Look at the Loss: Towards Robust Detection of False Positive Feature Interactions Learned by Neural Networks on Genomic Data

Mara Finkelstein[+ 1] **Avanti Shrikumar**[+* 1] **Anshul Kundaje**[* 1]

Abstract

Gene expression is modulated by cooperative binding of regulatory proteins called transcription factors (TFs) to DNA sequences. Recent work has demonstrated that neural networks show promise at identifying candidate pairs of TFs that have super-additive or sub-additive interaction effects, but the reliability of these predicted interactions remains unclear. Here, we design a simulated dataset to study the propensity of neural networks to learn false positive interactions. We find that feature interaction scores obtained from popular neural network architectures trained with multiple random initializations are consistently prone to identifying false positive interactions with large predicted interaction effects, and that previously-proposed null distributions based on the effect size of the interaction scores do not adequately control for false positives even if combining results across different model architectures. Instead, we find that the contribution of an interaction effect to the prediction loss - rather than the magnitude of the interaction itself - is a far more robust indicator of whether an interaction is likely to be real. Coupled with checking for consistency across different model architectures, our proposed tests based on loss improvement can reliably distinguish between positive and negative controls in our simulated data. To our knowledge, these are the first proposed statistical tests for detecting false positive interactions that leverage improvement in prediction loss on held-out data. We also perform analysis to shed insight on why models may learn large interaction effects in the absence of a ground-truth interaction. Anonymized code + trained models to replicate results available at https://github.com/kundajelab/feature_interactions.

1. Introduction

The cell-type specific activity of genes is precisely determined by the binding of regulatory proteins called Transcription Factors (TFs) to specific sequences in segments of genomic DNA called "regulatory elements". In recent years, Convolutional Neural Network (CNN) models trained to predict TF binding from genomic sequence have revealed insights into the "regulatory language of the genome" (Alipanahi et al., 2015; Zhou & Troyanskaya, 2015; Kelley et al., 2016; Avsec et al., 2020). Typically, these models take a one-hot encoded regulatory DNA sequence as input and try to predict the experimentally-measured TF binding signal as output. These models learn to recognize short (∼6-20 base-pair) DNA patterns (a.k.a. "motifs") in the input sequence that the TFs bind to. As TFs typically bind in a highly cooperative manner (Lambert et al., 2018), these models also learn *interactions* between motifs, whereby motifs contribute super-additively (in the case of synergistic TFs) or sub-additively (in the case of competition or inhibition between TFs) to the predicted output. Recent efforts have proposed techniques to infer these interaction scores from the trained models. Tsang et al. (2017) extracted feature interactions using feedforward neural network weights - however, their method was designed for tabular inputs and does not obviously generalize to the CNN architectures used for regulatory DNA sequence. More recently, Avsec et al. (2020); Greenside et al. (2018); Liu et al. (2019); Ullah & Ben-Hur (2020) proposed techniques to learn interactions from models of regulatory DNA sequence. However, the susceptibility of these techniques to picking up false positive interactions has not been extensively studied. While both Greenside et al. (2018) and Ullah & Ben-Hur (2020) used simulated datasets to show that motifs lacking predictive power are successfully identified as having no interaction, neither study considered a simulated negative control where motifs *do* have predictive power but contribute *independently* to the output (and should thus also be identified as having no interaction). This is discussed further in **Sec. A**.

In this work, we use simulations to study the behavior of neural networks on a negative control dataset where each motif independently contributes to the output. Surprisingly, we find that typical neural network architectures trained on this control dataset frequently predict large interaction effects between motifs, and previously-proposed null distributions that consider the effect size of the learned interaction (Greenside et al., 2018; Ullah & Ben-Hur, 2020) do not ad-

[+]Co-first Authors. [*]Co-Corresponding Authors. [1]Department of Computer Science, Stanford. Contact information: AS <avanti@cs.stanford.edu>, AK <akundaje@stanford.edu>, MF <marafinkelstein@gmail.com>.

equately control for this. Although Friedman et al. (2008) devised a general-purpose method for creating an empirical null distribution for interaction effects by training multiple models on synthetic altered datasets that lack ground-truth interactions, this approach is computationally expensive as each point in the empirical null distribution requires training a new model, and thus it has not been adopted for neural networks in genomics. Further, even when a model learns a strong interaction, it can be unclear whether this interaction benefits model predictions on held-out data (e.g. the interaction could be the result of overfitting or of convergence to a sub-optimal solution).

Motivated by the intuition that learning a true interaction should improve model predictions on unseen data, we explore statistical tests for whether the interaction effect significantly improves model prediction loss on held-out data. We identify a test that enables robust detection of false positive interactions while maintaining sensitivity to true interactions across different architectures and initializations in our simulated dataset. The core insight of our approach - to test for a significant improvement in loss on held-out data - can be applied to domains other than genomics. We hope our method brings us closer to reliable application of neural networks to derive novel insights from scientific data.

2. Methods

2.1. Simulation setup

We devised a regression task using a simulated dataset where motifs could contribute independently to the output. The simulation workflow is described in **Fig. 1**. Note that although the underlying inputs are the same for both the positive and negative control, the labeling scheme differs; for the negative control, the expected counts are linear in the sum of binding probabilities (implying additive/independent contributions between motifs), while for the positive control they are super-additive.

2.2. Model architectures and training

Prior to fitting models, we applied the Anscombe transform (Anscombe, 1948), given by $g(x) = 2\sqrt{x + 3/8}$, to the count labels in order to reduce the dynamic range of the counts and facilitate model training. Note that in the genomics literature, it is common to apply a transform to the raw counts prior to model fitting (e.g. a log transform in Avsec et al. (2020); Kelley (2019); Phuycharoen et al. (2020)). Because the Anscombe transform is a variance-stabilizing transform that converts Poisson noise to Gaussian noise, we trained our models using the mean squared error loss (which is suitable for Gaussian noise). We considered three different types of CNN model architectures with different numbers of layers, hidden units and filter widths. Each model architecture was trained with 3 different L1 regularization weights, and each of the 9 combinations of model architecture and regularization was trained with five different random seeds. This resulted in 45 models for the positive control and negative control datasets, for a total of 90 trained models. Details on the architectures and training are in **Sec. C**. As explained in **Sec. 3.1 & F**, to study the impact of padding we trained an additional 90 models with 'same' padding rather than the default 'valid' padding.

2.3. Computing interaction effects

Following standard convention, we defined the motif interaction effect as the difference between the joint effect and the sum of the main effects. Formally, let s_{GT} de-

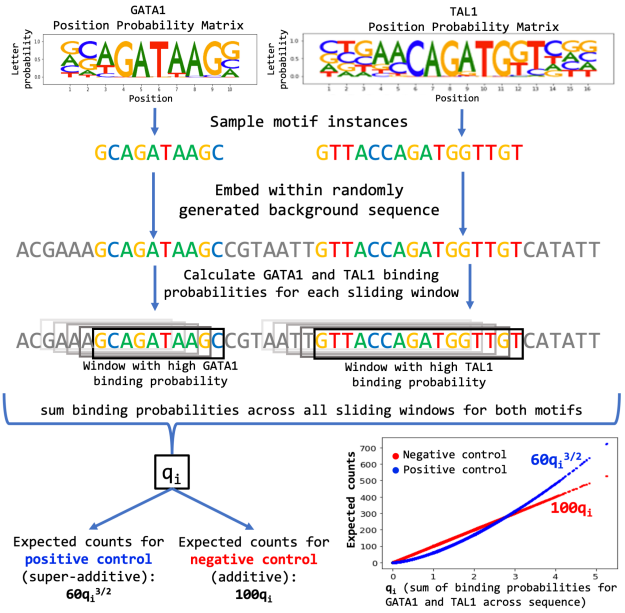

Figure 1. **Simulated sequence generation workflow.** GATA1 and TAL1 motif instances were sampled from their respective Position Probability Matrices and embedded randomly within a sequence background of length 100bp. 25% of sequences contained one embedded GATA1 motif, 25% contained one embedded TAL1 motif, another 25% contained one instance each of GATA1 and TAL1, and the remaining 25% contained no explicitly embedded motifs (but could contain motif matches by chance due to the randomly sampled surrounding sequence). A binding probability for GATA1 and TAL1 was computed for each motif-length window in each sequence using the method from Zhao & Stormo (2011), and the binding probabilities across all sliding windows in a sequence for both GATA1 and TAL1 were summed to obtain the quantity q_i for sequence i. In the case of the negative control (i.e. additive contributions), we set the expected number of observed counts in the sequence i to be $100q_i$, while for the positive control (where there is super-additivity) we set the expected counts to $60q_i^{3/2}$. Observed labels were then sampled from a Poisson distribution with λ equal to the expected counts. 100K sequences were generated (50K training, 50K test); the training set was further split into 40K for training and 10K for validation. See **Sec. B** for more details.

