Influence Decompositions For Neural Network Attribution

Kyle Reing* **Greg Ver Steeg** **Aram Galstyan**
Information Sciences Institute, University Of Southern California
{reing*, gregv, galstyan}@isi.edu

Abstract

Methods of neural network attribution have emerged out of a necessity for explanation and accountability in the predictions of black-box neural models. Most approaches use a variation of sensitivity analysis, where individual input variables are perturbed and the downstream effects on some output metric are measured. We demonstrate that a number of critical functional properties are not revealed when only considering lower-order perturbations. Motivated by these shortcomings, we propose a general framework for decomposing the orders of influence that a collection of input variables has on an output classification. These orders are based on the cardinality of input subsets which are perturbed to yield a change in classification. This decomposition can be naturally applied to attribute which input variables rely on higher-order coordination to impact the classification decision. We demonstrate that our approach correctly identifies higher-order attribution on a number of synthetic examples. Additionally, we showcase the differences between attribution in our approach and existing approaches on benchmark networks for MNIST and ImageNet.

1 Introduction

With an ever-increasing presence of deep learning in statistical modeling and prediction, the call for methods of explanation and interpretation of deep neural networks continues to grow. Methods of neural network attribution – whereby a score is assigned to individual inputs according to their influence on the output

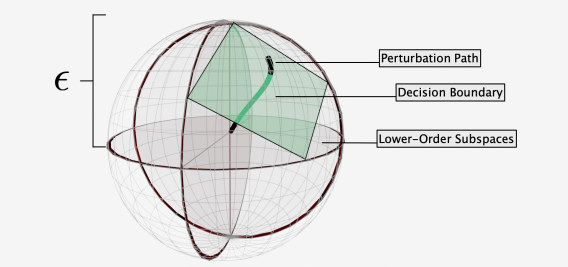

Figure 1: To measure importance of a joint coordinate system, we compare ϵ-bounded perturbations (based on some $\mathcal{L}_p$ norm) in high-dimensional space versus low-dimensional subspaces. If exclusively high-dimensional perturbation trajectories lead to misclassification (as above) then importance is assigned jointly.

– are one class of approaches seeking to shine a light inside these neural black-boxes. Measures of influence in these settings are frequently based on gradients [1, 2, 3, 4, 5, 6], or misclassification due to input perturbations [7, 8]. Note that both of these settings are inherently perturbation-based, as gradients measure the infinitesimal change around a particular data sample. A subject of contention within the attribution community which has garnered considerable criticism is the lack of insight gained by these methods [9, 10, 11]. Early work was evaluated qualitatively and through appeal to human intuition, until it was shown that many approaches ignore all details of the model [12]. To combat this evaluation problem, attempts have been made to introduce formal axioms (such as the sensitivity axiom [1], or input invariance [9]) which dictate whether a methods attributions are trustworthy. However, with axioms acting as a proxy for quantitative analysis, there has been a surge in candidate axioms without much work done to determine which are critical or desirable (see Appendix for axiom evaluation). A question worth asking is, does a method which satisfies more axioms tell us more about the underlying function?

Despite a perception of progress in the attribution literature, the use of gradients and input perturbations to probe black-box functions has been around for decades under the guise of *sensitivity analysis*[13] and *property*

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

testing [14]; though these connections are rarely (if ever) brought up in attribution papers. Through the wider lens afforded by this literature, it becomes apparent that many current attribution methods make critically limiting assumptions that can skew or hide important properties of the network. One example, which will be the primary focus of this work, is the assumption that higher-order influence can be disregarded. Almost all attribution methods only consider first-order influence (either single-variable partial derivatives, or single-variable input perturbations). This assumption is especially limiting for the highly nonlinear networks employed in modern vision and natural language tasks, where first-order perturbations (alt. partial derivatives) are loose approximations in the best case, and grossly misinformative in the worst. Recent work has begun to notice and address this limitation in a number of different ways. One example is the inclusion of second-order influence through computation of the Hessian [2], and another proposes a framework for measuring k_{th}-order contributions by modifying the Shapely interaction values [15]. Although these works take the first step in establishing higher-order influence, they do so by assuming access to an accurate and neutral baseline [16]. We wish to challenge this assumption, and demonstrate how the wrong choice of baseline can lead to pathological reporting of functional properties. In place of a fixed baseline, we propose to search for one within local regions of increasing volume, which is achieved by bounded adversarial perturbations.

