Error-controlled interaction detection in deep neural networks

Yang Lu[1], Winston Chen[3], and William Stafford Noble[1,2]

[1]Department of Genome Sciences, University of Washington
[2]Paul G. Allen School of Computer Science and Engineering, University of Washington
[3]Department of Electrical and Computer Engineering, University of Washington

November 30, 2021

1 Introduction

Despite their outstanding empirical performance, deep neural networks (DNNs) are frequently treated as black box models, preventing their adoption in many low-error-tolerant domains, such as healthcare, finance, robotics, and cybersecurity defense. In such contexts, interpreting DNN to reason about why certain decisions are made is crucial.

Inspired by existing works that interpret DNNs by selecting a subset of explanatory features subject to a controlled error rate [**?**], we are interested in detecting higher-order information captured by a DNN model, *i.e.,* interactions between features. Modelling the fact that features often have joint effects with other features is especially useful for scientific discoveries and hypothesis validation. For example, gene-gene, gene-environment, gene-drug and gene-disease interactions are key elements in explaining genetic mechanisms in biomedical applications.

2 Related work

2.1 Feature interaction detection

Our method falls into the paradigm of explaining feature interactions in DNNs. These interactions can be roughly divided into two categories: global (or model-level) interactions and local (or instance-level) interactions. Global interactions span the entire dataset, independent of any particular evaluation data sample. The most straightfroward way to detexct global interactions is via a method such as two-way ANOVA, which involves conducting hypothesis tests for each interaction candidate using F-statistics. A recently described method, neural interaction detection (NID) [**?**], detects global interactions by examining the weights of the neural network. Specifically, if some features are connected to the same neuron in the first hidden layer with large weights and if the paths from the node to the output also have large weights, then an interaction between the features is detected. Detecting local interactions, on the other hand, involves identifying interactions that lead the model to make a given prediction for a particular data sample. For example, local interactions between feature pairs can be obtained by leveraging the second-order derivative (*i.e.,* the Hessian) with respect to the input data sample [**?**]. Alternatively, the SHAP interaction score [**?**] extends the shapley value from game theory to detect local interactions, and it is efficiently implemented in ensemble tree methods.

2.2 The knockoff filter

Our method control the false discovery rate (FDR) among the detected feature interactions by leveraging the knockoffs framework [**? ?**], which was proposed recently in the setting of error-controlled feature selection, where the core idea is to generate knockoff features that perfectly mimic the empirical dependence structure among the original features. Specifically, assume that there exists a feature subset $\mathcal{S}_0 \subset \{1, \ldots, p\}$ such that, conditional on features in $\mathcal{S}_0$, the response Y_i is independent of features in the complement $\mathcal{S}_0^c$. Then for a subset of features $\widehat{S} \subset \{1, \ldots, p\}$ selected by some feature selection procedure, the FDR is defined as

$$\text{FDR} = \mathbb{E}[\text{FDP}] \text{ with } \text{FDP} = \frac{|\widehat{S} \cap \mathcal{S}_0^c|}{|\widehat{S}|},$$

where $|\cdot|$ stands for the cardinality of a set.

The knockoff filter achieves FDR control in two steps: construction of knockoff features followed by filtering using knockoff statistics. For the first step, we define the knockoff features as follows:

Definition 1 (Model-X knockoff [**?**]). Model-X knockoff features for the family of random features $\mathbf{x} = (X_1, \ldots, X_p)^T$ are a new family of random features $\tilde{\mathbf{x}} = (\tilde{X}_1, \ldots, \tilde{X}_p)^T$ that satisfy two properties:

1. $(\mathbf{x}, \tilde{\mathbf{x}})_{\text{swap}(\mathcal{S})} \overset{d}{=} (\mathbf{x}, \tilde{\mathbf{x}})$ for any subset $\mathcal{S} \subset \{1, \ldots, p\}$, where swap$(\mathcal{S})$ means swapping X_j and $\tilde{X}_j$ for each $j \in \mathcal{S}$ and $\overset{d}{=}$ denotes equal in distribution, and
2. $\tilde{\mathbf{x}} \perp\!\!\!\perp Y|\mathbf{x}$, i.e., $\tilde{\mathbf{x}}$ is independent of response Y given feature $\mathbf{x}$.

