# Chromatin interaction aware gene regulatory modeling with graph attention networks

Alireza Karbalayghareh[1], Merve Sahin[1], Christina S. Leslie[1*]

[1]Computational and Systems Biology Program, Memorial Sloan Kettering Cancer Center, New York, NY 10065

*Correspondence: cleslie@cbio.mskcc.org

**Abstract**

Linking distal enhancers to genes and modeling their impact on target gene expression are longstanding unresolved problems in regulatory genomics and critical for interpreting non-coding genetic variation. Here we present a new deep learning approach called GraphReg that exploits 3D interactions from chromosome conformation capture assays in order to predict gene expression from 1D epigenomic data or genomic DNA sequence. By using graph attention networks to exploit the connectivity of distal elements and promoters, GraphReg more faithfully models gene regulation and more accurately predicts gene expression levels than dilated convolutional neural networks (CNNs), the current state-of-the-art deep learning approach for this task. Feature attribution used with GraphReg accurately identifies functional enhancers of genes, as validated by CRISPRi-FlowFISH and TAP-seq assays, outperforming both CNNs and the recently proposed Activity-by-Contact model. GraphReg therefore represents an important advance in modeling the regulatory impact of epigenomic and sequence elements.

## Main

Transcriptional gene regulation involves the binding of transcription factors at both promoter and enhancer regions and the physical interaction of these bound complexes via DNA looping. Technological advances in chromosome conformation capture assays such as Hi-C [1] and HiChIP [2] now provide high-resolution data on 3D chromatin interactions, including regulatory interactions between enhancers and promoters. 3D interaction data sets combined with traditional 1D epigenomic profiles, including chromatin accessibility (DNase-seq and ATAC-seq [3]) and histone modifications (via ChIP-seq and CUT&RUN [4]), together map the chromatin state and connectivity of regulatory elements and should

provide rich training data for predictive gene regulatory models, including models that also incorporate the underlying genomic DNA sequence. Ultimately, such models could be used to infer the regulatory function of non-coding genetic variants in a cell type of interest.

Until now, machine learning models for the genome-wide prediction of gene expression in a single cell state, or gene expression changes in a cell state transition, have largely relied on 1D epigenomic data and/or genomic DNA sequence without using 3D genomic architecture. Prior models include regularized linear regression approaches [5, 6] as well as more recent work using convolutional neural networks [7, 8]. However, these previous linear models use a fixed assignment of regulatory elements to genes without incorporating 3D interaction data, while current deep learning models consider relatively local features, such as promoters and at most nearby enhancers, and therefore cannot capture the impact of distal regulatory elements, which can be 1Mb or farther away from gene promoters.

Here we propose a gene regulatory modeling approach called GraphReg that integrates 1D epigenomic data (Epi-GraphReg, **Fig. 1a**) or both epigenomic and DNA sequence data (Seq-GraphReg, **Fig. 1b**) with 3D chromatin interaction data from HiChIP or Hi-C via a graph neural network to predict gene expression. The 1D input data can include any standard epigenomic assays such as histone modification ChIP-seq, transcription factor ChIP-seq, or chromatin accessibility from DNase-seq or ATAC-seq. GraphReg models use convolutional neural network (CNN) layers to learn local representations from 1D inputs, followed by graph attention network (GAT) layers to propagate these representations over the 3D interaction graph, in order to predict gene expression (CAGE-seq) across genomic positions (bins). GraphReg is trained to predict CAGE-seq [9], a tag-based protocol for gene expression measurement and transcription start site (TSS) mapping, since it quantifies promoter output and does not depend on transcript length.

Our motivation for proposing GraphReg is two-fold. First we aim to improve the accuracy of predictive gene regulatory models by leveraging 3D genomic architecture to incorporate distal enhancer elements, and we demonstrate that GraphReg outperforms baseline CNN models for prediction of CAGE output. Second, we seek to interpret the model to assess the functional importance of distal enhancers for regulation of specific target genes. To this end, we use feature attribution methods on trained GraphReg models and show that we obtain a better ranking of functional enhancers as validated by CRISPRi-FlowFISH [10] and TAP-seq [11] than baseline CNN models and the recently proposed Activity-by-Contact (ABC) model [10].

## GraphReg: a deep graph neural network model for interaction aware gene expression prediction

In our experiments, we used a minimal set of 1D epigenomic data relevant to gene regulation: DNase-seq as a measure of chromatin accessibility, H3K4me3 ChIP-seq for promoter activity, and H3K27ac ChIP-seq for enhancer activity. For 3D interaction data, we found that H3K27ac HiChIP data provided an advantage over Hi-C due to its greater sensitivity for regulatory interactions, such as enhancer-enhancer (E-E) and enhancer-promoter (E-P) loops, rather than the structural (CTCF- and cohesin-mediated) interactions enriched in Hi-C. Since Epi-GraphReg is trained on 1D epigenomic and 3D interaction input data in a given cell type, in order to predict gene expression output in the same cell type, it learns a regulatory model that can generalize to other contexts. That is, given cell-type specific input data in a new cell type, the trained Epi-GraphReg model can predict gene expresion in that cell type. In this sense, the Epi-GraphReg model is *cell-type agnostic*. Seq-GraphReg uses DNA sequence as the input and performs multi-task learning, predicting DNase, H3K4me3, and H3K27ac in the CNN block and predicting CAGE-seq after the GAT block. Seq-GraphReg is therefore a *cell-type restricted* model because it learns the transcription factor (TF) binding signals specific to the trained cell type and consequently cannot generalize to another cell type. However, Seq-GraphReg can capture cell-type specific TF binding motifs in enhancer and promoter regions and potentially predict the impact of DNA sequence alterations on target gene expression.

We bin the epigenomic data (H3K4me3, K3K27ac, and DNase) at 100bp resolution. We process the 5Kb resolution H3K27ac HiChIP contact matrices with the HiC-DC+ package [12, 13] to identify significant interactions of genomic distance up to 2Mb, and use the 3D interactions satisfying a (permissive) $q$-value of 0.1 to define a graph on 5Kb genomic bins. For consistency with the HiChIP data, we also bin CAGE-seq at 5Kb resolution. In each batch of training, we extract 6Mb genomic regions and use the corresponding DNA sequences, epigenomic data, and interaction graphs from HiChIP as inputs, and the corresponding CAGE-seq as the output, for the model. For the next batch, we shift the entire 6Mb region by 2Mb and repeat this to cover the training chromosomes.

For Epi-GraphReg (**Fig. 1a**), each epigenomic signal track is treated as a different channel and fed to 1D CNN layers followed by max-pooling and ReLU activation functions. This produces latent local representations at 5Kb resolution, which serve as the node features for the HiChIP-derived interaction graphs. We then use several graph attention network (GAT) layers to propagate these local representations over the interaction graphs, so that promoter regions are influenced by the representations of their candidate distal enhancers. Finally, we predict the CAGE values (promoter activities) by a fully connected (FC) layer.

For Seq-GraphReg (**Fig. 1b**), we feed one-hot-coded genomic DNA sequences to a series of 1D CNN, max-pooling, and ReLU layers, similar to Epi-GraphReg. The resulting latent local representations are then supplied to two blocks corresponding to different prediction tasks: a GAT block, similar to the Epi-GraphReg model, uses several GAT layers followed by a FC layer to predict the CAGE values; and a dilated CNN block, containing several dilated CNN layers whose dilated rate is multiplied by two each layer, to predict the 1D epigenomic data. Dilated CNNs have been used previously to increase the receptive field of CNN layers in deep learning models that predict epigenomic data from DNA sequence, such as Basenji [7], BPNet [14], and Xpresso [15]. Therefore, Seq-GraphReg follows a multi-task learning approach to find more meaningful latent local representations to provide to the GAT block.