To measure how collections of variables with varying cardinality contribute to the total influence, we introduce the following formulation:

$$\mathcal{I}_{x_A} := \mathbb{E}[Z \mid do(x_A = [adv(\epsilon)]_k)]$$
$$\mathcal{I}_{(k)} := \frac{1}{\binom{n}{k}} \sum_{x_a \in X_A} \mathcal{I}_{x_A} - \frac{1}{\binom{n}{k-1}} \sum_{x_b \in X_B} \mathcal{I}_{x_B}$$

where $Z = |y - y_{adv}|$ measures the displacement in classification due to adversarial perturbation, and x_A is a set of input variables with cardinality k (alt. x_B with cardinality $k-1$). Intuitively, $\mathcal{I}_{(k)}$ measures the impact of ϵ-adversarial perturbations which is exclusive to subsets of size k. It does this by contrasting subsets of size k with subsets of size $k-1$ using average misclassification as a metric. Most of the paper will be spent justifying and expanding on these choices. Figure 1 illustrates an example of higher-order perturbations leading to a change in classification which would not be possible with a perturbation along any of the lower-order subspaces.

The following four objectives aptly summarize what this work sets out to achieve: 1) briefly review the literature on sensitivity analysis, and draw parallels to current methods of neural network attribution; 2) propose a framework for decomposing the total influence based on perturbations to the power set of input variables; 3) discuss how this decomposition can be used in practice to reveal functional properties not captured by existing attribution methods; 4) demonstrate the effectiveness of the decomposition on synthetic examples and in comparison with existing approaches to attribution.

2 Preliminaries

Implicit in most approaches to interpretability is the assumption that relevant details of the model can be distilled into something that is comprehensible to humans [17]. For attribution, an ideal description of the model should reveal how input features contribute to the final output. To measure this, it's necessary to investigate how changes in a feature lead to changes in the output. Historically, this investigation is referred to as *sensitivity analysis*. As a special case of sensitivity analysis, one may be interested in how the presence of a feature contrasts with its absence. This is the case with many attribution methods, which view the all-zero vector as a baseline that captures the absence of all features. This implies that the study of Boolean functions is relevant to the study of attribution. In the next section, we will briefly introduce literature on sensitivity analysis, and its connection with the analysis of Boolean functions. This discussion will lay a theoretical foundation, before transitioning to its application in neural networks.

2.1 Sensitivity Analysis

Analysis of black-box functions has a long history in theoretical computer science and learning theory. It is often desirable to introduce some measure of function complexity, or to test whether a function belongs to a restricted class (ex: linear, monotone) as a way to gain insight into a functions behavior [14]. We will start by considering the class of Boolean functions of the form $f : \{-1, 1\}^n \to \{-1, 1\}$. We use -1 instead of 0 for ease in later results. A fundamental measure of complexity for Boolean functions is *sensitivity* [18], which measures whether the output bit (y) is flipped when a particular input bit x_i is flipped.

$$Sens_{(i)} := \begin{cases} 1, & \text{if } flip(x_i) \to flip(y) \\ 0, & \text{otherwise} \end{cases}$$

When the sensitivity for a variable x_i is averaged over all samples, the resulting complexity measure is referred to as the *influence*:

$$Inf_{(i)} := \mathbb{E}_s\big[Sens_{(i)}\big]$$

Intuitively, sensitivity and influence measure the ability of individual changes in the input to propagate forward and affect the output. This notion of sensitivity was eventually broadened beyond Boolean functions to refer to any analysis of how input changes affect the function output [19]. This expanded definition of sensitivity analysis clearly encompasses the goals and definitions of neural network attribution.