According to Definition 1, the construction of knockoffs is independent of the response Y. If we can construct a set $\tilde{x}$ of model-X knockoff features, then by comparing the original features with these control features, FDR can be controlled at target level q. See [**?**] for theoretical guarantees of FDR control with knockoff filters.

With the constructed knockoff features $\tilde{\mathbf{x}}$, we quantify importance of each feature by computing the knockoff statistics $W_j = g_j(Z_j, \tilde{Z}_j)$ for $1 \leq j \leq p$, where Z_j and $\tilde{Z}_j$ represent feature importance measures for the jth feature X_j and its knockoff counterpart $\tilde{X}_j$, respectively, and $g_j(\cdot, \cdot)$ is an antisymmetric function satisfying $g_j(Z_j, \tilde{Z}_j) = -g_j(\tilde{Z}_j, Z_j)$. Note that the feature importance measures as well as the knockoff statistics depend on the specific algorithm used to fit the model. For example, in linear regression models one can choose Z_j and $\tilde{Z}_j$ as the Lasso regression coefficients of X_j and $\tilde{X}_j$, respectively, and a valid knockoff statistic could be $W_j = |Z_j| - |\tilde{Z}_j|$.

In principle, the knockoff statistics W_j should satisfy a coin-flip property such that swapping an arbitrary pair of X_j and its knockoff counterpart $\widetilde{X}_j$ only changes the sign of W_j but keeps the signs of other W_k ($k \neq j$) unchanged [**?**]. A desirable property for knockoff statistics W_j's is that important features are expected to have large positive values, whereas unimportant ones should have small magnitudes symmetric around 0.

Given the knockoff statistics as feature importance measures, we sort $|W_j|$'s in decreasing order and select features whose W_j's exceed some threshold T. In particular, two choices of threshold are suggested [**? ?**]

$$T = \min\left\{t \in \mathcal{W}, \frac{|\{j : W_j \leq -t\}|}{|\{j : W_j \geq t\}|} \leq q\right\}, \quad T_+ = \min\left\{t \in \mathcal{W}, \frac{1 + |\{j : W_j \leq -t\}|}{1 \vee |\{j : W_j \geq t\}|} \leq q\right\}, \tag{1}$$

where $\mathcal{W} = \{|W_j| : 1 \leq j \leq p\} \setminus \{0\}$ is the set of unique nonzero values obtained by $|W_j|$'s, and $q \in (0, 1)$ is the desired FDR level specified by the user. Candes *et al.* [**?**] proved that under mild conditions on the W_j's, the knockoff filter with T controls a slightly modified version of FDR, and the knockoff$_+$ filter with T_+ controls the exact FDR.

It is worth mentioning that the conventional knockoffs [**?**] are only restricted to the Gaussian settings, which is not widely-applicable in real-world dataset. Therefore, in practise, we use improved knockoff framework with no assumptions on the feature distribution, such as Deep knockoffs [**?**], KnockoffGAN [**?**], and DDLK [**?**].

3 Methods

In this section, we integrate the idea of knockoff filters with DNNs to achieve feature interaction detection with controlled FDR.

3.1 Problem setup

Consider a supervised learning task where we have n independent and identically distributed observations $(\mathbf{x}_i, Y_i)$, $i = 1, \cdots, n$, with $\mathbf{x}_i \in \mathbb{R}^p$ the feature vector and Y_i the scalar response. Here we consider the high-dimensional setting where the feature dimensionality p can be much larger than the sample size n. An interaction, $\mathcal{I} \subset \{1, \cdots, p\}$, is a subset of interacting features, where $|\mathcal{I}| \geq 2$. A k-th order interaction $\mathcal{I}$ satisfies $|\mathcal{I}| = k$. We will write $\mathbf{x}^{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ as the vector containing the subset of features selected by $\mathcal{I}$.

In this work, we aim to detect "non-additive" feature interactions, which are defined as follows.

