}_{\mathcal{V},s_i}^{(l+1)} = \sigma\left(\mathbf{X}_{\mathcal{V},s_i}^{(l)}\mathbf{W}_{\mathcal{V},s_i}^{(l)} + \tilde{\mathbf{H}}_F\hat{\mathbf{X}}_{\mathcal{E}}\mathbf{W}_{\mathcal{E},s_i}^{(l)}\right), \tag{23}$$

where $\mathbf{D}_v \in \mathbb{R}^{N\times N}$ is the node degree matrix, $\mathbf{D}_e \in \mathbb{R}^{M\times M}$ is the hyperedge degree matrix, and $\mathbf{W}_{\mathcal{V},s_i}^{(l)}$ and $\mathbf{W}_{\mathcal{E},s_i}^{(l)}$ are learnable weight matrices for the l-th HGNN layer, and s_i subscript denotes the i-th stream with $i \in \{1,2\}$. The term $\mathbf{X}_{\mathcal{V},s_i}^{(l)}\mathbf{W}_{\mathcal{V},s_i}^{(l)}$ transforms the previous layer's node features without any message passing, serving as a direct node-to-node path. The second term, $\tilde{\mathbf{H}}_F\hat{\mathbf{X}}_{\mathcal{E}}\mathbf{W}_{\mathcal{E},s_i}^{(l)}$, distributes the refined hyperedge features to their associated nodes, enabling higher-order feature propagation through the connections in $\mathbf{H}_F$. In conjunction, the first term captures node-specific features without structural context, while the latter captures higher-order neighborhood information lacking in the former. It is important to clarify that an arbitrary number of layers can be stacked sequentially. We use two layers throughout our experiments, with the layer index l taking values in $\{0,1\}$. Hence, the node features are initialized with $\mathbf{X}_{\mathcal{V},s_i}^{(0)} = \mathbf{X}$, and we obtain $\mathbf{X}_{\mathcal{V},s_i}^{(2)} \in \mathbb{R}^{N\times d_h}$ as the output node features after performing message passing through two HGNN layers, where d_h is the output node feature dimension.

Since the structure in $\mathbf{H}_L$ is dynamically learned as given in Eq. 21, some hyperedges may accumulate a disproportionate number of member nodes while others receive little to none. This imbalance can lead to oversmoothing, where node features become indistinguishable due to excessive representation mixing. To mitigate this effect, we incorporate a learnable residual weight matrix $\mathbf{W}_{r,s_i} \in \mathbb{R}^{C\times d_h}$ in each stream, which reintroduces raw node-specific information from the expression matrix $\mathbf{X}$ into the final representations. This promotes feature diversity and preserves individual gene characteristics. The new node representations for stream s_i are computed as:

$$\mathbf{X}_{\mathcal{V},s_i} = \mathbf{X}\mathbf{W}_{r,s_i} + \mathbf{X}_{\mathcal{V},s_i}^{(2)}. \tag{24}$$

To enable interaction between the two streams and allow each to benefit from the complementary information captured by the other, we incorporate a cross-attention mechanism. Specifically, we adopt a shared set of projections $q'(\cdot)$, $k'(\cdot)$, and $v'(\cdot)$, which map from d_h to d_h dimensions, and not to be confused with the mapping functions in Sec. IV-D. Given the node features $\mathbf{X}_{\mathcal{V},s_1}$ and $\mathbf{X}_{\mathcal{V},s_2}$ from the two streams, we compute cross-attended representations by attending each stream to the other. For instance, the refined node representations of the first stream are obtained with:

$$\boldsymbol{\alpha}_{s_1\to s_2} = \mathrm{Softmax}\left(q'(\mathbf{X}_{\mathcal{V},s_1})k'(\mathbf{X}_{\mathcal{V},s_2})^\top\right), \tag{25}$$

$$\hat{\mathbf{X}}_{\mathcal{V},s_1} = \mathbf{X}_{\mathcal{V},s_1} + \boldsymbol{\alpha}_{s_1\to s_2}v'\left(\mathbf{X}_{\mathcal{V},s_2}\right), \tag{26}$$

where $\boldsymbol{\alpha}_{s_1\to s_2} \in \mathbb{R}^{N\times N}$ is the cross-attention coefficient matrix, and the refined node features are stored in $\hat{\mathbf{X}}_{\mathcal{V},s_1} \in \mathbb{R}^{N\times d_h}$. Symmetrically, the update for the second stream is given by reversing the roles of s_1 and s_2:

$$\boldsymbol{\alpha}_{s_2\to s_1} = \mathrm{Softmax}\left(q'(\mathbf{X}_{\mathcal{V},s_2})k'(\mathbf{X}_{\mathcal{V},s_1})^\top\right), \tag{27}$$

$$\hat{\mathbf{X}}_{\mathcal{V},s_2} = \mathbf{X}_{\mathcal{V},s_2} + \boldsymbol{\alpha}_{s_2\to s_1}v'\left(\mathbf{X}_{\mathcal{V},s_1}\right). \tag{28}$$

The cross-attention enables information passing between node representations of two streams that capture distinct representations due to their independent initialization. The refined node features in the i-th stream $\hat{\mathbf{X}}_{\mathcal{V},s_i}$ are then processed through an MLP belonging to the i-th stream:

$$\mathbf{G}_{s_i} = \mathrm{MLP}_{s_i}\left(\hat{\mathbf{X}}_{\mathcal{V},s_i}\right), \tag{29}$$

where $\mathbf{G}_{s_i} \in \mathbb{R}^{N\times D}$ are the final set of gene representations in stream s_i. When we refer to $\mathbf{G}$, we implicitly refer to the set $\{\mathbf{G}_{s_1}, \mathbf{G}_{s_2}\}$, where both are gene embedding matrices obtained through separate streams with the same operations. For the loss optimization step discussed in Sec. IV-B, we draw embeddings of known TF-gene pairs from $\{\mathbf{G}_{s_1}, \mathbf{G}_{s_2}\}$ accordingly, and decode to predict the interaction scores.

V. Experiments and Results

A. Datasets

We use the cell-type-specific ChIP-seq datasets proposed in BEELINE [65]. There are seven distinct cell types: hHEP (human hepatocytes), hESC (human embryonic stem cells), mDC (mouse dendritic cells), mESC (mouse embryonic stem cells), mHSC-E (mouse hematopoietic stem cells, erythroid-biased), mHSC-GM (mouse hematopoietic stem cells, granulocyte-macrophage-biased), and mHSC-L (mouse hematopoietic stem cells, lymphoid-biased). We provide the raw and processed versions of the datasets.

In total, there are 14 datasets, of which seven are small-scale with 500 non-TF significantly varying genes, and seven are large-scale with $1,000$ non-TF significantly varying genes. In each scale, each cell type has scRNA-seq data and ChIP-seq data, which contain interactions exclusive to the specified cell type. Table I shows the statistics of ground truth ChIP-seq networks, such as the number of active TFs and genes that contain at least one interaction. The statistics of scRNA-seq data after pre-processing are given in the Supp. Table 1.

TABLE I: Summary of cell-type-specific ChIP-seq networks after pre-processing with the most varying 500 (and 1,000) genes.