A well-known property of Boolean functions is their unique expression as multilinear polynomials:

Theorem 1 (Fourier Expansion Theorem [18])**.** *Every function $f : \{-1, 1\}^n \to \mathbb{R}$ can be uniquely expressed with its* ***Fourier Expansion:*** $f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \cdot x^S$ *given* ***Fourier Coefficients:*** $\hat{f}(S) \in \mathbb{R}$*, and monomials* $x^S = \prod_{i \in S} x_i$*.*

This theorem has interesting consequences for attribution if it's possible to quantify the presence or absence of an input feature (as is claimed in the literature). As a quick example, the max function over two variables has the following Fourier expansion: $max(x_1, x_2) = \frac{1}{2} + \frac{1}{2}x_1 + \frac{1}{2}x_2 - \frac{1}{2}x_1x_2$. A method seeking to explain how inputs influence the output should be capable of revealing the second-order coordination between x_1 and x_2. Is this information attainable through perturbations alone?

Many properties of a Boolean function can be directly determined using its Fourier spectrum (the collection of Fourier coefficients), including influence [18]:

$$Inf_{(i)} := \sum_{S \ni i} \hat{f}(S)^2,$$

for a sum over subsets S containing the variable x_i. This relationship reveals that input perturbations of a Boolean function yield access to information about the monomials which uniquely define it. However, this information is confounded due to the sum over Fourier coefficients. To untangle this information, we now consider a higher-order formulation of influence. Instead of changes to individual variables x_i, we now consider changes to subsets x_A of cardinality k. This has the effect of including additional subsets S into the sum of Fourier coefficients which contain any variable present in x_A. If we consider the difference between average influences of order k and $k-1$:

$$\frac{1}{\binom{n}{k}} \sum_{x_A} \sum_{S \ni A} \hat{f}(S)^2 - \frac{1}{\binom{n}{k-1}} \sum_{x_B} \sum_{S \ni B} \hat{f}(S)^2,$$

the result captures the contribution of Fourier coefficients associated with monomials of a particular order. This implies that the relevance of higher-order interactions to the output decision can be determined through the use of higher-order perturbations. Our approach is inspired by this fact, and seeks to generalize to the case of continuous neural networks. The next section describes the steps we take towards this generalization, and challenges the assumptions of existing approaches that attempt to do something similar.

3 Method

3.1 Choosing the output metric

Formalizing a measure of influence in the case of continuous input variables requires a number of critical choices to be made. As a guiding principle, we will attempt to remain as close as possible to the notion of influence for Boolean functions introduced in Section 2.1. Every approach to attribution requires measuring a change in the function output relative to changes in the input. The two most common metrics for measuring output change are: a) the value of the loss function, and b) the output class. Due to its discrete nature, output class is closest to the definition for Boolean functions. If we introduce a single binary variable representing misclassification (0 for incorrect class prediction, 1 for correct), then the two settings are equivalent. Besides this similarity, there are a number of additional reasons to believe that output class is a preferred metric. Changes to the loss function are most commonly associated with derivative-based methods [3], which are not scalable to the higher-order settings we aim to test. Additionally, there are a number of cases where a change in the loss is not informative about misclassification (whereas misclassification is always informative about a change in the loss). One example is when the input sample is near a local minimum, then local changes to the loss may guide it away from a perturbation that leads to misclassification.

3.2 Measuring input changes: local versus global

The next choice is related to how input perturbations are defined for subsets of size k. Since we only care about a subsets ability to influence the output decision, we consider an adversarial perturbation instead of enumerating and averaging over many fixed perturbations. In the Boolean setting, there is a natural sense in which features are flipped on or off to perform a change. In the continuous case, perturbations are made along an interval, and are frequently classified as *local* or *global* [7]. Local perturbations can sometimes behave pathologically, such as in the example of Section 3.1, or when the local change along an axis is 0 (inspiring the sensitivity axiom [1]). Global perturbations (such as the use of the zero-vector as a baseline) are often

used to contrast the presence of a particular feature. Many popular attribution methods employ a hybrid approach that uses a combination of local and global perturbations. One example is Integrated Gradients [1], which measures derivatives along a straight-line interpolation between the sample and the zero-vector. We chose to use a hybrid approach which starts within a limited local region of possible perturbations that expands radially in volume up to a limit. This is accomplished by incrementally increasing the value of ϵ in a norm-bounded adversarial attack. By the nature of image data, if an adversarial perturbation is allowed to search without bounds, it will likely lead to a misclassification. This is true even when considering a small number of pixels, as evidenced by one-pixel attacks [20] and methods that attribute a reduced subset of features [21, 22]. Further comparison to these methods can be found in the Appendix. With this in mind, we start with small values of epsilon and grow until a saturation occurs.

