# A Comprehensive Evaluation of Self Attention for Detecting Feature Interactions

Saira Jabeen
saira.jabeen@colostate.edu

Asa Ben-Hur *
asa@colostate.edu

**Abstract**

Using deep learning in computational biology requires methods that are able to extract meaningful biological information from the trained models. Although deep learning models excel in their prediction performance, interpreting them presents a challenge. Recent work has suggested that self-attention layers can be interpreted to predict cooperativity between binding of transcription factors. We extend this earlier work and demonstrate that the addition of an entropy term to the loss function yields sparser attention values that are both easier to interpret and provide higher precision interpretations. Furthermore, we performed a comprehensive evaluation of the relative performance of different flavors of attention-based transcription factor cooperativity discovery methods, and compared methods that use raw attention scores to the use of attribution over the attention scores, and the earlier DFIM model. We found that the entropy-enhanced attention-based models performed similarly to each other, and exhibited improved accuracy and reduced computation time compared to DFIM.

## 1 Introduction

Deep learning has made a major impact in computational biology, addressing complex prediction problems, including the prediction of epigenomic features [1, 2], gene expression [3, 4, 5], deciphering protein-protein interactions [6] and predicting protein structures [7]. In computational biology, deriving biological insight from trained deep networks is critical. However, despite their strength in accurately modeling complex high-dimensional prediction problems, interpreting deep networks remains challenging. Interpretability has been a part of deep learning in computational biology from the very beginning where the authors of DeepBind have introduced a method to interpret the learned weights of the first convolutional layer as a motif model [8]. Another major effort has been the development of attribution methods which aim at discovering the parts of a sequence that most affect the classification results. These include *in-silico* mutagenesis [2] and gradient-based methods [9, 10].

The process of gene expression is regulated by transcription factors (TFs). The regulation of gene expression is orchestrated by the cooperative binding of TFs and other regulatory proteins in close proximity [11, 12]. Identifying these cooperative binding events can reveal insights that are important for understanding specific biological processes [13, 14]. Several methods have been proposed in order to capture regulatory interactions between TFs. The authors of DFIM [15] have proposed a method based on the DeepLIFT attribution method [16]. The more recent SATORI approach leverages the ability of self-attention layers to capture the relationship between different positions in a sequence [17]. We build on this earlier work in several ways:

- We improve model interpretability with the addition of an entropy term that promotes the sparsity of the attention weights, improving both interpretability and precision of the detected regulatory interactions.
- We extend the SATORI framework by scoring regulatory interactions using either raw attention values or attention attribution.
- We perform a comprehensive evaluation of the approach and its variants on simulated and real datasets in the context of deep learning models of varying levels of complexity, demonstrating the effectiveness of attention for this task.

*Corresponding author. Email: asa@colostate.edu

## 2 Related Work

While there has been extensive research on deep learning architectures for genomic data, efforts related to inferring cooperative regulatory elements have been somewhat limited. As mentioned above, Greenside et al. [15] introduced a method to identify interactions at both nucleotide and motif resolution levels. Their approach utilizes DeepLIFT by computing the disparity in the attribution scores between a sequence with and without mutation to quantify the relationship between a pair of features. A drawback of DFIM is its significant computational expense and the necessity of a post-processing step. The capability of attention layers to discern relationships between features regardless of their spatial distance, laid the groundwork for SATORI [17]. The authors demonstrated comparable results to DFIM with much smaller computational overhead. However, both methods have not been evaluated on realistic simulated datasets that can provide a clear indication of the accuracy of their predictions.

The authors of EPCOT [18] proposed a self-attention-based encoder-decoder architecture to establish a framework for predicting epigenetic state that is able to generalize to conditions that were not a part of their training set. They demonstrated that the learned epigenomic feature embeddings of co-occurring motifs are nearest neighbors in the embedding space, and validated selected interactions using the STRING database. However, the method has not been well tested for the purpose of discovering regulatory interactions. The ECHO model provides a very different approach based on attribution over Micro-C contacts [19]. This approach identifies collaborative binding events without distance limitations, an advantage over sequence-based methods. However, the method is limited by the scarcity of Micro-C data.

While SATORI and EPCOT uses raw attention values to score regulatory interactions, the ISANREG method performs attribution using integrated gradients over the attention scores to infer TF cooperativity [20]. However, their model is limited to detecting TF cooperativity with a single TF of interest, and is designed specifically for the analysis of TF ChIP-seq datasets. SATORI on the other hand is more general, supporting the detection of TF cooperativity for more complex datasets such as those that probe gene regulation on a global scale. In this work, we integrate the idea of using attribution over attention scores into the SATORI framework, to enable a direct comparison of the two approaches, making it a comprehensive tool for inferring TF cooperativity.

In conclusion, while several methods have been proposed to infer regulatory interactions and TF cooperativity, existing approaches have limitations. These include computational complexity (DFIM), reliance on scarce data types (ECHO), and the ability to identify interactions only with specific ChIP-seq targeted proteins (ISANREG). Moreover, there is a notable lack of comprehensive evaluation for these methods. This gap in thorough benchmarking limits our understanding of how well these computational methods perform in real-world scenarios. Our work addresses this by utilizing simulated benchmark datasets to examine the efficacy of existing methods. Furthermore, we leverage these findings to improve SATORI, as demonstrated in our Methods section (Section 5). By doing so, we not only contribute to the advancement of regulatory interaction prediction techniques but also establish a more robust evaluation paradigm for future developments in this field. Additionally, we evaluate the enhancements on two real-world datasets, verifying our findings through a comprehensive overlap analysis using ChIP-seq data.

## 3 Results

### 3.1 SATORI efficiently infers TF interactions embedded in simulated datasets

The recently proposed SATORI method provides a general framework for inferring regulatory interactions from a deep learning model that employs a self-attention layer (see Fig. 1). In what follows we compare the ability of various flavors of SATORI to for this task using simulated datasets of varying levels of complexity and demonstrate the accuracy of the approach. We introduce an extension that incorporates an entropy loss term for the self-attention layer, and demonstrate its ability to increase interpretation accuracy and robustness.

