Statistical correction of input gradients for black box models trained with categorical input features

Antonio Majdandzic[1*] **and Peter K. Koo**[1*]

[1]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA
[*]majdand@cshl.edu and koo@cshl.edu

ABSTRACT

Gradients of a deep neural network's predictions with respect to the inputs are used in a variety of downstream analyses, notably in post hoc explanations with feature attribution methods. For data with input features that live on a lower-dimensional manifold, we observe that the learned function can exhibit arbitrary behaviors off the manifold, where no data exists to anchor the function during training. This leads to a random component in the gradients which manifests as noise. We introduce a simple correction for this off-manifold gradient noise for the case of categorical input features, where input values are subject to a probabilistic simplex constraint, and demonstrate its effectiveness on regulatory genomics data. We find that our correction consistently leads to a significant improvement in gradient-based attribution scores.

1 Introduction

State-of-the-art generalization performance is a highly desirable property of predictive modeling with deep neural networks (DNNs). However, in the sciences, understanding the reasons behind a DNN's predictions is arguably more important as it can help to drive scientific discovery. Key to this alternative objective is *model interpretability*, which is a somewhat elusive term because it can mean different things to different communities[1,2]. Here we refer to a narrow definition of interpretability as the ability to explain which input features are important to a network's predictions[3].

One area where DNN interpretability based on post hoc feature attribution methods is natural is regulatory genomics[4,5]. In this scientific field, DNNs have demonstrated powerful predictive performance across a wide variety of tasks, taking DNA sequences as input and predicting an experimentally measured regulatory function, such as transcription factor (TF) binding sites[6,7], chromatin accessibility sites[8,9], chromatin conformations[10], and gene expression[11–13]. Attribution methods provide an importance score for each nucleotide in a given sequence; they often reveal biologically meaningful patterns, such as protein binding motifs, that are important for gene regulation[6,11]. Moreover, attribution methods provide a natural way of quantifying the effect size of single nucleotide mutations, both observed and counterfactual, which can then be utilized to prioritize disease-associated variants[14,15].

Some of the most popular attribution methods are gradient-based, where partial derivatives of input features with respect to the output are used as a measure of model sensitivity to input features, thus assigning numerical importance to features. Notable and widely used gradient-based methods include saliency maps[16], integrated gradients[17], SmoothGrad[18], expected gradients[19], and DeepSHAP[20]. In practice, attribution methods yield noisy feature importance maps[21,22]. Many factors that influence the efficacy of attribution maps have been identified empirically, such as the smoothness properties of the learned function[18,23,24] and learning robust features[25–27]. However, the origins of all noise sources that afflict attribution maps is not yet fully understood.

Here we identify a new source of noise in input gradients when the input features have a geometric constraint set by a probabilistic interpretation, such as one-hot-encoded DNA sequences. In such cases, all data lives on a lower-dimensional manifold – a simplex within a higher-dimensional space. For DNA, the data lives on a 3D plane within a 4D space. A DNN has freedom to express any function shape off of the simplex, because no data points exist to guide the behavior of the function. This randomness can introduce unreliable gradient components in directions off the simplex, which can manifest as spurious noise in the input gradients, thereby affecting explanations from gradient-based attribution methods. To address this issue, we introduce a simple correction to input gradients which minimizes the impact of this off-simplex-derived gradient noise. Through a systematic empirical investigation, we show that this correction significantly improves gradient-based attribution maps both quantitatively and qualitatively.

2 Related work

For model interpretability based on feature importance, approaches to understand what the model is learning include: attribution methods, which could be gradient or perturbation based[16,17,20,28–32]; directly understanding neuron activity such as 1st layer visualization[33–35] or input optimization to maximize neuron activity[16,36]; and finally it is possible to interrogate the network with *in silico* experiments by testing hypotheses about patterns learned via interventions based on occlusion[37] and treatments[6,34,38,39].

Approaches to address noise in attribution methods have focused on: (1) improving the attribution method itself[17,18,20,28], (2) improving the training procedure via robust training methods, such as adversarial training[26,40] and manifold-mixup[41], which in turn smoothen the learned function, or with attribution priors[24,42], which directly regularize input gradients during training, and (3) improving the DNN architecture[27,35,43,44], which can provide a stronger inductive bias to learn robust features. To the author's knowledge, there has been no previous work that discusses noise in the gradients that result from off-manifold function behavior.

3 Correcting input gradients for data that lives on a simplex

3.1 Background

As a prototypical example of data that lives on a feature manifold – a subspace within a higher-dimensional feature space – let us consider one-hot sequences as inputs to neural networks, with genomic sequences as an empirical example. Input features to DNNs in genomic prediction tasks are sequences represented as one-hot encoded arrays of size $L \times 4$, having 4 nucleotide variants (i.e. $\{$A, C, G, T$\}$) at each position of a sequence of length L (Fig. 1a). One-hot encoded data naturally lends itself to a probabilistic interpretation, where each position corresponds to the probability of 4 nucleotides for DNA or 20 amino acids for proteins. While the values here represent definite/binary values, these one-hot representations can also be relaxed to represent real numbers – this is a standard view for probabilistic modeling of biological sequences[45], where the real numbers represent statistical quantities like nucleotide frequencies. Each position is described by a vector of 4 real numbers, given by x, y, z , w. The probability axiom imposes that each variable is bound between 0 and 1 and their sum is constrained to equal 1, that is

$$x+y+z+w=1. \tag{1}$$

This restricts the data to a simplex (i.e. a simple manifold) of allowed combinations of (x,y,z,w), and Eq. 1 – being an equation of a 3D plane in a 4D space – defines this simplex.

Importantly, an issue arises with input gradients from how DNNs process this data. While a one-hot representation accurately reflects the fact that genomic sequence data lives on this simplex, when used as input features to a DNN, such as a convolutional neural network (CNN), which is a popular architectural choice in genomics, the function that the network learns is not necessarily confined to this simplex. During training, a DNN is going to learn a function that is supported only by data that solely lives on this simplex, but it will have freedom to express any function shape outside of this plane, for which no training data exists. Since all data, including held-out test data, lives on this simplex, such a DNN can still maintain good generalization performance, despite its unregulated behavior off of the simplex. Surprisingly, when a function behaves erratic outside of the simplex, especially at points near the simplex where data lies, this could substantially affect input gradients (Fig. 1b). Together, we hypothesize that off-simplex gradient components introduce noise to input gradients and thus affect downstream applications that rely on such input gradients, such as post hoc model interpretability with gradient-based attribution methods.

3.2 Correction for input gradients

The input gradients can be decomposed into two components: the component locally parallel to the simplex, which is supported by data, and the component locally orthogonal to this simplex (Fig. 1b), which we surmise is unreliable as the function behavior off of the simplex is not supported by any data. Thus, we conjecture that **removing the unreliable orthogonal component** from the gradient via a directional derivative, leaving only the parallel component that is supported by data, will yield more reliable input gradients. Without loss of generality, we now illustrate this procedure and derive a formula for this gradient correction in the case of widely used one-hot encoded genomic sequence data where the simplex is a 3D plane within a 4D space.

Given $\vec{n} = \frac{1}{2}(\hat{i}+\hat{j}+\hat{k}+\hat{l})$ is a normal vector to the simplex plane (Eq. 1) and $\vec{G}$ is the gradient of function f,

$$\vec{G} = \frac{\partial f}{\partial x}\,\hat{i} + \frac{\partial f}{\partial y}\,\hat{j} + \frac{\partial f}{\partial z}\,\hat{k} + \frac{\partial f}{\partial w}\,\hat{l}, \tag{2}$$

we can correct $\vec{G}$ by removing the unreliable orthogonal component, according to:

$$
\begin{aligned}
\vec{G}_{\text{corrected}} &= \vec{G}_{||} = \vec{G} - \vec{G}_{\perp} = \vec{G} - (\vec{G} \bullet \vec{n})\vec{n} \\
&= (\frac{\partial f}{\partial x} - \mu)\hat{i} + \ldots + (\frac{\partial f}{\partial w} - \mu)\hat{l}
\end{aligned} \tag{3}
$$

where $\mu = \frac{1}{4}\left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} + \frac{\partial f}{\partial z} + \frac{\partial f}{\partial w}\right)$. By comparing Eqs. 2 and 3, we see that the corrected gradient at each position is obtained by simply subtracting the original gradient components by the mean gradients across components.

4 Experimental overview

4.1 Regulatory genomics prediction task

To empirically test whether our proposed correction (Eq. 3) leads to more reliable input gradients, we systematically evaluated attribution maps before and after correction for various CNNs trained on a regulatory genomics task using synthetic data (with known ground truth). Knowing position-level ground truth thus allows us to quantitatively compare the efficacy of the attribution maps. We also evaluate the generalization of our proposed gradient correction method on *in vivo* data.

Synthetic data. The synthetic data reflects a simple billboard model of gene regulation taken from Ref.[27]. Briefly, 20,000 synthetic sequences, each 200 nucleotides long, were embedded with known motifs in specific combinations in a uniform sequence model. Positive class sequences were generated by sampling a sequence model embedded with 3 to 5 "core motifs", randomly selected with replacement from a pool of 10 position frequency matrices, which include the forward and reverse-complement motifs for CEBPB, Gabpa, MAX, SP1, and YY1 proteins from the JASPAR database[46]. Negative class sequences were generated following the same steps with the exception that the pool of motifs include 100 non-overlapping "background motifs" from the JASPAR database. Background sequences can thus contain core motifs; however, it is unlikely to randomly draw motifs that resemble a positive regulatory code. The dataset is randomly split into training, validation and test sets with a 0.7, 0.1, and 0.2 split, respectively.