note a sequence containing both GATA1 and TAL1 motifs, and let s_G, s_T and $s_\varnothing$ denote perturbed sequences in which only the TAL1 motif, only the GATA1 motif, and both TAL1 and GATA1 motifs have been "knocked out" respectively (a motif is "knocked out" by replacing it with a random sequence that is not a good motif match). Let $f(s)$ denote the model prediction on sequence s. The main effect contribution M_G of GATA1 is calculated as $M_G = f(s_G) - f(s_\varnothing)$, while the main effect contribution M_T of TAL1 is $M_T = f(s_T) - f(s_\varnothing)$. The joint contribution of both motifs is $J_{G,T} = f(s_{GT}) - f(s_\varnothing)$. The interaction effect $I_{G,T}$ is defined as $I_{G,T} = J_{G,T} - (M_G + M_T)$. Analogous approaches to scoring feature interactions were considered in Avsec et al. (2020); Greenside et al. (2018).

All interaction effects were calculated *after* mapping the model predictions in the Anscombe-transformed space back to the original counts space. This is consistent with recent work (Sanford et al., 2020) that tested for interaction effects between biological stimuli using quantities proportional to the count-based outputs of various experimental assays.

2.4. Selecting motif pairs to study for interaction effects

We considered the 25% of simulated sequences that contained embedded TAL1 and GATA1 motifs, and filtered for those sequences where both motif instances in the embedded pair were strong matches to the motif (match score log-odds > 2 relative to background). We used these pairs to study the interaction effect between TAL1 and GATA1. We enforced that the motifs in the pair be separated by at least the length of the longer motif to rule out cases where the motif instances jointly create a weak motif match due to their proximity (and would thus have a nonzero ground-truth interaction effect even in the case of the negative control). This resulted in 7,971 motif pairs from the 50K test sequences and 6,451 motif pairs from the 40K training set sequences. Note that a sequence can contain additional strong motif matches besides the pair considered due to the chance appearance of motifs in the background; however, the interaction effect was calculated only between the motifs in the embedded pair.

3. Results

3.1. The strength of an interaction effect is not a reliable predictor of ground truth interactions

Following Greenside et al. (2018) and Ullah & Ben-Hur (2020), we generated a "null distribution" dataset consisting of dinucleotide-preserving shuffled versions of the test sequences, and calculated the interaction effect between the locations that used to contain motifs in the original test sequences. The intuition behind this null distribution is that shuffled sequences are unlikely to contain strong motifs. We compared the magnitudes of the predicted interaction effects (**Sec. 2.3**) between the motif pairs from the test set to the predicted interaction effects from this shuffled-sequence null. Strikingly, we found that even for the negative control, the magnitudes of the predicted interactions between motifs in the test-set were significantly larger than the null (**Fig. 2**, top left). Curious why this might be, we plotted the predicted interaction effect $I_{G,T}$ against the sum of the

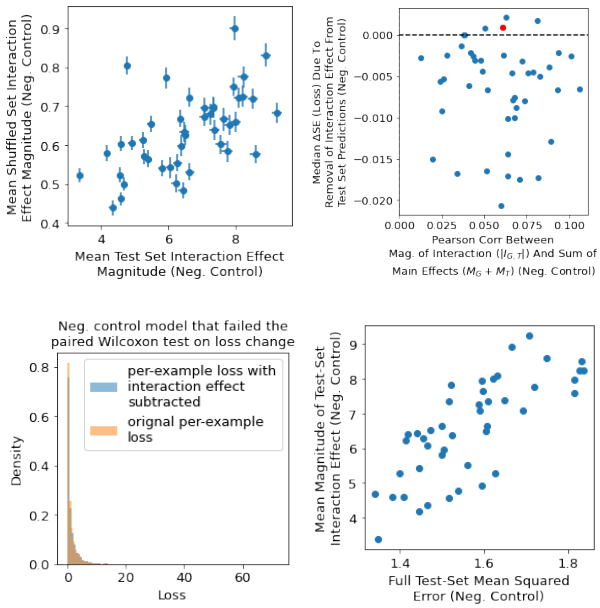

Figure 2. **Improvement in loss is a more reliable indicator than magnitude of interaction effect of whether an interaction is likely real.** All plots are for the negative control dataset (for which there is no ground-truth interaction) with models trained using 'valid' padding. **Top Left:** For all 45 models, magnitude of interactions on test set sequences greatly exceeds magnitude on shuffled sequence control as measured by a one-sided unpaired Mann-Whitney U test. **Top Right:** larger magnitude interactions are predicted for motif pairs that have large predicted main effects (positive Pearson correlation on x-axis), and the learned interaction is often detrimental to the median prediction loss (y-axis); point marked in red is the sole model where subtracting the interaction effect significantly worsens the loss compared to the original predictions according to a paired wilcoxon test (median loss improvement is 9e$-$4, $p = 0.011$). **Bottom Left:** histogram comparing original prediction loss with interaction effect included (orange) to prediction loss with interaction effect subtracted (blue) over 7,971 motif pairs in the test set for the model in red from the top-right scatterplot. The two distributions are not significantly different according to an unpaired Mann-Whitney U test ($p > 0.3$), even though they are different according to the paired test. **Bottom Right**: mean magnitude of interaction effect (computed over 7,971 motif pairs in test set) is inversely correlated with mean-squared-error calculated over all 50K test-set sequences (Spearman r = 0.78). See **Sec. E** for corresponding plots on positive control data and **Sec. F** for plots for models trained with 'same' padding rather than 'valid' padding.

predicted main effects $M_G + M_T$ (**Fig. D.1**). We observed that, across models, the largest interaction effects (which were consistently negative; **Fig. D.1**) were observed for motif pairs with large main effects, resulting in a positive correlation between the magnitudes of interaction effects & the main effects (**Fig. 2, top right**).

Surprisingly, removing the interaction effect from the predictions (i.e. replacing the prediction $f(s_{GT})$ with $f(s_{GT}) - I_{G,T}$) often tended to improve the average test-set loss on the negative control (**Fig. 2, top right**). Along with the fact that the magnitude of the learned interactions was inversely correlated with model performance for the negative control (**Fig. 2, bottom right**), this suggests the false-positive interactions results from convergence to a sub-optimal solution on the training set. Note that the models are not fit to the original raw counts; instead, models are fit to the Anscombe transformation of the counts, while the interaction effect is computed in the original counts space (**Sec. 2.2**). Any error in the learned interaction in the Anscombe-transformed space translates to an artifactual interaction in the raw counts space. This is especially important given that transforming raw counts prior to model fitting is standard practice in regulatory genomics literature (Avsec et al., 2020; Kelley, 2019; Phuycharoen et al., 2020), even though the original count-based output may be the appropriate space to test interaction effects between biological stimuli (Sanford et al., 2020).