Definition 2 (Non-additive interaction [**?**])**.** Let $\mathcal{F} = \{1, \cdots, p\}$ denote the input feature set. Consider a function $f : \mathbb{R}^p \mapsto \mathbb{R}$, an input feature vector $\mathbf{x} \in \mathbb{R}^p$, and an interaction $\mathcal{I} \subset \{1, \cdots, p\}$. Then $\mathcal{I}$ is a non-additive interaction of function f if and only if f can be expressed as the sum of two functions, $f_{\backslash i}$ and $f_{\backslash j}$, which are defined as functions not depending on $\mathbf{x}^{\{i\}}$ and $\mathbf{x}^{\{j\}}$, respectively:

$$f(\mathbf{x}) = f_{\backslash i}(\mathbf{x}^{\mathcal{F}\backslash i}) + f_{\backslash j}(\mathbf{x}^{\mathcal{F}\backslash j}) \tag{2}$$

For example, $x_1x_2 + \sin(x_2 + x_3 + x_4)$ contains a pairwise interaction $\{1, 2\}$ and a third order interaction $\{2, 3, 4\}$. In general, a k-th order interaction can only exist if all its corresponding $(k-1)$-th order interactions exist. For example, the interaction $\{2, 3, 4\}$ can only exist if interactions $\{2, 3\}$, $\{2, 4\}$, and $\{3, 4\}$ also exist.

The goal of interaction detection is to map a trained model into a set of learned interaction candidates and associated interaction strengths, where a larger interaction strength value indicates stronger confidence in the existence of a true interaction. Note that we are not interested in detecting additive interactions because they can be further decomposed into smaller interactions, whereas non-additive interactions cannot be further decomposed.

3.2 DNN architecture for interaction detection

The main idea of the proposed method is to first generate the knockoffs $\tilde{\mathbf{x}}$, and then feed an augmented feature vector $(\mathbf{x}^T, \tilde{\mathbf{x}}^T)^T$ into a DNN. Following the architecture of DeepPINK [**?**], the augmented feature vector is fed into the network through a plugin pairwise-coupling layer containing p filters, $\mathbf{F} = (F_1, \cdots, F_p)$, where the jth filter connects feature X_j and its knockoff counterpart $\tilde{X}_j$. Initialized equally, the corresponding filter weights z_j and $\tilde{z}_j$ compete against each other during training. The outputs of the filters are then fed into a fully connected feed-forward neural network that aims to capture the strength of individual feature interactions.

Parallel to the DeepPINK filter network and fully connected neural network, the augmented feature vector $(\mathbf{x}^T, \tilde{\mathbf{x}}^T)^T$ is also feed into a module with a collection of univariate networks that aim to model the marginal contribution of each individual feature by learning a feature-wise transformation. When training the neural networks, as suggested by Tsang *et al.* [**?**], a sparsity regularization is applied to the first module of the architectures so as to push the modeling of the main effects toward the second module.

For the feature interaction module we use a feed-forward neural network with L_1 hidden layers (*i.e.,* a multilayer perceptron). Let p_l be the number of hidden units in the l-th layer. The input feature filters are treated as the 0-th layer, and $p_0 = p$ is the number of input features. There are L_1 weight matrices, where the weight matrix connecting the $(l-1)$-th layer and the l-th layer is denoted as $\mathbf{W}^{(l)} \in \mathbb{R}^{p_l \times p_{l-1}}$, together with the bias vector $\mathbf{b}^{(l)} \in \mathbb{R}^{p_l}$. Let $\phi(\cdot)$ be the nonlinear activation function (*e.g.,* ReLU), then the hidden layer of the neural network, $\mathbf{h}^{(l)}$ with input feature filters $\mathbf{F} \in \mathbb{R}^p$, can be expressed as

$$\begin{aligned} \mathbf{h}^{(0)} &= \mathbf{F}, \\ \mathbf{h}^{(1)} &= \text{diag}(\mathbf{W}^{(0)})\mathbf{h}^{(0)} + \mathbf{b}^{(0)}, \\ \mathbf{h}^{(l)} &= \phi\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right) \end{aligned} \tag{3}$$

where diag$(\mathbf{W}^{(0)})$ indicates that $\mathbf{W}^{(0)}$ is a diagonal matrix.