***In vivo* data.** TF chromatin immunoprecipitation sequencing (ChIP-seq) data was processed and framed as a binary classification task. Positive-label sequences represent the presence of a ChIP-seq peak, which can be interpreted as TF binding to the sequence, and negative-label sequences represent peaks for non-overlapping DNase I hypersensitive sites from the same cell type that do not overlap with any ChIP-seq peaks. 10 representative TF ChIP-seq experiments in a GM12878 cell line and a DNase-seq experiment for the same cell line were downloaded from ENCODE[47], for details see Table 1. 200 nucleotide sequences about the center of each peak was converted to a one-hot representation. BEDTools[48] was used to identify non-overlapping DNase-seq peaks and the number of negative sequences were randomly down-sampled to exactly match the number of positive sequences, keeping the classes balanced. The dataset was split randomly into training, validation, and test set according to the fraction 0.7, 0.1, and 0.2, respectively.

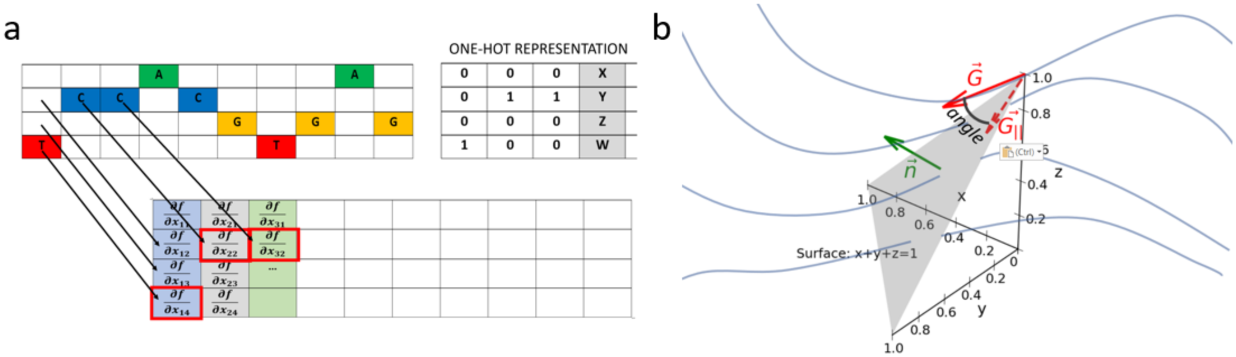
a
A
A
C
C
C
G
G
G
T
T
ONE-HOT REPRESENTATION
0 0 0 X
0 1 1 Y
0 0 0 Z
1 0 0 W
$\frac{\partial f}{\partial x_{11}}$
$\frac{\partial f}{\partial x_{21}}$
$\frac{\partial f}{\partial x_{31}}$
$\frac{\partial f}{\partial x_{12}}$
$\frac{\partial f}{\partial x_{22}}$
$\frac{\partial f}{\partial x_{32}}$
$\frac{\partial f}{\partial x_{13}}$
$\frac{\partial f}{\partial x_{23}}$
...
$\frac{\partial f}{\partial x_{14}}$
$\frac{\partial f}{\partial x_{24}}$
b
$\vec{G}$
angle
$\vec{G}_{||}$
$\vec{n}$
1.0
0.8
0.4
0.2
z
x
1.0
0.8
0.6
0.4
0.2
0
Surface: x+y+z=1
0.2
0.4
0.6
0.8
1.0
y

Figure 1. Toy diagram of input gradients. (a) One-hot encoded genetic sequence example. General values (x, y, z, w) can be interpreted as probabilities. (b) General geometric relation of the gradient and the simplex. Blue curves represent gradient lines of the learned function. Gray plane represents the data simplex. The red vector represents the gradient pointing off of the simplex.

4.2 Overview of models

We used two different base architectures, namely CNN-shallow and CNN-deep from Ref.[27], each with two variations – rectified linear units (ReLU) or exponential activations for the first convolutional layer, while ReLU activations are used for other layers – resulting in 4 models in total. CNN-shallow is a network that is designed with an inductive bias to learn interpretable motifs in first layer filters with ReLU activations; while, CNN-deep has been empirically shown to learn distributed motif representations[35]. Both networks learn robust motif representations in first layer filters when employing exponential activations[27]. Details of the model architecture and training procedure can be found Appendix A.

4.3 Evaluating attribution methods

Attribution methods. To test the efficacy of attribution-based interpretations of the trained models, we generated attribution scores by employing saliency maps[16], integrated gradients[17], SmoothGrad[18] and expected gradients[19]. Saliency maps were calculated by computing the gradient of the predictions with respect to the inputs. Integrated gradients were calculated by integrating the saliency maps generated from 20 linear interpolation points between a reference sequence and a query sequence. SmoothGrad was employed by averaging the saliency maps of 25 variations of a query sequence, which were generated by adding Gaussian noise (zero-centered with a standard deviation of 0.1) to all nucleotides – perturbing all possible nucleotides in each position. For expected gradients, we averaged the integrated gradients across 10 different reference sequences, generated from random shuffles of the query sequence.

Quantifying interpretability. Since synthetic data contains ground truth of embedded motif locations in each sequence, we can directly test the efficacy of the attribution scores. We calculated the similarity of the attribution scores with ground truth using 3 metrics: cosine similarity, area under the receiver-operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPR).

Cosine similarity uses a normalized dot product between vector of positions in a given attribution map and the corresponding ground truth vector; the more similar the two maps are, the closer their cosine similarity is to 1. This is done on a per sequence basis. We subtract 0.25 from the ground truth to "zero out" non-informative positions. Thus, cosine similarity focuses only on the positions where ground truth motifs are embedded.

AUROC and AUPR were calculated according to[27], by comparing the distribution of attribution scores where ground truth motifs have been implanted (positive class) and the distribution of attribution scores at positions not associated with any ground truth motifs (negative class). Briefly, we first multiply the attribution scores (S_{ij}) and the input sequence (X_{ij}) and reduce the dimensions to get a single score per position, according to $C_i = \sum_j S_{ij} X_{ij}$, where j is the alphabet and i is the position, a so-called grad-times-input. We then calculate the information of the sequence model, M_{ij}, according to $I_i = \log_2 4 - \sum_j M_{ij} \log_2 M_{ij}$. Positions that are given a positive label are defined by $I_i > 0.1$ (i.e. 5% of maximum information content for DNA), while positions with an information content of zero are given a negative label. The AUROC and AUPR is then calculated separately for each sequence using the distribution of C_i at positive label positions against negative label positions.

Each metric captures different aspects of the quality of the attribution maps. For instance, cosine similarity focuses on true positive positions and uses the full gradient vector associated with each sequence. On the other hand, AUROC and AUPR uses a single component of the gradient, i.e. the observed nucleotide, due to the grad-times-input. AUROC and AUPR also focus on a different balance between true positives with either false positives or recall, respectively.

5 Results

5.1 Better predictions do not necessarily imply better interpretability

Figure 2a-c shows a scatter plot of the classification AUC versus different interpretability metrics for each CNN (Sec. 4), namely CNN-shallow-relu, CNN-shallow-exp, CNN-deep-relu, and CNN-deep-exp. Strikingly, the classification performance does not perfectly correlate with interpretability performance for all 3 interpretability metrics, namely cosine similarity, AUROC, and AUPR. For instance, CNN-shallow-exp and CNN-deep-relu share a similar classification performance, but CNN-shallow-exp consistently yields a higher intepretability performance. Thus, classification performance is not a reliable metric for model selection when the desired application requires gaining scientific insights through model explanations with attribution methods. As expected, models that employ exponential activations in first layer filters tend to outperform their counterparts with ReLU activations – an observation that was previously described in Ref.[27], albeit across 50 initializations, a few runs led to poor interpretability performance with a small drop in classification performance. We also performed the same analysis using integrated gradients, SmoothGrad and expected gradients and observed similar results (Fig. 5 in Appendix B).

5.2 Input gradient correction significantly improves attribution maps

By comparing the efficacy of attribution maps before and after correction, we find that our gradient correction consistently leads to a significant improvement in the efficacy of saliency maps for each CNN; corrected attribution maps are consistently closer

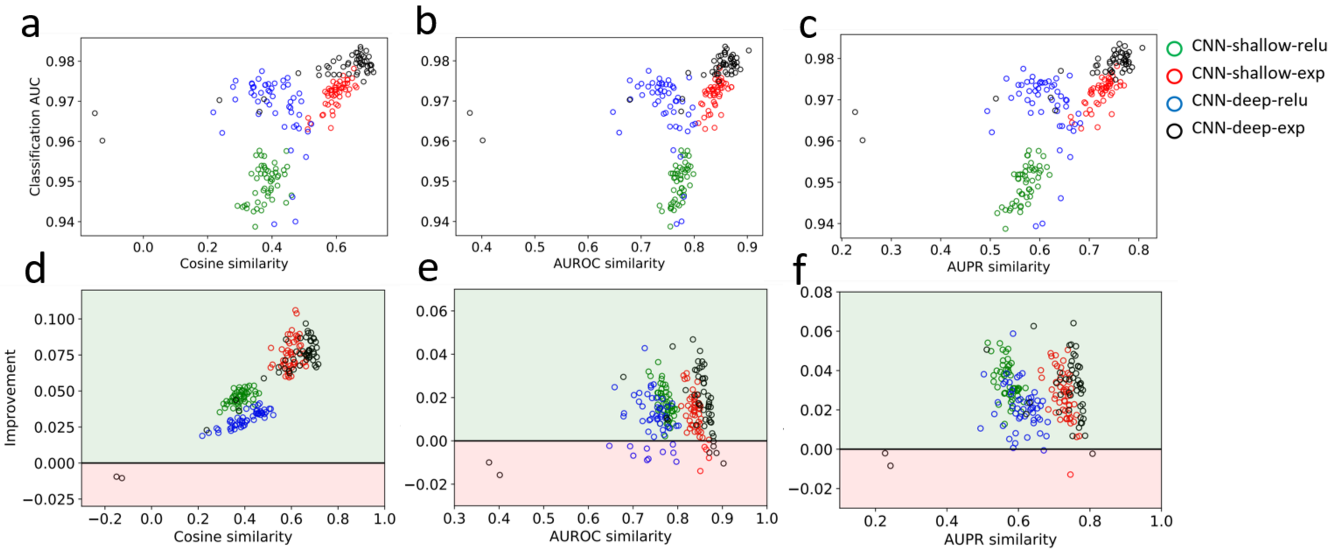

Figure 2. Performance comparison on synthetic data with saliency maps. (a-c) Scatter plot of interpretability performance measured by different similarity scores versus the classification performance (AUC) for saliency maps. (d-f) Interpretability improvement for saliency maps for different similarity metrics when using our gradient noise correction. Improvement represents the change in similarity score after and before the correction. Light green region highlights a positive improvement; light red is the region where the change in similarity score is worse. Each point represents 1 of 50 runs with a different random initialization for each model.

to the ground truth than the naive implementation across three similarity metrics (Fig. 2d-f). We find a similar improvement in attribution maps for integrated gradients, SmoothGrad, and expected gradients as well (Fig. 5 in Appendix B). This demonstrates that our gradient correction is effective and should always be applied to gradient-based attribution methods.