We also noticed the predicted interactions in the negative control tended to be strongest when motif instances were near the edges of the sequence (**Fig. F.1**). We speculated that the choice of "valid" padding in the conv layers (which is the default in keras and is used in many genomics papers) meant that the net may has difficulty identifying motif instances near the ends of sequences, and is thus learning artifactual interactions to compensate (this is explained further in **Sec. F**). We repeated experiments using "same" padding throughout, which removed the correspondence between motif position and interaction strength; however, we still found that strength of interaction did not reliably predict ground truth interactions **Fig. F.2**.

3.2. Impact of interaction effect on loss can distinguish simulated positive and negative controls.

For both 'same' and 'valid' padding, we first compared the test-set model loss on the original predictions to the test-set model loss on the predictions with the interaction effect $I_{G,T}$ removed using both a one-sided *paired* Wilcoxon test. At a p-value threshold of 0.05, this test identified all 45 models trained on the positive control data as having highly beneficial interactions. For the negative control, the paired test correctly identified 44/45 'valid' padding models and 26/45 'same' padding models as having no significantly beneficial interaction, suggesting that one way of identifying false-positive interactions is to verify that a learned interaction *consistently* improves model loss on held-out data across different architectures. We also observed that models with significantly beneficial interactions in the case of the negative control appeared to use the interactions to compensate for mis-predictions in the main effects (**Fig. D.3** & **Fig. F.6**); combined with the inverse correlation between model performance and magnitude of interaction effect (**Fig. 2, bottom right & Fig. F.2, bottom right**), this suggests that selecting for models that achieve the best performance may also be a way to decrease the likelihood of encountering false-positive interactions.

Interestingly, the *unpaired* Mann-Whitney U test at a p-value threshold of 0.05 perfectly classified all models trained on positive and negative controls for both 'same' and 'valid' padding. Consistent with this, we observe that even when the interaction effect tends to improve the loss on individual examples, the improvement is small compared to the overall variation in loss (**Fig. 2, bottom left & F.2, bottom left**). Implicitly, the unpaired test compares the loss improvement against a null generated by resampling the original examples. We also explored an empirical null distribution based on the interaction effect between randomly chosen locations in the original sequences, but this null didn't work as well as the implicit null of the unpaired test (**Sec. G**).

Finally, we observed that the increase in loss from removing interaction effects tended to be consistently higher on training data compared to held-out data (**Fig. D.2 & F.7**). This suggests that overfitting can also help explain some artifactual interaction effects - thus it is important to study the interaction effects on held-out data.

4. Discussion

We applied a standard definition of feature interactions to CNN architectures commonly used in regulatory genomics, and made the surprising finding that these models consistently identify strong false positive interactions when applied to simulated DNA sequences lacking ground-truth motif feature interactions. Our analysis suggests this was often due to convergence of the models to sub-optimal solutions (**Fig. 2, bottom right & Fig. D.1**), but could also be due to over-fitting (**Fig. D.2 & F.7**) or models that use interactions to compensate for mis-predictions in the main effects (**Fig. D.3**). We find one way to alleviate this problem is to check whether a learned interaction significantly improves prediction loss on held-out data across different model architectures. Alternative architectures, loss functions and optimization methods that stabilize models, avoid suboptimal minima and/or regularize feature attributions & interaction scores (Tseng et al., 2020; Avsec et al., 2020) may also improve the reliability of detected feature interactions. We hope this work brings us closer to robust application of neural networks to make discoveries in genomics.

5. Author Contributions

AS conceived of the idea of identifying false positive interactions by testing for a significant improvement in the loss on held-out data. AS & MF designed and conducted experiments. AK provided guidance and feedback. AS, MF & AK wrote the manuscript.

6. Acknowledgements

We would like to thank Kelly Cochran for feedback on the manuscript.

References

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

Anscombe, F. J. The transformation of poisson, binomial and Negative-Binomial data. *Biometrika*, 35(3/4):246–254, 1948.

Avsec, Ž., Weilert, M., Shrikumar, A., Krueger, S., Alexandari, A., Dalal, K., Fropf, R., McAnany, C., Gagneur, J., Kundaje, A., et al. Deep learning at base-resolution reveals cis-regulatory motif syntax. *bioRxiv*, pp. 737981, 2020.

Friedman, J. H., Popescu, B. E., et al. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2 (3):916–954, 2008.

Greenside, P., Shimko, T., Fordyce, P., and Kundaje, A. Discovering epistatic feature interactions from neural network models of regulatory dna sequences. *Bioinformatics*, 34(17):i629–i637, 2018.

Kelley, D. R. Cross-species regulatory sequence activity prediction. June 2019.

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

Keras. Keras. https://github.com/keras-team/keras/, 2020.

Kheradpour, P. and Kellis, M. Systematic discovery and characterization of regulatory motifs in encode tf binding experiments. *Nucleic acids research*, 42(5):2976–2987, 2014.

Lambert, S. A., Jolma, A., Campitelli, L. F., Das, P. K., Yin, Y., Albu, M., Chen, X., Taipale, J., Hughes, T. R., and Weirauch, M. T. The human transcription factors. *Cell*, 172(4):650–665, February 2018.

Liu, G., Zeng, H., and Gifford, D. K. Visualizing complex feature interactions and feature sharing in genomic deep neural networks. *BMC Bioinformatics*, 20(1):401, July 2019.

Maslova, A., Ramirez, R. N., Ma, K., Schmutz, H., Wang, C., Fox, C., Ng, B., Benoist, C., Mostafavi, S., and . Learning immune cell differentiation. *bioRxiv*, 2019. doi: 10.1101/2019.12.21.885814. URL https://www.biorxiv.org/content/early/2019/12/23/2019.12.21.885814.

Movva, R., Greenside, P., Marinov, G. K., Nair, S., Shrikumar, A., and Kundaje, A. Deciphering regulatory dna sequences and noncoding genetic variants using neural network models of massively parallel reporter assays. *PLOS ONE*, 14(6):1–20, 06 2019. doi: 10.1371/journal.pone.0218073. URL https://doi.org/10.1371/journal.pone.0218073.

Phuycharoen, M., Zarrineh, P., Bridoux, L., Amin, S., Losa, M., Chen, K., Bobola, N., and Rattray, M. Uncovering tissue-specific binding features from differential deep learning. *Nucleic Acids Res.*, 48(5):e27, March 2020.

Sanford, E. M., Emert, B. L., Coté, A., and Raj, A. Gene regulation gravitates towards either addition or multiplication when combining the effects of two signals. May 2020.

Tsang, M., Cheng, D., and Liu, Y. Detecting statistical interactions from neural network weights. May 2017.

Tseng, A. M., Shrikumar, A., and Kundaje, A. Fourier-transform-based attribution priors improve the interpretability and stability of deep learning models for genomics. *bioRxiv*, 2020. doi: 10.1101/2020.06.11.147272. URL https://www.biorxiv.org/content/early/2020/06/12/2020.06.11.147272.

Ullah, F. and Ben-Hur, A. A Self-Attention model for inferring cooperativity between regulatory features. February 2020.

Zhao, Y. and Stormo, G. D. Quantitative analysis demonstrates most transcription factors require only simple models of specificity. *Nature biotechnology*, 29(6):480–483, 2011.

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

Appendix

A. Limitations of Previous Feature Interaction Studies on Simulated Genomic Sequences

Previous analyses (Greenside et al., 2018; Ullah & Ben-Hur, 2020) studied a simulated binary TF binding classification task using a dataset configured as follows: the positive set consisted of synthetic DNA sequences containing motifs of the ELF1 and SIX5 TFs. The negative set consisted of synthetic DNA sequences containing motifs for either ELF1 or SIX5 (but not both). Further, motifs of the AP1 and TAL1 TFs were randomly embedded across the positive and negative sequences. These analyses demonstrated that the significant interactions inferred from the models did not include the AP1 or TAL1 motifs. However, the AP1 and TAL1 motifs were not predictive features i.e. not necessary for the model to discriminate between the two sets of sequences. A simulated dataset where two motifs are indeed important for the prediction task, but do not have an interaction, has not been studied prior to this work.