Cell Type	Active Genes	Active TFs	Interactions	Network Density
hESC	815 (1,260)	34 (34)	20,677 (32,065)	0.164 (0.165)
hHEP	874 (1,331)	30 (31)	19,002 (30,026)	0.379 (0.377)
mDC	443 (684)	20 (21)	10,969 (18,556)	0.085 (0.082)
mESC	977 (1,385)	88 (89)	65,895 (96,460)	0.345 (0.347)
mHSC-E	691 (1,177)	29 (33)	13,632 (26,565)	0.578 (0.566)
mHSC-GM	618 (1,089)	22 (23)	9,280 (17,406)	0.543 (0.565)
mHSC-L	525 (640)	16 (16)	5,976 (7,392)	0.525 (0.507)

B. Experimental Setup

Our proposed framework contains various hyperparameters to be specified. For the regularization coefficients λ_{KL} and λ_{var} described in Sec. IV-B, we use $\lambda_{\text{KL}} = 0.0001$ and $\lambda_{\text{var}} = 0.005$. For the k_1 and k_2 values in the DHI module, we use $k_1 = k_2 = 10$. For the hidden dimensions d_e and d_h, we

TABLE II: AUROC and AUPRC performance of scHyperLink and baselines on cell-type-specific ChIP-seq datasets with 500 non-TF genes.

AUROC	hESC	hHEP	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
DeepSEM [54]	0.569	0.548	0.496	0.501	0.508	0.527	0.541
HyperG-VAE [64]	0.556	0.535	0.538	0.495	0.494	0.500	0.546
GENELink [40]	0.845	0.837	0.743	0.883	0.874	0.891	0.844
GNNLink [38]	0.827	0.831	0.654	0.820	0.798	0.852	0.849
GATCL [58]	0.865	0.851	0.754	0.887	0.869	0.881	0.820
GMFGRN [60]	0.864	0.895	0.795	0.898	0.918	0.918	0.874
GRACE [41]	0.849	0.868	0.650	0.917	0.900	0.893	0.836
scMGATGRN [59]	0.872	0.886	0.768	0.910	0.894	0.897	0.862
GCLink [62]	0.845	0.859	0.780	0.901	0.887	0.900	0.853
LineGRN [63]	0.870	0.880	0.801	0.871	0.911	0.913	0.845
scHyperLink	**0.885**	**0.905**	**0.824**	**0.924**	**0.927**	**0.930**	**0.886**
AUPRC	hESC	hHEP	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
DeepSEM [54]	0.191	0.402	0.052	0.308	0.567	0.517	0.528
HyperG-VAE [64]	0.189	0.382	0.061	0.301	0.563	0.528	0.540
GENELink [40]	0.521	0.692	0.129	0.749	0.890	0.894	0.838
GNNLink [38]	0.470	0.701	0.132	0.617	0.791	0.842	0.816
GATCL [58]	0.563	0.714	0.133	0.752	0.880	0.881	0.809
GMFGRN [60]	0.558	0.826	0.149	0.793	0.940	0.928	0.874
GRACE [41]	0.529	0.790	0.132	0.833	0.915	0.891	0.815
scMGATGRN [59]	0.598	0.804	0.150	0.807	0.910	0.897	0.859
GCLink [62]	0.556	0.759	0.169	0.777	0.904	0.905	0.845
LineGRN [63]	0.567	0.802	0.144	0.725	0.932	0.922	0.843
scHyperLink	**0.614**	**0.847**	**0.220**	**0.854**	**0.947**	**0.940**	**0.888**

TABLE III: AUROC and AUPRC performance of scHyperLink and baselines on cell-type-specific ChIP-seq datasets with 1,000 non-TF genes.

AUROC	hESC	hHEP	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
DeepSEM [54]	0.583	0.565	0.496	0.507	0.538	0.535	0.580
HyperG-VAE [64]	0.565	0.544	0.520	0.507	0.532	0.525	0.553
GENELink [40]	0.830	0.850	0.739	0.893	0.892	0.891	0.840
GNNLink [38]	0.853	0.822	0.708	0.818	0.833	0.863	0.807
GATCL [58]	0.870	0.858	0.754	0.900	0.894	0.875	0.822
GMFGRN [60]	0.863	0.900	0.799	0.905	0.933	0.930	0.852
GRACE [41]	0.852	0.877	0.703	0.919	0.901	0.911	0.844
scMGATGRN [59]	0.874	0.896	0.760	0.917	0.909	0.914	0.851
GCLink [62]	0.847	0.867	0.771	0.907	0.907	0.912	0.846
LineGRN [63]	0.874	0.886	0.798	0.872	0.933	0.925	0.843
scHyperLink	**0.885**	**0.906**	**0.833**	**0.926**	**0.941**	**0.940**	**0.879**
AUPRC	hESC	hHEP	mDC	mESC	mHSC-E	mHSC-GM	mHSC-L
DeepSEM [54]	0.189	0.413	0.046	0.314	0.563	0.535	0.523
HyperG-VAE [64]	0.189	0.380	0.049	0.304	0.570	0.547	0.514
GENELink [40]	0.505	0.710	0.114	0.756	0.896	0.898	0.817
GNNLink [38]	0.501	0.670	0.139	0.602	0.807	0.853	0.749
GATCL [58]	0.573	0.732	0.132	0.774	0.901	0.868	0.802
GMFGRN [60]	0.563	0.835	0.135	0.796	0.947	0.942	0.826
GRACE [41]	0.565	0.803	0.135	0.829	0.904	0.921	0.809
scMGATGRN [59]	0.610	0.821	0.138	0.826	0.917	0.919	0.837
GCLink [62]	0.555	0.769	0.126	0.786	0.919	0.920	0.821
LineGRN [63]	0.573	0.811	0.133	0.716	0.947	0.938	0.825
scHyperLink	**0.611**	**0.850**	**0.186**	**0.850**	**0.954**	**0.951**	**0.864**

use $d_e = d_h = 128$. For the final gene embedding dimension, we use $D = 64$. As the optimizer, we use Adam [69] with a learning rate 0.002 for training. All hyperparameters were chosen based on the losses on the validation sets. The hyperparameter search is presented in the Supp. Fig. 1. All experiments were conducted on an RTX 4090 24GB GPU.

C. Results

We compare scHyperLink against seven state-of-the-art baseline methods: DeepSEM [54], HyperG-VAE [64], GENELink [40], GNNLink [38], GATCL [58], GMFGRN [60], GRACE [41], scMGATGRN [59], GCLink [62], and LineGRN [63]. We use AUROC (Area Under the Receiver Operating Characteristic Curve) and AUPRC (Area Under the Precision-Recall Curve) as our evaluation metrics. To ensure fair comparison and reproducibility, we use the same dataset splits across all methods, execute each method 10 times with an identical array of seeds on the same device, and obtain the average score in each metric. In Tables II & III, we provide the results of scHyperLink and other baseline methods on the 500 non-TF and 1,000 non-TF variants of the cell-type-specific ChIP-seq datasets discussed in Sec. V-A, where the best performances in terms of metrics are displayed in bold, and the closest competing results are underlined.

The results in Tables II & III indicate that our proposed method consistently outperforms existing baselines across all cell-type-specific ChIP-seq datasets in AUROC and AUPRC metrics, demonstrating the importance of higher-order message passing through hypergraphs in the context of GRN reconstruction, which we also validate via conducting ablation studies by altering or removing the hypergraph structures and comparing the performances.

D. Ablation Studies

We conducted a set of ablation experiments to determine the effect of the main components of scHyperLink under various sparsity conditions, which are simulated via randomly dropping out a portion of the positive samples in training. For reference, we utilize interaction drop rates $p = \{0.0,\ 0.3,\ 0.5,\ 0.7\}$. To examine the robustness of scHyperLink, we create five variants. The variant *w/o DHI* removes the DHI module, hence the learned hypergraph, leaving only the initial hypergraph structure for inference. *w/o PHI* removes the prior hypergraph, leaving only the DHI module to learn a hypergraph structure. *w/o DS* reduces the dual stream nature of the HGNN module to a single stream, removing the cross-attention block in the process. *w/o CA* removes the cross-attention block but retains the dual-stream, isolating the contribution of cross-attention. Lastly, *w/o HGNN* removes the HGNN module from the streams, instead replacing it with two streams of two-layered GCNs. We run all variants and scHyperLink with a fixed array of 10 seeds in each sparsity scenario, and record the mean and standard deviations for the metric scores in each dataset. Our results in Fig. 2 and Table IV demonstrate that scHyperLink remains robust in metric scores across various sparsity scenarios.

