Integrating long-range regulatory interactions to predict gene expression using graph convolutional networks

Jeremy Bigness[1,2,4], Xavier Loinaz[2], Shalin Patel[3], Erica Larschan[1,4], and Ritambhara Singh[1,2]

[1]*Center for Computational Molecular Biology, Brown University*
[2]*Department of Computer Science, Brown University*
[3]*Division of Applied Mathematics, Brown University*
[4]*Department of Molecular Biology, Cell Biology and Biochemistry, Brown University*

Abstract

Long-range spatial interactions among genomic regions are critical for regulating gene expression, and their disruption has been associated with a host of diseases. However, when modeling the effects of regulatory factors, most deep learning models either neglect long-range interactions or fail to capture the inherent 3D structure of the underlying genomic organization. To address these limitations, we present GC-MERGE, a **G**raph **C**onvolutional **M**odel for **E**pigenetic **R**egulation of **G**ene **E**xpression. It integrates measurements of both the spatial genomic organization and local regulatory factors, specifically histone modifications, for predicting gene expression. This formulation enables the model to incorporate crucial information about long-range interactions via a natural encoding of spatial interactions into a graph representation. We apply GC-MERGE to datasets for the GM12878 (lymphoblastoid), K562 (myelogenous leukemia), and HUVEC (human umbilical vein endothelial) cell lines and demonstrate its state-of-the-art predictive performance. More importantly, we show that our model is interpretable in terms of the observed biological regulatory factors, highlighting both the histone modifications and the interacting genomic regions contributing to a gene's predicted expression. We provide model explanations for several exemplar genes and validate them with evidence from the literature. Our model not only presents a novel setup for predicting gene expression by integrating multimodal datasets in a graph convolutional framework but it also enables interpretation of the biological mechanisms driving the model's predictions. Available at `https://github.com/rsinghlab/GC-MERGE`.

1 Introduction

Gene regulation determines the fate of every cell, and its disruption leads to diverse diseases ranging from cancer to neurodegeneration [17, 26]. Although specialized cell types – from neurons to cardiac cells – exhibit different gene expression patterns, the information encoded by the linear DNA sequence remains virtually the same in all non-reproductive cells of the body. Therefore, the observed differences in cell type must be encoded by elements extrinsic to sequence, commonly referred to as epigenetic factors. Epigenetic factors found in the local neighborhood of a gene typically include histone marks (also known as histone modifications). These marks are naturally occurring chemical additions to histone proteins that control how tightly the DNA strands are wound around the proteins and the recruitment or occlusion of transcription factors. However, the focus of attention in genomics has shifted increasingly to the study of long-range epigenetic regulatory interactions that result from the three-dimensional organization of the genome [25]. For example, one early study demonstrated that chromosomal rearrangements, some located as far as 125 kilo-basepairs (kbp) away, disrupted the region downstream of the PAX6 transcription unit causing Aniridia (absence of the iris) and related eye anomalies [16]. Thus, chromosomal rearrangement can not only directly affect the expression of proximal genes but can also indirectly affect a gene located far away by perturbing its regulatory (e.g., enhancer-promoter) interactions. This observation indicates that while local regulation of genes is informative, studying long-range

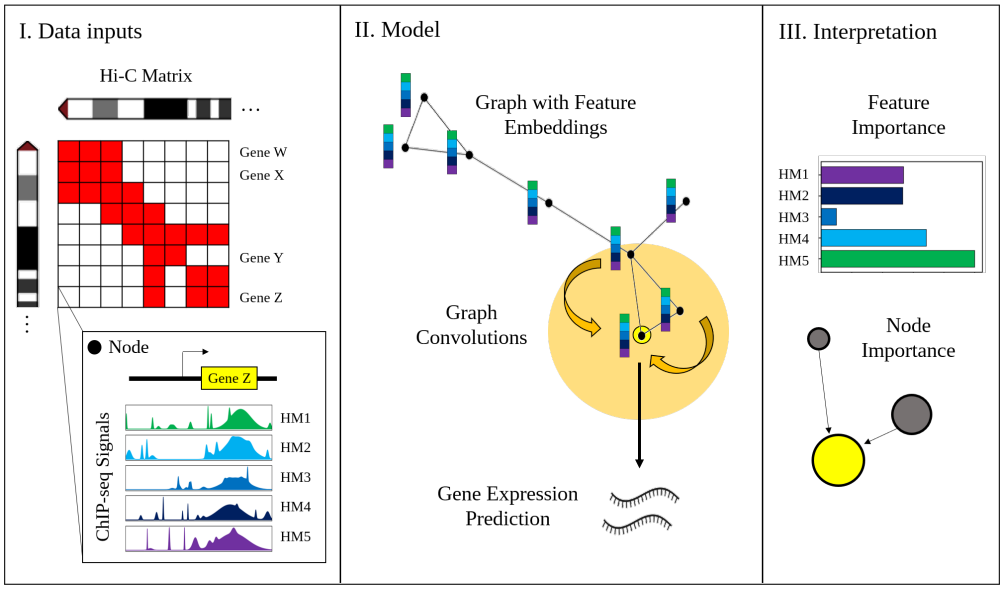

Figure 1: **Overview of GC-MERGE.** Our framework integrates both local histone mark (HM) signals and long-range spatial interactions to predict and understand gene expression. Inputs to the model include Hi-C maps for each chromosome, with chromosomal subregions corresponding to nodes in the graph, as well as the ChIP-seq readings of five core histone marks in each subregion, which constitute the initial feature embedding of the nodes. For nodes corresponding to subregions containing a gene, the model performs repeated graph convolutions over the neighboring nodes to yield either a continuous, real-valued prediction of expression level or a binarized class prediction of gene activity (either active or inactive). Finally, explanations for the model's predictions for any gene-associated node can be obtained by calculating the importance scores for each of the features as well as the relative contributions of neighboring nodes. Therefore, the model provides biological insight into the pattern of histone marks as well as the spatial regulatory interactions that work together to determine gene expression.

gene regulation is critical to understanding cell development and disease. However, experimentally testing for all possible combinations of long-range and short-range regulatory factors for $\sim 20,000$ genes is infeasible given the vast size of the search space. Therefore, computational and data-driven approaches are necessary to efficiently search this space and reduce the number of testable hypotheses due to the sheer scope of the problem.

Recently, deep learning frameworks have been applied to predict gene expression from histone modifications, and their empirical performance has often exceeded the previous machine learning methods [4, 6, 14]. Among their many advantages, deep neural networks perform automatic feature extraction by efficiently exploring feature space and then finding nonlinear transformations of the weighted averages of those features. This formulation is especially relevant to complex biological systems since they are inherently nonlinear. For instance, Singh *et al.* [29] introduced DeepChrome, which used a convolutional neural network (CNN) to aggregate five types of histone mark ChIP-seq signals in a $10,000$ bp region around the transcription start site (TSS) of each gene. Using a similar setup, they next introduced attention layers to their model [30], yielding a comparable performance but with the added ability to visualize feature importance within the local neighborhood of a gene. These methods framed the gene expression problem as a binary classification task in which the gene was either active or inactive. Agarwal *et al.* [1] introduced Xpresso, a CNN framework that operated on the promoter sequences of each gene and 8 other annotated features associated with mRNA decay to predict steady-state mRNA levels. This model

focused primarily on the regression task, such that each prediction corresponded to the logarithm of a gene's expression. While all the studies listed above accounted for combinatorial interactions among features at the local level, they did not incorporate long-range regulatory interactions known to play a critical role in differentiation and disease [17, 26].

Modeling these long-range interactions is a challenging task due to two significant reasons. First, we cannot confidently pick an input size for the genomic regions as regulatory elements can control gene expression from various distances. Second, inputting a large region will introduce sparsity and noise into the data, making the learning task difficult. A potential solution to this problem is to incorporate information from long-range interaction networks captured from experiments like Hi-ChIP [19] and Hi-C [34]. These assays use high-throughput sequencing to measure 3D genomic structure, in which each read pair corresponds to an observed 3D contact between two genomic loci. While Hi-ChIP focuses only on spatial interactions mediated by a specific protein, Hi-C captures the global interactions of all genomic regions. Recently, Zeng *et al.* [39] combined a CNN, encoding promoter sequences, with a fully connected network using Hi-ChIP datasets to predict gene expression values. The authors then evaluated the relative contributions of the promoter sequence and promoter-enhancer submodules to the model's overall performance. While this method incorporated long-range interaction information, its use of HiChIP experiments narrowed this information to spatial interactions mediated by H3K27ac and YY1. Furthermore, CNN models only capture the local topological patterns instead of modeling the underlying spatial structure of the data. Thus, the interpretation of their model was limited to local sequence features.