B. Simulation Details

A Position Probability Matrix (PPM) is a commonly-used representation of DNA sequence motifs that specifies the probability of observing a given nucleotide (one of A,C,G or T) at a given position. A PPM for a motif of length L is a 4 x L matrix where the rows represent the 4 nucleotides and the columns represent each position in the motif. The probabilities in each column sum to 1. See `https://en.wikipedia.org/wiki/Position_weight_matrix#Conversion_of_sequence_to_position_probability_matrix` for more details.

In our simulation, we used the the `GATA_disc1` PPM as the motif for the GATA1 TF, and the `TAL1_known1` PPM as the motif for the TAL1 TF. Both PPMs are available from ENCODE (Kheradpour & Kellis, 2014) and can be downloaded at `http://compbio.mit.edu/encode-motifs/`.

The 100 base-pair (bp) sequences in our dataset were created by randomly sampling nucleotides (A,C,G,T). A single instance of a GATA1 motif (sampled from its PPM) was embedded in 25% of the sequences at a random position, a single instance of a TAL1 motif was embedded into another 25%, one instance each of GATA1 and TAL1 were embedded into another 25%, and the remaining 25% contained no explicitly embedded motifs (but could contain motif matches by chance due to the randomly sampled surrounding sequence).

The generated sequences were then labeled under two different schemes: one in which the motifs contributed super-additively to the output (the positive control), and one in which the motifs contributed additively/independently to the output (the negative control). To generate the labels, the PPMs were converted to log-odds PWMs (`https://en.wikipedia.org/wiki/Position_weight_matrix#Conversion_of_position_probability_matrix_to_position_weight_matrix`) using the background frequencies for A, C, G, & T (which were set to be 27% A, 23% C, 27% T and 23% G). The log-odds PWMs were then multiplied by -1 and treated as $\Delta\Delta G$ PWMs. Following Zhao & Stormo (2011), the occupancy probabilities for GATA1 and TAL1 at each sliding window were computed by calculating the total $\Delta\Delta G$ for each window using the corresponding PWM and transforming it to a probability with the formula $1/(1+e^{\Delta\Delta G+\mu})$ (we set $\mu = 0$). The GATA1 and TAL1 binding probabilities over all sliding windows in a sequence were then summed to obtain the quantity q_i for sequence i. To simplify the simulation, reverse-complements were not scored. In the case of the negative control (i.e. no interaction), we set the expected number of observed counts in the sequence i to be $100q_i$, while for the positive control (where there is synergy) we set the expected counts to $60q_i^{3/2}$. Observed labels were then sampled from a Poisson distribution with λ equal to the expected counts. See **Fig. 1** for a an illustration.

C. Model Architecture Details

We considered three different types of model architectures. The first architecture ("arch1") consisted of 4 convolutional layers (each with 15 filters and kernel width 7), followed by global average pooling, followed by two dense layers with 50 hidden units, followed by the output layer. The second architecture ("arch2") resembled arch1, but with three dense layers of 30 hidden units. The third architecture ("arch3") resembled arch2, but with 5 convolutional layers of kernel width 5. All layers used ReLU nonlinearities except for the final output layer (which was linear). ReLU layers were initialized with the `he_normal` initialization scheme in Keras (Keras, 2020). Models were trained with the Adam optimizer and early stopping.

Each model architecture was trained with 3 different L1 weight regularizations levels: 0.0, 10^{-4} and 10^{-3}. Regularization was applied to all layers except the output layer. Each of the 9 combinations of model architecture and regularization was trained with five different random seeds. This resulted in 45 models for the positive control and negative control datasets, for a total of 90 trained models. All models achieved good MSE on the test set of around 1.8 (**Fig. 2 (bottom right) and E.3**).

D. Additional Figures for Models Trained on Negative Control Data ('Valid' Padding)

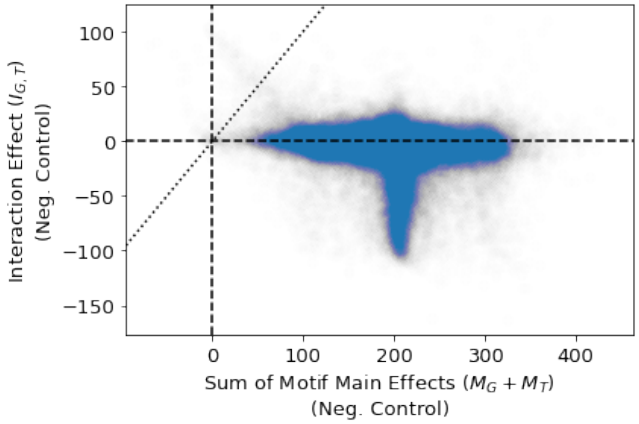

Figure D.1. **Largest-Magnitude Interaction Effects in Negative Control Correspond to Motif Pairs With Large Main Effects**. Each point represents a specific test-set example for a specific trained model; there are a total of 45×7971 points. $M_G + M_T = 200$ corresponds to two strong motif matches (one for TAL, one for GATA). Dotted line indicates x=y line. Large negative interaction effects are observed when the cumulative main effect is around 200. Because the original sequences are much more likely to contain strong motif matches compared to the corresponding locations in shuffled sequences, this explains why the mean magnitude of the interaction effect on the test set motif pairs greatly exceeds the mean magnitude in the corresponding shuffled sequences. Note that although there was no ground-truth interaction in the simulated raw counts space, the model was fit to the variance-stabilizing Anscombe transformation of the counts (we map the model output back to the raw counts space prior to computing interaction effects). For the model to correctly predict no interactions in the raw counts space, it must learn a negative (non-synergistic) interaction in the Anscombe-transformed space. Any error in the learned interaction in the transformed space manifests as an artifactual interaction in the raw counts space. As it is standard practice to transform raw counts (e.g. using a log transform) prior to fitting regression models on them, this analysis highlights the importance of carefully accounting for the transformations applied prior to calculating interaction effects.

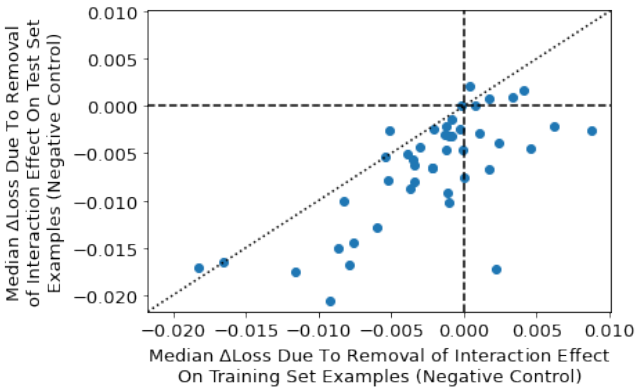

Figure D.2. **Some interaction effects can be the result of overfitting.** The median increase in the loss from excluding interactions in the training set tends to be higher than the median increase in the loss from excluding interactions in the test set.