TABLE IV: AUROC/AUPRC scores of model variants under different interaction drop rates p. Scores are averaged across seven cell types (10 runs each).

Variant	$p = 0.0$	$p = 0.3$	$p = 0.5$	$p = 0.7$
scHyperLink	**0.897 / 0.759**	**0.889 / 0.750**	**0.881 / 0.740**	**0.866 / 0.723**
w/o PHI	0.895 / 0.756	0.887 / 0.746	0.878 / 0.737	**0.866** / 0.720
w/o DHI	0.885 / 0.756	0.877 / 0.745	0.866 / 0.731	0.842 / 0.706
w/o DS	0.886 / 0.729	0.879 / 0.719	0.871 / 0.707	0.854 / 0.678
w/o CA	0.895 / 0.756	0.887 / 0.746	0.879 / 0.736	0.864 / 0.719
w/o HGNN	0.874 / 0.738	0.868 / 0.730	0.856 / 0.718	0.838 / 0.699

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2025.3650661

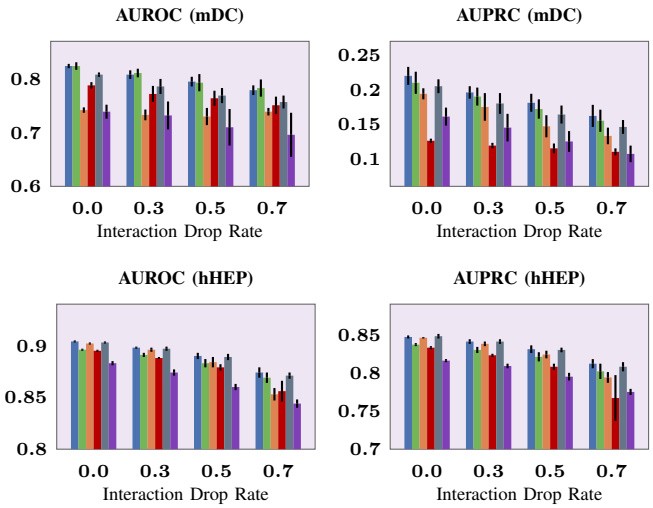

Fig. 2: Ablation study of scHyperLink under varying sparsity settings on two cell types (mDC & hHEP). Positive links are randomly dropped to simulate sparsity at rates shown on the x-axis. Bars represent mean AUROC/AUPRC across 10 runs, with standard deviations shown as black lines. From left to right, the variants are: scHyperLink (blue), w/o PHI (green), w/o DHI (orange), w/o DS (red), w/o CA (gray), and w/o HGNN (purple).

In the mDC dataset, which is the sparsest network [see Table I], the *w/o DHI* variant underperforms at any given sparsity condition compared to scHyperLink, while the *w/o PHI* variant does not affect the performance much. This confirms the expectation that with few prior interactions, the static hypergraph becomes noisy, and the dynamic hypergraph can complement and introduce beneficial TF-gene links for information propagation. As a denser network [see Table I], in the hHEP dataset, the removal of the prior hypergraph degrades the performance slightly more than the removal of the dynamic hypergraph, as there are useful prior interactions that can help with refining the representations.

To examine the overall performance of the variants across different datasets, we average the metric scores across all cell types for each variant, which are displayed in Table IV. Across each interaction drop rate, scHyperLink outperforms each variant, showing that the ablated components contribute to our model. The *w/o PHI* and *w/o DHI* variants perform well individually, and the best performance is obtained when they are used in conjunction. The *w/o DS* and *w/o HGNN* variants consistently underperform, showing the importance of using two separate streams to capture the distinct roles of TFs and target genes and using hypergraph neural networks to perform information propagation, respectively. Lastly, from the variant *w/o CA*, we reason that the cross-attention block is beneficial for the dual-stream nature of scHyperLink.

As an additional ablation study, we quantify the effect of the loss term $\mathcal{L}_{\text{var}}$ that acts as a regularizer. Specifically, we sweep its regularization coefficient λ_{var} over several orders of magnitude and report validation AUROC/AUPRC scores together with a Gini-based measure of hyperedge-membership imbalance computed on $\mathbf{H}_L$. We observe that decreasing λ_{var} weakens the regularization on $\mathbf{H}_L$, producing a more uneven distribution of hyperedge memberships (higher Gini) and a drop in performance scores. The degradation is strong in small-scale datasets, where insufficient PHI priors force the model to depend heavily on the learned DHI structure. The results of this analysis are presented in Supp. Fig. 1.

E. Few-Shot Transfer Learning

In many cell types or cellular states, the number of experimentally validated interactions is very limited, hindering GRN reconstruction. In such scenarios, leveraging knowledge from related, data-rich cell types becomes crucial. Hence, to test the generalization performance of scHyperLink, we consider a few-shot transfer setting, pretraining scHyperLink on a source cell type with ample data and transferring the learned weights to a new target cell type with only a small set of labeled TF–target interactions, where fine-tuning is performed.

To facilitate a few-shot scenario with our datasets, we randomly retain only 5% of the ground-truth interactions in the target cell type. Since the dataset of the source cell type for pretraining needs to be much larger than the target, we choose the mESC-1000 dataset as the source cell type, which has the largest number of ground-truth interactions [see Table I]. We ensure matching weight dimensions compatible for transfer via singular value decomposition on both the source and target cell type feature matrices. To ensure consistency with our previous experiments, we utilize the same set of hyperparameters, but decrease the number of epochs in the fine-tuning phase to avoid overfitting to the small target cell type.

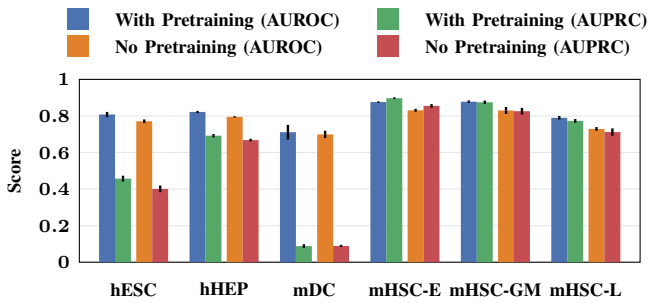

Fig. 3: Few-shot transfer of scHyperLink. In the pretrained setting, the learned weights on the source (mESC-1000) dataset are transferred to any other target dataset. Bars represent mean AUROC/AUPRC across 10 runs, with standard deviations depicted as black lines.

In Fig. 3, we show that pretraining scHyperLink on a large source dataset with broad interaction coverage and transferring the learned weights to a smaller target dataset consistently yields better performance than training directly on the limited target cell type. These results highlight scHyperLink's capacity to generalize across cell types and suggest its potential to extend to diverse perturbation conditions via transfer learning.

F. Computational Complexity Analysis

We benchmark computational cost against baselines using wall-clock training time and peak GPU memory. Supp. Table 2 reports results on the smaller mHSC-L-500 dataset, while Supp. Table 3 reports results on the larger hHEP-1000 dataset.