GATs have the advantage over other graph neural networks in that they learn to weight edges in the graph from node features. In this way, GATs weigh E-E and E-P interactions, based on the features learned in the promoter and enhancer bins, in order to predict CAGE values more accurately. We use different attention heads (**Fig. 1a,b**, **Methods**) in each GAT layer to enhance the flexibility of the model to learn distinct E-E and E-P interaction weights and to improve the prediction accuracy by integrating them together. **Fig. 1c** shows an example of a 6Mb region (11Mb-17Mb) of chromosome 19 in K562 cells with the corresponding true CAGE output, 1D epigenomic inputs, H3K27ac HiChIP interaction graph, and the predicted CAGE signals using Epi-GraphReg and Seq-GraphReg and their CNN counterparts, which we denote as Epi-CNN and Seq-CNN, respectively. The dashed middle 2Mb region in **Fig. 1c** indicates where we compute the loss in training and evaluate CAGE predictions. The inputs and outputs of Epi-CNN and Seq-CNN models are the same as those of Epi-GraphReg and Seq-GraphReg, respectively, with the exception that they use dilated CNN layers instead of GAT layers and do not use any 3D interaction data.

## GraphReg accurately predicts gene expression by using 3D interaction data

We trained GraphReg models for two ENCODE human cell lines, GM12878 and K562, and for mouse embryonic stem cells (mESC), for which complete 1D epigenomic data and 3D interaction data are available. To evaluate, we performed cross-validation experiments where we held out 2 chromosomes for testing, used 2 chromosomes for validation, and trained on all remaining autosomal chromosomes (**Methods**). Although our model predicts the CAGE signal at all genomic bins, we focused first on predictions for GENCODE-annotated TSS bins, where the CAGE signal can be non-zero. **Fig. 2a,b** and **Fig. S1a-c** show the CAGE prediction results for GM12878, K562, and mESC in their corresponding held-out validation and test chromosomes. We computed the Pearson correlation ($R$) of log-normalized predicted and true CAGE values over 10 cross-validation runs for four gene sets (**Fig. 2a,b**, **Fig. S1a,b**): all genes ($A$); expressed genes (defined as CAGE signal $\geq$ 5 TPM, $B$); expressed genes with at least one

E-P interaction ($C$); and expressed genes with at least five E-P interactions ($D$). We also reported the negative log-likelihood values ($NLL$), based on the Poisson distribution, of the predicted CAGE signals. In all cases, GraphReg models outperform the corresponding CNN models, achieving higher $R$ and lower $NLL$ (**Fig. 2a,b**, **Fig. S1a-c**). After restricting to expressed genes or to expressed genes with E-P interactions in the graph, the problem gets harder and the prediction improvement of GraphReg over CNN models increases. Epigenome-based models (Epi-GraphReg and Epi-CNN) have higher prediction accuracy than the sequence-based models (Seq-GraphReg and Seq-CNN) (**Fig. 2a,b**, **Fig. S1a-c**) since the epigenomic data are highly correlated with the CAGE output.

**Fig. S1c** shows an example scatter plot of log-normalized predicted versus true CAGE values of GraphReg and CNN models across all TSS bins on held-out chromosomes in mESC. Here TSS bins are color-coded by their number of HiChIP interactions ($n$). Analogous to our previous notion of *gene regulatory complexity* [5], we refer to genes with $n = 0$ as "simple genes" and those with $n > 0$ (at least one HiChIP interaction) as "complex genes". We hypothesized that by exploiting 3D interaction data, GraphReg would better model the regulation of complex genes. Indeed, Epi-GraphReg yields $R = 0.878$ and $NLL = 640.5$ for complex genes, as compared to Epi-CNN with $R = 0.859$ and $NLL = 827.0$ (**Fig. S1c**). Similarly for the sequence models, Seq-GraphReg yields $R = 0.757$ and $NLL = 917.5$ on complex genes, outperforming Seq-CNN with $R = 0.709$ and $NLL = 1197.0$ (**Fig. S1c**).

Notably, the extent of improvement of Seq-GraphReg over Seq-CNN is usually higher than that of the Epi-GraphReg over Epi-CNN (**Fig. 2a,b**, **Fig. S1c**), suggesting the importance of 3D interaction data for integrating distal regulatory elements and improving prediction accuracy when predicting from DNA sequence, especially for more complex genes with many E-P interactions (**Fig. S1a-c**). In particular, Seq-CNN models predict lower values for some highly expressed genes with high HiChIP interactions, while Seq-GraphReg can more accurately predict these higher expression values (**Fig. S1c**).

GraphReg's improved performance over CNNs can also be witnessed by calculating the log-fold change between CAGE predictions for two different cell types (**Fig. 2a,b**), and by comparing to true log-fold changes using Pearson correlation ($R$) and mean squared error ($MSE$). Evaluation of predicted log-fold changes in GM12878 versus K562 on held-out chromosomes in 10 cross-validation runs (**Fig. 2c**) confirmed the improvement of Epi-GraphReg and Seq-GraphReg over Epi-CNN and Seq-CNN, respectively. **Fig. 2c** shows example scatter plots of predicted versus true log-fold change in TSS bins on held-out chromosomes, color-coded by the minimum number of HiChIP interactions over the two cell types ($m$). For genes with $m > 0$, Epi-GraphReg achieves $R = 0.627$ and $MSE = 2.38$, compared to $R = 0.540$ and $MSE = 3.424$ using Epi-CNN. Even more striking for sequence-based models, for genes with $m > 0$, Seq-GraphReg achieves $R = 0.440$ and $MSE = 3.506$, respectively, compared to very poor $R = 0.031$ and $MSE = 5.355$ for Seq-CNN. Overall, these results confirm that using 3D chromatin interactions in

GraphReg leads to improved gene expression prediction. In particular, GraphReg outperforms dilated CNN models for complex genes that have at least one 3D interaction.

We also investigated how well the cell-type agnostic Epi-GraphReg and Epi-CNN models can generalize from one cell type to another. **Fig. 2d** shows the box plots of Pearson correlation ($R$) and loss ($NLL$) when we train on one cell type (K562 or GM12878) and test on the other, using 10 cross-validation runs with distinct sets of validation chromosomes (held out from training and used to assess performance in the test cell type). For the more complex genes (sets $C$ and $D$), Epi-GraphReg significantly outperforms Epi-CNN ($p < 0.05$, Wilcoxon rank test) for generalization from K562 to GM12878 and vice versa. Example scatter plots for the cross-cell-type generalization tasks show the improvement of Epi-GraphReg over Epi-CNN (**Fig. 2d**): when training on K562 and testing on GM12878, Epi-GraphReg attains $R = 0.826$ and $NLL = 241.1$, whereas Epi-CNN yields $R = 0.789$ and $NLL = 386.0$ for the more complex genes ($n > 0$); similarly, when training on GM12878 and testing on K562, Epi-GraphReg attains $R = 0.746$ and $NLL = 245.1$ compared to $R = 0.647$ and $NLL = 440.2$ for Epi-CNN on the more complex genes ($n > 0$). In particular, Epi-CNN-predicted values for more complex genes show greater dispersion, while the corresponding predictions for Epi-GraphReg have less variance (**Fig. 2d**).