**Impact of the Entropy Loss.** To comprehensively evaluate its impact on the quality of predicted interactions, we trained deep learning models with two architectures, Basic and Deep, with varying levels of depth, with and without the entropy loss. Our analysis reveals that the entropy loss maintains the classification accuracy of the models while enhancing precision by promoting sparsity in the attention profile (see Fig. 2). This reduction in false positives aligns with our expectations. We also observe gains in overall accuracy and in most cases it also contributed positively to recall. The marginal decrease in classification accuracy observed in some cases when using the entropy loss indicates a potential loss of information induced by the increased sparsity (see Fig. 2).

We also observe that as the complexity of the simulated dataset increases, both interpretation and classification accuracy decrease, as would be expected. It is also interesting to note that moving from Basic to Deep architecture

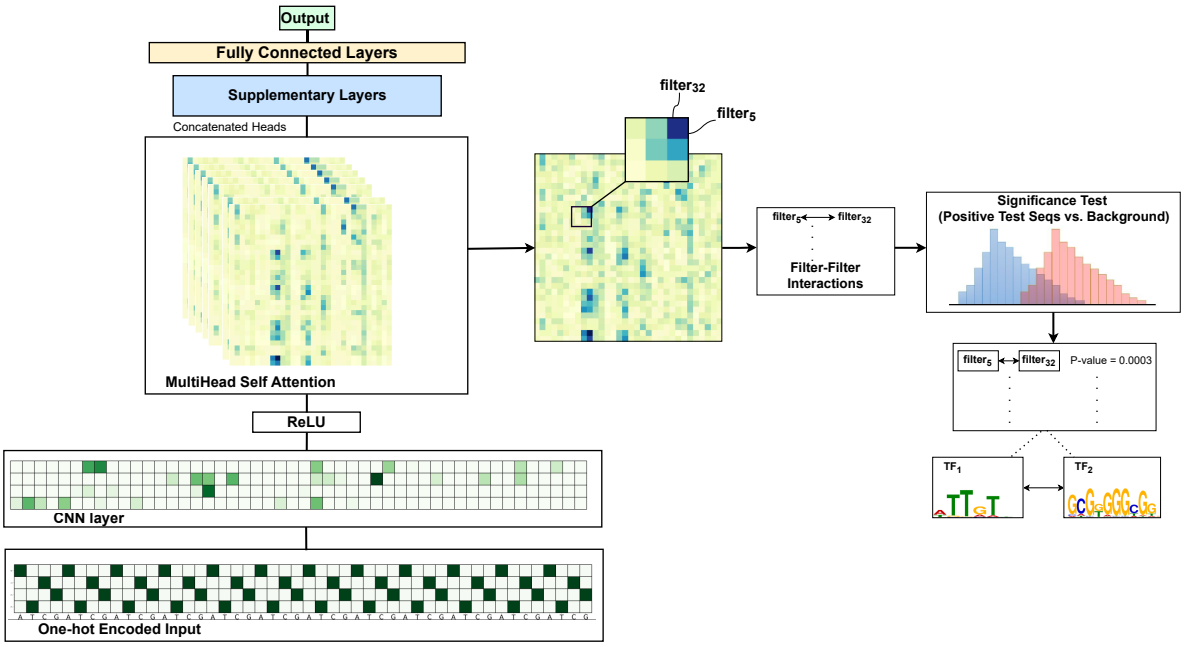


Figure 1: The SATORI framework. The deep learning architecture receives as input a one-hot encoded DNA sequence, which is then processed through a 1D-CNN layer, a self-attention layer, optional supplementary layers and a classification layer. SATORI combines the attention heads and compares the attention values between pairs of filters in positive examples to a background set. It then reports pairs of filters that demonstrate statistical significance.

increases classification AUC, as shown in Fig. 2a. However, interpretation performance demonstrated a slight decrease. This behaviour has been consistent across all three simulated datasets.

To obtain a better understanding of the contribution of the entropy term we looked at the distribution of the attention scores with and without entropy (see Fig. 3a). As expected, we observe that the inclusion of entropy results in a distribution of attention scores that is clearly bi-modal. SATORI scores interactions based on the top attention values. The addition of the entropy term has the added benefit of making it trivial to establish a cutoff between attention values that are used for computing feature interactions. The sparsity resulting from the use of entropy can also be observed in the heatmap representing the attention matrix (see Fig. 3b). The vertical lines observed in the heatmap correspond to positions of embedded motifs. Other positions in the sequence attend to these motif positions, suggesting that the model has successfully learned to attend to motif occurrences. In the figure we also observe that the signal in the attention map is more clear in a model trained with an entropy term. Note that the entropy term is multiplied by a regularization parameter whose value needs to be determined. In our experiments, values of the parameter between 0.001 and 0.01 have yielded similar levels of sparsity.

It is important to note that the accuracy of inferred interactions depends on the ability of TomTom to accurately match motifs. Motifs that bear resemblance to each other may be erroneously matched to a TF not included in our simulated dataset, thereby affecting the accuracy of the inferred interactions.

**Comparison with Attention Attribution and DFIM.** Next, we compare the interpretation accuracy of SATORI using raw attention scores with that of attention attribution scores and DFIM using the Basic deep learning architecture. In our experiments, we observed that raw attention scores and attention attribution scores give comparable results on simulated datasets as shown in Fig. 4. In fact, the attention attribution matrices exhibit a high similarity to the raw attention matrices (see Fig. S1 in the supplementary material) SATORI with entropy loss outperforms the corresponding model without entropy, both with raw attention scores and attention attribution scores. Both models also outperforms DFIM when used in conjunction with entropy.