3.3 Challenging neutral baselines

Most approaches to attribution measure the importance of a feature relative to a predefined notion of its absence. Previous work has highlighted some of the issues that can arise from this definition of importance [9, 16], but we wish to expand on some of them in this section, and discuss how our approach avoids them. If the presence/absence of a feature is measured using its linear distance to a baseline, the following cases would result in incorrect attribution: 1) features with opposite baselines. For example, assume our baseline is the zero-vector, and our image is a human eye. The presence of a pupil is defined by the absence of pixel intensity in a particular region, but this would be deemed unimportant relative to our baseline. 2) features with nonlinear importance relative to its distance to the baseline. 3) features with discontinuous regions of importance. For example, a feature may be "present" within a limited local range, and "absent" everywhere else. Measuring with respect to one direction inherently biases the search, and can lead to skewed results. The commonality between each of these examples is that a rich and potentially complex notion of importance along a continuous range is reduced to the simplest constant or linear approximation. The solution is to use more expressive ways of searching the space of global perturbations, and to redefine the notions of importance that use this information. Our work begins to do the former by expanding the search space radially without a bias in any particular direction, and through the use of adversarial search in this space. Under our measure, a variable is considered important if it's inclusion as a coordinate in the perturbation search space leads to a change in classification that would not have otherwise occurred (on average).

3.4 Implementation Details

We define the subset influence of order k for neural networks as the displacement $Z = |y - y_{adv}|$ in the original classification decision after adversarial intervention over a subset of k variables:

$$\mathcal{I}_{x_A} := \mathbb{E}[Z \mid do(x_A = [adv]_k)]$$
$$\mathcal{I}_{(k)} := \frac{1}{\binom{n}{k}} \sum_{x_a \in X_A} \mathcal{I}_{x_A} - \frac{1}{\binom{n}{k-1}} \sum_{x_b \in X_B} \mathcal{I}_{x_B}$$

We use $do(\cdot)$ notation to represent the act of perturbing a subset x_A, inspired by the notation for interventions in a related measure of Average Causal Effect/ Causal Attribution [7]. The expectation in $\mathcal{I}_{x_A}$ is taken over m samples, with the value of m differing in each experimental setting. The value of $\mathcal{I}_{(k)}$ is arrived at by taking averages of this expectation over all subsets of size k and $k-1$. Since it is intractable to consider all subsets if n and k are sufficiently large, we employ two task-dependent sampling strategies. The first is to simply sample l subsets uniformly at random, which is consistent with sampling strategies used by existing higher-order approaches (such as Shapley sampling [23, 24, 15]). The other is to consider contiguous blocks of size $k \times k$, which imposes an implicit bias of local contiguity that works well in certain settings (such as pixel-level image features). The attack strategies for adversaries in $\mathcal{I}_{x_A}$ are a subject of experimentation (see Section 5.4), but many experiments use a simple gradient attack (such as the fast gradient sign method [25], or its L_2 bounded variation).

4 Experiments

4.1 Higher-order Perturbations in Synthetic Boolean Circuits

We'll begin by introducing three small synthetic Boolean circuits with fundamentally different behavior. Using traditional Boolean sensitivity analysis, we will show that higher-order perturbations help to reveal the true logic underlying each function, which is not always possible with single-variable perturbations. Finally, we demonstrate that a standard practice in attribution – measuring influence in a single sample relative to a baseline – fails to distinguish any differences between the functions. This fact will be important for later experiments on neural networks.

The three Boolean functions of interest for our analysis are *parity*, *majority*, and the *OR gate*. We will consider the case of $n = 4$ input variables and one output variable. To briefly explain these functions: *parity*