For the univariate module, we consider a feature-wise transformation to model the marginal contribution of each individual feature filter. Specifically, each transformation function is modelled by a univariate network with L_2 hidden layers. For the i-th feature filter, its corresponding univariate network is denoted as $\mathbf{g}_i(F_i) : \mathbb{R} \mapsto \mathbb{R}$.

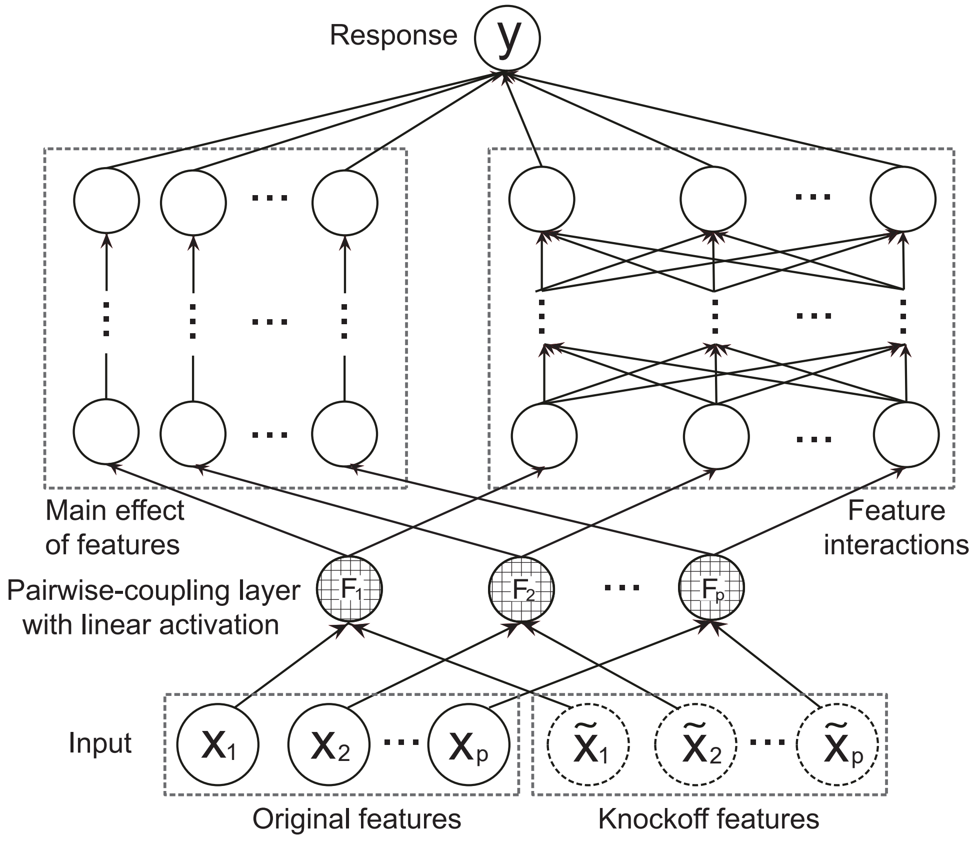

Figure 1: A graphical illustration of the proposed method.

Finally, the outputs of the two modules are concatenated and mapped to the response Y through a fully connected layer, represented as

$$Y = \phi_1\left(\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_p\right) + \phi_2\left(\mathbf{h}^{(L_1)}\right) \tag{4}$$

where both $\phi_1 : \mathbb{R}^p \mapsto \mathbb{R}$ and $\phi_2 : \mathbb{R}^p \mapsto \mathbb{R}$ are linear transformation functions. In particular, $\phi_2(\mathbf{h}^{(L_1)})$ can be written as

$$\phi_2(\mathbf{h}^{(L_1)}) = \mathbf{W}^y \mathbf{h}^{(L_1)} + \mathbf{b}^y \tag{5}$$

where $\mathbf{W}^y \in \mathbb{R}^{1 \times p_{L_1}}$ and $\mathbf{b}^y \in \mathbb{R}$ are the weight matrix and the bias for the layer.