An advantage of SATORI with raw attention scores is computation time (see Fig. S2 in the supplement). DFIM and SATORI with attention attribution have the added overhead of requiring the computation of gradients after the model has been trained. For the basic architecture using raw attention scores has only a slight advantage in terms of running time; for the deep architecture gradient computations dominate the running time, and using raw

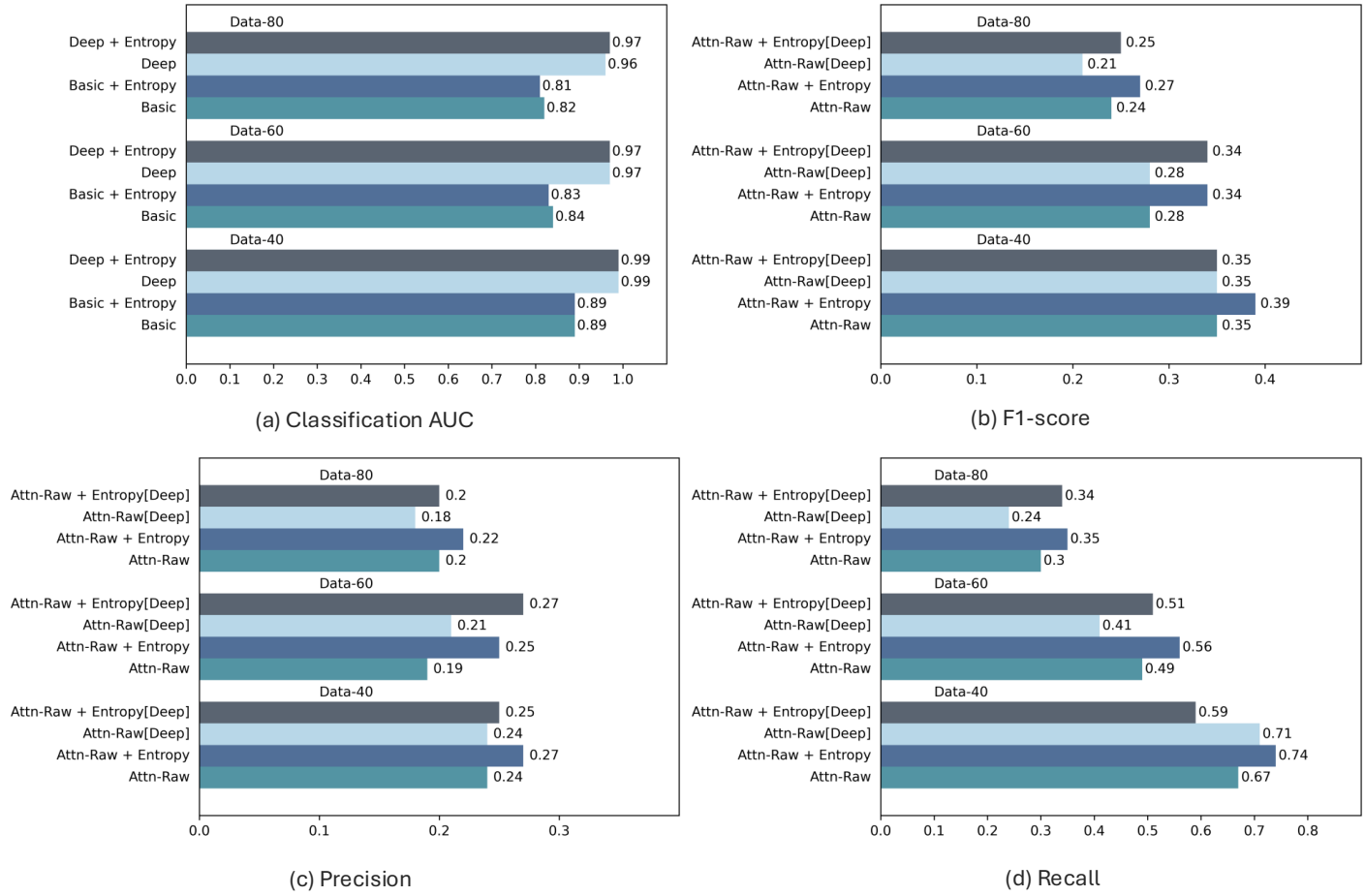


Figure 2: Comparative performance of SATORI on simulated datasets of varying levels of complexity using the Basic and Deep architectures. We show: Classification AUC (a) and interpretation accuracy: F1-score (b), Precision (c) and Recall (d). Results are shown for four models: Basic and Deep trained with and without the entropy loss; results are reported for datasets of increasing levels of complexity, comprising 40, 60, or 80 implanted TF motifs; performance numbers are averages over three instances of each dataset. Results are reported for the version of SATORI that uses raw attention values (Attn-Raw).

attention scores is around five times faster. The difference with respect to DFIM is even more dramatic, where SATORI with raw attention scores runs over a hundred times faster for the deep architecture.

**Incorporating RNNs.** In our original work we used a bidirectional recurrent neural network (RNN) layer after the initial CNN layer [17]. Although this design introduces a less direct relationship between the output of the CNN filters and the self-attention layer, we have found that this did not impact interpretability accuracy: on the Data-60 simulated dataset adding an RNN layer (a bidirectional LSTM) to the SATORI-Basic architecture produced an interpretation F1-score of 0.36 compared to 0.33 without the RNN (these represent an average over ten instances). Adding the RNN also improved classification accuracy measured by area under the ROC curve from 0.88 to 0.96. The use of an RNN led to a reduction in the reproducibility of the results: The interpretability F1-score with SATORI-Basic demonstrated a standard deviation of 0.0004 across ten runs of the model with different seed values. The addition of the RNN increased this to 0.004, which we would still consider stable.

## 3.2 TF Interactions in Accessible Chromatin in Human Promoters

To test the ability of SATORI to detect cooperation of TFs in real data, we explored the regulatory cooperation among TFs within human promoter regions leveraging DNase I hypersensitivity data (DHSs) collected from 164 immortalized cell lines and tissues [2] as in our earlier work [17]. The task is the prediction of the presence or