Computational Study	**Local Histone Marks**	**Additional Features (e.g. promoter sequence)**	**Feature-level Interpretation**	**Long-range Interactions**	**Edge-level Interpretation**
DeepChrome [29]	X		X		
AttentiveChrome [30]	X		X		
Xpresso [1]		X	X		
DeepExpression [39]		X	X	X	
GC-MERGE (this study)	X		X	X	X

Table 1: **Comparison of the properties of previous deep learning models predicting gene expression with GC-MERGE.** The proposed method integrates local and long-range regulatory interactions, capturing the underlying 3D genomic spatial structure, as well as highlights both the important node-level (histone modifications) and edge-level (long-range interactions) features.

To address these issues, we developed a **G**raph **C**onvolutional **M**odel of **E**pigentic **R**egulation of **G**ene **E**xpression (GC-MERGE), a graph-based deep learning framework that integrates 3D genomic data with existing histone mark signals to predict gene expression. Figure 1 provides a schematic of our overall approach. Unlike previous methods, our model incorporates genome-wide interaction information by using the Hi-C data. To accomplish this, we use a graph convolutional network (GCN) to capture the underlying spatial structure. GCNs are particularly well-suited to representing spatial relationships, as a Hi-C map can be represented as an adjacency matrix of an undirected graph $G \in \{V, E\}$. Here, V nodes represent the genomic regions and E edges represent their interactions. Our prediction task formulation captures the local and spatial relationships between the histone marks and gene expression. While some methods use many other types of features, such as promoter sequences [1, 39], we focus our efforts solely on histone modifications and extract their relationship to the genes. Even with this simplified set of features, we show that our model's performance exceeds that of the baseline methods for the gene expression prediction task.

Another significant contribution of our work is to enable biologists to determine at the genic level which regulatory interactions – local or distal – most affect the gene's expression and which histone

marks modulate these interactions. By making the model's predictive drivers more transparent, this information can suggest promising hypotheses and guide new research directions. To that effect, we perform an interpretation of the model's predictions that quantifies the relative importance of the underlying biological regulatory factors driving each gene's output. We integrate the GNNExplainer method [37] within our modeling framework to highlight not only the important node features (histone modifications) but also the important edges (long-range interactions) that contribute to determining a particular gene's predicted expression. In this paper, we apply our method to the three cell lines from Rao *et al.* [22] – GM12878 (lymphoblastoid), K562 (myelogenous leukemia), and HUVEC (human umbilical vein endothelial cells). While solving the gene expression prediction as a regression problem is more valuable for the community, our interpretation framework required us to formulate it as a classification task. Therefore, we perform both regression and classification tasks and demonstrate state-of-the-art performance. Furthermore, we show that our framework allows biologists to tease apart the cumulative effects of different regulatory mechanisms at the genic level. Table 1 places the proposed framework among state-of-the-art deep learning models and lists each model's properties.

2 Methods

Graph convolutional networks (GCNs) Graph convolutional networks (GCNs) are a generalization of convolutional neural networks (CNNs) to graph-based relational data that is not natively structured in Euclidean space [18]. Due to the expressive power of graphs, GCNs have been applied across a wide variety of domains, including recommender systems [12], and social networks [21]. The prevalence of graph-based datasets in biology has made these models a popular choice for tasks like modeling protein-protein interactions [36], stem cell differentiation [2], and chemical reactivity for drug discovery [32].

We use the GraphSAGE formulation [11] as our GCN for its relative simplicity and its capacity to learn generalizable, inductive representations not limited to a specific graph. The input to the model is represented as a graph $G \in \{V, E\}$, with nodes V and edges E, and a corresponding adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ [18], where N is the number of nodes. For each node v, there is also an associated feature vector $\boldsymbol{x_v}$. The goal of the network is to learn a state embedding $\boldsymbol{h}_v^K \in \mathbb{R}^d$ for v, which is obtained by aggregating information over v's neighborhood K times. Here, d is the dimension of the embedding vector. This state embedding is then fed through a fully-connected network to produce an output $\hat{y_v}$, which can then be applied to downstream classification or regression tasks.

Within this framework, the first step is to initialize each node with its input features. In our case, the feature vector $\boldsymbol{x}_v \in \mathbb{R}^m$ is obtained from the ChIP-seq signals corresponding to the five ($m = 5$) core histone marks (H3K4me1, H3K4me3, H3K9me3, H3K36me3, and H3K27me3) in our dataset:

$$\boldsymbol{h}_v^0 = \boldsymbol{x}_v \tag{1}$$

Next, to transition from the $(k-1)^{th}$ layer to the kth hidden layer in the network for node v, we apply an aggregation function to the neighborhood of each node. This aggregation function is analogous to a convolution operation over regularly structured Euclidean data such as images. A standard convolution function operates over a grid and represents a pixel as a weighted aggregation of its neighboring pixels. Similarly, a graph convolution performs this operation over the neighbors of a node in a graph. In our case, the aggregation function calculates the mean of the neighboring node features:

$$\boldsymbol{h}_{\mathcal{N}(v)}^k = \sum_{u \in \mathcal{N}(v)} \frac{\boldsymbol{h}_u^{k-1}}{|\mathcal{N}(v)|} \tag{2}$$

Here, $\mathcal{N}(v)$ represents the adjacency set of node v. We update the node's embedding by concatenating the aggregation with the previous layer's representation to retain information from the original embedding. Next, just as done in a standard convolution operation, we take the matrix product of this concatenated representation with a learnable weight matrix to complete the weighted aggregation step. Finally, we apply a non-linear activation function, such as ReLU, to capture the higher-order non-linear interactions among the features:

$$\boldsymbol{h}_v^k = \sigma\left(\mathbf{W}_k\left[\boldsymbol{h}_{\mathcal{N}(v)}^k \mid\mid \boldsymbol{h}_v^{k-1}\right]\right), \forall k \in \{1, ..., K\} \tag{3}$$

Here, $||$ represents concatenation, σ is a non-linear activation function, and $\mathbf{W}_k$ is a learnable weight parameter. After this step, each node is assigned a new embedding. After K iterations, the node embedding encodes information from the neighbors that are K-hops away from that node:

$$\boldsymbol{z}_v = \boldsymbol{h}_v^K \tag{4}$$

Here, $\boldsymbol{z}_v$ is the final node embedding after K iterations. For regression, we feed $\boldsymbol{z}_v$ into a fully connected network and output a prediction $\hat{y_v} \in \mathbb{R}$, representing a real-valued expression level. We use the mean squared error (MSE) as the loss function. The model architecture is summarized in Supplementary Figure S1.

Interpretation of GC-MERGE Although a model's architecture is integral to its performance, just as important is understanding how the model arrives at its predictions. Neural networks in particular have sometimes been criticized for being "black box" models, such that no insight is provided into how the model operates. Most graph-based interpretability approaches either approximate models with simpler models whose decisions can be used for explanations [23] or use an attention mechanism to identify relevant features in the input that guide a particular prediction [35]. In general, these methods, along with gradient-based approaches [28, 33] or DeepLift [27], focus on the explanation of important node features and do not incorporate the structural information of the graph. However, a recent method called *Graph Neural Net Explainer* (or GNNExplainer) [37], given a trained GCN, can identify a small subgraph as well as a small subset of features that are crucial for a particular prediction. The authors demonstrate its interpretation capabilities on simulated and real-world graphs.

In order to apply this method, the problem must be constructed as a classification task. Therefore, we feed the learned embedding $\boldsymbol{z}_v$ in Equation 4 into a fully connected network and output a prediction $\hat{y_v}$ for each target node using a $Softmax$ layer to compute probabilities for each class c. Here, class $c \in \{0, 1\}$ corresponds to whether the gene is either off/inactive ($c = 0$) or on/active ($c = 1$). We use the true binarized gene expression value $y_v \in \{0, 1\}$ by thresholding the expression level relative to the median as the target predictions (as done previously [4, 29, 30, 39]), using a negative log-likelihood (NLL) loss to train the model.

Next, we integrate the GNNExplainer module into our classifier framework. GNNExplainer maximizes the mutual information between the probability distribution of the model's class predictions over all nodes and the probability distribution of the class predictions for a particular node conditioned on some fractional masked subgraph of neighboring nodes and features. Subject to regularization constraints, it jointly optimizes the fractional node and feature masks, determining the extent to which each element informs the prediction for a particular node.