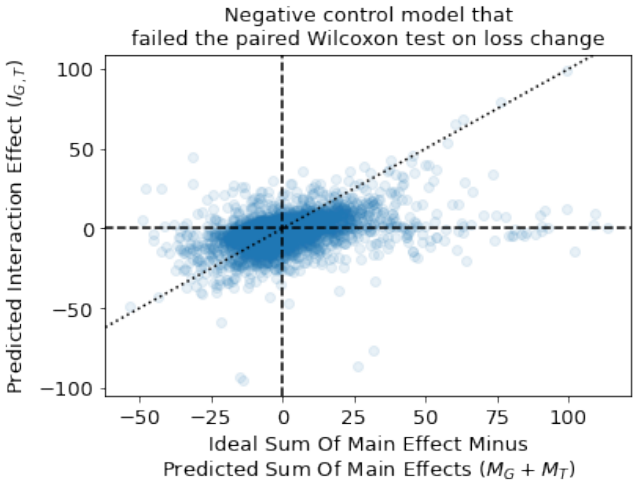

Figure D.3. **When the interaction effect consistently benefits prediction loss on the test set for the negative control, the model appears to use the interaction effect to compensate for mis-prediction in the main effect**. Shown is a scatterplot for error in the main effect prediction vs. the model's predicted interaction effect for the one model trained with valid padding on the negative control data that failed the paired Wilcoxon test for the loss change (this is the same model that was highlighted in red in **Fig. 2, top right**). The error in the main effect was calculated by subtracting the predicted sum of the main effects ($M_G + M_T$) from the ideal sum of the main effects according to an oracle model. Positive correlation between the predicted interaction effect and the main effect error indicates that the interaction effect is compensating for a mis-prediction in the main effects. This may help explain why removing the interaction effect worsens the prediction loss on held-out data, despite the absence of a ground-truth interaction.

E. Figures for Models Trained on Positive Control Data ('Valid' Padding)

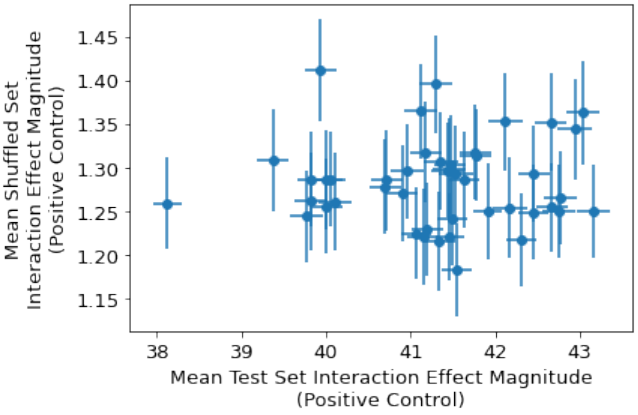

Figure E.1. **Counterpart of Fig. 2, top left, but on positive control data rather than negative control data**. As expected, the magnitudes of interaction effects between the original motif pairs greatly exceeds the magnitudes of interaction effects on shuffled sequences.

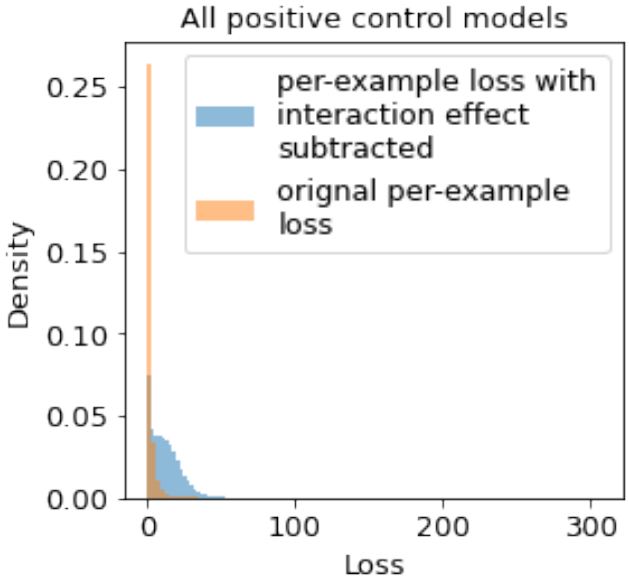

Figure E.2. **Counterpart of Fig. 2 (bottom left) but on positive control data rather than negative control data**. Histogram comparing original prediction loss with interaction effect included (orange) to prediction loss with interaction effect subtracted (blue) over all 45 positive control models for 7,971 motif pairs in the test set (i.e. the histogram was computed over 45×7971 points). Unlike for the models trained on the negative control, the two distributions in this case are visibly different.

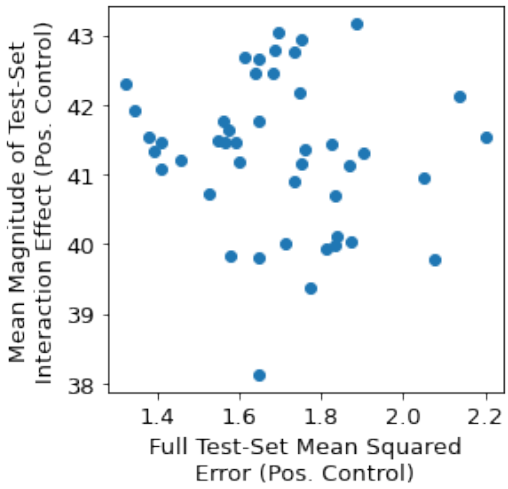

Figure E.3. **Magnitude of Inferred Interaction Effect is Not Inversely Correlated With Model Performance for the Positive Control**. Counterpart of **Fig. 2, bottom right**, but for the positive control dataset rather than the negative control dataset. X-axis: mean squared error of model predictions on 50K sequences from the test-set. Y-axis: mean magnitude of inferred interaction effect on 7,971 motif pairs in the test set (**Sec. 2.4**). Spearman correlation is -0.19.

F. Models Trained Using 'Same' Padding Rather than 'Valid' Padding

Say we have an input of length L that we would like to scan with a pattern detector that has width w. Absent any padding, there would be $L - w + 1$ complete windows of length w. Thus, if no padding is performed, the output from scanning with the pattern detector would have length $L - w + 1$. This situation is called 'valid' padding in the context of Convolutional Neural Networks (where the 'pattern detector' is a convolutional filter). If we would like the output to have the same length as the input, we can zero-pad the input on either side such that the padded input has length $L + w - 1$. Scanning the padded input with the pattern detector would result in an output that has length L.

Now consider the situation where the pattern detector, which is of length w, is able to recognize a motif of length l where $l < w$. Specifically, imagine the situation where the first l weights within the pattern detector are devoted to recognizing the motif, while the remaining $w - l$ weights have near-zero values. Now consider a motif instance that occupies positions $p...p + l$ in the input sequence. If $p > L - w$, then the pattern detector would be *unable* to identify this instance if valid padding is used, because it would not see the appropriate window that would cause it to recognize the motif. However, if same padding is used, then the pattern detector would successfully identify the motif instance.

When we initially conducted the simulation, we used 'valid' padding both during model training and when determining the ground-truth labels by scanning the sequences with the ground-truth PWMs. In the case of the negative control, we observed that this resulted in the network learning large interactions for motif instances that were near the ends of sequences (**Fig. F.1, left**). We speculated that this was because the network attempted to learn interactions between filters in order to compensate for the fact that 'valid' padding made it hard for the network to identify motif instances near the ends of sequences. Note that 'valid' padding is the default in many neural network packages, and many models trained on shorter genomic sequence inputs, such as Maslova et al. (2019) and Movva et al. (2019), could encounter motif instances near the flanks of sequences. For such models, our analysis suggests that it would be advisable to use 'same' padding to avoid learning artifactual interactions for motif instances near the ends of sequences.