### GraphReg accurately identifies functional enhancers of genes

Thanks to the feature attribution methods for machine learning models – e.g. saliency maps (gradient-by-input), DeepLIFT [16], DeepSHAP [17], and Integrated Gradients [18] – it is possible to derive the important input features for the prediction of a specific output. Since GraphReg allows each gene to be influenced by its potential enhancers through interactions in the H3K27ac HiChIP graph, we hypothesized that feature attribution analysis would allow the identification of functional distal enhancers, i.e. those that contribute to the regulation of target genes. CRISPRi-FlowFISH [10] is a recent enhancer screening approach that uses KRAB-dCas9 interference with candidate enhancer regions in pooled fashion and RNA FISH against a gene of interest as readout; this enables estimation of the effect size of perturbing each candidate enhancer on target gene expression. The developers of FlowFISH introduced a score called Activity-by-Contact (ABC) for finding and ranking enhancers of a gene, which is considered the current state-of-the-art for this problem. The ABC score is the product of "activity", defined as the geometric mean of DNase and H3K27ac signal in each candidate enhancer region, and "contact", defined by the KR-normalized Hi-C signal between the target promoter and the candidate enhancer, normalized so that the ABC scores of all candidate enhancers up to 5Mb from the gene sum to one.

To determine functional enhancers for genes, we used two feature attribution methods: the saliency map (gradient-by-input) and DeepSHAP [17]. For the epigenome-based models (Epi-GraphReg and Epi-

CNN), as the inputs are binned at 100bp resolution, both saliency and DeepSHAP scores are computed at 100bp resolution as well. For the sequence-based models (Seq-GraphReg and Seq-CNN), the feature attribution methods can provide scores at nucleotide resolution. However, for enhancer scoring and validation with FlowFISH analysis, as suggested in Basenji [7], we derived the saliency maps as the dot product of the 100bp bin representations (after the CNN layers and before the dilated CNN layers) and the gradient of the model prediction for the gene with respect to those bin representations. **Fig. 3a** shows the distribution of area under the precision-recall curve (auPR) values for 19 genes from the K562 FlowFISH dataset with more than 10 candidate enhancers for all models. The overall precision-recall curve across 2574 enhancer-gene (E-G) pairs for all 19 genes is shown in **Fig. 3b**. These results confirm that feature attribution applied to GraphReg models more accurately identifies the functional enhancers of the genes from a pool of candidate enhancers than the ABC score or feature attribution on the corresponding CNN models (**Fig. 3a,b**). The highest auPR among GraphReg models was 0.4236 (Epi-GraphReg with saliency), outperforming the best CNN model (auPR 0.366 for Epi-CNN with saliency) and ABC (auPR 0.364).

To assess the ability of models to find distal enhancers, we examined the example of *MYC* and considered only candidate enhancers more than 10Kb from TSS. There are 200 such candidate distal enhancer-gene (DE-G) pairs for *MYC*, including 8 true functional enhancers. Precision-recall analysis for *MYC* (**Fig. S2a**) found the highest auPR of GraphReg models to be 0.576 (Epi-GraphReg with DeepSHAP), strongly outperforming the best CNN model (auPR 0.0792, Epi-CNN with DeepSHAP) as well as ABC (auPR 0.3567). Visualization of the feature attribution scores gives insight into how 3D information allows GraphReg to access distal regulatory information that dilated CNNs cannot exploit, despite their large receptive field. **Fig. 3e** shows a 2.5Mb genomic region containing the *MYC* locus, with the corresponding epigenomic data, HiChIP graph edges, and true and predicted CAGE signals by GraphReg and CNN models in K562 cells, together with DeepSHAP and saliency scores of the four models for *MYC*, and ABC scores and FlowFISH results for *MYC*. Large negative scores in the FlowFISH track indicate true functional enhancers for *MYC*. Feature attribution tracks in **Fig. 3e** show that GraphReg models are able to capture the most distal enhancers of *MYC* (green boxes), about 2Mb away from its TSS, while the CNN models fail to capture these enhancers and produce false negatives (red boxes).

We further evaluated GraphReg using recent chromosome-wide enhancer screening data from TAP-seq (targeted perturb-seq) for chromosomes 8 and 11 in K562 cells [11]. We restricted to the 35 genes with at least one functional enhancer, considering all screened enhancers whose distance from the target gene TSS is less than 2Mb. **Fig. 3c** shows the distribution of auPR for these 35 genes. Similar to CRISPRi-FlowFISH results (**Fig. 3a**), GraphReg models have better median auPR than ABC scores or the CNN models using either DeepSHAP or saliency as the feature attribution method. Pooling all the E-G pair

of these 35 genes (4423 E-G pairs in total), precision recall analysis (**Fig. 3d**) shows that the highest auPR for the GraphReg models is 0.435 (Epi-GraphReg with saliency), outperforming the best CNN model (auPR 0.3936) and ABC (auPR 0.395).

As another example of distal enhancer discovery in the TAP-seq dataset, we considered screened enhancers more than 20Kb from TSS of *IFITM1* (79 distal DE-G pairs, 5 true functional enhancers). **Fig. S2b** shows precision-recall analysis for *IFITM1*, where the highest auPR among the GraphReg models is 0.894 (Epi-GraphReg with saliency), compared to 0.635 for the best CNN model (Epi-CNN with DeepSHAP) and 0.751 for ABC. **Fig. S2c** displays a 250Kb genomic region containing the *IFITM1* locus, with the corresponding K562 epigenomic data, HiChIP graph edges, true and predicted CAGE signals by GraphReg and CNN models, feature attribution scores for *IFITM1*, ABC scores for *IFITM1*, and TAP-seq results for *IFITM1*. Large negative scores in the TAP-seq track indicate true functional enhancers of *IFITM1*. Here all GraphReg, CNN, and ABC models are able to capture the distal enhancers of *IFITM1* (green boxes), but the CNN models also capture many non-functional enhancers as false positives (red boxes).

We have shown that GraphReg more effectively models the impact of distal enhancers on gene expression than 1D dilated CNNs, which represent the current state-of-the-art machine learning models in regulatory genomics. While dilated CNNs have a large receptive field, our feature attribution analyses show that only relatively promoter-proximal information influences gene expression prediction in these models. By contrast, GraphReg exploits 3D chromatin interactions to access distal information up to 2Mb from the gene promoter. While we currently use GraphReg to predict bulk CAGE-seq counts with Poisson loss, we envision extensions to single cell multi-omic data sets – training at the pseudo-bulk level where signals are aggregated over cell clusters or meta-cells, or even at the single-cell level. Here (aggregated) scRNA-seq gives a tag-based output signal similar to CAGE-seq. Therefore, we anticipate that GraphReg models will have broad applicability for interpreting the function of epigenomic and genomic variation on gene expression.

# Methods

### CNN layers for learning local representations of 1D data

The inputs to the first layer of the Epi-GraphReg and Seq-GraphReg models are 1D epigenomic data and genomic DNA sequence, respectively. Regardless of the 1D input type, we use several CNN layers to find local representations of 5Kb bins of the genome. We consider genomic regions of 6Mb in length as our input; hence, we have vectors of size $N = 1200$ after the CNN layers, representing 5Kb bins over

the 6Mb region. We define the set of local representations by $H = \{h_1, h_2, \cdots, h_N\}$, where $h_i \in \mathbb{R}^F$ and $F = 64$ is the number of CNN filters at the last layer of CNN block. Then $H$ is given to the next block, which is a graph neural network (GNN) [19, 20, 21], where $h_i$ is the node feature of node $i$.