3.3 Measuring the importance of feature interaction

In this section, given the DNN architecture introduced in Section 3.2, we quantify the importance of feature interactions, which is necessary for subsequent FDR estimation.

3.3.1 Global (or model-level) interaction importance

The global importance of a feature interaction is determined by two factors: (1) the relative importance of the jth feature filter relative to all p feature filters; and (2) the relative importance of X_j and its knockoff counterpart $\tilde{X}_j$, encoded by filter weights $\mathbf{z} = (z_1, \cdots, z_p)^T \in \mathbb{R}^p$ and $\tilde{\mathbf{z}} = (\tilde{z}_1, \cdots, \tilde{z}_p)^T \in \mathbb{R}^p$, respectively. Specifically, the relative importance of feature filters $\mathbf{w} = (w_1, \cdots, w_p)^T \in \mathbb{R}^p$ can be represented by the weight matrices from the first module:

$$\mathbf{w} = |\mathbf{W}^y| \cdot \left|\mathbf{W}^{(L)}\right| \cdots \left|\mathbf{W}^{(0)}\right| \tag{6}$$

Equation 6 is the upper bound of the gradient magnitudes of the DNN in the first module, where the gradients have been commonly used as variable importance measures in DNNs [**?**]. Thus, an upper bound on the gradient magnitude approximates how important the feature is, from a global perspective.

We now combine our definitions from $\mathbf{z} \in \mathbb{R}^p$, $\tilde{\mathbf{z}} \in \mathbb{R}^p$, and $\mathbf{W}^{(l)} \in \mathbb{R}^{p_l}$ to obtain the interaction strength of a potential interaction between the i-th and j-th original features, the i-th original feature and the j-th knockoff feature, the i-th knockoff feature and the j-th original feature, and the i-th and j-th knockoff features, respectively:

$$\begin{aligned}
\gamma_{ij} &= \sum \mu_{\text{NID}}(|W_i^{(0)} z_i|, |W_j^{(0)} z_j|) \cdot |W^{(1)}| \cdots |W^{(l)}|, \\
\gamma_{i\tilde{j}} &= \sum \mu_{\text{NID}}(|W_i^{(0)} z_i|, |W_j^{(0)} z_{\tilde{j}}|) \cdot |W^{(1)}| \cdots |W^{(l)}|, \\
\gamma_{\tilde{i}j} &= \sum \mu_{\text{NID}}(|W_i^{(0)} z_{\tilde{i}}|, |W_j^{(0)} z_j|) \cdot |W^{(1)}| \cdots |W^{(l)}|, \\
\gamma_{\tilde{i}\tilde{j}} &= \sum \mu_{\text{NID}}(|W_i^{(0)} z_{\tilde{i}}|, |W_j^{(0)} z_{\tilde{j}}|) \cdot |W^{(1)}| \cdots |W^{(l)}|,
\end{aligned} \tag{7}$$

where $\mu_{\text{NID}} : \mathbb{R}^2 \mapsto \mathbb{R}$ is an aggregating function for an interaction that represents the interaction strength due to feature weights. This function can be the arithmetic mean, geometric mean, harmonic mean, or minimum value [**?**]. These options can account for various properties of interaction strength: (1) interaction strength should be zero whenever an interaction does not exist (one of the features has zero weight); (2) interaction strength does not decrease with any increase in magnitude of feature weights; (3) interaction strength is less sensitive to changes in large feature weights.

Our experiments showed that interaction score $\gamma_{ij}, \gamma_{i\tilde{j}}, \gamma_{\tilde{i}j}, \gamma_{\tilde{i}\tilde{j}}$ computed with above procedure contains strong marginal effects. Specifically, interactions composed of two non-interacting univariate features would be assigned a large score because each feature individually has large weights. Inorder to remove the impact of marginal effect, we proposed the following two types of adjusted interaction scores.