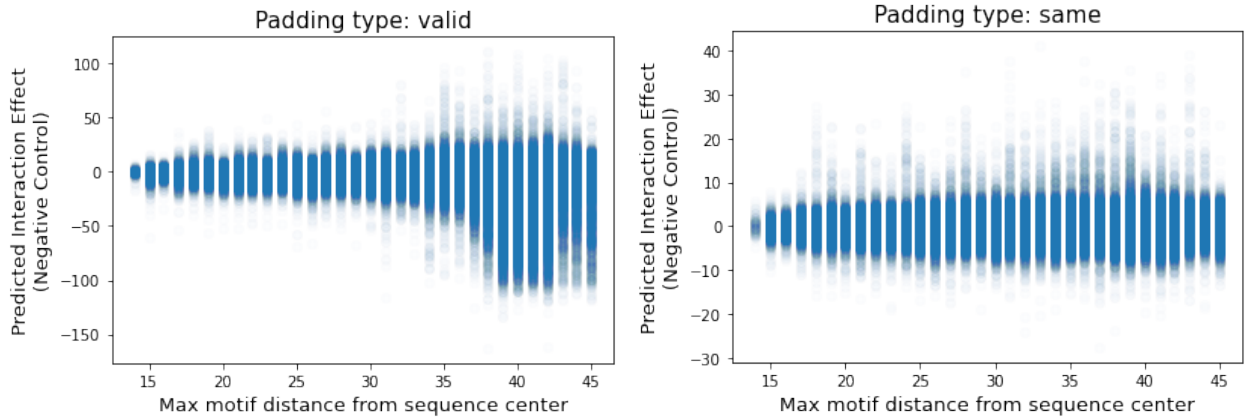

Figure F.1. **Dependence of interaction effect on motif position in the negative control dataset.** Each point in each scatterplot represents a specific test-set motif pair for a specific trained model; there are a total of 45×7971 points in each scatterplot (45 trained models, 7971 motif pairs). In the case of 'valid' padding (left), the model tends to predict strong negative interactions when the maximum distance of either motif from the center of the sequence is large. We hypothesize that this is because the model is learning interactions between multiple filters to compensate for the fact that valid padding makes it harder to identify motif instances near the ends of sequences. In the case of 'same' padding (right), this trend of strong interactions predicted between motif instances near the sequence ends is no longer apparent.

For completeness, we redid the simulations using 'same' padding - both during model training and when determining ground-truth labels by scanning sequences with the ground-truth PWMs. When we did this, we no longer observed the strong tendency of the network to learn large interactions for motif instances near the ends of sequences (**Fig. F.1, right**). Our qualitative results remained similar, in that:

1. All negative control models had significantly larger interactions on real data compared to a shuffled sequence null, indicating that the magnitude of interactions was not a reliable indicator of whether interactions were likely to be real

(**Fig. F.2, top left & top right**).

2. Even though all negative control models had significantly large interactions on real data compared to a shuffled sequence null, this interaction did not significantly improve model performance on held-out data according to a paired Wilcoxon test for a large fraction of models (26/45). For the remaining 19 negative control models that failed the paired Wilcoxon test, a scatterplot of the error in the main effect prediction against the learned interaction effect suggested that the interaction effect was being leveraged to compensate for mis-predictions in the main effect (**Fig. F.6**). As before, model performance on the negative control data was roughly inversely correlated with the average magnitude of the learned interaction effects, whereas no such inverse correlation was observed on the positive control data **Fig. F.2, bottom right & Fig. F.5**.

3. Using the unpaired Mann-Whitney U test instead of the paired Wilcoxon test on the negative control data identified all models as having learned no significantly beneficial interaction on held-out data (**Fig. F.2, bottom left**). We note, however, that this result may not generalize to all possible simulated settings, and so we recommend training multiple models to see if a learned interaction consistently improves performance across different architectures.

4. For all models trained on positive control data, the learned interactions significantly improved model performance on held-out data according to both the paired Wilcoxon test and the unpaired Mann-Whitney U test.

5. Improvement in prediction loss was higher on the training set compared to the heldout set, suggesting a role of overfitting (**Fig. D.2**).

In addition, we noticed that while the average learned interactions tended to be strongly negative for models trained with 'valid' padding, the sign of the average learned interactions was more mixed for models trained on 'same' padding data (**Fig. F.8**). This suggests that looking for consistency in the sign of the learned interactions across multiple trained models is another potential strategy for identifying false positive interactions, but (as the case of 'valid' padding shows), it is not completely reliable as there may be inductive biases due to the model architecture.