### Graph attention networks for integration of 3D data

We employ the graphs extracted from H3K27ac HiChIP data in order to capture gene regulatory interactions and predict gene expression (CAGE-seq). We processed HiChIP data sets at 5Kb resolution using HiC-DC+ [12, 13] and kept only the significant interactions having $q$-values less than 0.1. The nodes in the extracted graphs therefore represent 5Kb genomic bins and the edges indicate the most significant interactions between bins. Each 6Mb genomic input region thus corresponds to a graph with $N = 1200$ nodes whose features $H = \{h_1, h_2, \cdots, h_N\}$ have been learned in the previous CNN block.

The graph attention network (GAT) block receives a graph $G = (V, E)$ and a set of node features $H^t = \{h_1^t, h_2^t, \cdots, h_N^t\}$ from the previous layer $t$ and outputs an updated set of node features $H^{t+1} = \{h_1^{t+1}, h_2^{t+1}, \cdots, h_N^{t+1}\}$. Each GAT layer uses two weight matrices: $W_p^t \in \mathbb{R}^{F' \times F}$ for promoters (or self nodes) and $W_e^t \in \mathbb{R}^{F' \times F}$ for enhancers (or neighbor nodes). Note that here, unlike previous graph neural networks [21], we do not include self-loops in the graph $G$. We have decoupled self-loops and neighbor-loops due to the differing role of promoters and enhancers in the model. By only including the neighbor-loops in the graph $G$, the model focuses on distal enhancers that cannot be captured in local models such as CNNs.

We define the self-attention mechanism for the nodes $i$ and $j$ at layer $t$ as

$$\beta_{i,j}^t = \sigma\left((a_p^t)^T W_p^t h_i^t + (a_e^t)^T W_e^t h_j^t\right), \tag{1}$$

where $a_p^t \in \mathbb{R}^{F'}$ and $a_e^t \in \mathbb{R}^{F'}$ are two weight vectors, $\sigma(.)$ is the sigmoid function, $\beta_{i,j}^t \in \mathbb{R}$ is the attention weight from node $j$ to node $i$ at layer $t$, and $T$ denotes the transpose of a vector. By using a sigmoid function instead of a conventional softmax function, we give extra freedom to the model to discard edges that are not related to enhancers. We also account for the cardinality of the nodes by defining $\alpha_i^t = \sigma\left(a^t \sqrt{|\mathcal{N}_i|} + b^t\right)$, where $a^t \in \mathbb{R}^{F'}$ and $b^t \in \mathbb{R}^{F'}$ are two weight vectors. Finally, we define the updates of the node features at the next layer as

$$h_i^{t+1} = f\left(\alpha_i^t \circ \left(W_p^t h_i^t + \sum_{j \in \mathcal{N}_i} \beta_{i,j}^t W_e^t h_j^t\right)\right), \tag{2}$$

where $f$ is a nonlinearity function and $\circ$ is the element-wise product, $\mathcal{N}_i$ is the set of neighbors of node $i$ (not including itself), $|\mathcal{N}_i|$ is the number of neighbors of node $i$. We use an exponential linear unit

(ELU) for $f$. As in [21], we use $K$ heads and concatenate the features as

$$h_i^{t+1} = \|_{k=1}^{K} f\left(\alpha_i^{t,k} \circ \left(W_p^{t,k} h_i^t + \sum_{j \in \mathcal{N}_i} \beta_{i,j}^{t,k} W_e^{t,k} h_j^t\right)\right), \tag{3}$$

where $\|_{k=1}^{K}$ means concatenation of $K$ independent heads. Therefore, $h_i^{t+1}$ will have $KF'$ features.

### Poisson regression

After the GAT layers, the last layer is a fully connected layer with exponential nonlinearity in order to predict the CAGE-seq signal in each bin. As the CAGE values are counts, we used Poisson regression, meaning that the expected value of each (CAGE-seq TSS) output given the inputs is the mean of a Poisson distribution.

For the Epi-GraphReg model, we let $X$ denote the 1D epigenomic inputs for a 6Mb region, $G$ the corresponding graph, $Y = [y_1, \cdots, y_N]$ the observed CAGE signal across 5Kb bins, and $f_i^\theta(X, G)$ the predicted CAGE signal for the bin $i$, where $\theta$ represents parameters of the model. Now we assume that $y_i|X, G \sim \text{Poisson}(\lambda_i)$, where $\lambda_i = f_i^\theta(X, G)$ and $E(Y|X, G) = f^\theta(X, G)$. Hence, the loss function is the negative log-likelihood of the Poisson distribution,

$$L_E = \frac{3}{N} \sum_{i=N/3}^{2N/3} \Big(\log \Gamma(y_i + 1) + f_i^\theta(X, G) - y_i \log f_i^\theta(X, G)\Big), \tag{4}$$

where $\Gamma(.)$ is the gamma function. Note that we train our model for the middle one third region (2Mb or middle $N/3$ bins) instead of the whole 6Mb region (all $N$ bins) in each batch. Since we include HiChIP interactions of genomic distance up to 2Mb, promoters in the middle 2Mb region can see the effects of distal enhancers in the full 6Mb input region. In test phase we also restrict to predictions on the middle 2Mb regions. We shift input regions by 2Mb across genome, so there is no overlap between predictions in two different batches.

For the Seq-GraphReg model, we also predict the 1D epigenomic data with a resolution of 100bp after the dilated CNN layers from DNA sequence. Here we let $X$ denote the DNA sequence of the 6Mb input region, $G$ the corresponding graph, $Y^e = [y_1^e, \cdots, y_{N'}^e]$ the observed 100bp epigenomic inputs (over $N' = 60000$ bins), and $g_i^e(X)$ the predicted epigenomic signal $e$ for the bin $i$. The loss for each epigenomic signal $e$ is given by

$$L_S^e = \frac{1}{N'} \sum_{i=1}^{N'} \Big(\log \Gamma(y_i^e + 1) + g_e^\theta(X) - y_i^e \log g_i^e(X)\Big), \tag{5}$$

and the CAGE loss $L_S^{cage}$ is the same as (4). The final loss function to be minimized is the summation

of the CAGE loss and the epigenomic losses,

$$L_S = \lambda L_S^{cage} + (1-\lambda)\sum_e L_S^e, \tag{6}$$

where $\lambda$ is a hyperparameter to control the weights of the CAGE loss versus the epigenomic losses. We use a sigmoid-like schedule for $\lambda$ given by $\lambda = \max(.01, \frac{1}{1+\exp -(epoch-10)})$, which starts with a small value $\lambda = 0.01$ and approaches a value close to one at higher epochs. This schedule lets the model to first learn meaningful motif representations from DNA sequences in order to predict the epigenomic data and then puts higher weights on predicting the CAGE signals from those motif representations at 100bp resolution.

To improve prediction accuracy in sequence-based models, it can be preferable to first train the CNN and dilated CNN layers to predict the epigenomic data from the DNA sequence inputs (first column in **Fig. 1b**) and then train the GAT block to predict CAGE values. However, for feature attribution at nucleotide level, it is preferable to train the end-to-end model to be able to backpropagate the output CAGE gradients back to the input DNA nucleotides

### Model architecture and training

For Epi-GraphReg and Seq-GraphReg we used respectively 3 GAT layers and 4 heads in each layer. For Epi-CNN and Seq-CNN we used 8 dilated CNN layers to increase their receptive field up to 2.5Mb. For each of GM12878 and K562, we held out chromosomes $i$ and $i+10$ for validation and chromosomes $i+1$ and $i+11$ for test, $i = 1, \cdots, 10$, and trained on all remaining chromosomes except X and Y. For mESC, we held out chromosomes $i$ and $(i+10 \bmod 20) + \text{sign}(i \geq 10)$ for validation and chromosomes $i+1$ and $(i+11 \bmod 20) + \text{sign}(i \geq 9)$ for test, $i = 1, \cdots, 10$, and trained on all remaining chromosomes except X and Y.