Specifically, given a node v, the goal is to learn a subgraph $G_s \subseteq G$ and a feature mask $X_s = \{x_j \mid v_j \in G_s\}$ that contribute the most to driving the full model's prediction of $\hat{y}_v$. To achieve this objective, the algorithm learns a mask that maximizes the mutual information (MI) between the original model and the masked model. Mathematically, this objective function is as follows:

$$\max_{G_s} MI(Y, (G_s, X_s)) = H(Y) - H(Y \mid G_s, X_s) \tag{5}$$

where H is the entropy of a distribution. Since this is computationally intractable with an exponential number of graph masks, GNNExplainer optimizes the following quantity using gradient descent:

$$\min_{M,N} -\sum_{c=1}^{C} \mathbb{1}_{\{y=c\}} \log(P_\phi(Y = y | G = A_c \odot \sigma(M), X = X_c \odot \sigma(N)) \tag{6}$$

where c represents the class, A_c represents the adjacency matrix of the computation graph, M represents the subgraph mask, and N represents the feature mask. The importance scores of the nodes and features are obtained by applying the sigmoid function to the subgraph and feature masks, respectively. Finally, the element-wise entropies of the masks are calculated and inserted as regularization terms into the loss function. Therefore, in the context of our model, GNNExplainer learns which genomic regions (via the subgraph mask) and which features (via the feature mask) are most important in driving the model's predictions.

3 Experimental Setup

Overview of the datasets GC-MERGE requires the following information: (1) Interactions between the genomic regions (Hi-C contact maps); (2) Histone mark signals representing the regulatory signals (ChIP-seq measurements); (3) Expression levels for each gene (RNA-seq measurements). For each gene in a particular region, the first two datasets are the inputs into our proposed model, whereas gene expression is the predicted target. We formulate the problem as both regression and classification tasks. We take the base-10 logarithm of the gene expression values for the regression task, adding a pseudo-count of 1. For the classification task, we binarize the gene expression values as either 0 (off) or 1 (on) using the median as the threshold, consistent with previous studies [4, 29, 30, 39]. Constructing a binary classifier enables us to integrate the GNNExplainer interpretive mechanism with our framework.

We focused on three human cell lines from Rao *et al.* [22]: (1) GM12878, a lymphoblastoid cell line with a normal karyotype, (2) K562, a myelogenous leukemia cell line, and (3) HUVEC, a human umbilical vein endothelial cell line. For each of these cell lines, we accessed RNA-seq expression and ChIP-Seq signal datasets for five uniformly profiled histone marks from the REMC repository [24]. These histone marks include (1) H3K4me1, associated with enhancer regions; (2) H3K4me3, associated with promoter regions; (3) H3K9me3, associated with heterochromatin; (4) H3K36me3, associated with actively transcribed regions; and (5) H3K27me3, associated with polycomb repression. We chose these marks because of the wide availability of the relevant data as well as for ease of comparison with previous studies [29, 30, 39].

Graph construction and data integration Our main innovation is formulating the graph-based prediction task to integrate two very different data modalities (histone mark signals and Hi-C interaction frequencies). We represented each genomic region with a node and connected edges between it and the nodes corresponding to its neighbors (bins with non-zero entries in the adjacency matrix) to construct the graph. Due to the large size of the Hi-C graph, we subsampled neighbors to form a subgraph for each node we fed into the model. While there are methods to perform subsampling on large graphs using a random node selection approach (e.g., [38]), we used a simple strategy of selecting the top j neighbors with the highest Hi-C interaction frequency values. We empirically selected the value $j = 10$ for the number of neighbors. A smaller number of neighbors (i.e., $j = 5$) resulted in decreased performance while selecting more neighbors proved prohibitive due to memory constraints.

To integrate the Hi-C datasets (preprocessing details in Supplementary Section S2.1) with the RNA-seq and ChIP-seq datasets, we obtained the average ChIP-seq signal for each of the five core histone marks over the chromosomal region corresponding to each node. In this way, a feature vector of length five was associated with each node. For the RNA-seq data, we took each gene's transcriptional start site (TSS) and assigned it to the node corresponding to the chromosomal region in which the TSS is located. We applied a mask during the training phase so that the model made predictions only on nodes corresponding to chromosomal regions with genes. If multiple genes were assigned to the same node, we took the median of the expression levels. We assigned 70% of the nodes to the training set, 15% to the validation set, and 15% to the testing set. We provide the details of the hyperparameter tuning in Supplementary Section S1.2.

Baselines models We compared GC-MERGE with the following deep learning baselines for gene expression prediction formulated as both regression and classification tasks:

- **Multi-layer perceptron (MLP)**: A simple MLP comprised of three fully-connected layers. In this framework, the model predictions for each node do not incorporate feature information from the node's neighbors.
- **Shuffled neighbor model**: GC-MERGE applied to shuffled Hi-C matrices, such that the neighbors of each node are randomized. The shuffled neighbor and MLP baselines can be viewed as proxies for the importance of including information from long-range regulatory interactions for similarly processed inputs.
- **Convolutional neural network (CNN):** A convolutional neural network based on DeepChrome [29]. This model takes 10 kb regions corresponding to the genomic regions demarcated in the Hi-C data and subdivides each region into 100 bins. Each bin is associated with five channels, which correspond to the ChIP-seq signals of the same five core histone marks in the present study. A standard convolution is applied to the channels, followed by a fully-connected network.

For the regression task, the range of the outputs is the set of continuous real numbers. For the classification task, a $Softmax$ function is applied to the models' output to yield a binary prediction. For the CNN baseline, genomic regions are subdivided into smaller 100-bp bins, consistent with Singh *et al.* [29]. However, GC-MERGE and the baselines other than the CNN average the histone modification signals over the entire 10 kb region.

We also implemented GC-MERGE on higher resolution ChIP-seq datasets (1000-bp bins), which we fed through a linear embedding module to form features for the Hi-C nodes. We did not observe an improvement in the performance for the high-resolution input (Supplementary Figure S2). Additionally, we compared our results to the published results of two other recent deep learning methods, Xpresso by Agarwal *et al.* [1] and DeepExpression by Zeng *et al.* [39], when such comparisons were possible, since in some cases the experimental data sets were unavailable or the code provided did not run

Evaluation metrics We measured the regression task performance of all the models by calculating the Pearson correlation coefficient (PCC), which quantifies the correlation between the true and predicted gene expression values in the test set. For interpretation, we adapted our model to perform classification. Therefore, we also evaluated the classification performance using two metrics: the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPR).

4 Results

GC-MERGE gives state-of-the-art performance for the gene expression prediction task We evaluate GC-MERGE and the baseline models on the regression task for the GM12878, K562, and HUVEC cell lines. Figure 2(a) shows the predicted versus true gene expression values for GC-MERGE and Figure

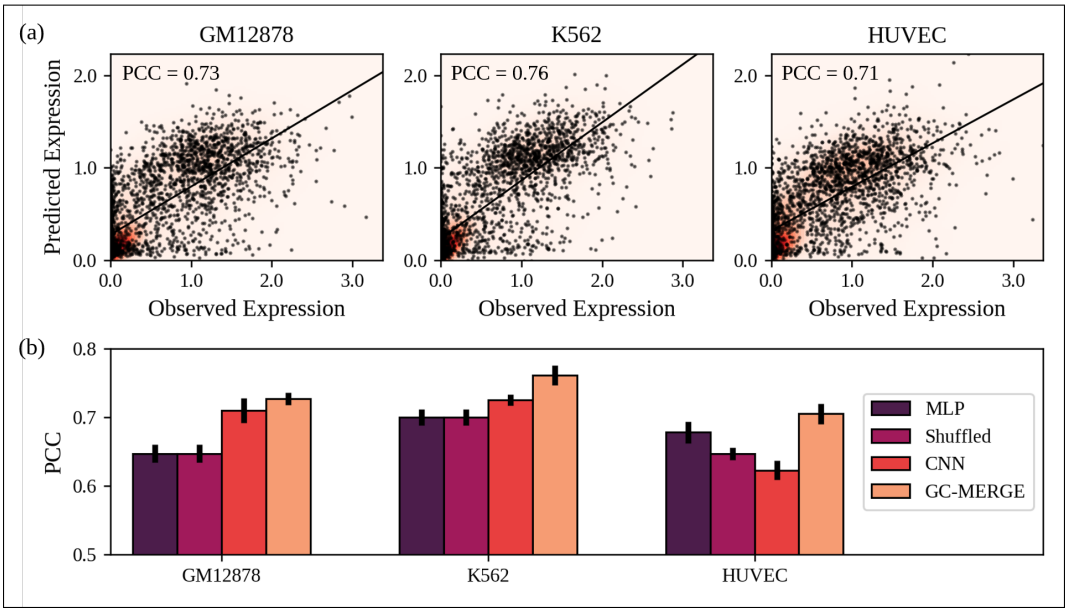

Figure 2: **Model comparison and evaluation.** GC-MERGE gives state-of-the-art performance for the regression task on all three studied cell lines. (a) The Pearson correlation coefficients (PCC) obtained by running GC-MERGE on each cell line are displayed. (b) GC-MERGE outperforms all the baseline models for each of the three cell lines. Scores are calculated as the average of 10 runs and standard deviations are denoted by error bars.