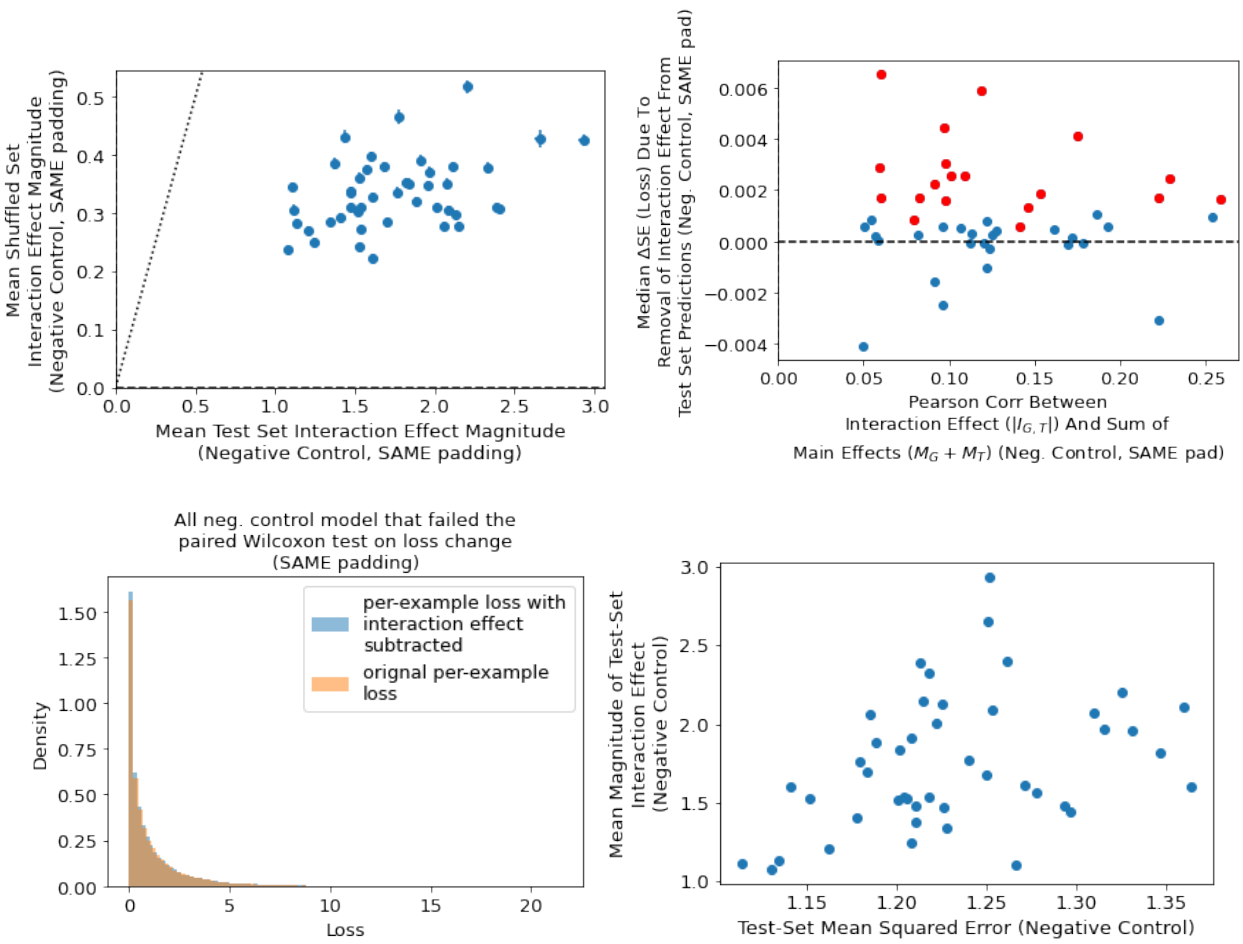

Figure F.2. **Counterpart of Fig. 2, but for same padding instead of valid padding**. **Top Left:** As before, for all 45 models trained on the negative control data, the magnitudes of the interaction effects on motif pairs from the original test-set sequences greatly exceeds the magnitudes of the interaction effects on a shuffled sequence control as measured by a one-sided unpaired Mann-Whitney U test (dotted line indicates the x=y line). Thus, the magnitude of an interaction effect is not a reliable indicator of whether an interaction is likely real, at least when compared to a null distribution derived from shuffled sequences. **Top Right:** As before, larger magnitude interactions are predicted for motif pairs that have large predicted main effects (positive Pearson correlation on x-axis), explaining why the magnitudes of interaction effects on real sequences greatly exceed the magnitudes on the shuffled sequence null. The learned interactions are also often detrimental to the median prediction loss (y-axis); points marked in red are the 19 models for which subtracting the interaction effect significantly worsens the loss compared to the original predictions according to a pairs Wilcoxon test at $p < 0.05$. **Bottom Left**: histogram comparing original prediction loss with interaction effect included (orange) to prediction loss with interaction effect subtracted (blue) over 7,971 motif pairs in the test set for all 19 models in red from the top-right scatterplot (i.e. the histogram was computed over 19×7971 points. The two distributions are not significantly different according to an unpaired Mann-Whitney U test ($p = 0.062$), even though they are very different according to a paired Wilcoxon test ($p < 1e-47$). **Bottom Right**: mean magnitude of interaction effect (computed over 7,971 motif pairs in test set) is roughly inversely correlated with mean-squared-error calculated over all 50K test-set sequences (Spearman r = 0.40, $p < 0.0069$). Corresponding figures for the positive control data are **Fig. F.3,**

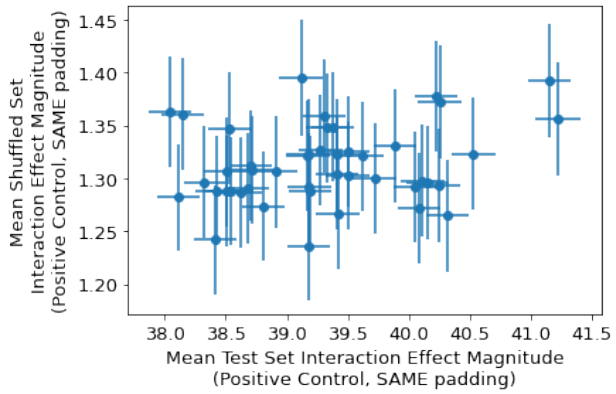

Figure F.3. **Counterpart of Fig. F.2 (top left) but on positive control data rather than negative control data.** As expected, the magnitudes of interaction effects between the original motif pairs greatly exceeds the magnitudes of interaction effects on shuffled sequences.

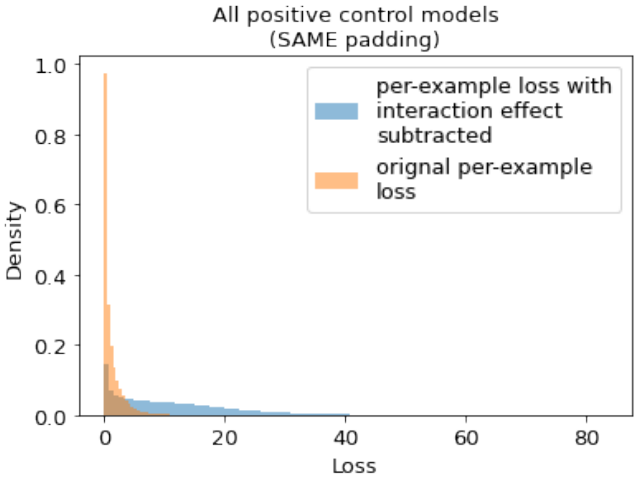

Figure F.4. **Counterpart of Fig. F.2 (bottom left) but on positive control data rather than negative control data**. Histogram comparing original prediction loss with interaction effect included (orange) to prediction loss with interaction effect subtracted (blue) over all 45 positive control models for 7,971 motif pairs in the test set (i.e. the histogram was computed over 45×7971 points). Unlike for the models trained on the negative control, the two distributions in this case are visibly different.

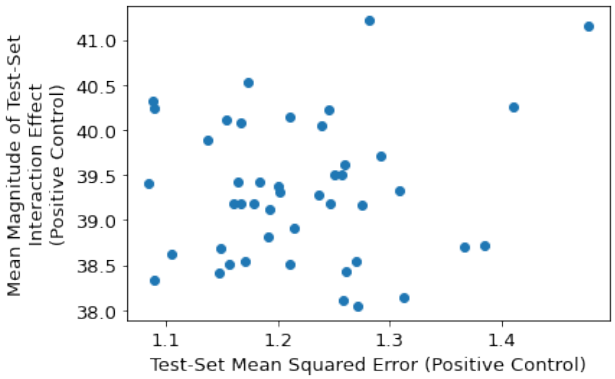

Figure F.5. **Magnitude of Inferred Interaction Effect is Not Inversely Correlated With Model Performance for the Positive Control (same padding)**. Counterpart of Fig. F.2 (bottom right), but for the positive control dataset rather than the negative control dataset. X-axis: mean squared error of model predictions on 50K sequences from the test-set. Y-axis: mean magnitude of inferred interaction effect on 7,971 motif pairs in the test set (**Sec. 2.4**). Spearman correlation is -0.0018.

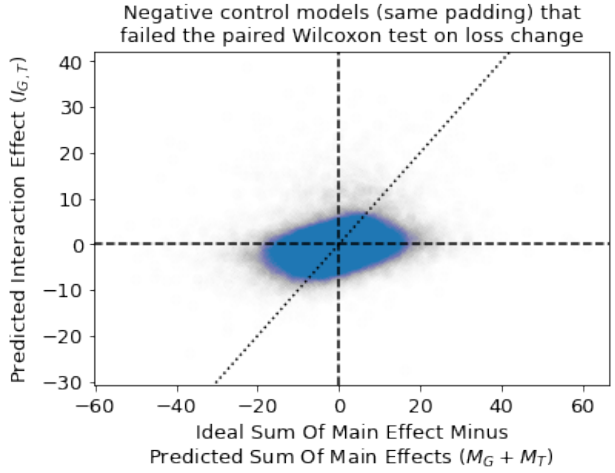

Figure F.6. **Corresponding scatterplot of Figure D.3, but for same padding rather than valid padding**. Shown is a scatterplot for error in the maineffect prediction vs. the model's predicted interaction effect for the 19 models trained with valid same on the negative control data that failed the paired Wilcoxon test for the loss change (these are the same models that are highlighted in red in **Fig. F.2, top right**). The error in the main effect was calculated by subtracting the predicted sum of the main effects $(M_G + M_T)$ from the ideal sum of the main effects according to an oracle model. Positive correlation between the predicted interaction effect and the main effect error indicates that the interaction effect is compensating for a mis-prediction in the main effects. This may help explain why removing the interaction effect worsens the prediction loss on held-out data, despite the absence of a ground-truth interaction.

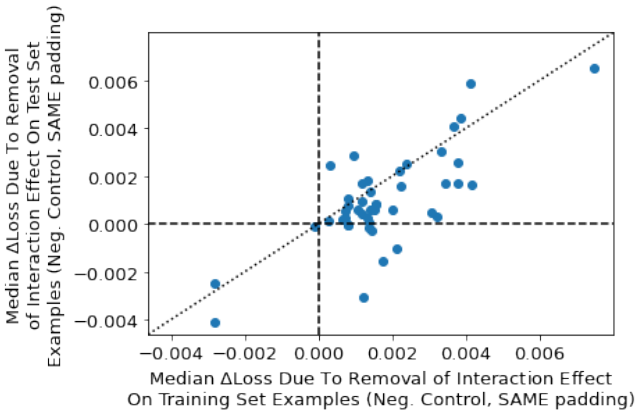

Figure F.7. **Counterpart of Fig. D.2, but for same padding rather than valid padding.** The median increase in the loss from excluding interactions in the training set tends to be higher than the median increase in the loss from excluding interactions in the test set.