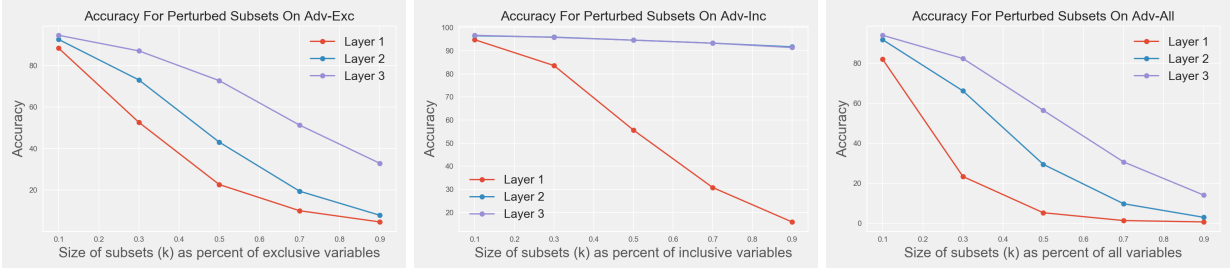

Figure 2: Plots of accuracy versus subset perturbation: "unimportant" (left) and "important" (middle) variables from Integrated Gradients, contrasted with all variables (right)

outputs 1 if the total number of 1's in the input is odd; *majority* outputs 1 if the total number of 1's in the input is $\geq \frac{n}{2}$; and the *OR gate* outputs 1 in all cases, except when all inputs are 0 (alt., -1). Performing standard measurements of sensitivity and influence (as defined in Section 2.1) yield the following results:

Function	Inf_i	*Total Influence*
Parity	1.0	4.0
Majority	0.375	1.5
OR	0.125	0.5

Note that influence is the same for each variable i in these functions, so we only report one value. While these results demonstrate that there are measurable differences in the influence of each function, a single scalar is not always enough to reveal their core properties. For example, the importance of $\frac{n}{2}$ as a threshold in *majority* cannot be determined from 0.375. If we instead use the proposed influence decomposition, where adversarial perturbations are based on some sequence of bit flips, we get:

Function	$\mathcal{I}_{(4)}$	$\mathcal{I}_{(3)}$	$\mathcal{I}_{(2)}$	$\mathcal{I}_{(1)}$
Parity	0.0	0.0	0.0	1.0
Majority	0.0	0.0	0.625	0.375
OR	0.5	0.25	0.125	0.125

Looking at each function independently, we can see that the influence decomposition does a better job of showcasing relevant properties. For *parity*, all of the influence is concentrated around first-order perturbations, owing to the fact that it's a higher-order function, and single bit flips change the global parity. For *majority*, there is a clear separation in the decomposition around $k = \frac{n}{2}$. Once the order of perturbations exceed this halfway point, there is always a way to change the majority, and thus change the output bit. This results in a concentration of influence around the first $\frac{n}{2}$ terms. Lastly, most of the influence in the *OR gate* comes from perturbations to higher-order subsets, since all 1's must be eliminated from the input to change the output.

To contrast the insight gained from the influence decomposition on these synthetic circuits, we will compare to what is learned from standard tools in the attribution literature. We start by assuming that these functions represent the input-output mapping of a neural network. For an example of neural architectures that implement these functions, see the Appendix. Common practice in attribution is to select a sample, and assign scores to individual input variables relative to some baseline. If we assume $\{0, 0, 0, 0\}$ is our baseline, $\{0, 1, 1, 1\}$ is our input sample, and Integrated Gradients is the attribution method, then X_1, X_2, and X_3 will all receive the same non-zero attribution score. The problem is that these variables will be deemed important regardless of which of the three functions we are attributing. It becomes clear that in this setting, Integrated Gradients is simply assigning importance to non-zero inputs, which tells us very little about the function. Given that the primary goals of neural network attribution are interpretability and increased understanding of network properties, such non-informative results should be a cause for concern. Since these are Boolean functions, one might make the case that this behavior is atypical of attribution methods, and not representative of their informativeness on continuous image data. However, in the next section we provide evidence that this is not the case.