Adjusted interaction score by subtracting feature importance scores: This adjustment method computes a importance score for each feature within the interactions in a fasion similar DeepPINK [**?**]. Then by subtracting the importance scores of both features from the interaction score, we obtain the adjusted interaction score. Details of the algorithm is shown below:

$$\begin{aligned}
w_i &= \sum |W_i^{(0)}|z_i| \cdot |W^{(1)}| \cdots |W^{(l)}|, \\
w_j &= \sum |W_j^{(0)}|z_j| \cdot |W^{(1)}| \cdots |W^{(l)}|, \\
w_{\tilde{i}} &= \sum |W_i^{(0)}|z_{\tilde{i}}| \cdot |W^{(1)}| \cdots |W^{(l)}|, \\
w_{\tilde{j}} &= \sum |W_i^{(0)}|z_{\tilde{j}}| \cdot |W^{(1)}| \cdots |W^{(l)}|, \\
\gamma'_{ij} &= 2 \times \gamma_{ij} - \mu_{\text{DeepPINK}}(w_i, w_j), \\
\gamma'_{i\tilde{j}} &= 2 \times \gamma_{i\tilde{j}} - \mu_{\text{DeepPINK}}(w_i, w_{\tilde{j}}), \\
\gamma'_{\tilde{i}j} &= 2 \times \gamma_{\tilde{i}j} - \mu_{\text{DeepPINK}}(w_{\tilde{i}}, w_j), \\
\gamma'_{\tilde{i}\tilde{j}} &= 2 \times \gamma_{\tilde{i}\tilde{j}} - \mu_{\text{DeepPINK}}(w_{\tilde{i}}, w_{\tilde{j}}),
\end{aligned} \tag{8}$$

where w scores are the individual features' importance score. γ score is the same feature interaction score described in the previous paragraph. $\mu_{\text{DeepPINK}} : \mathbb{R}^2 \mapsto \mathbb{R}$ is another aggregating function for feature importance scores. This function can be euclidian norm, geometric mean, harmonic mean,

Adjusted interaction score by incorporating weight correlation: This second adjustemnt incorporates pearson correlation of DNN's input layer weights, $W^{(0)}$, into the interaction score. For features with ground truth interactions, we can expect their first layer weights to be highly correlated, and for feaatures that has strong marginal effect but no interactions, they should exhibit low correlation. Therefore, we modified the interaction score equation as follows:

$$\begin{aligned}
\rho_{ij} &= \mathrm{corr}(W_i^{(0)} z_i, W_j^{(0)} z_j) \\
\rho_{i\tilde{j}} &= \mathrm{corr}(W_i^{(0)} z_i, W_{\tilde{j}}^{(0)} z_{\tilde{j}}) \\
\rho_{\tilde{i}j} &= \mathrm{corr}(W_{\tilde{i}}^{(0)} z_{\tilde{i}}, W_j^{(0)} z_j) \\
\rho_{\tilde{i}\tilde{j}} &= \mathrm{corr}(W_{\tilde{i}}^{(0)} z_{\tilde{i}}, W_{\tilde{j}}^{(0)} z_{\tilde{j}}) \\
\gamma'_{ij} &= \sum \frac{(|W_i^{(0)} z_i| + |W_j^{(0)} z_j|) \cdot \rho_{ij}}{2} \cdot |W^{(1)}| \cdots |W^{(l)}|, \\
\gamma'_{i\tilde{j}} &= \sum \frac{(|W_i^{(0)} z_i| + |W_j^{(0)} z_{\tilde{j}}|) \cdot \rho_{i\tilde{j}}}{2} \cdot |W^{(1)}| \cdots |W^{(l)}|, \\
\gamma'_{\tilde{i}j} &= \sum \frac{(|W_i^{(0)} z_{\tilde{i}}| + |W_j^{(0)} z_j|) \cdot \rho_{\tilde{i}j}}{2} \cdot |W^{(1)}| \cdots |W^{(l)}|, \\
\gamma'_{\tilde{i}\tilde{j}} &= \sum \frac{(|W_i^{(0)} z_{\tilde{i}}| + |W_j^{(0)} z_{\tilde{j}}|) \cdot \rho_{\tilde{i}\tilde{j}}}{2} \cdot |W^{(1)}| \cdots |W^{(l)}|,
\end{aligned} \tag{9}$$

where ρ_{ij} is the pearson correlation of the first layer weights multiplied by pairwise input weights between feature i and feature j.