## References


[1] Erez Lieberman-Aiden, Nynke L Van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R Lajoie, Peter J Sabo, Michael O Dorschner, et al. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science*, 326(5950):289–293, 2009.

[2] Maxwell R Mumbach, Adam J Rubin, Ryan A Flynn, Chao Dai, Paul A Khavari, William J Greenleaf, and Howard Y Chang. HiChIP: efficient and sensitive analysis of protein-directed genome architecture. *Nature Methods*, 13(11):919–922, 2016.

[3] Jason D Buenrostro, Beijing Wu, Howard Y Chang, and William J Greenleaf. ATAC-seq: a method for assaying chromatin accessibility genome-wide. *Current Protocols in Molecular Biology*, 109(1):21–29, 2015.

[4] Peter J Skene and Steven Henikoff. An efficient targeted nuclease strategy for high-resolution mapping of DNA binding sites. *eLife*, 6:e21856, 2017.

[5] Alvaro J González, Manu Setty, and Christina S Leslie. Early enhancer establishment and regulatory locus complexity shape transcriptional programs in hematopoietic differentiation. *Nature Genetics*, 47(11):1249–1259, 2015.

[6] Hatice U Osmanbeyoglu, Fumiko Shimizu, Angela Rynne-Vidal, Direna Alonso-Curbelo, Hsuan-An Chen, Hannah Y Wen, Tsz-Lun Yeung, Petar Jelinic, Pedram Razavi, Scott W Lowe, et al. Chromatin-informed inference of transcriptional programs in gynecologic and basal breast cancers. *Nature Communications*, 10(1):1–12, 2019.

[7] David R Kelley, Yakir A Reshef, Maxwell Bileschi, David Belanger, Cory Y McLean, and Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Research*, 28(5):739–750, 2018.

[8] Ritambhara Singh, Jack Lanchantin, Gabriel Robins, and Yanjun Qi. DeepChrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics*, 32(17):i639–i648, 08 2016.

[9] Toshiyuki Shiraki, Shinji Kondo, Shintaro Katayama, Kazunori Waki, Takeya Kasukawa, Hideya Kawaji, Rimantas Kodzius, Akira Watahiki, Mari Nakamura, Takahiro Arakawa, et al. Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. *Proceedings of the National Academy of Sciences*, 100(26):15776–15781, 2003.

[10] Charles P Fulco, Joseph Nasser, Thouis R Jones, Glen Munson, Drew T Bergman, Vidya Subramanian, Sharon R Grossman, Rockwell Anyoha, Benjamin R Doughty, Tejal A Patwardhan, et al. Activity-by-contact model of enhancer–promoter regulation from thousands of CRISPR perturbations. *Nature Genetics*, 51(12):1664–1669, 2019.

[11] Daniel Schraivogel, Andreas R Gschwind, Jennifer H Milbank, Daniel R Leonce, Petra Jakob, Lukas Mathur, Jan O Korbel, Christoph A Merten, Lars Velten, and Lars M Steinmetz. Targeted perturb-seq enables genome-scale genetic screens in single cells. *Nature methods*, 17(6):629–635, 2020.

[12] Mark Carty, Lee Zamparo, Merve Sahin, Alvaro González, Raphael Pelossof, Olivier Elemento, and Christina S Leslie. An integrated model for detecting significant chromatin interactions from high-resolution Hi-C data. *Nature Communications*, 8(1):1–10, 2017.

[13] Merve Sahin, Wilfred Wong, Yingqian Zhan, Kinsey Van Deynze, Richard Koche, and Christina S Leslie. HiC-DC+: systematic 3D interaction calls and differential analysis for Hi-C and HiChIP. *bioRxiv*, 2020.

[14] Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, and Julia Zeitlinger. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, 2021.

[15] Vikram Agarwal and Jay Shendure. Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks. *Cell Reports*, 31(7):107663, 2020.

[16] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.

[17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[18] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

[19] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

[20] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *ICLR*, 2019.