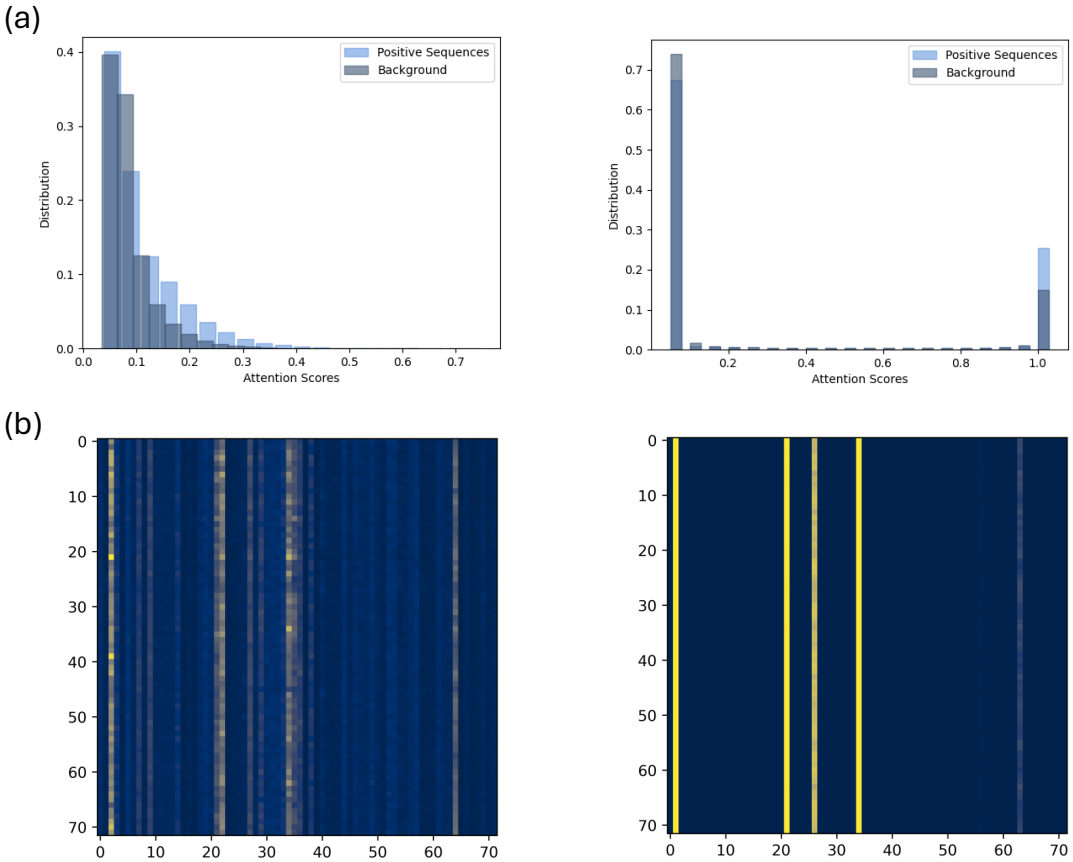


Figure 3: The effect of entropy on attention weights, for a basic model trained without (left) and with (right) the entropy loss using a regularization parameter of 0.01 for Data-80. (a) shows the overall distribution across accurately predicted test sequences. (b) shows the heatmaps of attention weights (max over all attention heads) for a true positive sequence. X-axis and Y-axis in heatmaps are the positions in the simulated DNA sequence after max pooling.

absence of a DHS across each of the 164 cell lines, leading to a multi-label classification problem. The dataset consists of 20,613 examples across 164 cell types and each sequence is 600bp in length. We partitioned the dataset into training, testing, and validation sets at an approximate ratio of 80%, 10%, and 10%, respectively, while ensuring entire chromosomes were allocated to each set. Specifically, the training set includes chromosomes 1-14 and 18-19, the validation set includes chromosomes 17, 20-21, and the test set contains the remaining chromosomes. The mean classification AUC-ROC achieved for the basic and deep SATORI models without entropy were 0.82 and 0.84, respectively. The addition of the entropy term resulted in a slight decrease in classification accuracy, yielding an AUC of 0.79 for the basic model, and 0.81 for the deep model. However, the entropy term was crucial for detecting interactions: the addition of the entropy term increased the number of interactions detected for both architectures without affecting the quality of the detected interactions (see Fig. 5).

In order to validate the predicted interactions we ran a Locus Overlap Analysis [21] using ChIP-Seq data for the identified TF pairs downloaded from the ChIP-Atlas website [22]. All the methods exhibited a very high rate of confirmation by Locus Overlap Analysis, even higher than the precision shown in our simulated data, demonstrating the validity of the predicted interactions (see Fig. 5(a)). The deep architecture with raw attention scores produced the largest number of verified interactions and precision of 0.99. This demonstrates the value of attention scores even for deeper networks. We note that not all detected interactions could be tested due to the lack of availability of ChIP-Seq data for all the TFs. For further validation of the predictions of Attn-Raw with the deep architecture, we compared those interactions with known interactions in the TrrustV2 database [23]. Out of the 344 interactions we detected, 16 interactions matched entries in TrrustV2 having a p-value of $9.82 \times 10^{-7}$ using the hypergeometric test. We also searched these interactions against the HIPPIE protein-protein interactions database [24] since the co-occorrence of TF binding can be the result of physical interactions between the TFs; we found 14 matches in