Additionally, to examine how scHyperLink scales to large-scale datasets, we perform a parametric analysis. Since asymptotic analysis focuses on order-of-growth, we adopt two practical simplifications consistent with our implementation [see

Sec. V-B]: we set $k_1{=}k_2{=}k$ and $d_e{=}d_h{=}d$ as in our experimental setup. The PHI module sets $\mathbf{H}_0 = \mathbf{A}_{tf}^{\top}$ at negligible cost. In the DHI module, the two GCN passes and the node-to-hyperedge and hyperedge-to-node attention computations dominate. In the HGNN module, the cross-attention operation between streams and the per-layer projections dominate. Putting these together, the forward pass scales as:

$$\mathcal{O}\big(N^2 d + (N+M)d^2 + N(C+M+D)d + NM\log k\big). \quad (30)$$

Here, the $N^2 d$ term comes from cross-attention across streams, $(N{+}M)\,d^2{+}N(C{+}M{+}D)\,d$ is from linear projections/MLPs (e.g., computing query/key/value matrices, per-layer weight multiplications); and $NM\log k$ is from the top-k selections used to sparsify node–hyperedge interactions. Since $\log k \ll \min\{N, M\}$, the other terms dominate, yielding an order of:

$$\mathcal{O}\big(N^2 d + (N+M)d^2 + N(C+M+D)d\big). \quad (31)$$

This analysis demonstrates that scHyperLink's complexity scales quadratically with respect to the number of genes N and the hidden dimension d, which denotes d_e or d_h, which are practically similar. The hidden dimensions in the DHI and HGNN modules can be kept relatively small and do not need to scale aggressively with dataset size. On the other hand, the $N^2 d$ term remains the principal bottleneck for scaling to large-scale datasets. When applying scHyperLink to datasets with lots of genes, several optimization strategies can be considered. We discuss some potential techniques in Sec. VI.

G. Interpretability Analysis on Learned Hyperedges

Having demonstrated the superior inference performance of scHyperLink in Tables II & III, and demonstrated the importance of the DHI module through ablation results in Fig. 2 and Table IV, we now turn to an interpretability analysis to assess the biological relevance of the learned regulatory hyperedges. To perform the biological plausibility evaluation, we employ Grad-CAM [52], a gradient-based localization method that highlights the regions of a layer's activation map that are most informative towards output predictions. To adapt Grad-CAM into our framework, we track the gradients of the learned hypergraph incidence matrix $\mathbf{H}_L \in \mathbb{R}^{N\times M}$ with respect to the binary interaction predictions for each TF-gene pair in the test set. This yields an importance matrix of size $N \times M$, where each entry (i, j) quantifies the contribution of target gene i and TF j toward the model's predictions in the test set. Then we take the column-wise mean of the importance matrix to obtain a $1 \times M$ vector, where each entry is a scalar value representing the average importance of the corresponding TF in the test set. We then choose the Top-5 contributing TFs and their target genes, which are effectively the Top-5 most significant learned hyperedges, and employ GO (Gene Ontology) [70] & KEGG (Kyoto Encyclopedia of Genes and Genomes) [71] term enrichment analysis to recover the most statistically significant terms in each TF's hyperedge gene set. For each TF, we retain one GO and one KEGG term.

For biological plausibility assessment, we use the mHSC-E dataset due to its highest AUROC and AUPRC performance. In Fig. 4, Top-5 TFs recovered by Grad-CAM (left) connect to their Top-5 targets (middle) and flow to significantly enriched terms (right) for that learned hyperedge, demonstrating crucial

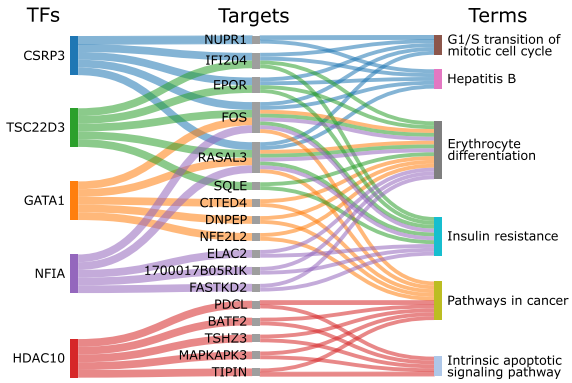

Fig. 4: Top-5 TFs recovered by Grad-CAM, their associated Top-5 target genes from the learned hyperedges, and the most statistically significant terms enriched per TF. For each TF, we display one GO term and one KEGG term. Each color-tracked path highlights one TF-to-term regulatory trajectory via its learned hyperedge set.

biological relevance in hematopoietic stem cells with erythroid differentiation tendencies. GATA1 serves as the master regulator in erythropoiesis, promoting erythroid lineage commitment by activating genes essential for red blood cell development while repressing alternative lineage programs [72]. Through interactions with cofactors like FOG1, it facilitates chromatin remodeling at erythroid-specific enhancers [73]. Conversely, NFIA plays an antagonistic role in erythroid differentiation. The NFIA-ETO2 fusion suppresses gene expression programs in erythroid maturation and differentiation, inducing erythroleukemia [74]. In broader hematopoietic context, HDAC10 regulates DNA replication and genome stability in malignant lymphoid cells. HDAC10 inhibition leads to DNA damage accumulation and apoptosis in cancerous lymphoid cells [75]. The findings in Fig. 4 align with experimental validations in literature, demonstrating that scHyperLink learns biologically plausible and context-specific hyperedge sets that propagate biologically consistent information across TFs and targets.

As a case study, we also evaluate whether scHyperLink captures known multi-TF regulatory modules. Specifically, we examine the well-characterized OCT4/POU5F1–SOX2–NANOG (OSN) core circuit that maintains pluripotency in human embryonic stem cells [76]. Using the hESC dataset and the same hyperparameter configuration as in our main experiments, we extracted the hyperedges inferred for the OSN TFs in each training seed and quantified their degree of co-regulation by computing the mean of the pairwise overlap ratios among the three hyperedges. This produced one co-targeting score per seed for the OSN triad, which we then averaged across seeds. To evaluate whether this co-targeting was greater than expected by chance, we generated a size-matched null by randomly resampling gene sets of the same cardinality for each TF, creating random hyperedges. This preserves the individual hyperedge sizes of POU5F1, SOX2, and NANOG while removing any coordinated targeting among them. Applying the overlap statistic to $10,000$ resampled triads yielded a null distribution, against which the observed OSN co-targeting score was compared. The OSN co-targeting score exceeded the null expectation with high significance ($p = 5.99 \times 10^{-4}$), and was $1.863\times$ larger than the null mean. Taken together, this experiment demonstrates that scHyperLink does not merely as-

This article has been accepted for publication in IEEE Journal of Biomedical and Health Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2025.3650661

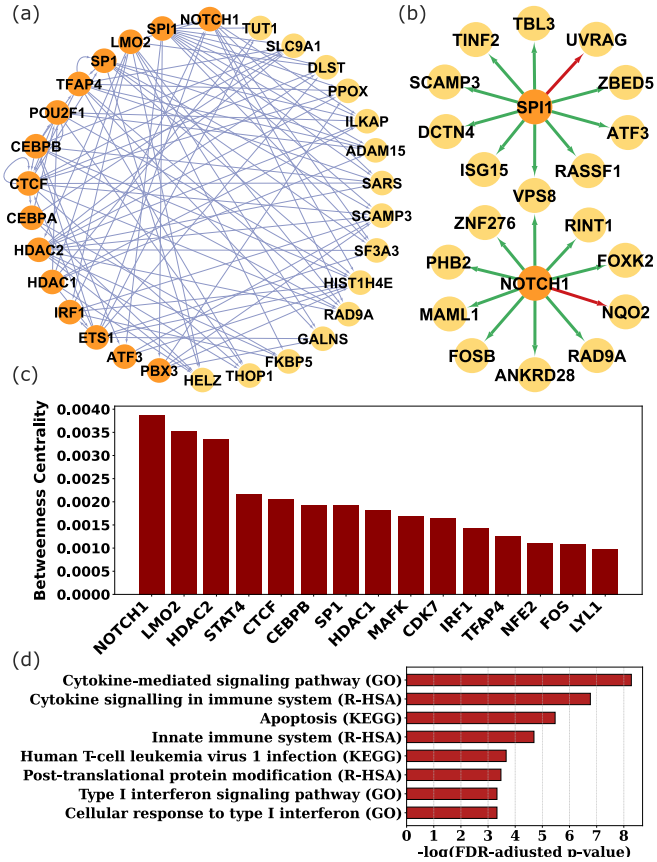

Fig. 5: (a) The reconstructed GRN of the Top-15 TFs and Top-15 target genes ranked by node degree. (b) The Top-10 predictions for the TF SPI1 and NOTCH1, each with nine true positive predictions. (c) The Top-15 TFs, ranked by the betweenness centrality coefficient. (d) Significantly enriched biological terms on the set of TFs found in (c) and their target genes. The horizontal axis displays the negative log of the adjusted p-values for the terms.