5.3 Gradient correction in action

To demonstrate how the gradient correction qualitatively affects attribution maps, we highlight a sequence patch of a representative positive-label sequence from the synthetic dataset (Fig. 3a). This shows that the uncorrected saliency maps for CNN-deep-exp exhibit spurious noise throughout; positions within and directly flanking the ground truth motif pattern have a high degree of spurious noise. This is a pervasive issue when using gradient-based attribution maps to gain insights into the sequence mechanisms of biological functions (i.e. motifs). After the correction, the motif definition is improved, while spurious saliency scores in background positions, including the positions flanking the ground truth motifs, are driven towards zero, resulting in a (corrected) saliency map that better reflects the ground truth. Notice the large angles that coincide with large noise in the uncorrected saliency map. The improvements in the attribution maps from the gradient correction are not only statistically significant, but they are also visually discernible. We observe a similar qualitative improvement across all models and datasets, both synthetic (Figs. 6 in Appendix B) and *in vivo* (Fig. 3b and Fig. 7 in Appendix B).

5.4 Larger angles between gradients and simplex introduce correctable attribution noise

To better understand the gradient correction, we performed a statistical analysis of angles between gradients and the simplex for CNN-deep-exp in the main text and the results for other models are shown in Appendix B. As predicted, the probability density of gradient angles with respect to the simplex for positions that contain a ground truth motif (Fig. 3c, top) and background positions (Fig. 3d, top) shows the distribution is centered around zero, with a standard deviation of around 28 and 37 degrees, respectively. Surprisingly, most angles are not large. Even with the enormous freedom to express arbitrary functions outside of the simplex, the function that is often learned largely aligns with the simplex, producing gradients that naturally are close to zero. Interestingly, we observe that the width of the background distribution of angles is broader than the distribution for ground truth positions. This suggests that background positions are more prone to off-simplex gradient noise, which creates spurious importance scores – a common feature observed in attribution maps for genomics. A similar distribution of angles was observed across other CNNs trained on the synthetic dataset (Fig. 8 in Appendix B) as well as the *in vivo* datasets (Fig. 3e, top and 9 in Appendix B).

Next, we measured the extent that the correction improves saliency map interpretability at different angles for ground truth positions (Fig. 3c, bottom). Strikingly, we found that the positions that have large gradient angles with respect to the simplex are associated with a much larger improvement of attribution scores. The amount of correction is directly related to the angle. For zero angle, the correction is also zero, and this geometry results in the observed envelope where the true improvement (with respect to the ground truth) cannot exceed the amount of correction itself. We see that for most ground truth positions that have large angles, the improvement is near maximal – points are concentrated by the upper envelope with positive improvements. This highlights how this correction only addresses off-simplex gradient noise.

We performed a similar analysis for background positions that do not contain a ground truth motif. However, instead of quantification of improvement based on cosine similarity, which are not defined for null-vectors (i.e. ground truth of background positions), we directly compare the grad-times-input saliency scores at each position, which should be zero. As expected, the saliency scores for background positions with a large angle become smaller after the correction, which means that spurious attribution noise is getting reduced when caused by large gradient angles. Interestingly, we observed a large set of false positive saliency scores near small angles, for which our correction method cannot address (Fig. 3d, bottom). We believe these false positive motifs arise throughout this dataset simply by chance but are not considered ground truth, despite matching a ground truth motif pattern. We obtain similar results with other models (Fig. 8 in Appendix B).

For *in vivo* data, a similar quantitative analysis is challenging due to a lack of ground truth. Instead, we trained an ensemble

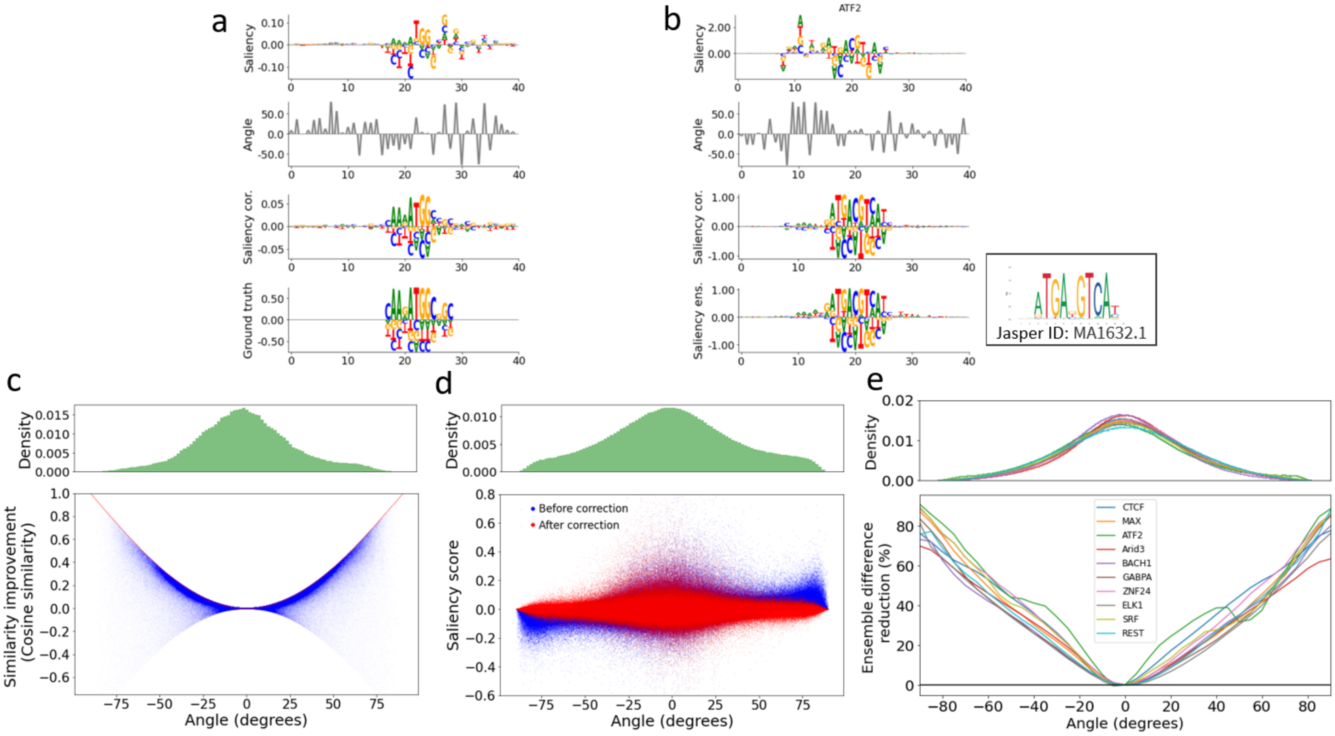

Figure 3. Gradient correction performance. (a,b) Uncorrected saliency map (top row), gradient angles at each position (second row), corrected saliency map (third row), and a sequence logo of the ground truth (bottom row) for a patch from a representative test sequence using CNN-deep-exp for (a) synthetic data and (b) *in vivo* ChIP-seq data for ATF2 protein. (b) An ensemble average saliency map is shown in lieu of ground truth (bottom row), and a known ATF2 motif from JASPAR database for comparison. Sequence logos were created using Logomaker[49]. (c-e) Probability density of input gradient angles across all test sequences for CNN-deep-exp (top). (c) Scatter plot of the saliency map improvement based on cosine similarity (after correction minus before correction) versus the gradient angles for ground truth positions in synthetic data analyzed by CNN-deep-exp. Red line indicates the theoretical limit for our correction, i.e. $1-\cos(\text{angle})$. (d) Scatter plot of the grad-times-input saliency map before (blue) and after (red) correction for background positions (i.e. positions without any ground truth motifs). (c-d) Each dot represents a different nucleotide position. (e) Improvement in saliency maps *in vivo* measured as a percentage decrease of the L2-difference between the single-model saliency score vectors and the corresponding ensemble saliency vectors serving as ground truth. Each line shows averaged results across all positions at a given angle for a different protein (shown in a different color).

of 50 models – each with a slightly different architecture (i.e. different numbers of convolutional filters) and different random initializations – and averaged their saliency maps. We treated the ensemble-averaged saliency maps as a proxy for the ground truth and performed a similar analysis. Indeed the distribution of gradient angles for saliency maps generated with an ensemble average of models is narrower compared to individual models (Figs. 9 and 10 in Appendix B). Unlike our previous analysis, we opted not to make any assumptions about ground truth positions with motifs or background; we used all positions. Cosine similarity is not appropriate here for the same reason as for background positions, so instead we calculated an L2-norm of the difference between the saliency vector and the ensemble-average saliency vector separately for each position.

Strikingly, we found that there is a noticeable decrease in the L2-norm after the correction; saliency maps are closer to the ensemble average (i.e. lower L2-norm) especially for larger angles. Figure 3e shows the percentage by which the L2-norm is decreased (called *Ensemble difference reduction*), for 10 proteins using CNN-deep-exp; results for other models are shown in Fig. 9 in Appendix B. Moreover, significant gradient angles are also observed in larger-scale models trained in a multitask setting, such as the Basset model[34], which has over 4 million parameters and is trained in a multi-task setting to simultaneously predict chromatin accessibility sites across 164 cell-types/tissues (see Fig. 11).