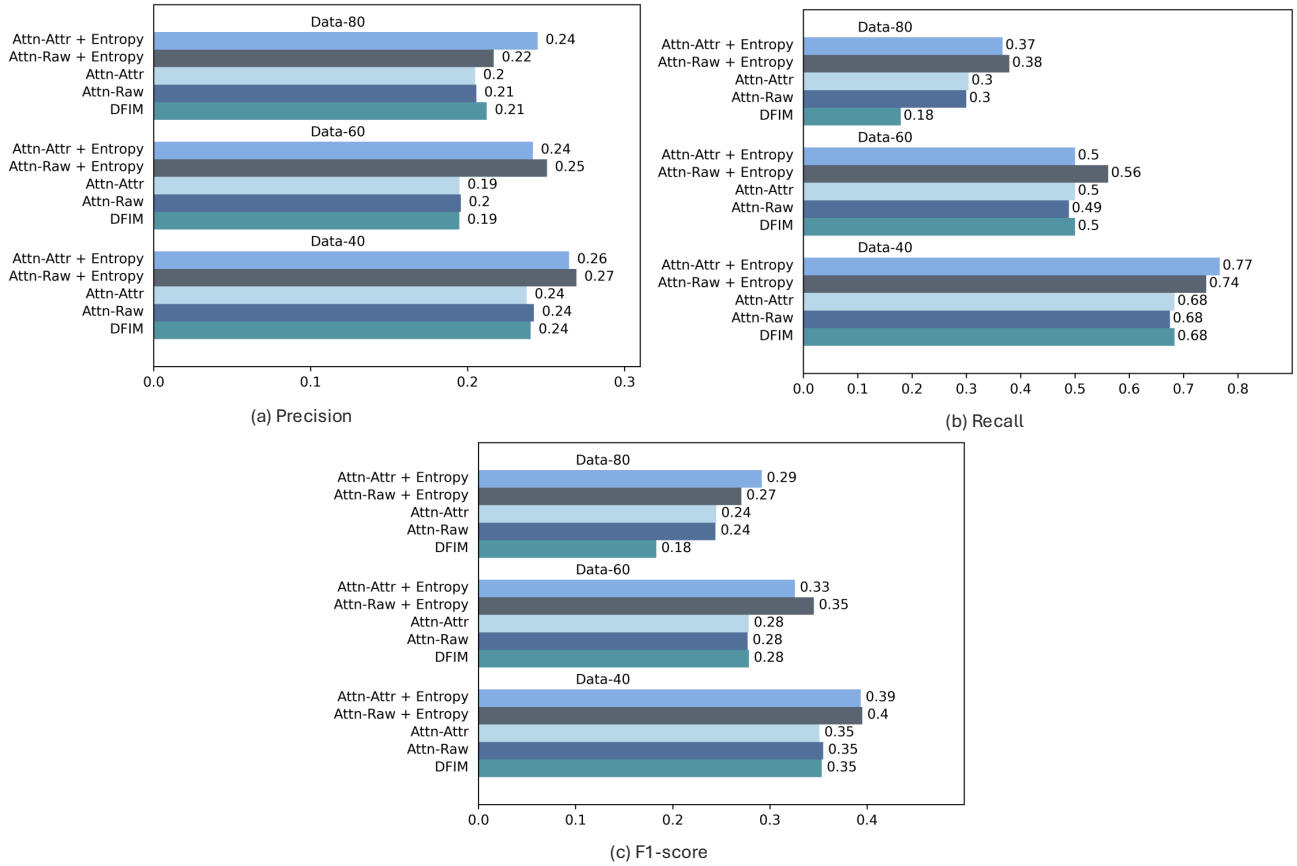


Figure 4: Interpretation accuracy. We compare performance of SATORI using raw attention scores (Attn-Raw), attention attribution (Attn-Attr), and DFIM based on precision (a), recall (b), and F1-score (c). All methods use the same basic architecture; datasets comprise 40, 60, or 80 implanted TF motifs; performance numbers are averages over three instances of each dataset.

the HIPPIE database with a hypergeometric test resulting in a p-value of $1.61 \times 10^{-16}$; four of them were also confirmed in TrrustV2. Tables S4 and S5 in the supplementary material provide the list of matches identified.

### 3.3 TF Interactions in Accessible Chromatin in Arabidopsis

For the next experiment, we predicted chromatin accessibility across 36 Arabidopsis samples gathered from DNase I-Seq and ATAC-Seq studies used in our earlier work [17]. This is also a multi-label classification problem with a total of 80,000 instances, each of length 600bp. We partitioned the dataset into training, validation, and testing subsets in an 80%, 10%, and 10% ratio, respectively, while preserving chromosomal distribution. As a result, chromosome 2 was allocated to the validation set, chromosome 4 to the test set, and all other chromosomes were included in the training set. We trained two SATORI models, basic and deep, with and without employing the entropy loss. The mean classification AUC achieved by both models without entropy on this dataset were 0.82 and 0.83, respectively. For the models trained with entropy loss, basic and deep SATORI models achieved an AUC of 0.77 and 0.82, respectively.

For this dataset we also performed a comprehensive validation experiment based on a large collection of ChIP-seq datasets retrieved from ChIP-Hub [25]. For each pair of TFs for which we were able to find data on ChIP-Hub we ran a a Locus Overlap Analysis [21] to indicate whether the two TFs demonstrate statistically significant co-localization of their peaks in the Arabidopsis genome. Overall, the Arabidopsis dataset exhibited lower precision compared to the human promoters dataset Fig. 5(b). We believe this is due to the size of the Arabidopsis genome being ten times smaller than the human genome, resulting in fewer binding sites and therefore less power to detect significant locus overlap. Although lower, these levels of validation for a real-world dataset are similar to the levels of precision we observed in our simulated datasets.

In this dataset the addition of the entropy term was not as critical for detecting interactions, and in some cases

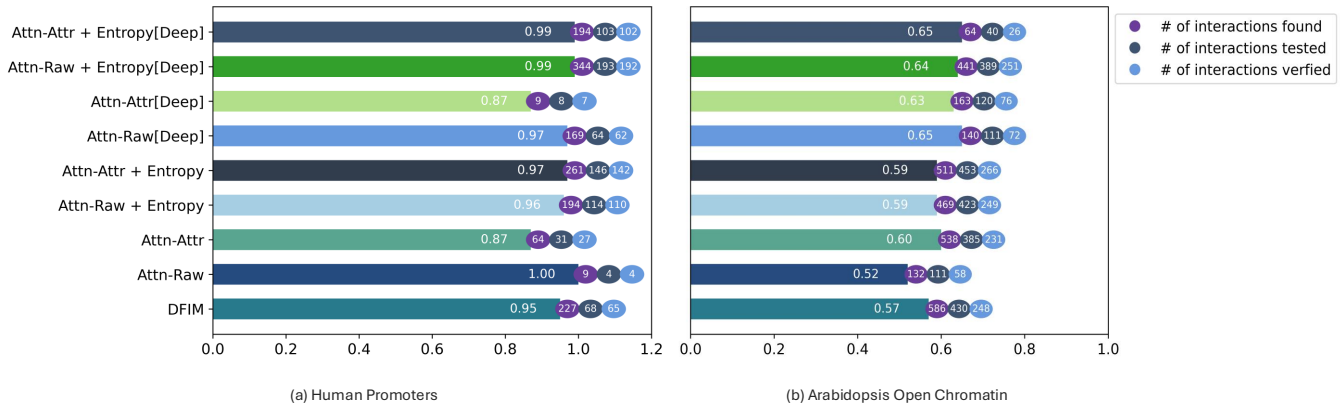


Figure 5: Validation of inferred interactions using ChIP-Seq overlap analysis. We performed an overlap analysis over interactions inferred by SATORI basic and deep architectures with and without entropy loss, compared against DFIM on the Human Promoter dataset (a) and accessible chromatin in Arabidopsis (b). Horizontal bars represent precision values of the predicted interactions with respect to the interactions verified by overlap analysis.

we obtained a higher number of interactions for the model without entropy (Fig. 5(b)). We believe this is a result of the much larger number of examples available in this dataset, making it easier for SATORI to detect significant interactions from the noisy attention weights, thereby requiring less help form the entropy term. In comparison to DFIM, the different variants of SATORI exhibited higher precision. In terms of number of interactions, DFIM had an advantage in the Arabidopsis dataset, and was competitive on the human promoters dataset.