sign targets to each TF independently, but captures coordinated target sharing among TFs known to function jointly within a regulatory module, consistent with the expected role of the OSN triad in maintaining pluripotency. Additional results of this analysis are presented in Supp. Table 4.

H. Inferring Tissue-Specific Regulatory Networks

Across the primary analyses reported here, the TF-gene interaction inference task has been performed in cell-type-specific contexts. However, certain physiological and pathological processes, such as inflammation or immune responses, involve tissue-level phenomena that emerge from coordinated regulation across multiple cell types within a shared microenvironment [77]. Therefore, to evaluate how scHyperLink scales to broader interaction contexts, such as tissue-level specificity, we gather blood tissue interactions from the hTFtarget database [78]. As scRNA-seq expression data, we utilize the PBMC8K dataset, which comprises various cell types such as CD14+ monocytes, CD8 T cells, CD4 T cells, and B cells. The data pre-processing details are described in Supp. Materials.

After training, we apply scHyperLink to reconstruct a blood tissue-specific GRN at inference. We first analyze the reconstructed GRN at a global scale. Using the correctly predicted positive samples in the test set as edges and taking the Top-15 TFs and Top-15 target genes (ranked by node degree), Fig. 5-(a) displays a portion of the reconstructed GRN, where orange nodes are TFs and yellow nodes are target genes. As a global statistic of the reconstructed GRN, we examine the betweenness centrality coefficients of each node, a metric that identifies nodes serving as bottlenecks in the GRN [79]. Fig. 5-(c) illustrates that most information of the reconstructed GRN is localized in a few TFs, with NOTCH1 having the highest score. NOTCH1 is known to enhance B cell activation and antibody secretion [80], and regulate peripheral T cell activation, proliferation, and cytokine production [81], making it a highly significant regulator in peripheral blood mononuclear cells (PBMCs). To evaluate the biological relevance of the bottleneck TFs in Fig. 5-(c), we consider the bottleneck TFs and their target genes as a set and perform term enrichment analysis to extract significant biological processes. Fig. 5-(d) displays the GO & KEGG & Reactome [82] enriched terms and their adjusted p-values, which are highly consistent with general PBMC functions, such as sensing pathogens, producing cytokines, and coordinating early immune responses.

Additionally, we analyze the reconstructed GRN at a local scale, illustrated in Fig. 5-(b). Considering both the true positives and the false positives, we recover the Top-10 target prediction scores of two key TFs, NOTCH1 and SPI1, both of which identify nine out of 10 known targets. An examination shows that the NOTCH1-MAML1 pair is correctly predicted, which is a well-studied interaction. NOTCH1 and MAML1 work together as part of a key regulatory complex in PBMCs. MAML1 acts as a transcriptional coactivator that is directly recruited by the intracellular domain of activated NOTCH1. Once assembled into the NOTCH1–RBPJ–MAML1 complex, MAML1 enhances transcriptional activation and promotes the acetylation of NOTCH1 by p300, a modification that stabilizes NOTCH1 by preventing ubiquitination [83], [84]. In the SPI1 hub, the SPI1-ISG15 is another well-studied regulatory link. ISG15 contains a promoter region with a specific binding site for SPI1 (PU.1), a key TF in myeloid and lymphoid differentiation. SPI1 activates ISG15 by forming heterocomplexes with immune-restricted IRFs such as IRF8 and IRF4, promoting its expression in hematopoietic cells [85], [86].

Overall, this reconstruction of a blood tissue-specific GRN demonstrates that scHyperLink can be applied to broader contexts like tissue-level specificity while retaining biological relevance in the inferred networks, and hints at potentially novel targets yet to be experimentally validated.

VI. Discussion

The technical novelties of scHyperLink center on its hybrid HGNN architecture that uniquely combines static prior-informed hypergraphs with learnable dynamic hypergraphs to capture higher-order regulatory dependencies beyond conventional pairwise interactions. Unlike existing methods relying on correlation-based adjacency measures, scHyperLink adaptively learns context-specific hyperedges from gene expression data while leveraging experimentally validated regulatory knowledge, enabling comprehensive modeling of multi-way transcriptional dependencies. Clinically, this enhanced capability to reconstruct biologically plausible and context-specific

GRNs holds translational potential for precision medicine, including identification of novel therapeutic targets, prediction of drug responses based on patient-specific regulatory profiles, and understanding disease mechanisms through tissue-specific regulatory perturbations. The demonstrated scalability to tissue-level analyses and superior performance across diverse cell types position scHyperLink as a promising tool for clinical genomics, where accurate GRN inference could inform personalized treatment strategies and facilitate the discovery of regulatory biomarkers for disease diagnosis and prognosis.

Several limitations merit consideration for enhancing the utility of hypergraphs in GRN inference. Our static hypergraph construction in the PHI module relies on curated experimental interaction data to form hyperedges, providing reliable ground-truth links for information propagation. However, such curated resources are inherently incomplete and noisy, and they can exhibit biases toward TFs with many validated interactions. Conversely, TFs that are less studied often have sparse or missing annotations, causing a degree imbalance between TFs [87]. This imbalance can hinder effective information propagation for low-degree TFs, limiting the quality of learned representations and model performance. One particular direction to mitigate this issue is transfer learning (as demonstrated in Sec. V-E, where pretraining on large-scale interaction data can provide broader TF coverage, and enhance downstream performance following fine-tuning on cell–type–specific datasets. Another complementary strategy is to incorporate sequence-based regulatory priors, such as enhancer predictions. Approaches such as Enhancer-MDLF [88] identify cell-type-specific enhancer regions from DNA sequences, which could supply additional regulatory information for the PHI module and improve the biological grounding of inferred interactions.

Our dynamic hypergraph construction requires each gene to select and join a fixed number of TF-associated hyperedges. Without external regularization—such as the column-wise variance term $\mathcal{L}_{\text{var}}$ we employ—this scheme can lead to imbalanced membership distributions across TFs. This may be biologically plausible in GRN reconstruction, as cell-type-specific datasets typically feature a small number of active TFs [see Table I]. However, when seeking improved stability and uniformity, beyond our own column-wise variance regularization term that promotes uniformity, alternative dynamic hypergraph learning strategies such as HSLS [89] and SPHINX [90] can be considered. HSLS promotes smooth variation of signals across hyperedges for balanced memberships through regularization, and SPHINX uses k-subset sampling to ensure each hyperedge includes exactly k members. These alternative methods can help alleviate this imbalance. Beyond balancing strategies, more expressive graph-learning architectures may also strengthen DHI. The transformer-based TREE framework [91] captures long-range dependencies through multi-head relational attention, which could improve TF–gene interaction modeling in our setting. The Bayesian formulation in LCAAG [92] models link-level uncertainty and latent structural patterns, and incorporating a similar Bayesian prior for hyperedge formation could yield more reliable regulatory inferences.