Together, this demonstrates that the correction does not alter information about learned motif features in the gradients, but rather corrects for noise that arises from the randomness of gradients that deviate significantly from the simplex.

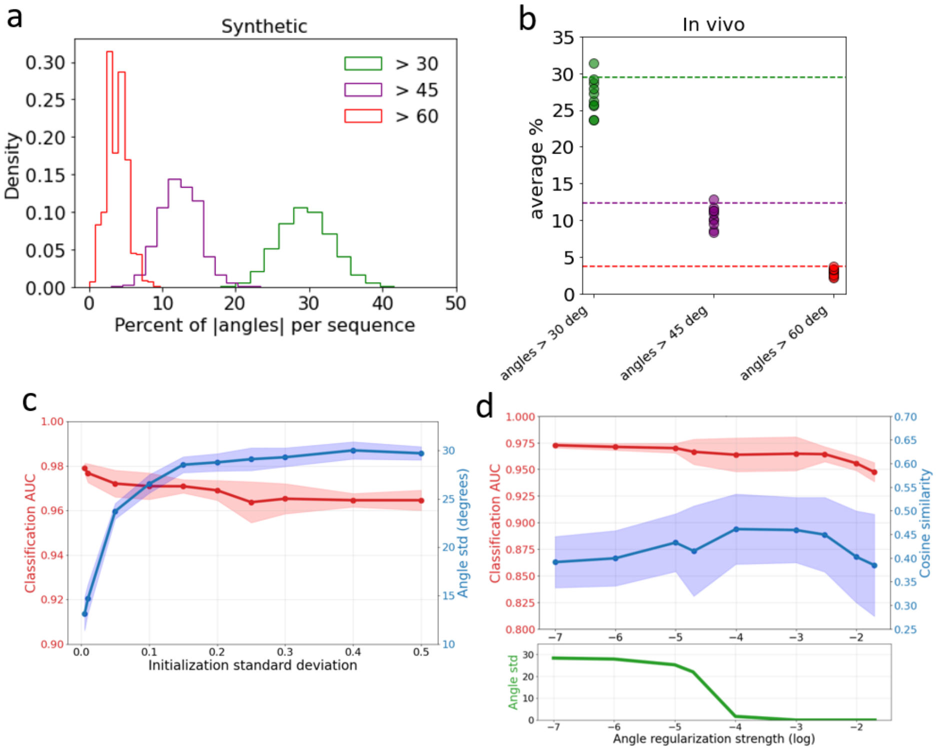

Figure 4. Investigation of input gradient angles using CNN-deep-relu. (a) Histogram of the percentage of positions in a sequence with a gradient angle larger than various thresholds for CNN-deep-relu trained on synthetic data. (b) Scatter plot of the percentage of positions in a sequence with a gradient angle larger than various thresholds for CNN-deep-relu. Each point represents the average angle across all test sequences for each TF ChIP-seq experiment. For comparison, horizontal dashed lines indicate the mean value from the corresponding synthetic data distributions in (a). (c) Plot of the average classification performance (red) and the average standard deviation of the distribution of gradient angles (blue) for different variances of the random normal initializations for CNN-deep-relu trained on synthetic data. (d) Plot of classification performance (red) and standard deviation of the distribution of gradient angles (blue) for CNN-deep-relu trained with a loss that includes a different gradient angle regularization strength. Below is a plot the the mean gradient angle distribution at the end of training for models trained with each regularization strength. (c-d) Dots represent the average across 10 trials and the shaded region represents the standard deviation of the mean.

5.5 Distribution of large angle gradients

To investigate how gradients with large angles are distributed across the data, we generated histograms of the fraction of positions in each sequence that have angles larger than 30, 45, and 60 degrees. We found that each sequence, depending on the model, has about 2-15 percent of positions with a gradient angle larger than 60 degrees; about 10-20 percent of positions have angles greater than 45 degrees; and about 20-40 percent of positions have angles greater than 30 degrees (Fig. 4a and Fig. 12 in Appendix B). We observed a similar distribution of angles across 10 transcription factors (Fig. 4b shows results for CNN-deep-relu and other models are shown in Fig. 13 in Appendix B). Thus, large gradient angles are pervasive in most sequences and result in gradient-based attribution maps that are prone to exhibiting a substantial amount of spurious noise.

5.6 Role of initialization

We hypothesize that initialization may play a large role in the behavior of the function off of the simplex. To elaborate, if the initialization is set poorly, the initial function may already be pointing far away from the simplex, thereby introducing a larger gradient angle. Since models are trained to minimize the loss, they are only concerned with predictions of observed data (that resides on the simplex). Thus, a highly expressive model, such as a deep neural network, may have limited ability to correct this arbitrary behavior as no data exists off the simplex to fix it during training. To investigate the effect of initialization, we explored how random normal initializations with zero mean and different standard deviations affected the gradient angle of a trained CNN-deep-relu model. In agreement with our hypothesis, we found that the standard deviation in the gradient angle distribution is narrower for smaller initializations and the width of the distribution increases dramatically with larger initializations, with only a marginal drop-off in the classification performance (Fig. 4c). We noticed a similar trend for other models (Fig. 14 in Appendix B). This suggests that initialization largely drives the randomness of the function off of the simplex; CNNs are sufficiently expressive enough to maintain their complex initial functions off of the simplex throughout training. Hence, it may be beneficial to find a new initialization strategy better suited for categorical inputs such that the initial function better aligns with the simplex.

5.7 Angle regularization during training

Our proposed correction is ideally suited for post hoc analysis of already trained models. However, it is possible to consider using it as an attribution prior, similar to previous work[19,42,50], to directly regularize the angle of the input gradients during training to drive the model to actively learn a function that removes this undesirable behavior. To test this idea, we trained several versions of CNN models with different angle regularization penalties (Fig. 4d and Fig. 15 in Appendix B). We found that when sufficient regularization penalty is applied, the gradient angles get driven to zero as expected. We observed a bump in interpretability performance as the angle drops to zero – a small but significant improvement, which is a result of preventing off-manifold gradient noise. Interestingly, the classification performance largely remains constant, until the regularization strength is too large, which then detracts the model from the proper objective. In conclusion, we suggest post-training gradient correction as perhaps a more robust way of correcting the angle noise, as it is simpler to implement (i.e. a single line of code) and does not require carefully tuning an additional hyperparameter.

6 Conclusion

Input gradients are used in a variety of applications such as gradient-based model interpretability. Here we identify a new type of noise source for input gradients, we call it *off-manifold noise*, which arises from an unconstrained, expressive model that fits data that lives on a lower dimensional manifold. We propose a simple gradient correction for data that lives on a probabilistic simplex, i.e. one-hot features, and we demonstrate its effectiveness with gradient-based attribution methods. While not intended to improve the classification performance of the network itself, our proposed input gradient correction provides a consistent improvement in the interpretability of gradient-based attribution methods, leading towards better transparency of its decision-making process and, more importantly, providing clearer insights into the underlying biology of our empirical example.

We emphasize that the noise removed is only the noise associated with erratic function behavior off of the simplex. This correction is not a "magic bullet" that can correct other kinds of gradient noise, such as if the model learns a noisy (i.e. not smooth) function, or if the model learns non-robust (i.e. short-cut) representations. The fact that the off-the-simplex gradient angles are typically small is itself an interesting property of the functions trained on categorical data with constraints. This largely supports the utility of the naive implementation of gradient-based attribution methods, albeit with a certain degree of off-simplex gradients that is realized as spurious noise.

Although our gradient correction formula was explicitly derived for widely used one-hot data, our correction method – removing the components of the gradient orthogonal to the data simplex – is general and thus should be applicable to any data structure with well defined geometric constraints[51,52]. For instance, our derived correction can be extended to address applications that use input gradients for all data types that live on a probabilistic simplex, including RNA and protein sequences.

Even data that resides on a lower dimensional manifold, such as a sphere or cylinder, could also suffer from similar issues and our correction proposes that input gradients projected onto the manifold would improve its efficacy, albeit a new correction factor would need to be derived as it depends on the normal vector.

Gradient-based attributions only provide a first-order, local explanation that reveals effect sizes of individual features in an independent manner. Nevertheless, such applications are being used both in science, such as data analysis to understand mechanisms of protein-DNA interactions[6], protein-RNA or RNA-RNA interactions[38,53], protein structure prediction[54,55], sequence design[56–58], in addition to many clinical applications[59,60], where interpretability is critical to ensure trustworthy decision making. These applications, along with many others not described here, would benefit from more reliable input gradients.

Data and Code Availability

Data and code to reproduce results can be found here: https://doi.org/10.5281/zenodo.6506787

Acknowledgements

This work was supported in part by funding from the Simons Center for Quantitative Biology at Cold Spring Harbor Laboratory. This work was performed with assistance from the US National Institutes of Health Grant S10OD028632-01. The authors would like to thank Justin Kinney, David McCandlish, and Anna Posfai for helpful discussions.

References

1. Guidotti, R. *et al.* A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* **51** (2018).
2. Zhang, Y., Tiňo, P., Leonardis, A. & Tang, K. A survey on neural network interpretability. *IEEE Transactions on Emerg. Top. Comput. Intell.* **5**, 726–742 (2021).
3. Doshi-Velez, F. & Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv* (2017).
4. Eraslan, G., Avsec, Ž., Gagneur, J. & Theis, F. J. Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.* **20**, 389–403 (2019).
5. Koo, P. K. & Ploenzke, M. Deep learning for inferring transcription factor binding sites. *Curr. Opin. Syst. Biol.* (2020).
6. Avsec, Ž. *et al.* Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat. Genet.* **53**, 354–366 (2021).
7. Li, J., Pu, Y., Tang, J., Zou, Q. & Guo, F. DeepATT: a hybrid category attention neural network for identifying functional effects of DNA sequences. *Briefings Bioinforma.* (2020).
8. Maslova, A. *et al.* Deep learning of immune cell differentiation. *Proc. Natl. Acad. Sci.* **117**, 25655–25666 (2020).
9. Atak, Z. K. *et al.* Interpretation of allele-specific chromatin accessibility using cell state-aware deep learning. *Genome Res.* gr–260851 (2021).
10. Fudenberg, G., Kelley, D. R. & Pollard, K. S. Predicting 3D genome folding from DNA sequence with Akita. *Nat. Methods* **17**, 1111–1117 (2020).
11. Kelley, D. R. *et al.* Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research* **28**, 739–750 (2018).
12. Agarwal, V. & Shendure, J. Predicting mrna abundance directly from genomic sequence using deep convolutional neural networks. *Cell Reports* **31**, 107663 (2020).
13. Avsec, Z. *et al.* Effective gene expression prediction from sequence by integrating long-range interactions. *bioRxiv* (2021).
14. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning–based sequence model. *Nat. methods* **12**, 931–934 (2015).
15. Zhou, J. *et al.* Whole-genome deep-learning analysis identifies contribution of noncoding mutations to autism risk. *Nat. genetics* **51**, 973–980 (2019).
16. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv:1312.6034* (2013).
17. Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, 3319–3328 (PMLR, 2017).