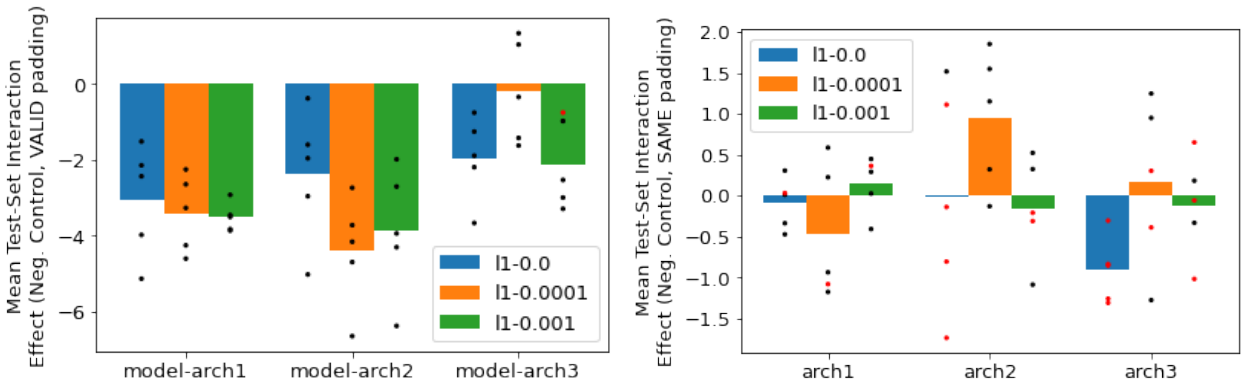

Figure F.8. **Mean interaction effect learned by models trained on negative control data.** Each point represents the average test set interaction effect (computed over 7971 motif pairs) for a single trained model. Note that, unlike in **Fig. 2 (bottom right)**, in this figure we *do not* take the absolute value of the interaction effect prior to averaging. With valid padding (left), the average interaction effect has a negative sign for all but two models. With 'same' padding (right), the sign of the average interaction effect is more varied. Red points indicate models for which the interaction effect significantly improved the loss on test-set examples according to the paired Wilcoxon test at $p < 0.05$. Model architectures are described in **Sec. C**.

G. Empirical Null Distribution For Loss Improvement

We explored a way to generate an empirical null distribution to test whether the increase in loss due to removal of an interaction effect is statistically significant. The approach we took was as follows: rather than looking at the interaction effect between the actual motif locations in the original sequences, we considered the interactions between two randomly chosen locations within the original sequences that were still separated by at least the length of the longer motif. The intuition behind using this for a null distribution is that two randomly chosen positions are unlikely to contain a pair of interacting motifs. While this choice of empirical null distribution worked well in the case of 'valid' padding, it did not work as well for 'same' padding, as illustrated in **Fig. G.1**.

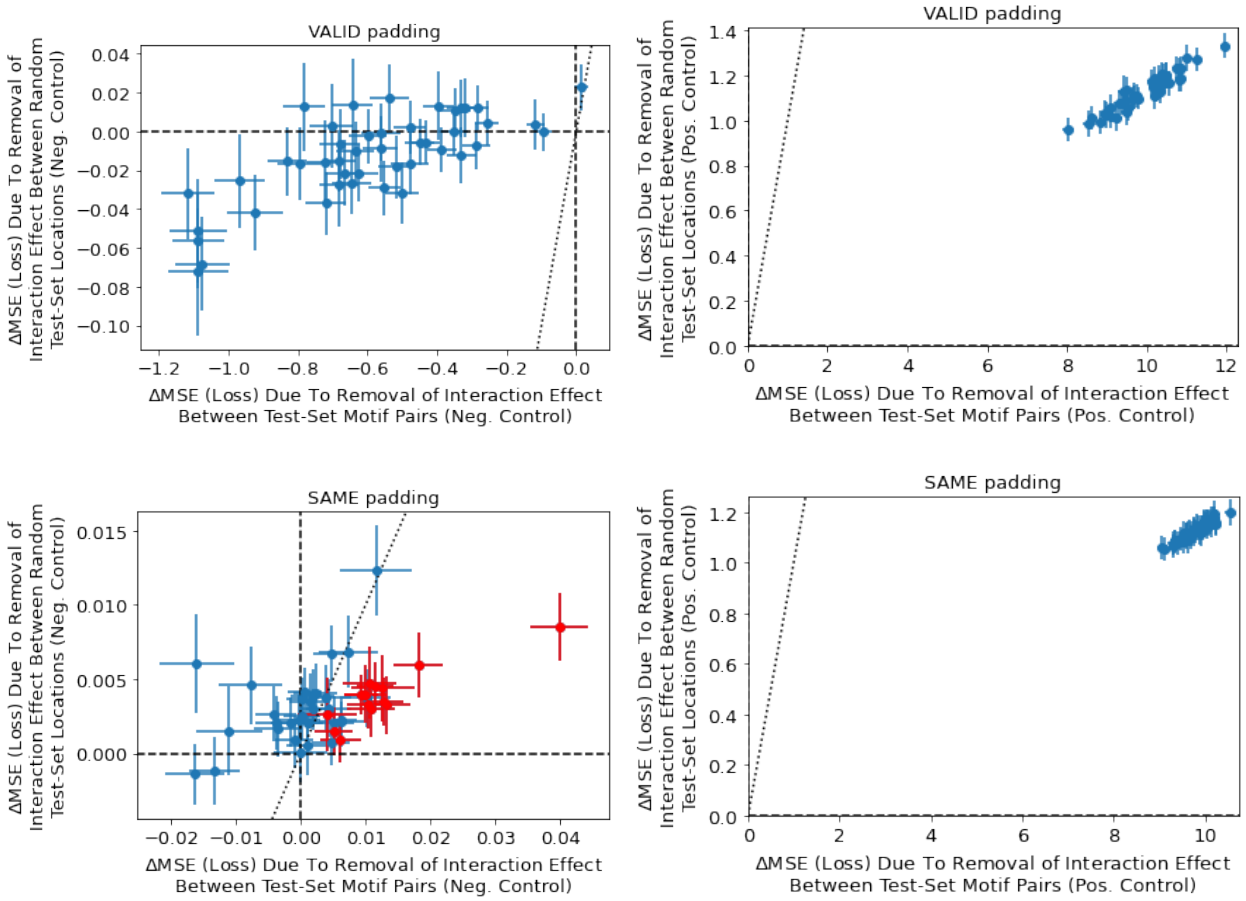

Figure G.1. **Results from comparing change in loss due to removal of interaction effect between motif pairs in test set to an empirical null consisting of change in loss due to removal of interaction effect between random positions in the test-set sequence**. Each point consists of one of 45 models trained on the respective dataset (top left: negative control with valid padding, top right: positive control with valid padding, bottom left: negative control with same padding, bottom right: positive control with same padding). X-axis shows increase in MSE from excluding the interactions between 7,971 motif pairs in the test set (**Sec. 2.4**), and y-axis shows the increase in MSE from excluding interactions between randomly chosen locations in the test-set sequences. Error bars indicate the standard error of the mean. We tested whether the squared error from excluding interactions between motif pairs in the test set was significantly higher compared to the increase in squared error from excluding interactions between randomly chosen locations in the test-set using a one-sided unpaired Mann-Whitney U test. For all models trained on the positive control data, the difference was highly significant. For all models trained on the negative control data with valid padding, the difference was not statistically significant. However, for 14/45 models trained on the negative control data with same padding, the difference *was* statistically significant; these models are highlighted in red, and can be considered false positives.