[21] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

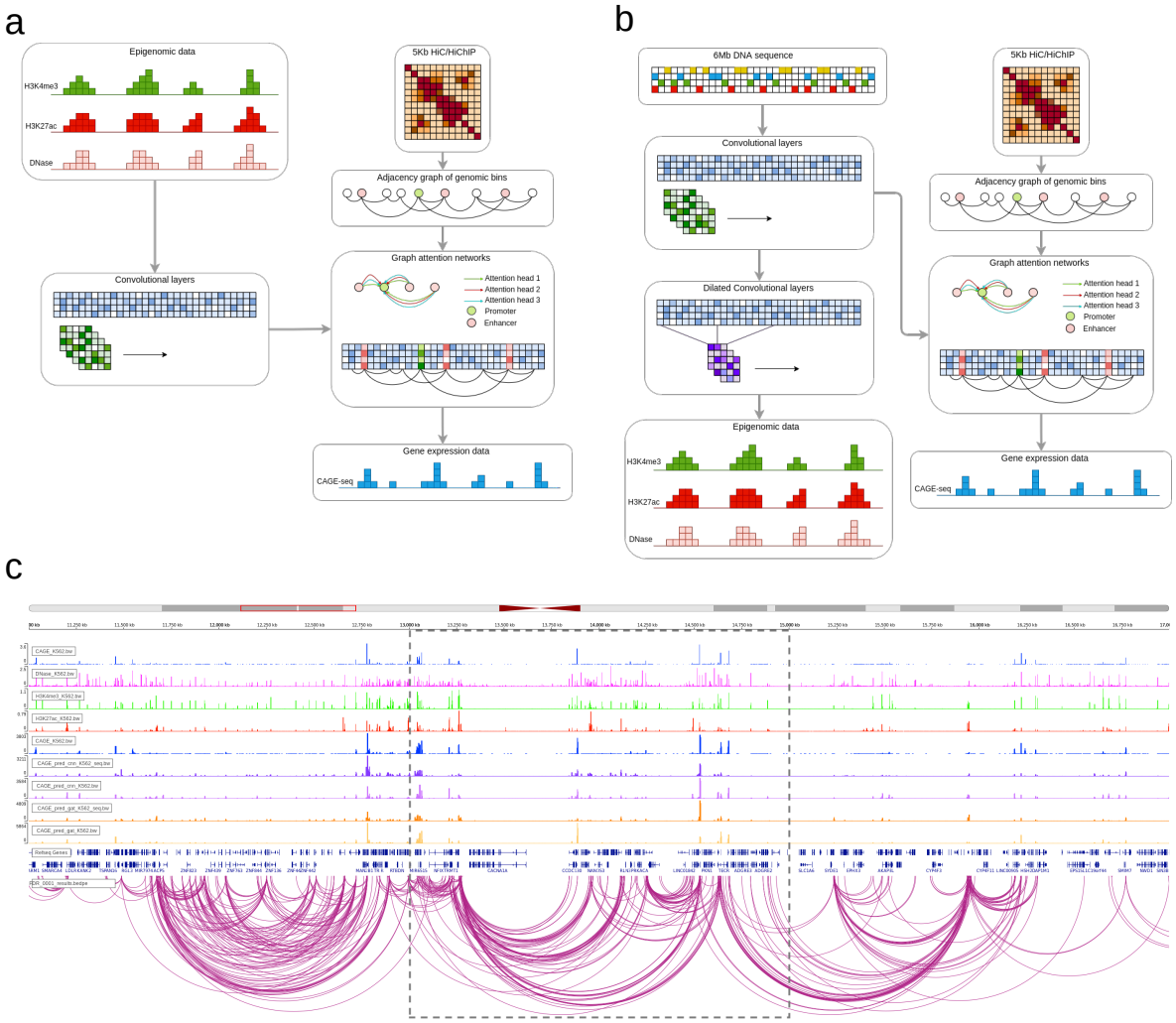


Figure 1: **A schematic overview of GraphReg models. a.** The Epi-GraphReg model uses 1D epigenomic data, such as H3K4me3 and H3K27ac ChIP-seq and DNase-seq (or ATAC-seq) to learn local features of genomic bins via convolutional neural networks, and then propagates these features over adjacency graphs extracted from HiC/HiChIP contact matrices using graph attention networks, in order to predict gene expression (CAGE-seq) across genomic bins. **b.** The Seq-GraphReg model uses DNA sequence as input, and after some convolutional and dilated convolutional layers predicts epigenomic data. This helps to learn useful latent representations of genomic DNA sequences that are then passed to the graph attention networks to be integrated over the adjacency graphs derived from Hi-C/HiChIP contact matrices and to predict gene expression values (CAGE-seq). **c.** A 6Mb genomic region (11Mb-17Mb) of ch19 showing input and output signals and predictions in K562 cells, including epigenomic data (H3K4me3, H3K27ac, DNase), CAGE, HiChIP interaction graph, and predicted CAGE values for GraphReg and CNN models. Training and evaluations of the models are performed in the dashed middle 2Mb (here 13Mb-15Mb) region so that all genes can see the effects of their distal enhancers up to 2Mb.

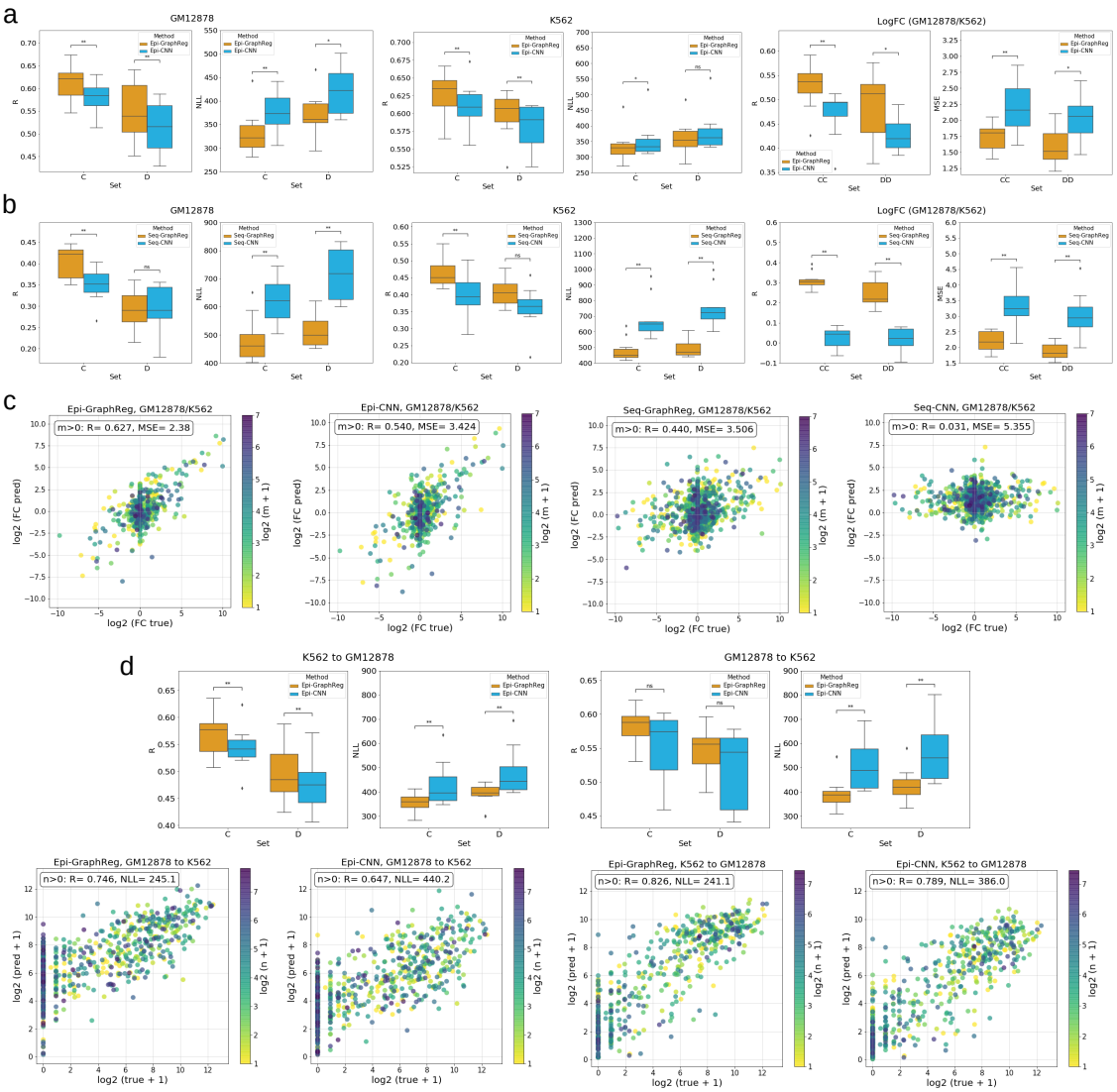


Figure 2: **GraphReg models outperform their CNN counterparts for gene expression prediction. a,b.** Pearson correlation ($R$) and negative log-likelihood ($NLL$) between true and predicted CAGE signals of epigenome-based (**a**) and sequence-based (**b**) GraphReg and CNN models over 10 runs with different training, test, and validation chromosomes. Box plots show the distributions of $R$ and $NLL$ on held-out validation and test chromosomes in the cell lines GM12878 and K562 for the two gene sets: $C$, expressed genes with at least one HiChIP interaction; and $D$, expressed genes with at least five HiChIP interactions. The last column shows $R$ and mean squared error ($MSE$) of the true and predicted log-fold gene expression changes between GM12878 and K562 in the held-out chromosomes over 10 runs. The sets $CC$, and $DD$ in the last column denote the intersections of the sets $C$, and $D$, respectively, in GM12878 and K562 cells. Both Epi- and Seq-GraphReg models have better prediction accuracy than their CNN counterparts. **c.** One of the scatter plots (out of 10 runs) of the true log-fold gene expression changes and the log-fold changes derived from the predicted CAGE values, between GM12878 and K562 in the heldout chromosomes. TSS bins are color-coded by the minimum number of HiChIP interactions in GM12878 and K562 ($m$). Both Epi-GraphReg and Seq-GraphReg have higher $R$ and lower $MSE$ than Epi-CNN and Seq-CNN for the more complex genes ($m > 0$) involving extensive enhancer-promoter interactions. **d.** Epi-GraphReg models demonstrate higher cell to cell generalization capability than Epi-CNN models. Box plots show the distributions of $R$ and $NLL$ on the held-out cell type (K562 or GM12878) when trained on the other cell type over 10 runs with different validation chromosomes on the trained cell. The models are evaluated on the same validation chromosomes in the unseen test cell. The generalization from cell to cell for complex genes (sets $C$ and $D$) is always significantly better ($p < 0.05$, Wilcoxon paired rank test) using Epi-GraphReg than Epi-CNN, both from K562 to GM12878 and from GM12878 to K562. One scatter plot (out of 10) for each case (K562 to GM12878 and GM12878 to K562) is shown for both Epi-GraphReg and Epi-CNN. Epi-GraphReg yields higher $R$ and lower $NLL$ than Epi-CNN for complex genes ($n > 0$) in both cases. The predictions of Epi-CNN are more dispersed, while the predictions of Epi-GraphReg have less variance.