18. Smilkov, D., Thorat, N., Kim, B., Viégas, F. & Wattenberg, M. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017).

19. Erion, G., Janizek, J. D., Sturmfels, P., Lundberg, S. M. & Lee, S.-I. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nat. Mach. Intell.* 1–12 (2021).

20. Lundberg, S. & Lee, S.-I. A unified approach to interpreting model predictions. *arXiv:1705.07874* (2017).

21. Adebayo, J. *et al.* Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292* (2018).

22. Hooker, S., Erhan, D., Kindermans, P.-J. & Kim, B. A benchmark for interpretability methods in deep neural networks. *arXiv preprint arXiv:1806.10758* (2018).

23. Labelson, E. L., Tripathy, R. & Koo, P. K. Towards trustworthy explanations with gradient-based attribution methods. In *NeurIPS 2021 AI for Science Workshop* (2021).

24. Ross, A. S. & Doshi-Velez, F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Thirty-second AAAI conference on artificial intelligence* (2018).

25. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A. & Madry, A. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152* (2018).

26. Etmann, C., Lunz, S., Maass, P. & Schönlieb, C.-B. On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172* (2019).

27. Koo, P. K. & Ploenzke, M. Improving representations of genomic sequence motifs in convolutional networks with exponential activations. *Nat. Mach. Intell.* **3**, 258–266 (2021).

28. Shrikumar, A., Greenside, P. & Kundaje, A. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, 3145–3153 (PMLR, 2017).

29. Bach, S. *et al.* On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS One* **10**, e0130140 (2015).

30. Zintgraf, L. M., Cohen, T. S., Adel, T. & Welling, M. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595* (2017).

31. Fong, R. C. & Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, 3429–3437 (2017).

32. Selvaraju, R. R. *et al.* Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626 (2017).

33. Alipanahi, B., Delong, A., Weirauch, M. *et al.* Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat Biotechnol* **33**, 831–838 (2015).

34. Kelley, D., Snoek, J. & Rinn, J. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* **26**, 990–9 (2016).

35. Koo, P. K. & Eddy, S. R. Representation learning of genomic sequence motifs with convolutional neural networks. *PLoS Comput. Biol.* **15**, e1007560 (2019).

36. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. & Lipson, H. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579* (2015).

37. Zeiler, M. D. & Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833 (Springer, 2014).

38. Koo, P. K., Majdandzic, A., Ploenzke, M., Anand, P. & Paul, S. B. Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLoS Comput. Biol.* **17**, e1008925 (2021).

39. Hammelman, J. & Gifford, D. K. Discovering differential genome sequence activity with interpretable and efficient deep learning. *PLoS Comput. Biol.* **17**, e1009282 (2021).

40. Madry, A., Makelov, A., Schmidt, L., Tsipras, D. & Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

41. Verma, V. *et al.* Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, 6438–6447 (PMLR, 2019).

42. Tseng, A., Shrikumar, A. & Kundaje, A. Fourier-transform-based attribution priors improve the interpretability and stability of deep learning models for genomics. *Adv. Neural Inf. Process. Syst.* **33** (2020).

43. Dosovitskiy, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

44. Ghotra, R. S., Lee, N. K. & Koo, P. K. Uncovering motif interactions from convolutional-attention networks for genomics. In *NeurIPS 2021 AI for Science Workshop* (2021).

45. Durbin, R., Eddy, S. R., Krogh, A. & Mitchison, G. *Biological sequence analysis: probabilistic models of proteins and nucleic acids* (Cambridge university press, 1998).

46. Mathelier, A. *et al.* JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic Acids Res.* **44**, D110–D115 (2016).

47. ENCODE Project Consortium *et al.* The ENCODE (encyclopedia of DNA elements) project. *Science* **306**, 636–640 (2004).

48. Quinlan, A. R. & Hall, I. M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**, 841–842 (2010).

49. Tareen, A. & Kinney, J. B. Logomaker: beautiful sequence logos in Python. *Bioinformatics* **36**, 2272–2274 (2020).

50. Ross, A. S., Hughes, M. C. & Doshi-Velez, F. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717* (2017).

51. Cayton, L. Algorithms for manifold learning. *Tech. Rep.* **CS2008** (2005).

52. Narayanan, H. & Mitter, S. Sample complexity of testing the manifold hypothesis (NIPS, 2010).

53. Singh, J., Hanson, J., Paliwal, K. & Zhou, Y. Rna secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nat. communications* **10**, 1–13 (2019).

54. Yang, J. *et al.* Improved protein structure prediction using predicted interresidue orientations. *Proc. Natl. Acad. Sci.* **117**, 1496–1503 (2020).

55. Gligorijevic, V., Renfrew, P., Kosciolek, T. & et al. Structure-based protein function prediction using graph convolutional networks. *Nat. Commun.* **12**, 3168 (2021).

56. Bogard, N., Linder, J., Rosenberg, A. B. & Seelig, G. A deep neural network for predicting and engineering alternative polyadenylation. *Cell* **178**, 91–106 (2019).

57. Anishchenko, I., Chidyausiku, T. M., Ovchinnikov, S., Pellock, S. J. & Baker, D. De novo protein design by deep network hallucination. *bioRxiv* (2020).

58. Norn, C. *et al.* Protein sequence design by conformational landscape optimization. *Proc. Natl. Acad. Sci.* **118** (2021).

59. Singh, A., Sengupta, S. & Lakshminarayanan, V. Explainable deep learning models in medical image analysis. *J. Imaging* **6**, 52 (2020).

60. Huff, D. T., Weisman, A. J. & Jeraj, R. Interpretation and visualization techniques for deep learning models in medical imaging. *Phys. Medicine & Biol.* **66**, 04TR01 (2021).

61. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 448–456 (PMLR, 2015).

62. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The J. Mach. Learn. Res.* **15**, 1929–1958 (2014).

63. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv:1412.6980* (2014).

64. He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*.

A Model architecture

All models take as input one-hot-encoded sequences (200 nucleotides) and have a fully-connected output layer with a single sigmoid output for this binary prediction task. The hidden layers for each model are:

1. CNN-shallow
    1. convolution (24 filters, size 19, stride 1, activation)
       max-pooling (size 50, stride 50)
    2. convolution (48 filters, size 3, stride 1, ReLU)
       max-pooling (size 2, stride 2)
    3. fully-connected layer (96 units, stride 1, ReLU)
2. CNN-deep
    1. convolution (24 filters, size 19, stride 1, activation)
    2. convolution (32 filters, size 7, stride 1, ReLU)
       max-pooling (size 4, stride 4)
    3. convolution (48 filters, size 7, stride 1, ReLU)
       max-pooling (size 4, stride 4)
    4. convolution (64 filters, size 3, stride 1, ReLU)
       max-pooling (size 3, stride 3)
    5. fully-connected layer (96 units, stride 1, ReLU)

We incorporate batch normalization[61] in each hidden layer prior to activations; dropout[62] with probabilities corresponding to: CNN-shallow (layer1 0.1, layer2 0.2) and CNN-deep (layer1 0.1, layer2 0.2, layer3 0.3, layer4 0.4, layer5 0.5); and $L2$-regularisation on all parameters of hidden layers (except batch norm) with a strength equal to 1e-6.

We uniformly trained each model by minimizing the binary cross-entropy loss function with mini-batch stochastic gradient descent (100 sequences) for 100 epochs with Adam updates using default parameters[63]. The learning rate was initialized to 0.001 and was decayed by a factor of 0.2 when the validation area under the curve (AUC) of the receiver-operating characteristic curve did not improve for 3 epochs. All reported performance metrics are drawn from the test set using the model parameters from the epoch which yielded the highest AUC on the validation set. Each model was trained 50 times with different random initializations according to Ref.[64]. All models were trained using a single P100 GPU; each epoch takes less than 2 seconds.

B Additional Figures and Tables

Table 1. *In vivo* data details. Ten representative TF ChIP-seq experiments in a GM12878 cell line and a DNase-seq experiment (ENCODE file accession ENCFF235KUD) for the same cell line were downloaded from ENCODE[47]. Table shows ENCODE file accession codes for all transcription factor proteins.