The improved performance of scHyperLink in sparse regimes can be understood through the way it models TF-gene interactions as a hypergraph instead of a simple graph. In scHyperLink, each TF is assigned a hyperedge that links it to a group of potential target genes, processing relationships collectively rather than through isolated TF-gene pairs. Thus, information from a limited number of observed interactions can extend to other genes connected to the same TF, yielding more robust inference under data scarcity. Furthermore, the asymmetric design of the hypergraph, where only TFs initiate hyperedges, introduces a biologically plausible inductive bias that mirrors the direction of regulation observed in real gene networks. By constraining information flow to proceed from TFs to genes, only biologically plausible interactions are considered, reducing the risk of overfitting to coincidental correlations. Together, these properties allow scHyperLink to generalize more effectively in sparse regimes while maintaining an interpretable regulatory structure.

The HGNN layers in our model rely on the conventional formulation, with a forward-pass equation modification as proposed in TDHNN [68], which enables hyperedge features and node features to be used concurrently as in Eq. 23. While effective, future work could enhance this component by incorporating attention-based mechanisms [93] or exploring variants like WHNN [94], which employ Wasserstein distance to model geometric node relationships, offering a richer and more flexible representational capacity. Additionally, other dynamic hypergraph frameworks such as DHHNN [95] can be utilized, which integrates hyperbolic geometry to create dynamic hyperedges. The PHI module constructs static hypergraphs from experimentally-derived TF-gene interactions, which may be noisy, incomplete, or biased toward well-studied regulators. This can result in sparse and potentially misleading hypergraph structures [see Fig. 2]. A promising direction would be to adopt multi-view hypergraph construction strategies that integrate orthogonal biological data types—such as gene ontology (GO) annotations—to group functionally related genes into more biologically meaningful hyperedges, following approaches similar to HGTCGRN [61].

While our method is designed with efficiency in mind, scaling to large datasets such as whole-tissue atlases or multi-organ expression atlases remains a challenge. Constructing large dynamic hypergraphs with dense TF-gene relationships incur high memory and computational costs. Model optimization strategies and sparse training routines could facilitate broader applicability to large-scale genomics datasets [96], [97]. Various methods employ approximation strategies specifically aimed at improving the scalability of HGNNs. For instance, HyperGCN [98] reduces the cost of clique expansion by linking each node only to two representative nodes per hyperedge, and Ada-HGNN [99] accelerates training by adaptively sampling neighbors to replace the standard two-step aggregation. scHyperLink can also reduce computational costs by distributed/data-parallel training, sparse incidence matrix representations to cut storage costs, and gradient checkpointing to trade compute for lower peak memory.

Our model is trained and evaluated on cell-type–specific datasets, and we have further demonstrated a few-shot transfer

learning setting that enables adaptation to new cell types or conditions with limited labeled interactions. Nonetheless, broader generalization across diverse biological contexts, such as developmental stages or perturbation conditions, remains an open challenge. Future extensions could build on this foundation through domain adaptation, cross-condition, or multi-task training frameworks [100], [101], which may allow the model to capture both shared and context-specific regulatory mechanisms simultaneously. Integrating such designs with perturbation-based single-cell datasets or developmental atlases could further enhance robustness and interpretability across heterogeneous settings, extending the applicability of our approach beyond the static, cell-type–specific regime.

VII. Conclusion

We proposed scHyperLink, an HGNN-based model that fuses biological prior knowledge and dynamic learnable hypergraphs to create refined gene representations and predict cell-type-specific TF-gene interactions for GRN reconstruction. By utilizing hypergraphs, scHyperLink effectively captures higher-order interactions in gene regulation beyond standard pairwise graph structures. Our experiments showed scHyperLink outperforms state-of-the-art methods in link prediction accuracy across 14 benchmarks. Ablation studies across varying sparsity conditions validated complementary contributions of each module to scHyperLink's effectiveness. Interpretability analyses indicated that learned hyperedges capture biologically relevant information. Finally, application to tissue-scale GRN reconstruction with human blood tissue interactions demonstrated effective capture of important pathways and TFs.

References

[1] G. Karlebach *et al.*, "Modelling and analysis of gene regulatory networks," *Nat Rev Mol Cell Biol*, vol. 9, pp. 770–780, 2008.

[2] A. J. Singh *et al.*, "Differential gene regulatory networks in development and disease," *Cell Mol Life Sci*, vol. 75, pp. 1013–1025, 2018.

[3] L. Fei *et al.*, "Systematic identification of cell-fate regulatory programs using a single-cell atlas of mouse development," *Nat Gen*, vol. 54, pp. 1051–1061, 2022.

[4] P. U. Avila *et al.*, "Gene regulatory networks in disease and ageing," *Nat Rev Nephrol*, vol. 20, pp. 616–633, 2024.

[5] M. Ben Guebila *et al.*, "GRAND: A database of gene regulatory network models across human conditions," *Nucleic Acids Res*, vol. 50, no. D1, 2022.

[6] L. Fang *et al.*, "GRNdb: Decoding the gene regulatory networks in diverse human and mouse conditions," *Nucleic Acids Res*, vol. 49, no. D1, 2021.

[7] V. A. Huynh-Thu *et al.*, "Inferring regulatory networks from expression data using tree-based methods," *PLOS ONE*, vol. 5, no. 9, 2010.

[8] A. A. Margolin *et al.*, "ARACNE: An algorithm for the reconstruction of gene regulatory networks in a Mammalian cellular context," *BMC Bioinf*, vol. 7, no. 1, 2006.

[9] M. Banf *et al.*, "Computational inference of gene regulatory networks: Approaches, limitations and opportunities," *BBA Gen Reg Mech*, vol. 1860, no. 1, pp. 41–52, 2017.

[10] M. Zhao *et al.*, "A comprehensive overview and critical evaluation of gene regulatory network inference technologies," *Brief Bioinf*, vol. 22, no. 5, 2021.

[11] P. Parsana *et al.*, "Addressing confounding artifacts in reconstruction of gene co-expression networks," *Gen Biol*, vol. 20, no. 1, 2019.

[12] B. Van de Sande *et al.*, "Applications of single-cell RNA sequencing in drug discovery and development," *Nat Rev Drug Dis*, vol. 22, pp. 496–520, 2023.

[13] Y. Lei *et al.*, "Applications of single-cell sequencing in cancer research: Progress and perspectives," *J Hematol Oncol*, vol. 14, no. 1, 2021.

[14] D. T. Paik *et al.*, "Single-cell RNA sequencing in cardiovascular development, disease and medicine," *Nat Rev Cardiol*, vol. 17, no. 8, pp. 457–473, 2020.

[15] E. Boxer *et al.*, "Emerging clinical applications of single-cell RNA sequencing in oncology," *Nat Rev Clin Oncol*, vol. 22, pp. 315–326, 2025.

[16] P. V. Kharchenko, "The triumphs and limitations of computational methods for scRNA-seq," *Nat Methods*, vol. 18, pp. 723–732, 2021.

[17] L. Yuan *et al.*, "iCircDA-NEAE: Accelerated attribute network embedding and dynamic convolutional autoencoder for circRNA-disease associations prediction," *PLoS Comput Biol*, vol. 19, no. 8, 2023.

[18] L. Yuan *et al.*, "iCRBP-LKHA: Large convolutional kernel and hybrid channel-spatial attention for identifying circRNA-RBP interaction sites," *PLOS Comput Biol*, vol. 20, no. 8, 2024.