3.3.2 Local (or instance-level) interaction importance

In addition to capturing the global importance of a feature interaction (Equation 7), we aim to incorporate instance-level importance, which identifies the interactions that lead the model to make a particular prediction on a particular data sample. To do so, we replace the relative importance of feature filters in Equation 6 with a variant that is dependent on a specific instance $\mathbf{x} \in \mathbb{R}^p$:

$$\mathbf{w_x} = |\mathbf{W}^y| \cdot \left|\mathbf{W}_{\mathbf{x}}^{(L)}\right| \cdots \left|\mathbf{W}_{\mathbf{x}}^{(0)}\right| \tag{10}$$

where the weight matrix $\mathbf{W}_{\mathbf{x}}^{(l)}$ is modified from $\mathbf{W}^{(l)}$ by setting the columns, which correspond to the inactivated neurons, to be all zero vectors.

3.4 FDR control in detected interactions

We obtain a set of feature interactions of size $2p \cdot (2p - 2)$, denoted as $\Gamma = \Big\{\gamma_{ij}, \gamma_{i\tilde{j}}, \gamma_{\tilde{i}j}, \gamma_{\tilde{i}\tilde{j}} | i \in \{1, \cdots, p\}, j \in \{1, \cdots, p\}, i \neq j\Big\}$. We further sort the set by decreasing importance (Section 3.3.1) and let Γ_j denote the j-th interaction in the sorted set.

We aim to select feature interactions whose importance exceed some threshold T such that the selected interactions are subject to a specified FDR. A complication arises from the fact that the feature interactions contain a heterogeneous set of identifications, potentially comprised of original-original feature interactions, original-knockoff feature interactions, knockoff-original feature interactions, and knockoff-knockoff feature interactions. Following Walzthoeni *et al.* [**?**], we choose the threshold T by

$$T = \min \left\{ t \in \mathcal{T}, \frac{|\{j : \Gamma_j \geq t, j \in \mathcal{D}\}| - 2 \cdot |\{j : \Gamma_j \geq t, j \in \mathcal{DD}\}|}{|\{j : \Gamma_j \geq t\}|} \leq q \right\} \tag{11}$$

where $\mathcal{D}$ and $\mathcal{DD}$ refer to the set of interactions each of which contain at least one knockoff feature and both knockoff features, respectively. $\mathcal{T} = \{|\Gamma_j| : 1 \leq j \leq |\Gamma|\} \setminus \{0\}$ is the set of unique nonzero values obtained by $|\Gamma_j|$'s, and $q \in (0, 1)$ is the desired FDR level specified by the user.

4 Results

In this section, we discuss our experiments on both simulated and real-world datasets to study the performance of our approach to interaction detection

Univariate function	**Bivariate Interaction**	**Tri-variate Interaction**
$F_1(x) = x$	$G_1(x_1, x_2) = x_1 x_2$	$H_1(x_1, x_2, x_3) = x_1 x_2 x_3$
$F_2(x) = \log(x)$	$G_2(x_1, x_2) = \log(x_1 + x_2)$	$H_2(x_1, x_2, x_3) = \log(x_1 + x_2 + x_3)$
$F_3(x) = \arctan(x)$	$G_3(x_1, x_2) = \arctan(x_1 + x_2)$	$H_3(x_1, x_2, x_3) = \arctan(x_1 + x_2 + x_3)$
$F_4(x) = \sin(x)$	$G_4(x_1, x_2) = \sin(x_1 + x_2)$	$H_4(x_1, x_2, x_3) = \sin(x_1 + x_2 + x_3)$
$F_5(x) = 2^x$	$G_5(x_1, x_2) = 2^{x_1 + x_2}$	$H_5(x_1, x_2, x_3) = 2^{x_1 + x_2 + x_3}$
$F_6(x) = \pi^x$	$G_6(x_1, x_2) = \pi^{x_1 x_2}$	$H_6(x_1, x_2, x_3) = \pi^{x_1 x_2 x_3}$
$F_7(x) = \frac{1}{1+x^2}$	$G_7(x_1, x_2) = \frac{1}{1+x_1^2+x_2^2}$	$H_7(x_1, x_2, x_3) = \frac{1}{1+x_1^2+x_2^2+x_3^2}$
$F_8(x) = \sqrt{1+x^2}$	$G_8(x_1, x_2) = \sqrt{1+x_1^2+x_2^2}$	$H_8(x_1, x_2, x_3) = \sqrt{1+x_1^2+x_2^2+x_3^2}$