4.2 Analysis of Intermediate MLP Layers on MNIST

Having demonstrated the ability of our influence decomposition to reveal meaningful properties on small synthetic circuits, we now move on to its application in neural networks. In this first setup, we train a three layer MLP with *ReLU* activations on MNIST, where each intermediate layer contains 256 neurons. The network was trained for 3 epochs using the standard MNIST train/test split until a test accuracy of 97 percent was reached. This simple network provides a test-

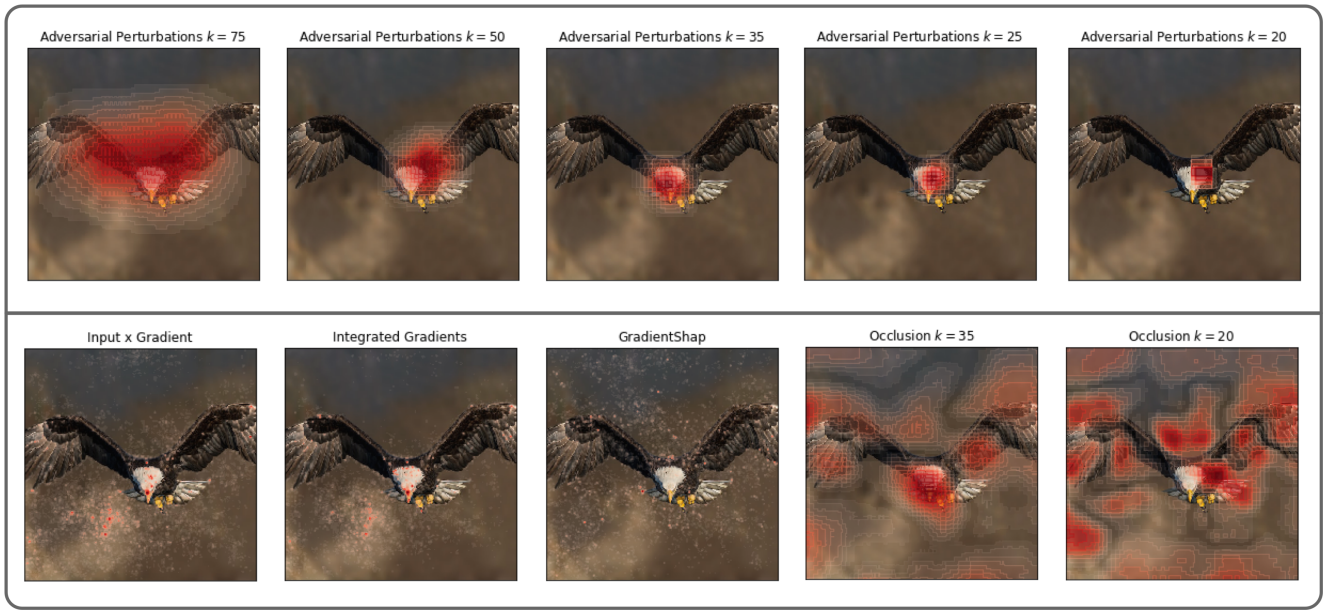

Figure 3: Attribution in our approach (top row) for varied k versus existing approaches (bottom row). For $k = 75$ and 50, $\epsilon = 1.0$, otherwise $\epsilon = 4.0$. Intensity of red represents the absolute value of an attribution.

bed for exploring hypotheses and comparing against existing attribution approaches prior to investigating modern networks.

Consider the task of trying to understand the input-output relationships in this network. For each layer, we will use standard attribution techniques to try to identify which neurons are most important. As an example, we will use Integrated Gradients (also known as Total Conductance or Neuron Integrated Gradients when used on intermediate layers [26]). Using the zero-vector as a baseline, we apply this method and record the importance scores for pre-activation values of all 256 neurons. Next, we measure the average importance for each sample, and use it to threshold the scores. If a neurons importance is greater than the mean for a given sample, we record a 1, otherwise 0. On average, about 20 percent of the neurons were marked as important under this thresholding scheme, which was consistent across each layer. If we ablate neurons not deemed important (Ablation-Exclusive), the accuracy of the network actually increases. Additionally, if we ablate important neurons (Ablation-Inclusive), the accuracy of the network reduces to 0. While this may seem like evidence in favor of Integrated Gradients, the full story is a bit more complicated. If we threshold the pre-activation neurons such that values > 0 are marked as 1 (i.e, we create an activation mask) and compare this with the importance mask of the previous experiment, the two masks are equal for roughly 90 percent of neurons on average (consistent across layers). This implies that for Integrated Gradients, importance is nearly synonymous with activation. This result is similar to the findings on synthetic Boolean circuits. To challenge this, Table 1 shows the results of applying adversarial perturbations to all neurons deemed "important" (Adv-Inc) versus all neurons deemed "unimportant" (Adv-Exc) by Integrated gradients. In general, we observe that even small perturbations ($\epsilon = 0.5$) to "unimportant" variables leads to drastic changes in classification accuracy. In addition to exploring the