PROTEIN	ENCODE FILE ACCESSION	CELL LINE
CTCF	ENCFF710VEH	GM12878
MAX	ENCFF083KVY	GM12878
ATF2	ENCFF127GYQ	GM12878
ARID3	ENCFF027VZK	GM12878
BACH1	ENCFF012JXJ	GM12878
GABPA	ENCFF116EXQ	GM12878
ZNF24	ENCFF103OOV	GM12878
ELK1	ENCFF556JBS	GM12878
SRF	ENCFF909FRA	GM12878
REST	ENCFF677KJB	GM12878

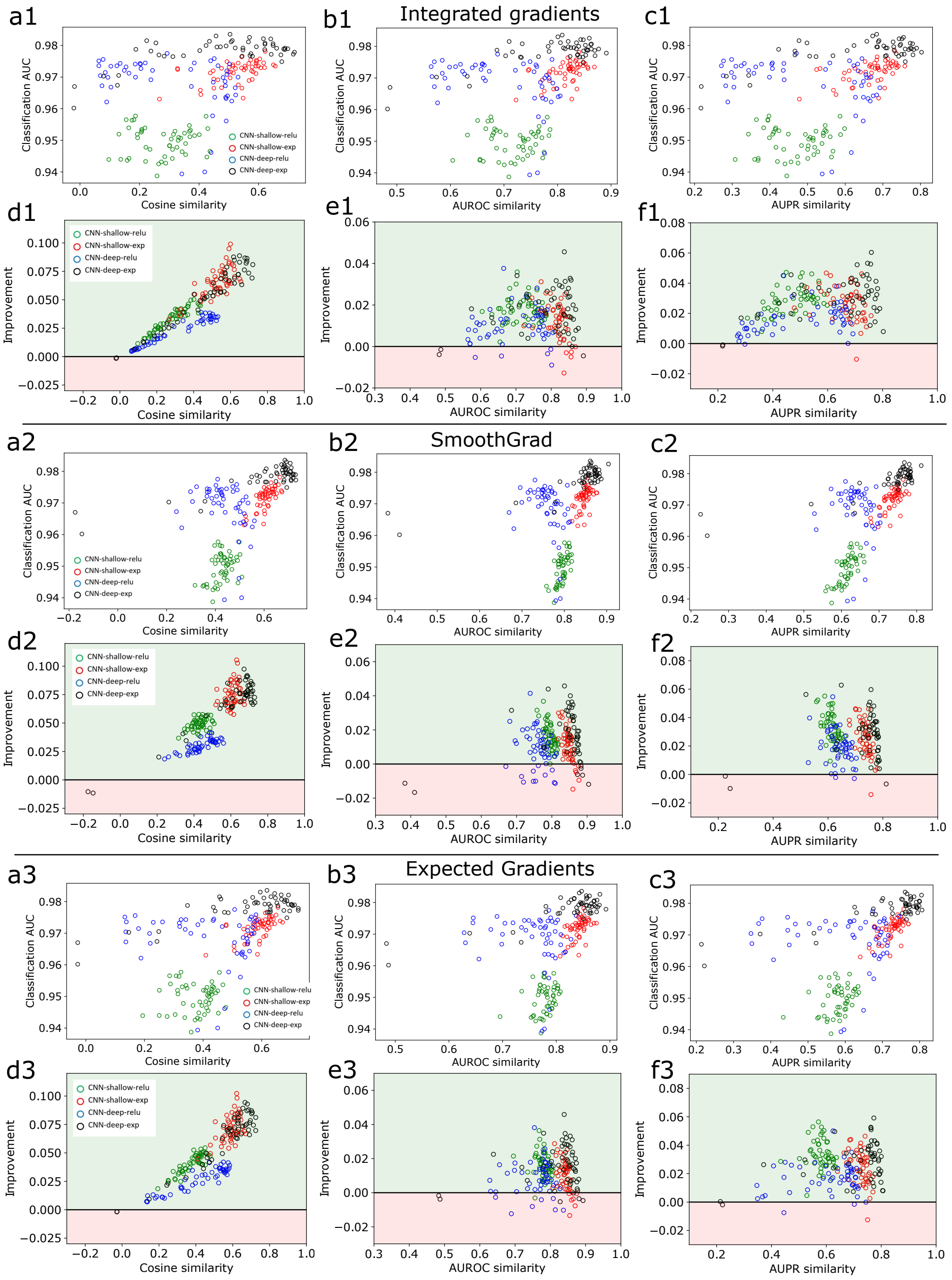

Figure 5. Performance comparison on synthetic data with integrated gradients (top), SmoothGrad (middle) and expected gradients (bottom). Scatter plot of interpretability performance measured by different similarity scores versus the classification performance (AUC) for integrated gradients (a1-c1), SmoothGrad (a2-c2) and expected gradient maps (a3-c3). Interpretability improvement for integrated gradients (d1-f1), SmoothGrad (d2-f2) and expected gradient maps (d3-f3) for different similarity metrics, when using our gradient noise correction. Improvement represents the change in similarity score after and before the correction. Green region highlights a positive improvement; light red is the region where the change in similarity score is worse. Each point represents 1 of 50 runs with a different random initialization for each model.

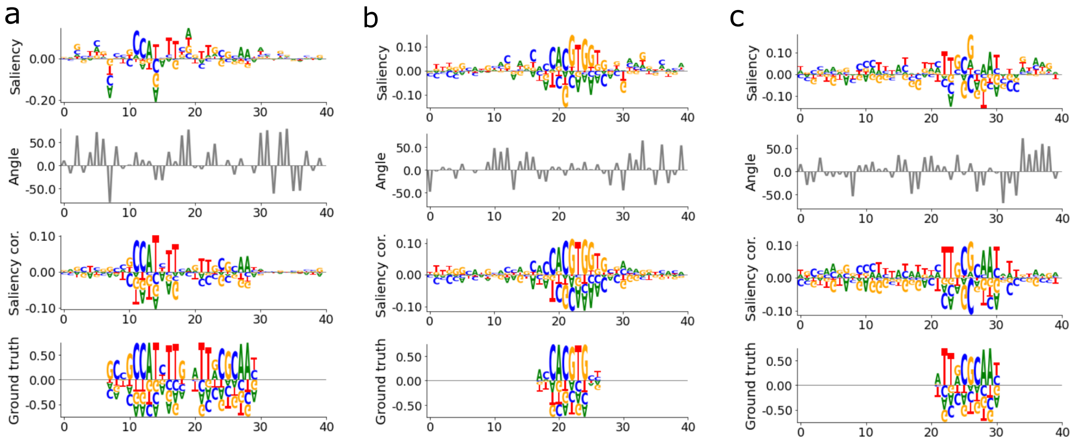

Figure 6. Attribution comparison before and after correction for (a) CNN-shallow-exp, (b) CNN-deep-relu and (c) CNN-shallow-relu. (a-c) Representative patches from positive label sequences that shows a sequence logo of the saliency scores at each position (top row), a plot of the angle between gradients and the simplex at each position (second row), a sequence logo of the corrected saliency scores (third row), and a sequence logo of the ground truth (bottom row). Sequence logos for ground truth are subtracted by 0.25 prior to plotting to remove uninformative positions.

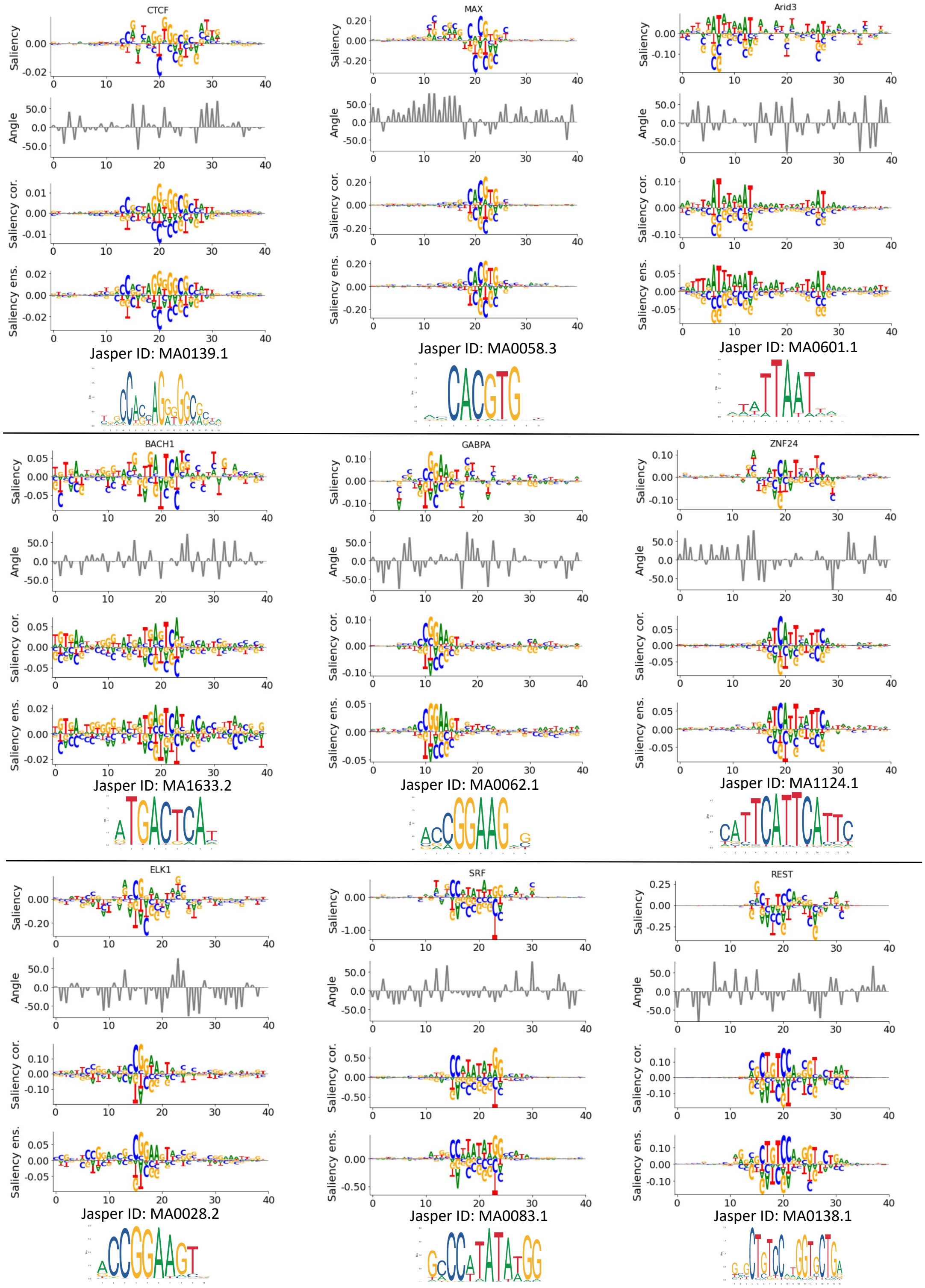

Figure 7. Attribution comparison before and after correction for CNN-deep-exp model for ChIP-seq proteins CTCF, MAX, Arid3, BACH1, GABPA, ZNF24, ELK1, SRF and REST. Subplots are representative patches from positive label sequences that show a sequence logo of the saliency scores at each position (top row), a plot of the angle between gradients and the simplex at each position (second row), a sequence logo of the corrected saliency scores (third row), and a sequence logo of the ensemble average (bottom row). Known motifs from literature (JASPAR database) are shown below.