Table 1: Test suite of univariate, bivariate, and trivariate functions.

4.1 Simulation studies

For simulated experiments, we use a test suite of synthetic functions, ranging from univariate functions to trivariate interactions (Table 1). The test functions were designed to have a mixture of interactions with varying order, strength, and nonlinearity. Because our proposed method is tailored to detect pairwise interactions, even for the high-order interaction function such as $H_1(x_1, x_2, x_3) = x_1 x_2 x_3$, we only expect to detect pairwise interactions (x_1, x_2), (x_1, x_3), and (x_2, x_3).

We generated simulated data as follows. from each row in table 1, we specify the following model, comprised of a mixture of univariate, pairwise and tri-variate interactions:

$$f = f_i(x_1) + \cdots + f_i(x_5) + g_i(x_6, x_7) + \cdots + g_i(x_{14}, x_{15}) + h_i(x_{16}, x_{17}, x_{18}) + \cdots + h_i(x_{28}, x_{29}, x_{30}) \quad (12)$$

where f contains an equal number of univariate, pairwise and tri-variate interactions and features are shared by multiple functions. all feature values are uniformly distributed between -1 and 1. we generate 30,000 data points, which are then randomly split into train, validation, and test sets containing 10,000 points each.

4.2 Drug combination response

We consider one of the largest publicly available datasets measuring drug combination response in acute myeloid leukemia [?]. Each one of $n = 12,362$ samples consists of the measured response of a two-drug pair tested in the cancer cells of a patient. The $p = 1,235$ input features are split between features describing the drug combinations and features describing the cancerous cells. We aim to use the proposed method to re-discover the important two-drug pairs.

4.3 Interactions between regulatory variants

The Allen Institute dataset on aging brains [?] consists of $n = 337$ samples from 107 brains collected from four areas in the brain (parietal cortex, temporal cortex, frontal white matter, and hippocampus). The gene-expression data, with $p = 50,281$ contains expression values in units of fragments per kilobase of transcript per million (fpkm), batch-corrected by using the RSEM pipeline [?].

4.4 Pneumonia risk

The pneumonia risk prediction dataset [?] contains data from $n = 14,199$ pneumonia patients, including $p = 46$ features describing each patient. These features range from history features such as age and gender, to simple measurements taken at a routine physical such as heart rate and blood pressure, to lab tests such as white blood cell count and blood urea nitrogen, to features read from a chest x-ray such as lung collapse or pleural effusion. The modeling task is to predict the probability of death (POD) for patients with pneumonia so that high-risk patients can be admitted to the hospital while low-risk patients are treated as outpatients.

4.5 Interactions between regulatory variants

DNNs mapping regulatory DNA sequences to TF binding and chromatin accessibility have been previously used to score the predicted allelic effects of putative regulatory genetic variants based on *in silico* mutagenesis [**? ?**]. Greenside *et al.* [**?**] discovered statistically significant binding QTLs (bQTLs) feature interactions for different TFs from a single DNN model trained to predict chromatin accessibility (instead of TF binding) from sequence. We aim to use the proposed method to re-discover their findings.

5 Discussion

The proposed method may be extensible to convolutional neural networks (CNNs) by replacing the CNN with an effective MLP-Mixer architecture [**?**].