2(b) compares our model's performance with the baselines. Note that we determine the Pearson correlation coefficient (PCC) by taking the average of ten runs and denote the standard deviation by the error bars on the graph. For GM12878, the Pearson correlation coefficient of GC-MERGE predictions (PCC = 0.73) exceeds that of the other baselines. Furthermore, we note that our model performance also compares favorably to Xpresso (PCC $\approx$ 0.65) [1], a CNN model that uses promoter sequence and 8 features associated with mRNA decay to predict gene expression. For K562, GC-MERGE again outperforms all alternative baseline models (PCC = 0.76). In addition, GC-MERGE performance also exceeds that of Xpresso (PCC $\approx$ 0.71) [1] as well as DeepExpression (PCC = 0.65) [39], a CNN model that uses promoter sequence data as well as spatial information from H3K27ac and YY1 Hi-ChIP experiments. Our model gives better performance (PCC = 0.71) for HUVEC as well. Neither Xpresso nor DeepExpression studied this cell line.

These results strongly suggest that including spatial information can improve gene expression predictive performance over methods solely using local histone mark features as input. We emphasize that this prediction task allows us to model the relationships between the histone marks, 3D structure of the DNA, and gene expression. Therefore, a good performance indicates that the model can leverage the existing data to learn these connections. One of our main goals is to extract these relationships from the model and present GC-MERGE as a hypothesis driving tool for understanding epigenetic regulation.

Interpretation of GC-MERGE highlights long-range interactions and histone modification profiles To determine the underlying biological factors driving the model's predictions, we integrate the GNNExplainer method, designed for classification tasks, into our modeling framework. Adapting our GC-MERGE model to the classification task also resulted in state-of-the-art performance (Supplementary Figure S3) achieving 0.87, 0.88, and 0.85 AUPR scores for GM12878, K562, and HUVEC, respectively. Once trained, we show that our classification model can determine which spatial interactions are

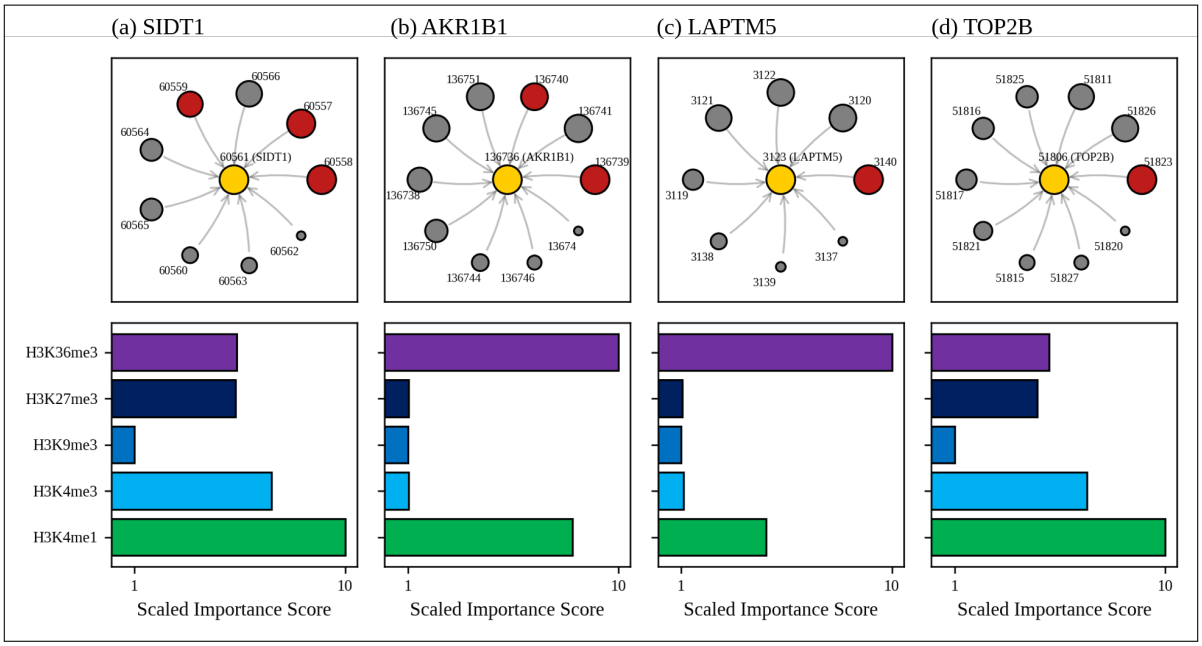

Figure 3: **Model explanations for exemplar genes. Top:** For (a) SIDT1, designated as node 60561 (yellow circle), the subgraph of neighbor nodes is displayed. The size of each neighbor node correlates with its predictive importance as determined by GNNExplainer. Nodes in red denote regions corresponding to known enhancer regions regulating SIDT1 [13] (note that multiple interacting fragments can be assigned to each node, see Supplementary Tables S2 and S3). All other nodes are displayed in gray. Nodes with importance scores corresponding to outliers have been removed for clarity. **Bottom:** The scaled feature importance scores for each of the five core histone marks used in this study are shown in the bar graph. Results also presented for (b) AKR1B1, (c) LAPTM5, and (d) TOP2B.

most critical to a gene's expression and the histone marks that are most important. For GM12878, a lymphoblastoid cell line, we selected four genes: SIDT1, AKR1B1, LAPTM5, and TOP2B as exemplar genes. These genes are among the most highly expressed genes in our data set, and they have also been experimentally shown to be controlled by several long-range promoter-enhancer interactions [13]. To illustrate the validity of our approach, we perform analyses for each of these genes and corroborate our results using previous studies from the literature. Supplementary Table S2 lists the chromosomal coordinates and corresponding node identifiers for each gene.

- **SIDT1** encodes a transmembrane dsRNA-gated channel protein and is part of a larger family of proteins necessary for systemic RNA interference [7, 20]. This gene has also been implicated in chemoresistance to the drug gemcitabine in adenocarcinoma cells [7] and is regulated by at least three chromosomal regions [13, 20]. In Figure 3(a), we show that for SIDT1, the model makes use of all three genomic regions known to have regulatory effects by assigning high importance scores to those nodes (indicated by the size of the node). In addition, we plot the importance scores assigned to the histone marks (node features) that are most important in driving the model's predictions. From the bar graph, it is apparent that H3K4me1 and H3K4me3 are the two most important features in determining the model's prediction. This histone mark profile has been associated with regions flanking transcription start sites (TSS) in highly expressed genes [8, 24].
- **AKR1B1** encodes an enzyme that belongs to the aldo-keto reductase family. It has also been identified as a key player in complications associated with diabetes [5, 20] and is regulated by at least two chromosomal regions [13]. As seen in Figure 3(b),the model strongly bases its predictions

for AKRB1 on both of the regions known to have regulatory effects (location information in Supplementary Table S3). We also show that H3K36me3 and H3K4me1 are the two histone marks with the highest scaled importance scores. This chromatin state signature is correlated with genic enhancers of highly expressed genes [24].

- **LAPTM5** encodes a receptor protein that spans the lysosomal membrane [20]. It is highly expressed in immune cells and plays a role in the downregulation of T and B cell receptors and the upregulation of macrophage cytokine production [10] as well as interacts with at least one regulatory sequence [13]. In Figure 3(c), the genomic region corresponding to the node with the highest scaled importance score has been experimentally shown to interact with LAPTM5 (see Supplementary Table S3) [13] and its histone mark profile is characteristic of genic enhancers [8].
- **TOP2B** encodes DNA topoisomerase II beta, a protein that controls the topological state of DNA during transcription and replication [20]. It transiently breaks and then reforms duplex DNA, relieving torsional stress. Mutations in this enzyme can lead to B cell immunodeficiency [3] and it has been shown to interact with at least two regulatory regions [13]. Figure 3(d) shows that the most important neighbor node has been corrobrated by experiments to have a regulatory role for that gene [13] and its histone mark profile is indicative of regions flanking TSS [8].