# 4 Conclusions

In this work we presented SATORI 2.0 which adds the option of an entropy penalty term and incorporates the option of scoring feature interactions based on attribution scores over the attention matrix as an alternative to scoring based on the raw attention values. We performed a comprehensive evaluation of its performance over realistic simulated datasets and real-world datasets with architectures with varying levels of complexity. The results show that the self-attention layers can effectively and accurately discover regulatory interactions across different architectures and datasets. We found that the entropy loss helps remove false positives, and in most cases, it also increases the method's recall. In our experiments we found that the method that uses attention attribution scores performed similarly to the one that uses raw attention values; the earlier DFIM method performed at a simialr level as well, but required significantly higher computation times. In summary, the SATORI framework provides a comprehensive tool for interpreting neural network layers and obtaining valuable predictions about regulatory relationships that can be further tested in the laboratory.

# 5 Materials and Methods

## 5.1 Simulated Datasets

To obtain reliable estimates of interpretability accuracy we created simulated datasets with implanted interactions. This addresses the lack of ground truth interactions in real-world datasets. In generating our simulated datasets we aimed to replicate the characteristics of real-world datasets. These datasets are designed to be sufficiently complex to challenge the ability of the network to discover interactions. We have generated binary labeled datasets with varying levels of complexity. In each dataset, the positive class is distinguished from the negative class by the presence of interacting pairs of motifs. All sequences are randomly generated based on the nucleotide frequency of the human genome. For each dataset we randomly selected motifs from the JASPAR database [26] and implanted motif occurrences according to their PWMs. In the positive class, for each sequence three interacting pairs are randomly chosen from the predefined set of interacting pairs. These pairs are embedded into the sequence using its PWM, ensuring PWM scores exceeding half of the maximum PWM score. Motifs from an interacting pair are placed in close proximity, within 8-15 base pairs. In the negative class, one to three motifs are randomly selected from the same pool of motifs. These motifs are uniformly embedded using their PWM, with each motif positioned

Table 1: Dataset details. We provide the number of TFs and interacting pairs in the simulated datasets.

| Dataset Name | No of Unique TFs | No of pairs |
|---|---|---|
| Data-40 | 20 | 40 |
| Data-60 | 25 | 60 |
| Data-80 | 30 | 80 |

at least 40 base pairs away from every other motif within the sequence. The generated datasets are balanced, consist of a total of 60,000 instances per dataset and each sequence is 300bp long. Table 1 describes the number of TFs embedded in the three simulated datasets along with the number of interactions.

## 5.2 Interpreting Self-Attention Layers with SATORI

SATORI uses a network with a structure that is composed of two or more blocks: an initial block composed of a 1D convolutional feature extraction layer followed by a multi-head self-attention layer; the initial block is followed by additional (optional) deep learning layers and a fully connected output block (see Fig. 1). The attention layer in the primary block serves as a pivotal element for interpretation. Any additional layers serve to provide improved model accuracy, and we demonstrate that this framework can handle models of varying levels of complexity.

The input to the model is a DNA sequence that is represented by its one-hot encoding. The first layer in the network is a one-dimensional convolutional layer that facilitates feature extraction from the DNA sequence. The output of convolution is passed through an activation function. In this work we use the Rectified linear unit function (ReLU) defined as

$$\mathrm{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

After applying the ReLU function, a max pooling operation is employed to downsample the spatial dimensions of the feature maps. This max-pooling step is optional, and can help in reducing the computation time and memory usage by reducing the length of the sequences that need to be processed by the attention layer, which is the most time and memory consuming aspect of the model. In our experiments we used a very small value for the pooling parameter (4 or less), to ensure that the attention layer receives accurate spatial information about the sequence. The output of the max-pooling operation is fed into the multi-head self-attention layer, which captures relationships between elements within the sequence. This allows the model to weigh the importance of different elements, enabling it to focus on relevant information. Self-attention is computed using three matrices known as query, key, and value defined as:

$$Q = XW_q, \quad K = XW_k, \quad V = XW_v, \tag{2}$$

where $X$ is the max-pooled output of first CNN layer, $W_q$ , $W_k$ and $W_v$ are learnable weight matrices used to transform the input vector $X$ into the query ($Q$), key ($K$), and value ($V$) matrices. By computing the dot product between the query and key vectors of specific positions in the sequence, we can assess the relationship between the selected positions. We then compute dot product attention as follows:

$$A = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) \tag{3}$$

where, $d_k$ is the embedding dimension of $K$, $Q$, and $V$. The softmax function is used to transform the dot products into probabilities arranged in a row-wise manner. This allows us to interpret the output of the attention layer as the likelihood of a relation between different parts of the input.

The output from the softmax function is employed as weights to compute the value matrix $V$ to generate the output of a given attention head:

$$Z = A \cdot V \tag{4}$$

Multi-head attention enables the model to simultaneously attend to different aspects of the input from diverse subspaces across various positions. Subsequently, the outputs of the attention heads are concatenated to produce the final output of the attention layer.

We use a similar process as described by Ullah et al. [17] to discover filter-filter interactions, and infer motif interactions from those filter interactions. To infer interactions we first combine the values of the attention heads

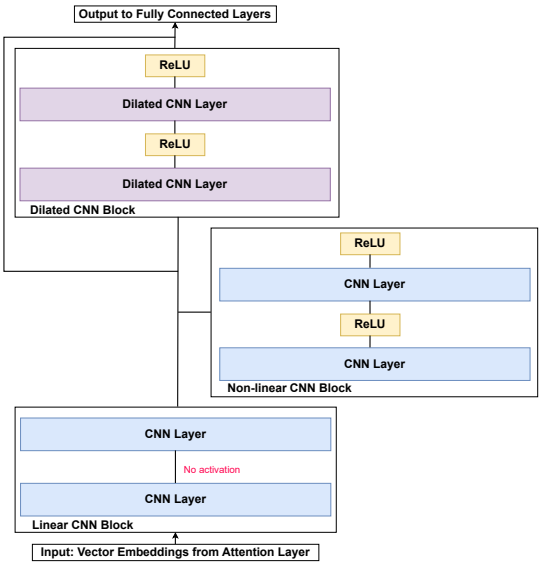


Figure 6: Additional layers included in the SATORI deep architecture after the self-attention layer.

by computing the maximum value across heads. Instead of employing a fixed threshold value as was used in the original SATORI, we utilized the mean value of attention scores as a cutoff threshold for selecting interacting attention values. Following this, the statistical significance of the attention values is evaluated using the non-parametric Mann-Whitney U Test [27] relative to the background attention scores computed over the negative examples for binary classification or shuffled sequences in the context of multi-label classification. This process is illustrated in Fig. 1.