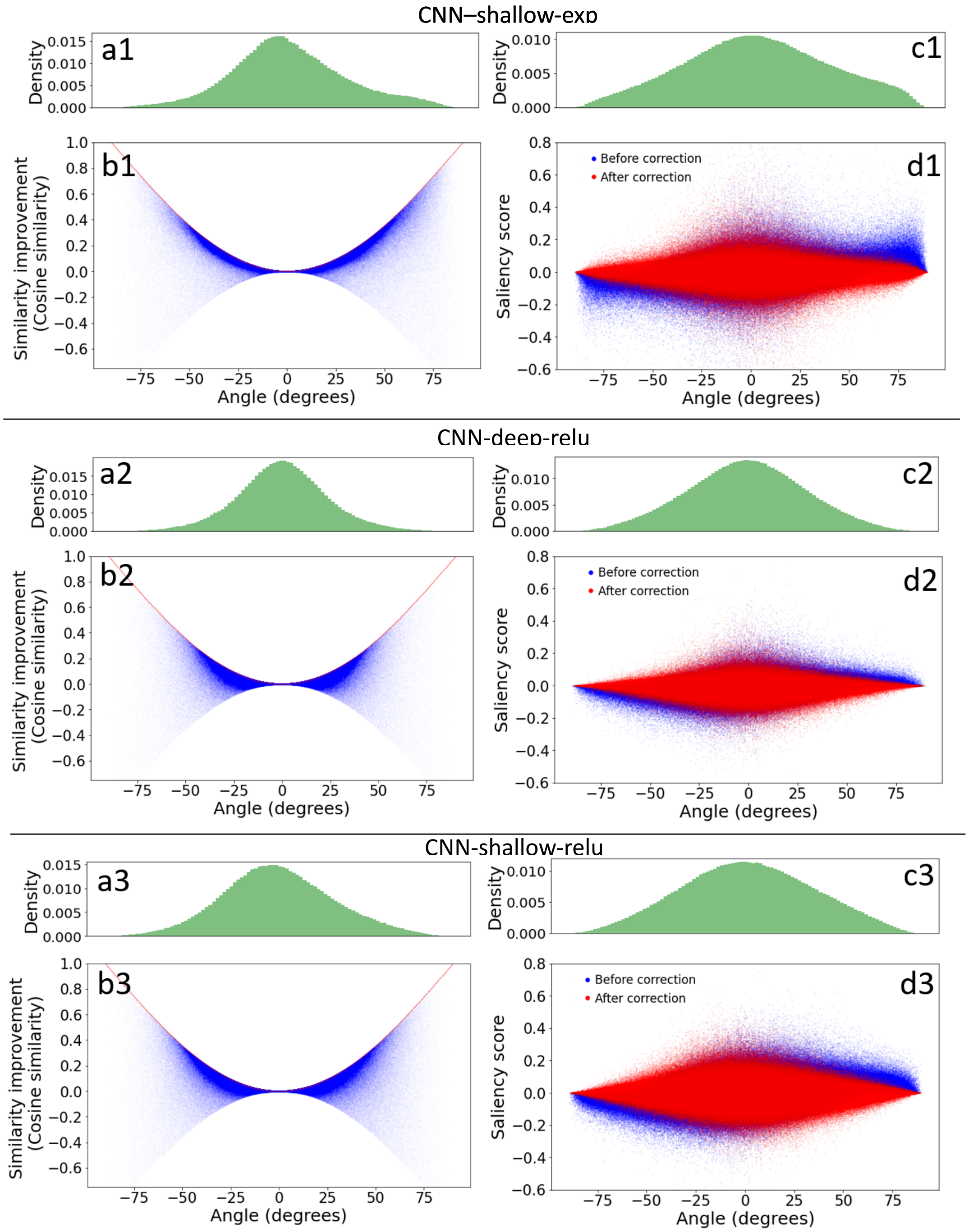

Figure 8. Analysis of gradients at different angles for CNN-shallow-exp (top row - 1), CNN-deep-relu (middle row - 2) and CNN-shallow-relu (bottom row - 3). (a, c) Probability density of input gradient angles for positions where ground truth motifs are embedded (a) and other background positions (c). (b) Scatter plot of attribution score improvements based on cosine similarity (after correction minus before correction) versus the gradient angles for ground truth positions. Red line indicates the theoretical limit for a correction, i.e. $1 - \cos(\text{angle})$. (d) Scatter plot of saliency scores versus gradient angles before (blue) and after (red) correction for background positions (i.e. positions without any ground truth motifs). (b,d) Each dot represents a different nucleotide position.

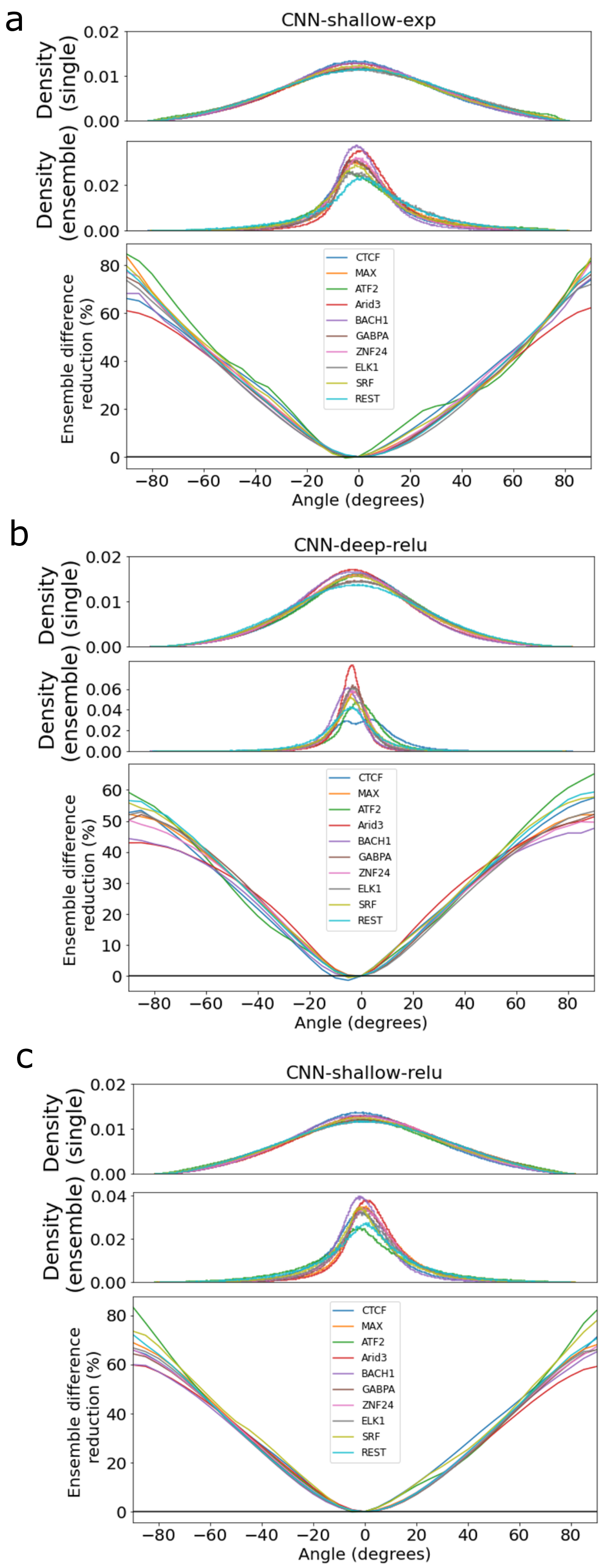

Figure 9. Angle-noise analysis for (a) CNN-shallow-exp, (b) CNN-deep-relu and (c) CNN-shallow-relu for 10 different ChIP-seq proteins: CTCF, MAX, ATF2, Arid3, BACH1, GABPA, ZNF24, ELK1, SRF and REST. (Top and middle rows in a-c) Probability density of input gradient angles are shown, using colors corresponding to legends below. (Bottom rows in a-c) Improvement in saliency maps *in vivo* measured as a percentage decrease of the L2-difference between the single-model saliency score vectors and the corresponding ensemble saliency vectors serving as ground truth. Each line shows averaged results across all positions at a given angle for a different protein (shown in a different color).

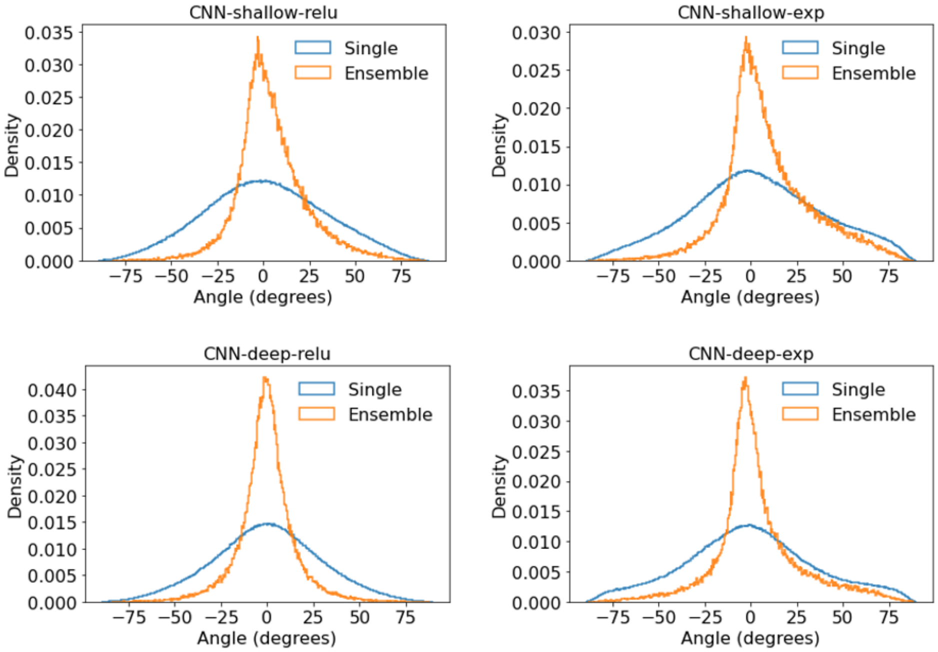

Figure 10. Comparison of angle distributions for the synthetic data. Comparison of the distribution of angles between gradients and the simplex for individual positions for individual models (blue) and an ensemble averaged model across 50 random initializations (orange) for different CNN models.