To confirm that the node importance scores obtained from GNNExplainer do not merely reflect the relative magnitudes of the Hi-C counts or the distances between genomic regions, we investigated the relationships among the Hi-C counts, genomic distances, and scaled importance scores for all four exemplar genes (Supplementary Figures S4 – S7). We observe that the scaled importance scores do not correspond to the Hi-C counts or the pairwise genomic distances. For example, for SIDT1, the three experimentally validated interacting nodes achieve the highest importance scores ($10, 9.55,$ and 7.73). However, they do not correspond to the regions with the highest Hi-C counts ($154.78, 412.53,$ and 170.55 for each of the three known regulatory regions while the highest count is 602.84). In addition, although they are close to the SIDT1 gene region (40, 20, and 30 kbp away), there are other nodes at the same or closer distances that do not have promoter-enhancer interactions. Therefore, we show that by modeling the histone modifications and the spatial configuration of the genome, GC-MERGE infers connections that could serve as important hypothesis-driving observations for gene regulatory experiments.

5 Discussion

We present GC-MERGE, a graph-based deep learning model, which integrates both local and long-range epigenetic data using a graph convolutional network framework to predict gene expression and explain its drivers. We demonstrate its state-of-the-art performance for the gene expression prediction task, outperforming the baselines on the GM12878, K562, and HUVEC cell lines. We also determine the relative contributions of histone modifications and long-range interactions for four highly expressed genes, showing that our model recapitulates known experimental results in a biologically interpretable manner.

With respect to future work, we plan on applying our framework on additional cell lines as high-quality Hi-C data sets become available. Incorporating other features, such as promoter sequence, would also be natural extensions. One avenue of particular importance would be to develop more robust methods for interpreting GCNs. For example, while the GNNExplainer model is a theoretically sound framework and yields an unbiased estimator for the importance scores of the subgraph nodes and features, there is variation in the interpretation scores generated over multiple runs. Furthermore, with larger GCNs, the optimization function utilized in GNNExplainer is challenging to minimize in practice. For some iterations, the importance scores converge with little differentiation and the method fails to arrive at a compact representation. This may be due to the relatively small penalties the method applies with re-

spect to constraining the optimal size of the mask and the entropy of the distribution. We plan to address this issue in the future by implementing more robust forms of regularization.

In summary, GC-MERGE demonstrates proof-of-principle for using GCNs to predict gene expression using both local epigenetic features and long-range spatial interactions. Interpretation of this model allows us to propose plausible biological explanations of the key regulatory factors driving gene expression as well as provide guidance regarding promising hypotheses and new research directions.

Acknowledgments We are grateful to members of the COBRE-CBHD Computational Biology Core (CBC) at Brown University for helpful discussions and suggestions.

Funding Research reported in this publication was supported by an Institutional Development Award (IDeA) from the National Institute of General Medical Sciences of the National Institutes of Health under grant number P20GM109035.

References

[1] V. Agarwal and J. Shendure. Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks. 31(7):107663.

[2] I. Bica, H. Andrés-Terré, A. Cvejic, and P. Liò. Unsupervised generative and graph representation learning for modelling cell differentiation. 10(1):9790.

[3] L. Broderick, S. Yost, D. Li, M. D. McGeough, et al. Mutations in topoisomerase II result in a b cell immunodeficiency. 10(1):3644.

[4] C. Cheng, K.-K. Yan, K. Y. Yip, J. Rozowsky, R. Alexander, C. Shou, and M. Gerstein. A statistical framework for modeling gene expression using chromatin features and application to modENCODE datasets. 12(2):R15.

[5] K. C. Donaghue, S. H. Margan, A. K. F. Chan, B. Holloway, M. Silink, T. Rangel, and B. Bennetts. The association of aldose reductase gene (AKR1b1) polymorphisms with diabetic neuropathy in adolescents. 22(10):1315–1320.

[6] X. Dong, M. C. Greven, A. Kundaje, S. Djebali, J. B. Brown, C. Cheng, T. R. Gingeras, M. Gerstein, R. Guigó, E. Birney, and Z. Weng. Modeling gene expression using chromatin features in various cellular contexts. 13(9):R53.

[7] M. O. Elhassan, J. Christie, and M. S. Duxbury. *Homo sapiens* systemic RNA interference-defective-1 transmembrane family member 1 (SIDT1) protein mediates contact-dependent small RNA transfer and MicroRNA-21-driven chemoresistance. 287(8):5267–5277.

[8] J. Ernst and M. Kellis. Chromatin-state discovery and genome annotation with ChromHMM. 12(12):2478–2492.

[9] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[10] W. K. Glowacka, P. Alberts, R. Ouchida, J.-Y. Wang, and D. Rotin. LAPTM5 protein is a positive regulator of proinflammatory signaling pathways in macrophages. 287(33):27691–27702.

[11] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc.

[12] B. Jin, C. Gao, X. He, D. Jin, and Y. Li. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–668. ACM.

[13] I. Jung, A. Schmitt, Y. Diao, A. J. Lee, et al. A compendium of promoter-centered long-range chromatin interactions in the human genome. 51(10):1442–1449.

[14] R. Karlic, H.-R. Chung, J. Lasserre, K. Vlahovicek, and M. Vingron. Histone modification levels are predictive for gene expression. 107(7):2926–2931.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[16] D. A. Kleinjan, A. Seawright, A. Schedl, R. A. Quinlan, S. Danes, and V. van Heyningen. Aniridia-associated translocations, dnase hypersensitivity, sequence comparison and transgenic analysis redefine the functional domain of pax6. *Human molecular genetics*, 10(19):2049–2059, 2001.

[17] P. H. L. Krijger and W. de Laat. Regulation of disease-associated gene expression in the 3d genome. 17(12):771–782.

[18] Z. Liu and J. Zhou. Introduction to graph neural networks. 14(2):1–127.

[19] M. R. Mumbach, A. J. Rubin, R. A. Flynn, C. Dai, P. A. Khavari, W. J. Greenleaf, and H. Y. Chang. Hichip: efficient and sensitive analysis of protein-directed genome architecture. *Nature methods*, 13(11):919–922, 2016.

[20] N. C. f. B. I. National Library of Medicine (US). Entrez gene. `https://www.ncbi.nlm.nih.gov/gene/`, 1988-. Accessed: 2020-10-22.

[21] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang. DeepInf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2110–2119. ACM.

[22] S. Rao, M. Huntley, N. Durand, et al. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. 159(7):1665–1680.

[23] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[24] Roadmap Epigenomics Consortium. Integrative analysis of 111 reference human epigenomes. 518(7539):317–330.

[25] M. J. Rowley and V. G. Corces. Organizational principles of 3d genome architecture. 19(12):789–800.

[26] S. Schoenfelder and P. Fraser. Long-range enhancer–promoter contacts in gene expression control. 20(8):437–455.

[27] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.

[28] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[29] R. Singh, J. Lanchantin, G. Robins, and Y. Qi. Deepchrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics*, 32(17):i639–i648, 2016.

[30] R. Singh, J. Lanchantin, A. Sekhon, and Y. Qi. Attend and predict: Understanding gene regulation by selective attention on chromatin. 30:6785–6795.

[31] H. Sobhy, R. Kumar, J. Lewerentz, L. Lizana, and P. Stenberg. Highly interacting regions of the human genome are enriched with enhancers and bound by DNA repair proteins. 9(1):4577.

[32] M. Sun, S. Zhao, C. Gilvary, O. Elemento, J. Zhou, and F. Wang. Graph convolutional networks for computational drug development and discovery. 21(3):919–935.

[33] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

[34] N. L. Van Berkum, E. Lieberman-Aiden, L. Williams, M. Imakaev, A. Gnirke, L. A. Mirny, J. Dekker, and E. S. Lander. Hi-c: a method to study the three-dimensional architecture of genomes. *JoVE (Journal of Visualized Experiments)*, (39):e1869, 2010.

[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[36] F. Yang, K. Fan, D. Song, and H. Lin. Graph-based prediction of protein-protein interactions with attributed signed graph embedding. 21(1):323.

[37] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. Gnnexplainer: Generating explanations for graph neural networks. 2019.

[38] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*, 2019.

[39] W. Zeng, Y. Wang, and R. Jiang. Integrating distal and proximal information to predict gene expression via a densely connected convolutional neural network. page btz562.

Supplementary Information

S1 GC-MERGE model details

S1.1 Model architecture and training

The model architecture is represented in Figure S1. Here, the first layer of the model performs a graph convolution on the initial feature embeddings with an output embedding size of 256, followed by application of ReLU, a non-linear activation function. The second layer of the model performs another graph convolution with the same embedding size of 256 on the transformed representations, again followed by application of ReLU. Next, the output is fed into three successive linear layers of sizes 256, 256, and 2, respectively. A regularization step is performed by using a dropout layer with probability 0.5. The model was trained using ADAM, a stochastic gradient descent algorithm [15]. We used the PyTorch Geometric package [9] to implement our code.