We leveraged the weights of the initial CNN layer and attention layer to map filters to putative TFs using an established protocol [2]. Each substring yielding an activation score exceeding 0.65 of the maximum score of the filter is used the final computation of the position weight matrix (PWM) for that filter. The resulting PWMs are then compared against the known motif databases (JASPAR [26] (simulated datasets), Human CISBP [28] (for the human promoters dataset) and Arabidopsis DAP [29]) using the TomTom tool [30], and the top matches are assigned to each filter.

**A deeper architecture.** Our baseline architecture (basic SATORI) directly feeds the output of the attention layer to a fully connected layer. In order to demonstrate the effectiveness of SATORI with more complex architectures we explored the addition of an additional block of convolutional layers whose structure was inspired by the SEI model [31]. The basic component of this block is a residual dual linear and non-linear CNN sub-block shown in Fig. 6. The linear pathway enables rapid training, while the nonlinear pathway is intended to enhance its representational power. Furthermore, a residual connection is established between the stacked linear and nonlinear blocks, thereby permitting computation to traverse either the linear or nonlinear pathway. The last sub-block uses dilated CNN layers augmented with residual connections, to capture longer-range relationships across the sequence. Subsequently, the resulting feature maps are averaged along the sequence dimension prior to processing through a fully connected layer and final output layer.

**Improving network interpretability with an entropy loss term.** In order to improve the interpretability of the attention values we incorporated an entropy term in our loss function in addition to the classification loss. The entropy loss is calculated from the attention scores to promote sparsity of the attention matrix. The resulting loss function is defined as follows:

$$Loss = Loss_c + \lambda Loss_{.} \tag{5}$$

Here, $Loss_c$ is a classification loss (e.g. binary cross entropy) and $Loss_e$ is the entropy term defined as:

$$Loss_e = -\sum_{k=1}^{h}\left(\frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{n} A_{ij}^{(k)} \log A_{ij}^{(k)}\right),$$

where $n$ is number of rows or columns in the attention head $A^{(k)}$, and $h$ is number of heads. The parameter $\lambda$ in Equation 5 is the regularization parameter for the entropy loss and is selected as part of our hyperparameter search.

**Attention Attribution Scores.** To compare the interpretation performance of raw attention scores with attributed attention scores, as used by [20], we incorporated their approach into the SATORI framework. Attention attribution scores are computed by calculating the gradients of the target output with respect to self-attention scores. This method was first proposed by [32] to interpret the information learned by a transformer model. The attribution scores are approximated from a zero-baseline, where no filters attend to each other, to the actual input by gradually increasing the attention scores' weight. This can be expressed via the Riemann approximation of the integration [33] as follows:

$$\text{Attr}(A^{(k)}) = \frac{A^{(k)}}{m} \odot \sum_{i=1}^{m} \frac{\partial F\left(\frac{i}{m}A\right)}{\partial A^{(k)}} \tag{6}$$

where $\odot$ is element-wise multiplication, $A^{(k)}$ describes a single attention head among a set of $h$ attention heads. $\frac{\partial F(A)}{\partial A^{(k)}}$ is the gradient of the model $F(\cdot)$ with respect to $A^{(k)}$; $m$ is the number of approximation steps, which is set to 20. To identify significant interactions, we process the attention attribution scores in a similar manner to the way SATORI processes raw attention scores. When using the entropy loss with the deep architecture we incorporated residual connections from the output of the feature extraction CNN layer in the initial block to the output of the multihead attention layer before applying additional convolution layers. This is done to maintain the accuracy of the model, which can be impacted by the increased sparsity induced by the entropy loss term.

## 5.3 Model Selection and Training

We used two primary architectures in our experiments: a basic model, which is a two-layered architecture comprising a CNN layer, self-attention layer, and the more complex model which includes additional CNN layers. For model selection for the basic architecture, we employed a randomized grid search algorithm using the predefined search space shown in Table S1 in the supplementary material. This algorithm tunes multiple hyperparameters including the number of CNN filters, CNN filter size, size of the attention head, number of attention heads, and the size of the linear layer after the multi-head attention layer. In addition to these, the optimizer, learning rate and weight decay are also determined through hyperparameter search. An instance of the simulated dataset is used for hyperparameter selection based on its AUC on the validation set. Details of selected models for training are provided in Table S2 of the supplementary material. To obtain an optimal value for the regularization parameter $\lambda$ of the entropy loss, we used a values on a logarithmic scale ranging from $10^{-5}$ to $10^{-2}$. We selected the entropy value corresponding to the best interaction performance (F1-score) on the dataset used for model selection for further experimentation. Datasets that are used for model and hyperparameter selection are later discarded, and new instances are utilized to evaluate the performance of the final selected model. For the real datasets we used architecture settings similar to our previous work [17]; the deep architecture used the same hyperparameters for the initial block; details of the SEI-inspired additional block are given in Table S3 in supplementary material. The simulated datasets were divided into train, validation and test sets with 80%, 10% and 10% ratios, respectively. For each dataset size we generated three dataset instances. Subsequently, we trained a model three times using different seed values on each instance of the dataset, and the best-performing model, determined by validation AUC was selected. The final results for each dataset type are reported by averaging over the three dataset instances. Model accuracy was measured using area under the ROC curve (AUC) and interpretation accuracy of the predicted interactions was measured using precision, recall, and F1-score. All methods were implemented using PyTorch [34] and all the experiments were conducted on a Linux workstation with a 12GB TITAN V GPU. For DFIM implementation, we have used the implementation provided in the original SATORI code base [17].