[19] L. Yuan *et al.*, "Identification of ferroptosis-related lncRNAs for predicting prognosis and immunotherapy response in non-small cell lung cancer," *Future Gen Comput Syst*, vol. 159, pp. 204–220, 2024.

[20] X. Zhou *et al.*, "scHiClassifier: A deep learning framework for cell type prediction by fusing multiple feature sets from single-cell Hi-C data," *Brief Bioinf*, vol. 26, no. 1, 2024.

[21] L. Yuan *et al.*, "scAMZI: Attention-based deep autoencoder with zero-inflated layer for clustering scRNA-seq data," *BMC Genomics*, vol. 26, 2025.

[22] X. Yang *et al.*, "scCross: A deep generative model for unifying single-cell multi-omics with seamless integration, cross-modal generation, and in silico exploration," *Genome Biol*, vol. 25, no. 1, 2024.

[23] Z.-H. Guo *et al.*, "scCorrector: A robust method for integrating multi-study single-cell data," *Brief Bioinf*, vol. 25, no. 2, 2024.

[24] M. A. Skinnider *et al.*, "Evaluating measures of association for single-cell transcriptomics," *Nat Methods*, vol. 16, pp. 381–386, 2019.

[25] H. Matsumoto *et al.*, "SCODE: An efficient regulatory network inference algorithm from single-cell RNA-seq during differentiation," *Bioinf*, vol. 33, no. 15, pp. 2314–2321, 2017.

[26] B. Yang *et al.*, "Reverse engineering gene regulatory network based on complex-valued ordinary differential equation model," *BMC Bioinf*, vol. 22, no. 3, 2021.

[27] Y. Yuan *et al.*, "Deep learning for inferring gene relationships from single-cell expression data," *Proc Nat Acad Sci*, vol. 116, no. 52, pp. 27 151–27 158, 2019.

[28] J. Chen *et al.*, "DeepDRIM: A deep neural network to reconstruct cell-type-specific gene regulatory network using single-cell RNA-seq data," *Brief Bioinf*, vol. 22, no. 6, 2021.

[29] F. Scarselli *et al.*, "The graph neural network model," *IEEE Trans Neural Net*, vol. 20, no. 1, pp. 61–80, 2009.

[30] T. N. Kipf *et al.*, "Semi-supervised classification with graph convolutional networks," *arXiv:1609.02907*, 2016.

[31] E. Koç *et al.*, "GraphTeacher: Transductive fine-tuning of encoders through graph neural networks," *IEEE Trans Artif Intell*, pp. 1–15, 2025.

[32] E. Koç *et al.*, "scGraPhT: Merging transformers and graph neural networks for single-cell annotation," *IEEE Trans Signal Inf Process Networks*, vol. 11, pp. 505–519, 2025.

[33] J. Wang *et al.*, "scGNN: A novel graph neural network framework for single-cell RNA-seq analyses," *Nat Commun*, vol. 12, no. 1, 2021.

[34] L. Yuan *et al.*, "scRGCL: A cell type annotation method for single-cell RNA-seq data using residual graph convolutional neural network with contrastive learning," *Brief Bioinf*, vol. 26, no. 1, 2025.

[35] W. Yang *et al.*, "Deciphering cell-cell communication at single-cell resolution for spatial transcriptomics with subgraph-based graph attention network," *Nat Commun*, vol. 15, 2024.

[36] D. S. Fischer *et al.*, "Modeling intercellular communication in tissues using spatial graphs of cells," *Nat Biotechnol*, vol. 41, pp. 332–336, 2023.

[37] J. Liu *et al.*, "scDD: A novel single-cell RNA-seq imputation method with diffusion and denoising," *Brief Bioinf*, vol. 23, no. 5, 2022.

[38] G. Mao *et al.*, "Predicting gene regulatory links from single-cell RNA-seq data using graph neural networks," *Brief Bioinf*, vol. 24, no. 6, 2023.

[39] A. Karaaslanli *et al.*, "scSGL: Kernelized signed graph learning for single-cell gene regulatory network inference," *Bioinf*, vol. 38, no. 11, pp. 3011–3019, 2022.

[40] G. Chen *et al.*, "Graph attention network for link prediction of gene regulations from single-cell RNA-sequencing data," *Bioinf*, vol. 38, no. 19, pp. 4522–4529, 2022.

[41] J.-C. Wang *et al.*, "GRACE: Unveiling gene regulatory networks with causal mechanistic graph neural networks in single-cell RNA-