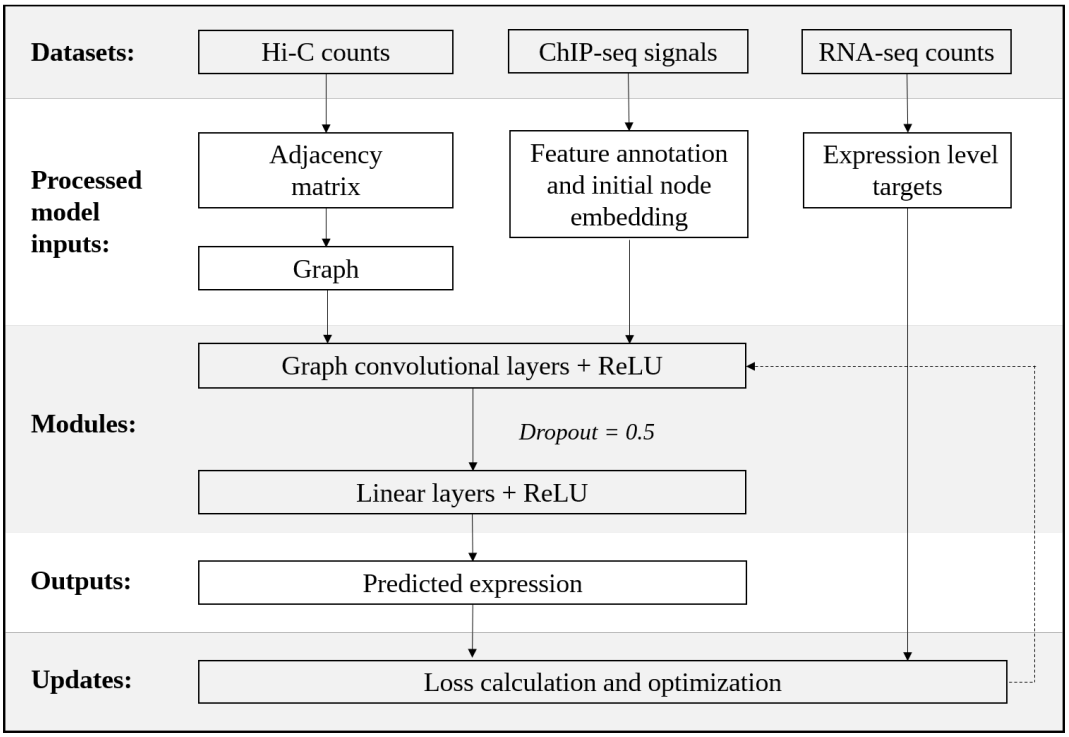

Figure S1: **Overview of the GCNN model architecture.** The datasets used in our model are Hi-C maps, ChIP-seq signals, and RNA-seq counts. A binarized adjacency matrix is produced from the Hi-C maps by subsampling from the Hi-C matrix, such that only the top 10 neighbors of each node are preserved. The nodes in the graph are annotated with features from the ChIP-seq datasets. Two graph convolutions, each followed by ReLU, are performed. The output is fed into a dropout layer (probability = 0.5), followed by a linear module comprised of three dense layers, in which the first two layers are followed by ReLU. For the regression model, the final output represents the base-10 logarithm of the expression level (with a pseudocount of 1). For the classification model, the output is fed through a *Softmax* layer and then the *argmax* is taken to make the final prediction.

Hyperparameter	Values
Number of graph convolutional layers	1, 2
Number of linear layers	1, 2, 3
Graph convolutional layer embedding sizes	64, 128, 256, 384
Linear layer sizes	Keep sizes of all linear layers constant; alternatively, for each subsequent layer, divide size by 2
Activation function after graph convolutional layers	Include; alternatively, do not include

Table S1: **Hyperparameter combinations used for tuning in grid search.** A grid search was conducted by varying the following hyperparameters: number of graph convolutional layers, number of linear layers, embedding size for graph convolutional layers, linear layer sizes, and inclusion/exclusion of activation function after the graph convolutional layers.

S1.2 Hyperparameter tuning

Table S1 details the hyperparameters and the range of values we used to conduct a grid search to determine the optimized model. Specifically, we varied the number of graph convolutional layers, number of linear layers, embedding size for graph convolutional layers, linear layer sizes, and inclusion (or exclusion) of an activation function after the graph convolutional layers. Through earlier iterations of hyperparameter tuning, we also tested the number of neighbors for each node (5 or 10), type of activation functions used for the linear layers of the model (ReLU, LeakyReLU, sigmoid, or tanh), method for accounting for background Hi-C counts, as well as dropout probabilities. Some combinations of hyperparameters were omitted from our grid search because the corresponding model's memory requirements did not fit on the NVIDIA Titan RTX and Quadro RTX GPUs available to us on Brown University's Center for Computation and Visualization (CCV) computing cluster. We recorded the loss curves for the training and validation sets over 1000 epochs, by which time the model began to overfit. In addition, the data was split into sets of 70% for training, 15% for validation, and 15% for testing. The optimal hyperparameters for our final model that also proved to be computationally feasible are as follows: 2 graph convolutional layers, 3 linear layers, graph convolutional layer embedding size of 256, linear layer sizes that match that of the graph convolutional layers, and using an activation function (ReLU) after all graph convolutional layers and all linear layers except for the last.

S2 Additional dataset details and results

S2.1 Hi-C Data preprocessing

For chromosome capture data, we used previously published Hi-C maps at 10 kilobase (kb) resolution for all 22 autosomal chromosomes [22]. We obtained an N x N symmetric matrix, where each row or column corresponds to a 10 kb chromosomal region. Therefore, each bin coordinate (*row, column*) corresponds to the interaction frequency between two respective genomic regions. We applied VC-normalization on the Hi-C maps. In addition, because chromosomal regions located closer together will contact each other more frequently than regions located farther away simply due to chance (rather than due to biologically significant effects), we made an additional adjustment for this background effect. Following Sobhy *et al.* [31], we took the medians of the Hi-C counts for all pairs of interacting regions located the same distance away and used this as a proxy for the background. We subtracted the appropriate median from each Hi-C bin and discarded negative values.

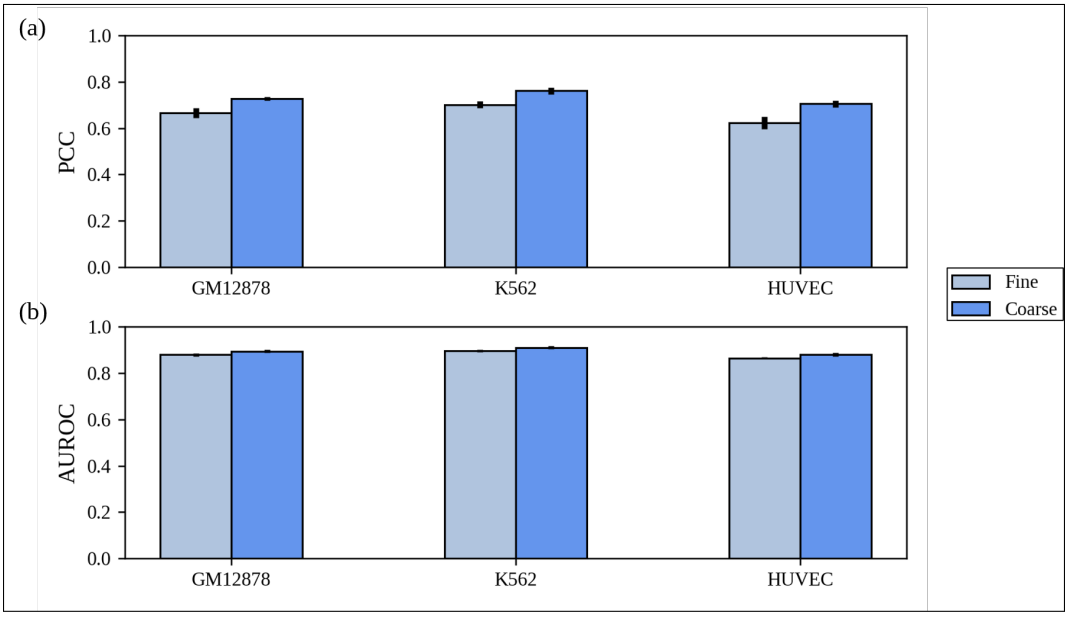

Figure S2: **Comparison of fine-grained versus coarse-grained ChIP-seq signals.** For the coarse-grained resolution, ChIP-seq signals were averaged over the entire Hi-C bin (10000 bp resolution). For the fine-grained resolution, ChIP-seq signals were first averaged over 1000 bp bins and then fed into two embedding linear layers followed by ReLU. The output of these embedding layers was then was used to feature annotate each node. (a) For the regression task, the fine-grained resolution ChIP-seq data produces performance worse than or comparable to the coarse-grained resolution ChIP-seq data as measured by PCC. (b) For the classification task, the fine-grained resolution ChIP-seq data performs slightly worse than or comparable to that of the coarse-grained resolution ChIP-seq data as measured by AUROC.