## 5.4 ChIP-seq overlap analysis

To validate interactions predicted on the real datasets we performed a locus overlap analysis using the R package LOLA [21] For each of the datasets we downloaded ChIP peaks from ChIP-Atlas [22] (for the human promoters

dataset) and ChIP-Hub [25] (for the arabidopsis open chromatin dataset). From each resource we downloaded ChIP-seq datasets for all TFs involved in the interactions identified by SATORI and DFIM. The LOLA package requires three inputs: 1) A query set, which consists of one of the TFs in the interacting pair; 2) A set of genomic regions in which to evaluate potential overlap, accessible DNA in our case; and 3) A reference database built using the other TF from the interacting pair. We used LOLA's default settings to run the overlap analysis, which counts a single shared base pair between regions as overlapping.

# 6 Data Availability

The source code and datasets are available at https://github.com/sairajab/satoriv2. A snapshot of the repository is available at https://doi.org/10.5281/zenodo.13362826. Tables S6 and S7 in the supplement provides the list of all accession numbers for ChIP-seq datasets used in the overlap analysis.

# 7 Funding

This material is based upon work supported by the U.S. Department of Energy Office of Science, Biological and Environmental Research award DE-SC0024459.

# References

[1] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

[2] David R Kelley, Jasper Snoek, and John L Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 26(7):990–999, 2016.

[3] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

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

[5] Alireza Karbalayghareh, Merve Sahin, and Christina S Leslie. Chromatin interaction–aware gene regulatory modeling with graph attention networks. *Genome Research*, 32(5):930–944, 2022.

[6] Somaye Hashemifar, Behnam Neyshabur, Aly A Khan, and Jinbo Xu. Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics*, 34(17):i802–i810, 2018.

[7] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[8] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

[9] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[10] Anupama Jha, Joseph K. Aicher, Matthew R. Gazzara, Deependra Singh, and Yoseph Barash. Enhanced integrated gradients: improving interpretability of deep learning models using splicing codes as a case study. *Genome biology*, 21:1–22, 2020.

[11] Wyeth W Wasserman and James W Fickett. Identification of regulatory regions which confer muscle-specific gene expression. *Journal of molecular biology*, 278(1):167–181, 1998.

[12] Satyanarayan Rao, Kami Ahmad, and Srinivas Ramachandran. Cooperative binding of transcription factors is a hallmark of active enhancers. *bioRxiv*, pages 2020–08, 2020.

[13] Priya Sudarsanam, Yitzhak Pilpel, and George M Church. Genome-wide co-occurrence of promoter elements reveals a cis-regulatory cassette of rRNA transcription motifs in saccharomyces cerevisiae. *Genome research*, 12(11):1723–1731, 2002.

[14] Ignacio L Ibarra, Nele M Hollmann, Bernd Klaus, Sandra Augsten, Britta Velten, Janosch Hennig, and Judith B Zaugg. Mechanistic insights into transcription factor cooperativity and its impact on protein-phenotype interactions. *Nature communications*, 11(1):124, 2020.

[15] Peyton Greenside, Tyler Shimko, Polly Fordyce, and Anshul Kundaje. Discovering epistatic feature interactions from neural network models of regulatory DNA sequences. *Bioinformatics*, 34(17):i629–i637, 2018.

[16] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[17] Fahad Ullah and Asa Ben-Hur. A self-attention model for inferring cooperativity between regulatory features. *Nucleic acids research*, 49(13):e77–e77, 2021.

[18] Zhenhao Zhang, Fan Feng, Yiyang Qiu, and Jie Liu. A generalizable framework to comprehensively predict epigenome, chromatin organization, and transcriptome. *Nucleic Acids Research*, 51(12):5931–5947, 2023.

[19] Zhenhao Zhang, Fan Feng, and Jie Liu. Characterizing collaborative transcription regulation with a graph-based deep learning approach. *PLOS Computational Biology*, 18(6):e1010162, 2022.

[20] Anil Prakash and Moinak Banerjee. An interpretable block-attention network for identifying regulatory feature interactions. *Briefings in Bioinformatics*, 24(4):bbad250, 2023.

[21] VP Nagraj, Neal E Magee, and Nathan C Sheffield. LOLAweb: a containerized web server for interactive genomic locus overlap enrichment analysis. *Nucleic acids research*, 46(W1):W194–W199, 2018.

[22] Zhaonan Zou, Tazro Ohta, and Shinya Oki. ChIP-Atlas 3.0: a data-mining suite to explore chromosome architecture together with large-scale regulome data. *Nucleic Acids Research*, page gkae358, 2024.

[23] Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al. TRRUST v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, 46(D1):D380–D386, 2018.

[24] Gregorio Alanis-Lobato, Miguel A Andrade-Navarro, and Martin H Schaefer. HIPPIE v2. 0: enhancing meaningfulness and reliability of protein–protein interaction networks. *Nucleic acids research*, page gkw985, 2016.

[25] Liang-Yu Fu, Tao Zhu, Xinkai Zhou, Ranran Yu, Zhaohui He, Peijing Zhang, Zhigui Wu, Ming Chen, Kerstin Kaufmann, and Dijun Chen. ChIP-Hub provides an integrative platform for exploring plant regulome. *Nature communications*, 13(1):3413, 2022.

[26] Oriol Fornes, Jaime A Castro-Mondragon, Aziz Khan, Robin Van der Lee, Xi Zhang, Phillip A Richmond, Bhavi P Modi, Solenne Correard, Marius Gheorghe, Damir Baranašić, et al. JASPAR 2020: update of the open-access database of transcription factor binding profiles. *Nucleic acids research*, 48(D1):D87–D92, 2020.

[27] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[28] Matthew T Weirauch, Ally Yang, Mihai Albu, Atina G Cote, Alejandro Montenegro-Montero, Philipp Drewe, Hamed S Najafabadi, Samuel A Lambert, Ishminder Mann, Kate Cook, et al. Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*, 158(6):1431–1443, 2014.

[29] Ronan C O'Malley, Shao-shan Carol Huang, Liang Song, Mathew G Lewsey, Anna Bartlett, Joseph R Nery, Mary Galli, Andrea Gallavotti, and Joseph R Ecker. Cistrome and epicistrome features shape the regulatory DNA landscape. *Cell*, 165(5):1280–1292, 2016.

[30] Shobhit Gupta, John A Stamatoyannopoulos, Timothy L Bailey, and William Stafford Noble. Quantifying similarity between motifs. *Genome biology*, 8:1–9, 2007.

[31] Kathleen M Chen, Aaron K Wong, Olga G Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature genetics*, 54(7):940–949, 2022.

[32] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12963–12971, 2021.

[33] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 06–11 Aug 2017.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.