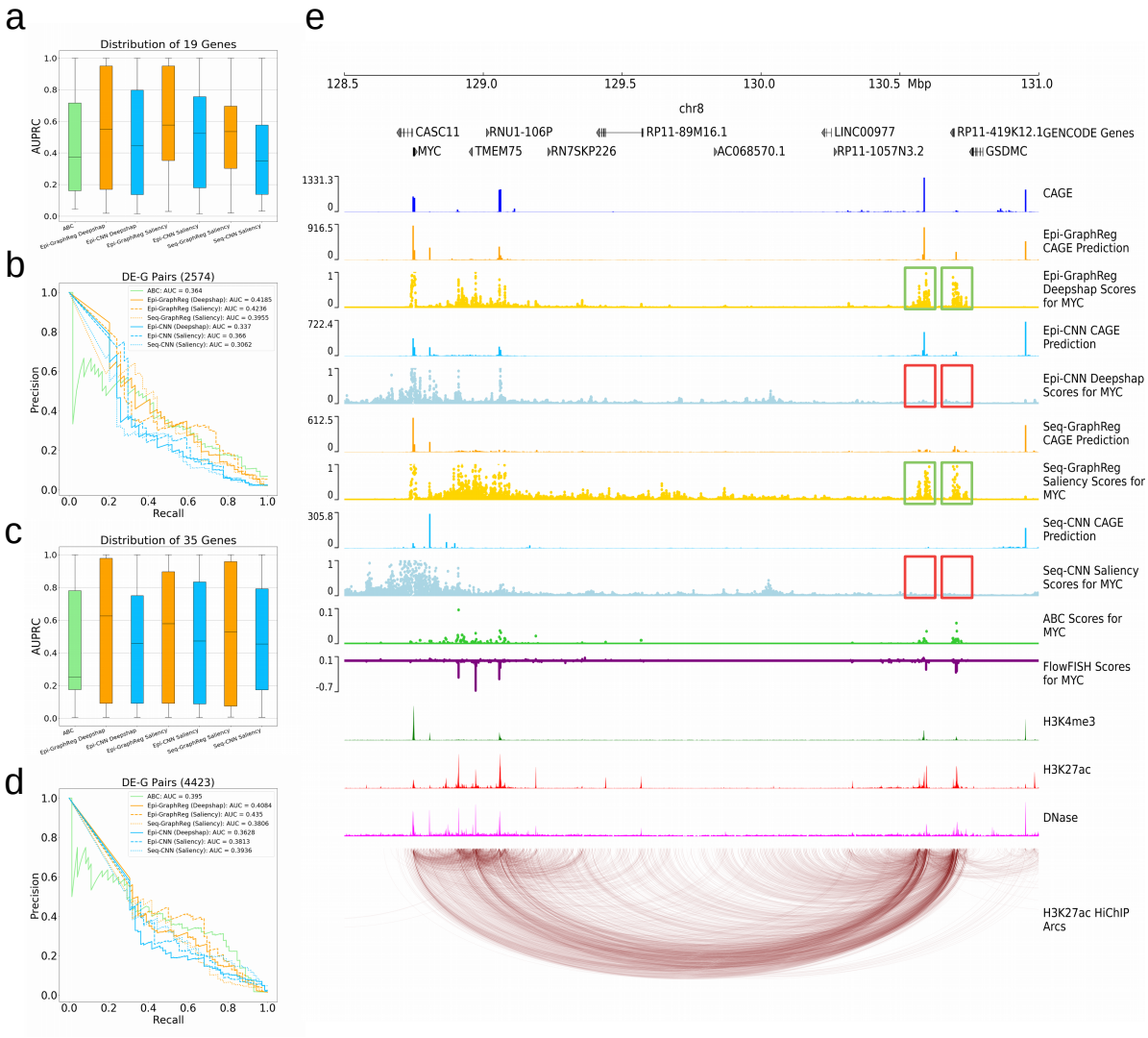


Figure 3: **GraphReg models more accurately identify functional enhancers of genes. a.** Distribution of the area under the precision-recall curve (auPR) for 19 genes in K562 cells based on CRISPRi-FlowFISH data. GraphReg models have higher median of auPR than both CNN and Activity-by-Contact (ABC) models. **b.** Precision-recall curves of the GraphReg, CNN, and ABC models for identifying enhancers of 19 genes screened by CRISPRi-FlowFISH. **c.** Distribution of auPR for 35 genes in K562 cells based on TAP-seq data. GraphReg models have higher median of auPR than both CNN and ABC models. **d.** Precision-recall curves for the GraphReg, CNN, and ABC models for identifying functional enhancers of 35 genes as determined by TAP-seq. **e.** *MYC* locus (2.5Mb) on chr8 with epigenomic data, true CAGE, predicted CAGE using GraphReg and CNN models, HiChIP interaction graph, and the saliency maps of the GraphReg and CNN models, all in K562 cells. Experimental CRISPRi-FlowFISH results and ABC values are also shown for *MYC*. Feature attribution shows that GraphReg models exploit HiChIP interaction graphs to find the distal enhancers, while CNN models find only promoter-proximal enhancers. Green and red boxes show true positives and false negatives, respectively. CNN models miss the distal enhancers and consequently lead to false negatives in very distal regions.