Gene	Node Identifier	Node Coordinates	Gene Coordinates
SIDT1	60561	chr3:113249241-113259241	chr3:113532296-113629579
AKR1B1	136736	chr7:134253323-134263323	chr7:134127127-134144036
LAPTM5	3123	chr1:31230000-31240000	chr3:31205316-31230667
TOP2B	51806	chr3:25699241-25709241	chr3:25639475-25706398

Table S2: **Node coordinates for all exemplar genes: SIDT1, AKR1B1, LAPTM5, and TOP2B.** For each gene, the second and third columns list the corresponding node identifiers and the chromosome coordinates, respectively. The fourth column lists the gene's actual chromosomal coordinates. Note that the transcription start site was used as the basis for assigning each gene to a node.

Gene	Neighbor Node Identifier	Neighbor Node Coordinates	Interacting Fragment Coordinates
SIDT1	60557	chr3:113209241-113219241	chr3:113251143-113348425
	60558	chr3:113219241-113229241	chr3:113228501-113232053
	60559	chr3:113229241-113239241	chr3:113228501-113232053
	60560	chr3:113239241-113249241	
	60562	chr3:113259241-113269241	
	60563	chr3:113269241-113279241	
	60564	chr3:113279241-113289241	
	60565	chr3:113289241-113299241	
	60566	chr3:113299241-113309241	
AKR1B1	136738	chr7:134273323-134283323	
	136739	chr7:134283323-134293323	chr7:134293046-134298798
	136740	chr7:134293323-134303323	chr7:134293046-134298798
	136741	chr7:134303323-134313323	
	136744	chr7:134333323-134343323	
	136745	chr7:134343323-134353323	
	136746	chr7:134353323-134363323	
	136747	chr7:134363323-134373323	
	136750	chr7:134393323-134403323	
	136751	chr7:134403323-134413323	
LAPTM5	3119	chr1:31190000-31200000	
	3120	chr1:31200000-31210000	
	3121	chr1:31210000-31220000	
	3122	chr1:31220000-31230000	
	3137	chr1:31370000-31380000	
	3138	chr1:31380000-31390000	
	3139	chr1:31390000-31400000	
	3140	chr1:31400000-31410000	chr1:31401583-31405576
TOP2B	51811	chr3:25749241-25759241	
	51815	chr3:25789241-25799241	
	51816	chr3:25799241-25809241	
	51817	chr3:25809241-25819241	
	51820	chr3:25839241-25849241	
	51821	chr3:25849241-25859241	
	51823	chr3:25869241-25879241	chr3:25878006-25881223
	51825	chr3:25889241-25899241	
	51826	chr3:25899241-25909241	
	51827	chr3:25909241-25919241	

Table S3: **Neighbor coordinates for SIDT1, AKR1B1, LAPTM5, and TOP2B.** The second column lists the node identifiers for all neighboring nodes of the relevant gene, including neighboring nodes that contain interacting fragments as well as those that do not. The third column third lists the corresponding chromosome coordinates for the node identifier. The fourth column lists the regulatory fragments that interact with each gene as described in Jung *et al.* [13].

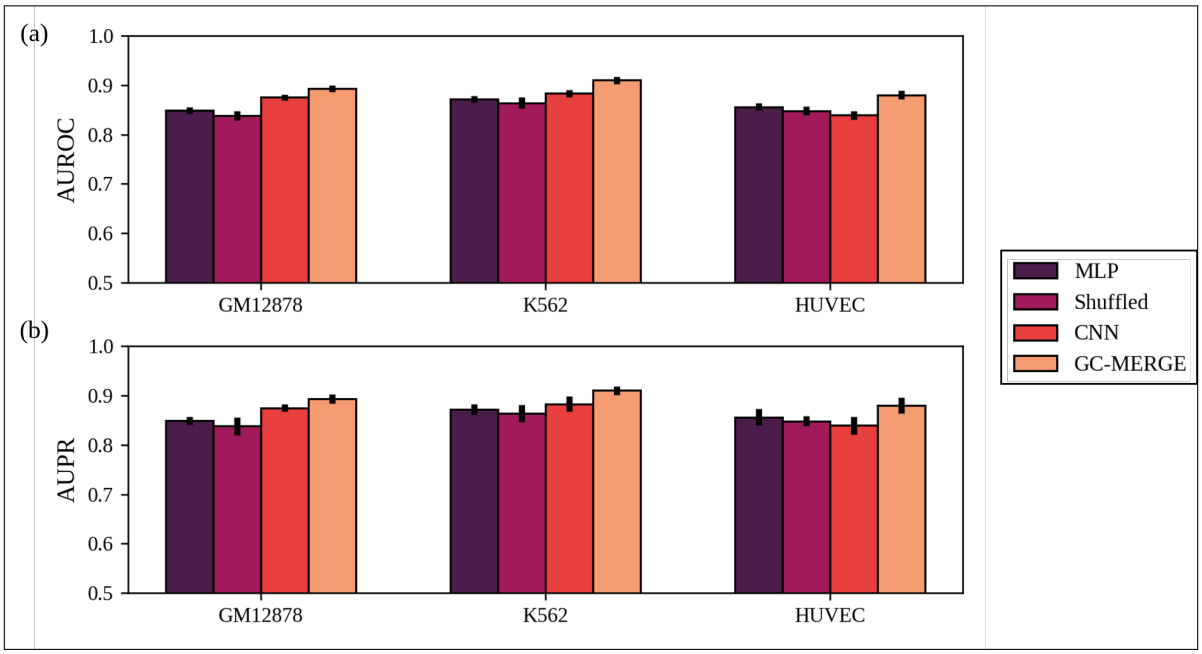

Figure S3: **Comparison of AUROC and AUPR scores for all models.** GC-MERGE gives state-of-the-art performance for classifying genes as on/active or off/inactive. (a) The AUROC metrics for GM12878, K562, and HUVEC were 0.893, 0.910, and 0.880, respectively. For each of these cell lines, GC-MERGE performance exceeded all other baselines. (b) Using the AUPR metric, GC-MERGE obtains scores of 0.865, 0.884, and 0.848 for GM12878, K562, and HUVEC, respectively. As with the AUROC metric, our model's performance was the highest among the baselines. Additionally, our AUROC score for K562 (0.91) is comparable to that reported by Zeng *et al.* [39] (0.91). We could not compare scores for the other two cell lines as they do not provide Hi-ChIP data for the cell line to run their model.

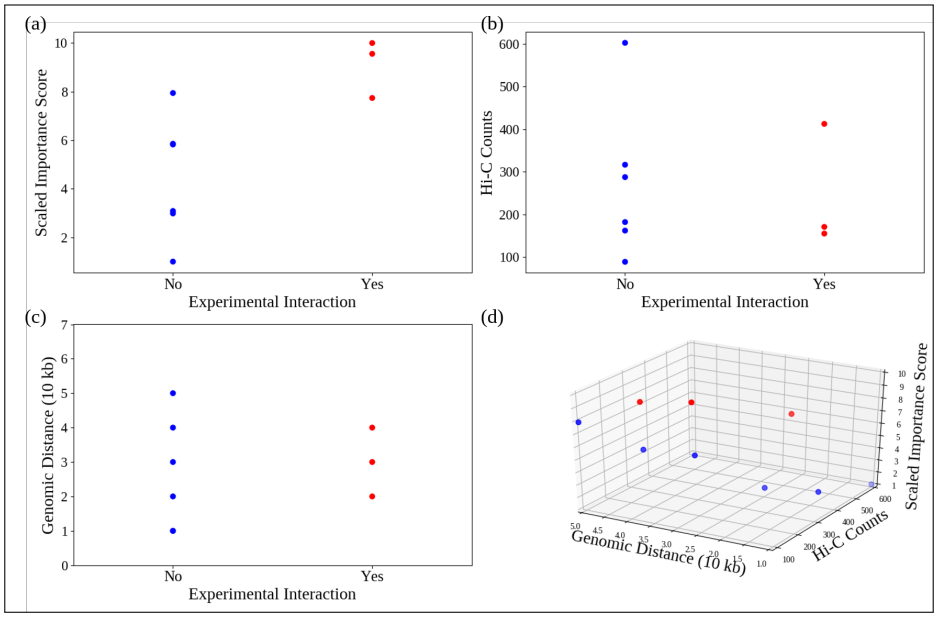

Figure S4: **Relationships among scaled importance scores, genomic distances, and Hi-C counts for all SIDT1 neighbors.** Nodes corresponding to experimentally validated interacting fragments are denoted in red and all others are denoted in blue. (a) Scaled importance score versus experimental interaction. Experimentally validated interacting fragments are ranked higher on average than non-interacting fragments. (b) Hi-C counts versus experimental interaction. Hi-C counts by themselves are not sufficient to explain the presence of experimentally validated interactions. (c) Genomic distance versus experimental interaction. Genomic distance does not correlate with experimentally validated interactions. (d) 3D plot displaying the relationships among scaled importance scores, genomic distances, and Hi-C counts.