sequencing data," *IEEE Trans Neural Net Learn Syst*, vol. 36, no. 5, pp. 9005–9017, 2025.
[42] D. Panne, "The enhanceosome," *Cur Opi Struct Biol*, vol. 18, no. 2, pp. 236–242, 2008.
[43] Y. Pan *et al.*, "The role of response elements organization in transcription factor selectivity: The IFN-β enhanceosome example," *PLoS Comput Biol*, vol. 7, no. 6, 2011.
[44] Y. Feng *et al.*, "Hypergraph neural networks," in *AAAI*, vol. 33, no. 1, 2019, pp. 3558–3565.
[45] X. Sun *et al.*, "Hyperedge representations with hypergraph wavelets: Applications to spatial transcriptomics," in *ICASSP*, 2025.
[46] C. Deng *et al.*, "Identifying new cancer genes based on the integration of annotated gene sets via hypergraph neural networks," *Bioinf*, vol. 40, 2024.
[47] W. Tao *et al.*, "Prediction of multi-relational drug–gene interaction via dynamic hypergraph contrastive learning," *Brief Bioinf*, vol. 24, no. 6, 2023.
[48] K. M. Saifuddin *et al.*, "HyGNN: Drug-drug interaction prediction via hypergraph neural network," in *ICDE*, 2023.
[49] Y. Qiao *et al.*, "Identifying novel therapeutic targets of natural compounds in traditional Chinese medicine herbs with hypergraph representation learning," *Brief Bioinf*, vol. 26, no. 4, 2025.
[50] W. Li *et al.*, "scHyper: Reconstructing cell–cell communication through hypergraph neural networks," *Brief Bioinf*, vol. 25, no. 5, 2024.
[51] S. Wu *et al.*, "Inference of gene regulatory networks based on multi-view hierarchical hypergraphs," *Int Sci Comput Life Sci*, vol. 16, no. 2, pp. 318–332, 2024.
[52] R. R. Selvaraju *et al.*, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017, pp. 618–626.
[53] T. Moerman *et al.*, "GRNBoost2 and Arboreto: Efficient and scalable inference of gene regulatory networks," *Bioinf*, vol. 35, no. 12, pp. 2159–2161, 2019.
[54] H. Shu *et al.*, "Modeling gene regulatory networks using neural network architectures," *Nat Comput Sci*, vol. 1, pp. 491–501, 2021.
[55] C. S. Gibbs *et al.*, "PMF-GRN: A variational inference approach to single-cell gene regulatory network inference using probabilistic matrix factorization," *Gen Biol*, vol. 25, no. 1, 2024.
[56] J. Xu *et al.*, "STGRNS: An interpretable transformer-based method for inferring gene regulatory networks from single-cell transcriptomic data," *Bioinf*, vol. 39, no. 4, 2023.
[57] K. K. C. *et al.*, "GNE: A deep learning framework for gene network inference by aggregating biological information," *BMC Syst Biol*, vol. 13, no. 2, p. 38, 2019.
[58] J. Liu *et al.*, "Graph attention network with convolutional layer for predicting gene regulations from single-cell ribonucleic acid sequence data," *Eng Appl Artif Intell*, vol. 136, 2024.
[59] L. Yuan *et al.*, "scMGATGRN: A multiview graph attention network–based method for inferring gene regulatory networks from single-cell transcriptomic data," *Brief Bioinf*, vol. 25, no. 6, 2024.
[60] S. Li *et al.*, "GMFGRN: A matrix factorization and graph neural network approach for gene regulatory network inference," *Brief Bioinf*, vol. 25, no. 2, 2024.
[61] W. Cui *et al.*, "Hierarchical graph transformer with contrastive learning for gene regulatory network inference," *IEEE J Biomed Health Inf*, vol. 29, no. 1, pp. 690–699, 2025.
[62] W. Yu *et al.*, "GCLink: A graph contrastive link prediction framework for gene regulatory network inference," *Bioinf*, vol. 41, no. 3, 2025.
[63] Z. Wang *et al.*, "LineGRN: A line graph neural network for gene regulatory network inference," *IEEE J Biomed Health Inf*, pp. 1–13, 2025.
[64] G. Su *et al.*, "Inferring gene regulatory networks by hypergraph generative model," *Cell Rep Methods*, vol. 5, no. 4, 2025.
[65] A. Pratapa *et al.*, "Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data," *Nat Methods*, vol. 17, no. 2, pp. 147–154, 2020.
[66] D. P. Kingma *et al.*, "Auto-encoding variational bayes," *arXiv:1312.6114*, 2013.
[67] A. Vaswani *et al.*, "Attention is all you need," in *NeurIPS*, 2017.
[68] P. Zhou *et al.*, "Totally dynamic hypergraph neural networks," in *IJCAI*, 2023, pp. 2476–2483.
[69] D. P. Kingma *et al.*, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.
[70] T. G. O. Consortium, "The Gene Ontology resource: 20 years and still GOing strong," *Nucleic Acids Res*, vol. 47, no. D1, pp. 330–338, 2019.
[71] M. Kanehisa *et al.*, "KEGG: New perspectives on genomes, pathways, diseases and drugs," *Nucleic Acids Res*, vol. 45, no. D1, pp. 353–361, 2017.
[72] L. Gutiérrez *et al.*, "Regulation of GATA1 levels in erythropoiesis," *IUBMB Life*, vol. 72, no. 1, pp. 89–105, 2020.
[73] E. Mancini *et al.*, "FOG-1 and GATA-1 act sequentially to specify definitive megakaryocytic and erythroid progenitors," *EMBO J*, vol. 31, no. 2, pp. 351–365, 2012.
[74] M.-R. Piqué-Borràs *et al.*, "The NFIA-ETO2 fusion blocks erythroid maturation and induces pure erythroid leukemia in cooperation with mutant TP53," *Blood*, vol. 141, no. 18, pp. 2245–2260, 2023.
[75] A. O. Mieland *et al.*, "The protein deacetylase HDAC10 controls DNA replication in malignant lymphoid cells," *Leukemia*, 2025.
[76] L. A. Boyer *et al.*, "Core transcriptional regulatory circuitry in human embryonic stem cells," *Cell*, vol. 122, no. 6, pp. 947–956, 2005.
[77] A. R. Sonawane *et al.*, "Understanding tissue-specific gene regulation," *Cell Rep*, vol. 21, no. 4, pp. 1077–1088, 2017.
[78] Q. Zhang *et al.*, "hTFtarget: A comprehensive database for regulations of human transcription factors and their targets," *Gen Proteo Bioinf*, vol. 18, no. 2, pp. 120–128, 2020.
[79] G. Ausiello *et al.*, "The (betweenness) centrality of critical nodes and network cores," in *IWCMC*, 2013, pp. 90–95.
[80] J.-A. Kang *et al.*, "Notch1 is an important mediator for enhancing of B-cell activation and antibody secretion by Notch ligand," *Immunology*, vol. 143, no. 4, pp. 550–559, 2014.
[81] T. N. Eagar *et al.*, "Notch 1 signaling regulates peripheral T cell activation," *Immunity*, vol. 20, no. 4, pp. 407–415, 2004.
[82] M. Gillespie *et al.*, "The Reactome pathway knowledgebase 2022," *Nucleic Acids Research*, vol. 50, no. D1, pp. 687–692, 2022.
[83] A. E. Popko-Scibor *et al.*, "Ubiquitination of Notch1 is regulated by MAML1-mediated p300 acetylation of Notch1," *Biochem Biophys Res Commun*, vol. 416, no. 3–4, pp. 300–6, 2011.
[84] M. Kitagawa, "Notch signalling in the nucleus: roles of Mastermind-like (MAML) transcriptional coactivators," *J Biochem*, vol. 159, no. 3, pp. 287–294, 2016.
[85] J. A. Kang *et al.*, "The diverse repertoire of ISG15: more intricate than initially thought," *Exp Mol Med*, vol. 54, pp. 1779–1792, 2022.
[86] D. Zhang *et al.*, "Interferon-Stimulated Gene 15 and the protein ISGylation system," *J Interfer Cytokine Res*, vol. 31, no. 1, pp. 119–130, 2011.
[87] P. Weidemüller *et al.*, "Transcription factors: Bridge between cell signaling and gene regulation," *Proteomics*, vol. 21, no. 23-24, 2021.
[88] Y. Zhang *et al.*, "Enhancer-MDLF: A novel deep learning framework for identifying cell-specific enhancers," *Brief Bioinf*, vol. 25, no. 2, 2024.
[89] B. T. Brown *et al.*, "Scalable hypergraph structure learning with diverse smoothness priors," *IEEE Trans Signal Inf Process Networks*, vol. 11, pp. 1072–1086, 2025.
[90] I. Duta *et al.*, "SPHINX: Structural prediction using hypergraph inference network," *arXiv:2410.03208*, 2024.
[91] X. Su *et al.*, "Interpretable identification of cancer genes across biological networks via transformer-powered graph representation learning," *Nat Biomed Eng*, vol. 9, pp. 371–389, 2025.
[92] Y. Yang *et al.*, "Link-based attributed graph clustering via approximate generative bayesian learning," *IEEE Trans Syst Man Cybern*, vol. 55, no. 8, pp. 5730–5743, 2025.
[93] S. Bai *et al.*, "Hypergraph convolution and hypergraph attention," *arXiv:1901.08150*, 2019.
[94] I. Duta *et al.*, "Wasserstein hypergraph neural network," *arXiv:2506.09682*, 2025.
[95] Z. Mei *et al.*, "DHHNN: A dynamic hypergraph hyperbolic neural network based on variational autoencoder for multimodal data integration and node classification," *Inf Fusion*, vol. 119, 2025.
[96] Y. Gao *et al.*, "Hypergraph learning: Methods and practices," *IEEE Trans Pattern Anal Mach Intell*, vol. 44, no. 5, pp. 2548–2566, 2022.
[97] Z. Lin *et al.*, "Automatic hypergraph generation for enhancing recommendation with sparse optimization," *IEEE Trans Multimedia*, vol. 26, pp. 5680–5693, 2024.
[98] N. Yadati *et al.*, "HyperGCN: A new method for training graph convolutional networks on hypergraphs," in *NeurIPS*, 2019, pp. 1509–1520.
[99] S. Wang *et al.*, "Ada-HGNN: Adaptive sampling for scalable hypergraph neural networks," *arXiv preprint arXiv:2405.13372*, 2024.
[100] D. Seçilmiş *et al.*, "Knowledge of the perturbation design is essential for accurate gene regulatory network inference," *Sci Rep*, vol. 12, 2022.
[101] R. Shen *et al.*, "Graph domain adaptation–based framework for gene expression enhancement and cell type identification in large-scale spatially resolved transcriptomics," *Brief Bioinf*, vol. 25, no. 6, 2024.