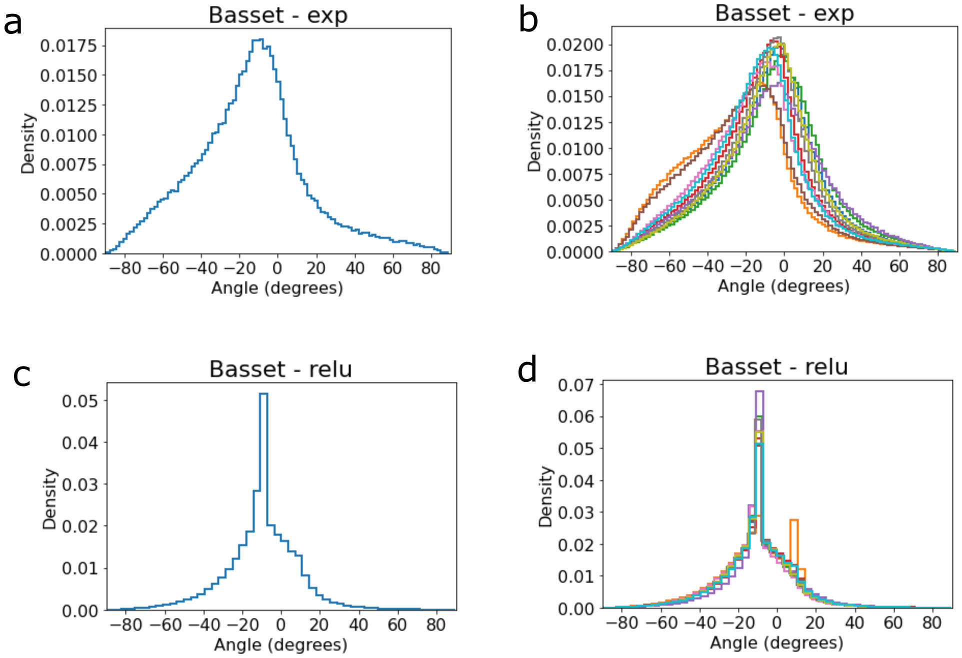

Figure 11. Gradient angle analysis for a large-scale multi-task CNN. Histogram of the gradient angles for test sequences in the Basset dataset[34], which consists of a multi-task classification of 161 chromatin accessibility datasets, using a Basset model with exponential activations (a,b) and ReLU activations (c,d) in first layer filters (see Task 4 in[27] for model specifications). (a,c) represent the gradient with respect to the class with the highest prediction across the entire test set, while (b,d) represent the test sequences with a positive label for the first 10 classes (each shown in a different color).

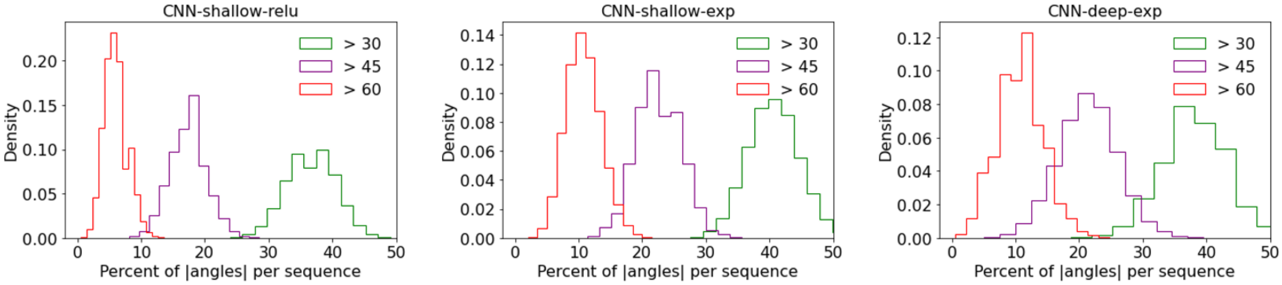

Figure 12. Distribution of large gradient angles for CNN-shallow-relu, CNN-shallow-exp and CNN-deep-exp. Histogram of the percentage of positions in a sequence with a gradient angles larger than various thresholds for each CNN model.

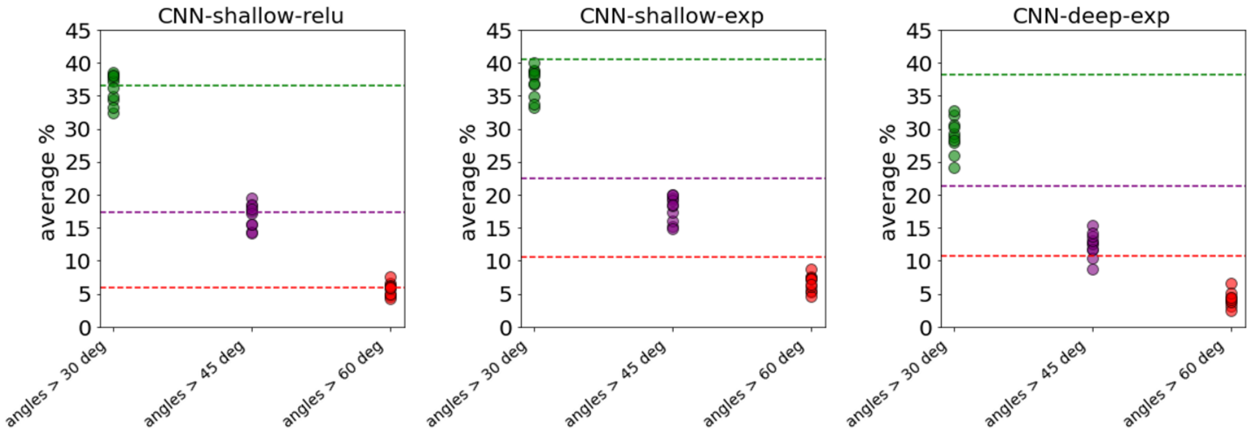

Figure 13. Box plots of the average percentage of positions in a sequence with a gradient angle larger than various thresholds for CNN-shallow-relu, CNN-shallow-exp and CNN-deep-exp, across 10 TF ChIP-seq experiments (scatter plots, where each point is one protein). For comparison, horizontal dashed lines indicate the mean value from synthetic experiments using the corresponding models.

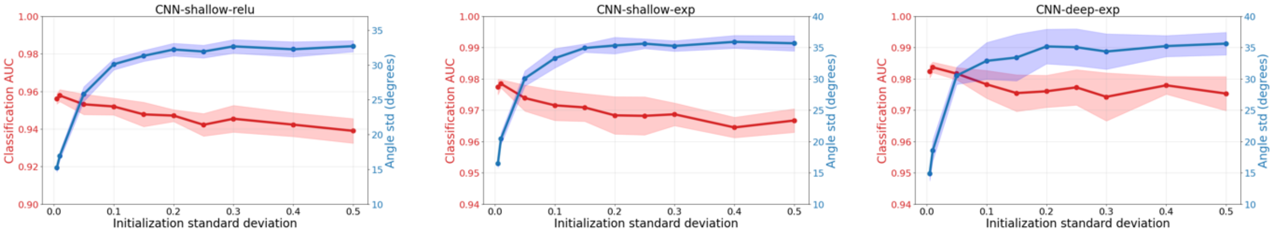

Figure 14. Plot of classification performance (red) and standard deviation of the distribution of gradient angles (blue) for different random normal initializations for CNN-shallow-relu, CNN-shallow-exp and CNN-deep-exp, trained on synthetic data.

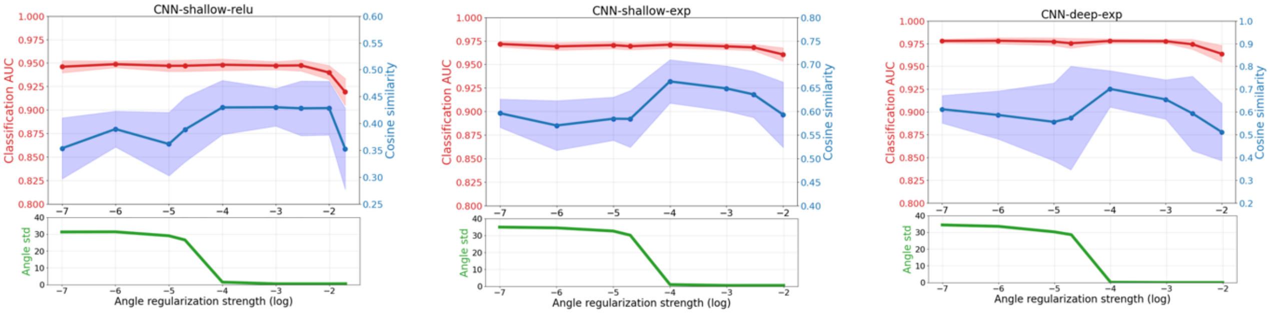

Figure 15. Plot of classification performance (red) and standard deviation of the distribution of gradient angles (blue) for CNN-shallow-relu, CNN-shallow-exp and CNN-deep-exp, trained with different gradient angle regularization strengths. Below is a plot the the mean gradient angle distribution for models trained with each regularization strength. Dots represent the average across 10 trials and the shaded region represents the standard deviation of the mean.