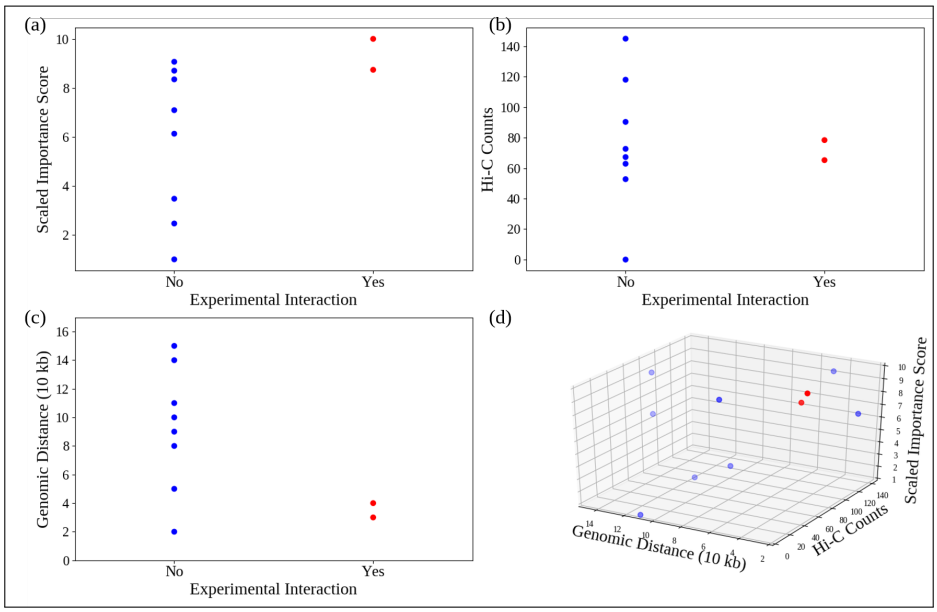

Figure S5: **Relationships among scaled importance scores, genomic distances, and Hi-C counts for all AKR1B1 neighbors.** Nodes corresponding to experimentally validated interacting fragments are denoted in red and all others are denoted in blue. (a) Scaled importance score versus experimental interaction. Experimentally validated interacting fragments are ranked higher on average than non-interacting fragments. (b) Hi-C counts versus experimental interaction. Hi-C counts by themselves are not sufficient to explain the presence of experimentally validated interactions. (c) Genomic distance versus experimental interaction. Genomic distance does not correlate with experimentally validated interactions. (d) 3D plot displaying the relationships among scaled importance scores, genomic distances, and Hi-C counts.

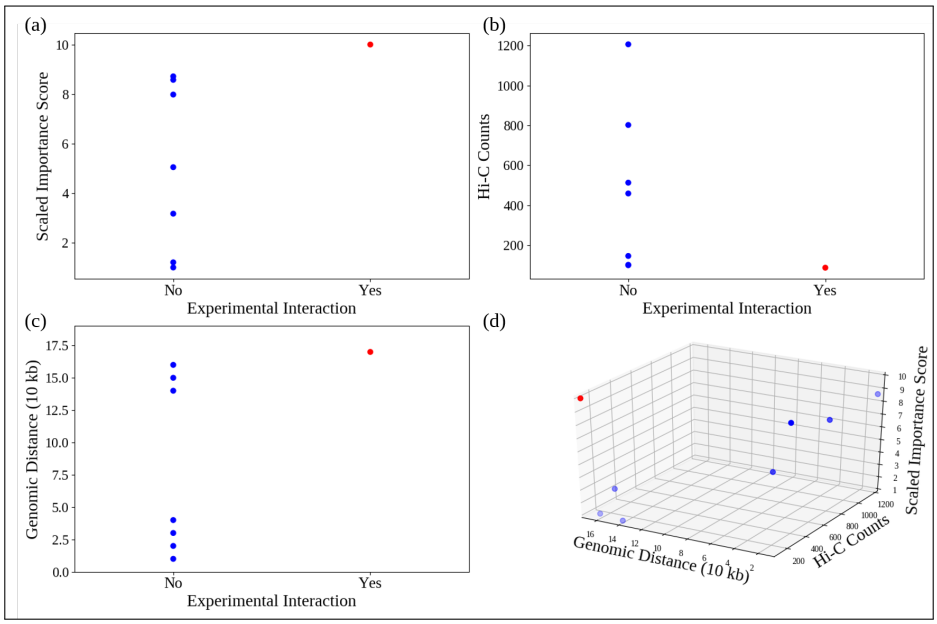

Figure S6: **Relationships among scaled importance scores, genomic distances, and Hi-C counts for all LAPTM5 neighbors.** Nodes corresponding to experimentally validated interacting fragments are denoted in red and all others are denoted in blue. (a) Scaled importance score versus experimental interaction. Experimentally validated interacting fragments are ranked higher on average than non-interacting fragments. (b) Hi-C counts versus experimental interaction. Hi-C counts by themselves are not sufficient to explain the presence of experimentally validated interactions. (c) Genomic distance versus experimental interaction. Genomic distance does not correlate with experimentally validated interactions. (d) 3D plot displaying the relationships among scaled importance scores, genomic distances, and Hi-C counts.

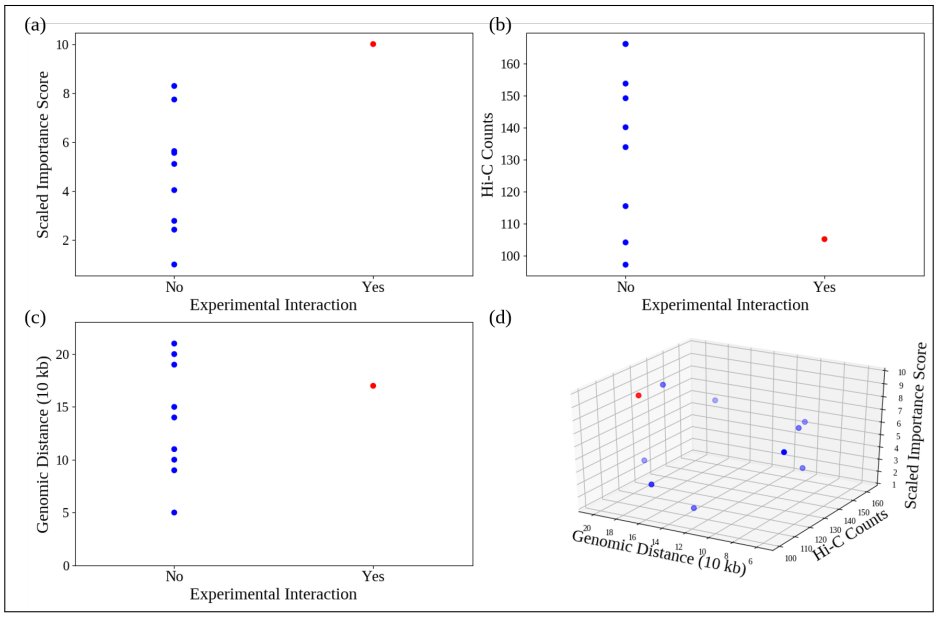

Figure S7: **Relationships among scaled importance scores, genomic distances, and Hi-C counts for all TOP2B neighbors.** Nodes corresponding to experimentally validated interacting fragments are denoted in red and all others are denoted in blue. (a) Scaled importance score versus experimental interaction. Experimentally validated interacting fragments are ranked higher on average than non-interacting fragments. (b) Hi-C counts versus experimental interaction. Hi-C counts by themselves are not sufficient to explain the presence of experimentally validated interactions. (c) Genomic distance versus experimental interaction. Genomic distance does not correlate with experimentally validated interactions. (d) 3D plot displaying the relationships among scaled importance scores, genomic distances, and Hi-C counts.