## Supplementary Figures

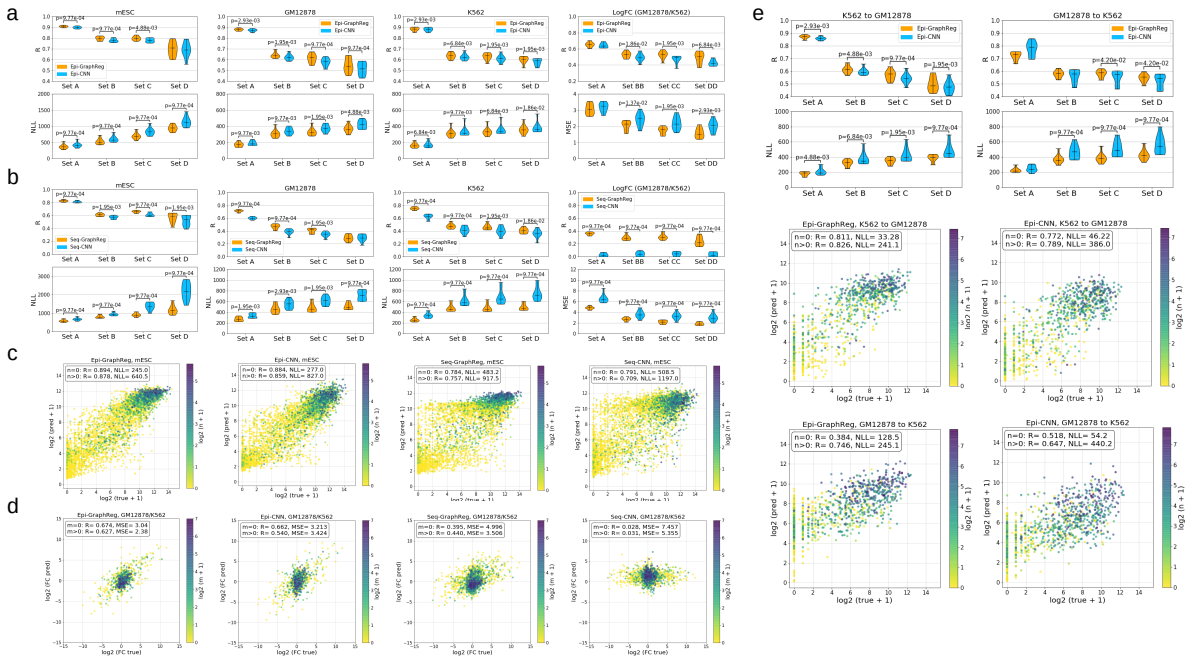


Figure S1: **GraphReg models outperform their CNN counterparts for gene expression prediction. a,b.** Pearson correlation ($R$) and negative log-likelihood ($NLL$) between true and predicted CAGE signals of epigenome-based (**a**) and sequence-based (**b**) GraphReg and CNN models over 10 runs with different training, test, and validation chromosomes. Violin plots show the distributions of $R$ and $NLL$ on held-out validation and test chromosomes in three cell lines, GM12878, K562, and mESC, for the four gene sets: $A$, all genes; $B$, expressed genes; $C$, expressed genes with at least one HiChIP interaction; and $D$, expressed genes with at least five HiChIP interactions. The last column shows $R$ and mean squared error ($MSE$) of the true and predicted log-fold gene expression changes between GM12878 and K562 in the held-out chromosomes over 10 runs. The sets $BB$, $CC$, and $DD$ in the last column denote the intersections of the sets $B$, $C$, and $D$, respectively, in GM12878 and K562 cells. Both Epi- and Seq-GraphReg models have better prediction accuracy than their CNN counterparts. **c.** One of the scatter plots (out of 10 runs) of the log-normalized predicted and true CAGE values in all TSS bins of the heldout chromosomes for mESC. TSS bins are color-coded by their number of HiChIP interactions ($n$). Both Epi-GraphReg and Seq-GraphReg have higher $R$ and lower $NLL$ than Epi-CNN and Seq-CNN for complex genes ($n > 0$) having higher enhancer-promoter interactions. **d.** One of the scatter plots (out of 10 runs) of the true log-fold gene expression changes and the log-fold changes derived from the predicted CAGE values, between GM12878 and K562 in the heldout chromosomes. TSS bins are color-coded by the minimum number of HiChIP interactions in GM12878 and K562 ($m$). Both Epi-GraphReg and Seq-GraphReg have higher $R$ and lower $MSE$ than Epi-CNN and Seq-CNN for the more complex genes ($m > 0$) involving extensive enhancer-promoter interactions. **e.** Epi-GraphReg models demonstrate higher cell to cell generalization capability than Epi-CNN models. Violin plots show the distributions of $R$ and $NLL$ on the held-out cell type (K562 or GM12878) when trained on the other cell type over 10 runs with different validation chromosomes on the trained cell. The models are evaluated on the same validation chromosomes in the unseen test cell. The generalization from cell to cell for complex genes (sets $C$ and $D$) is always significantly better ($p < 0.05$, Wilcoxon rank test) using Epi-GraphReg than Epi-CNN, both from K562 to GM12878 and from GM12878 to K562. One scatter plot (out of 10) for each case (K562 to GM12878 and GM12878 to K562) is shown for both Epi-GraphReg and Epi-CNN. Epi-GraphReg yields higher $R$ and lower $NLL$ than Epi-CNN for complex genes ($n > 0$) in both cases. The predictions of Epi-CNN are more dispersed, while the predictions of Epi-GraphReg have less variance.

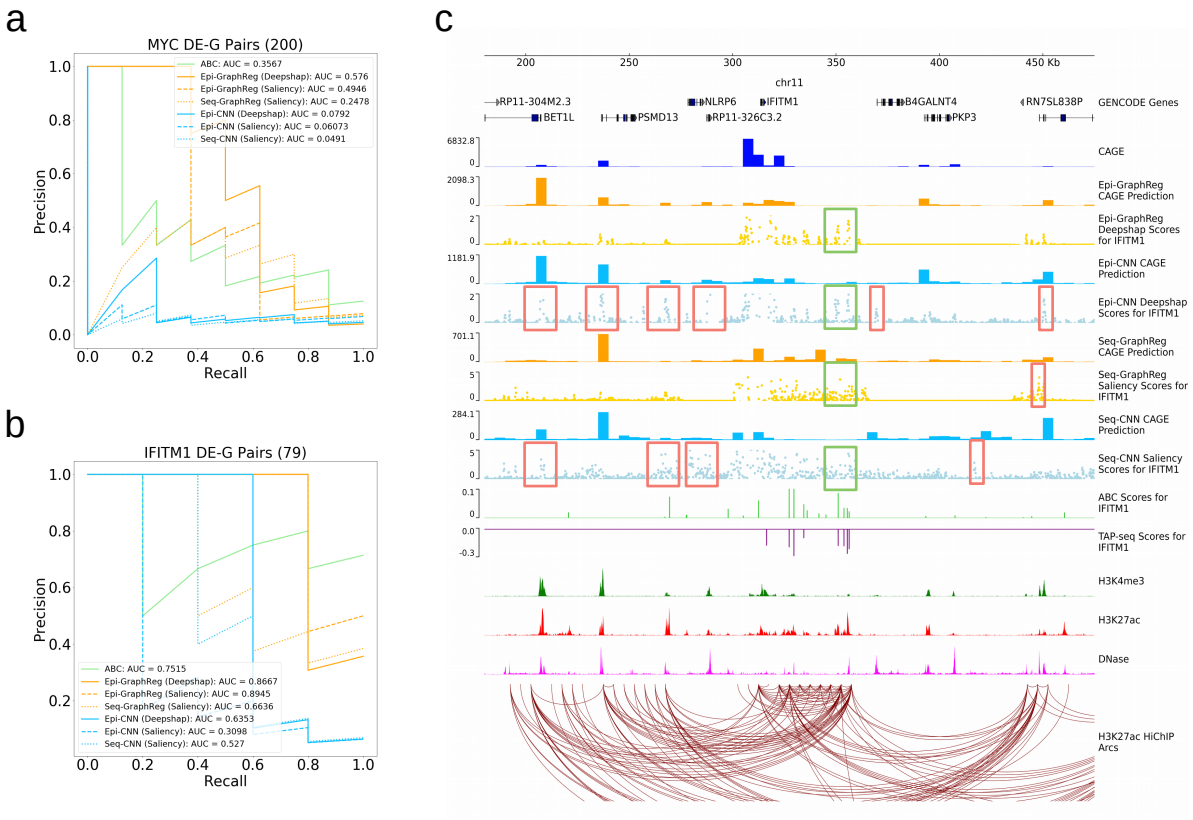


Figure S2: **GraphReg models more accurately identify functional enhancers of genes. a.** Precision-recall curves of the GraphReg, CNN, and ABC models for identifying enhancers of *MYC*. **b.** Precision-recall curves for the GraphReg, CNN, and ABC models for identifying functional enhancers of IFITM1. **c.** *IFITM1* locus (250Kb) in chr11 with epigenomic data, true CAGE, predicted CAGE using GraphReg and CNN models, HiChIP interaction graph, and the saliency maps of the GraphReg and CNN models, all in K562 cells. Experimental TAP-seq results and ABC values are also shown for *IFIM1*. Green and red boxes show the true positives and false positives, respectively. CNN models assign most of the nearby peaks as the true enhancers, leading to many false positives in promoter-proximal regions. GraphReg avoids this problem by explicitly attending to the likely regulatory regions based on HiChIP arcs in the graph.