Methods	Layer 1	Layer 2	Layer 3
Ablation-Exclusive	99.0	99.7	99.6
Ablation-Inclusive	0.0	0.0	3.0
Adv-Exc($\epsilon = 0.1$)	**88.4**	92.4	94.6
Adv-Inc($\epsilon = 0.1$)	94.5	96.5	96.5
Adv-Exc($\epsilon = 0.5$)	**22.9**	**43.4**	**72.9**
Adv-Inc($\epsilon = 0.5$)	**54.0**	94.2	94.1
Adv-Exc($\epsilon = 1.0$)	**3.2**	**4.9**	**25.4**
Adv-Inc($\epsilon = 1.0$)	**9.9**	89.5	89.2

Table 1: Perturbations to important vs. unimportant neurons

effects of global perturbations to the set of important/unimportant neurons, we can explore perturbations to subsets of these variables through our proposed decomposition. The results of this experiment can be found in Figure 2, which plots how accuracy changes as a function of the subset size. The cardinality of subsets was determined based on a percentage of neurons in a

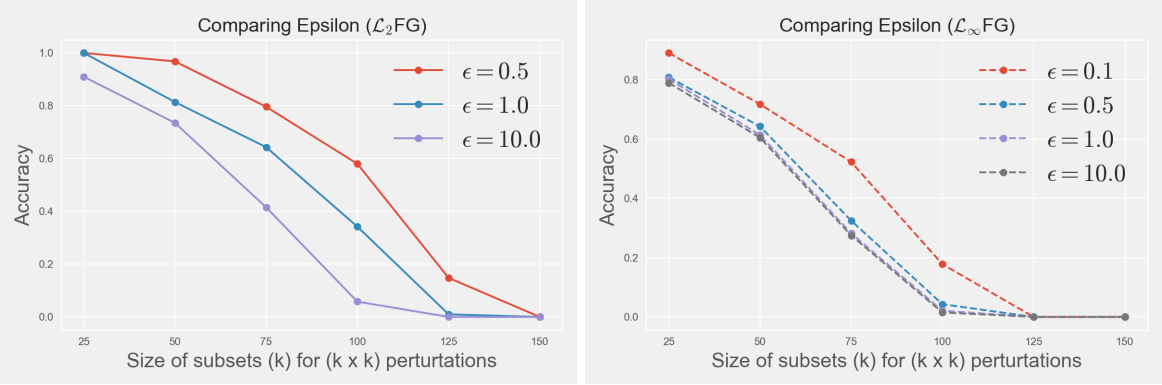

Figure 4: Plots showing the effect of increasing ϵ on adversary strength for fixed norm and attack strategy

particular set (Adv-Inc/ Adv-Exc), since the set size changed for every sample. 500 subsets were chosen uniformly at random for each sample of the test set. We also used this procedure to select subsets from all neurons in a given layer, which appears as the rightmost plot. As the layers progress, neurons deemed "important" (Adv-Inc) by Integrated Gradients are unaffected by small perturbations. Additionally neurons deemed "unimportant" (Adv-Exc) steadily increased in robustness to higher-order perturbations as a function of the layers. This trend was mirrored when all neurons were considered, but to a lesser degree.

Taking local differences based on the accuracies reported in Figure 2 yields an influence decomposition, which is reported in Table 2. If we compare this to the decomposition reported for synthetic circuits, there are a number of similarities. At layer one, most of the influence is concentrated around lower-order perturbations, which resembles the logic of a parity function. At the second layer, most of the influence is concentrated in the first half, which resembles the logic of a majority function. Finally, at the last layer, the logic resembles an OR gate, or reversed Majority. While these comparisons are not exact, they help to illustrate how the logic of network changes as a function of the layer. Small changes in the input can lead to big changes of the output, but once the network forms an intermediate representation, it becomes robust.

4.3 Pixel-level Attributions for ResNet-18 on ImageNet

We now demonstrate the use of our approach in a standard attribution setting, where the goal is to highlight the pixel-level features in a single image that contribute the most to the decision. Plotting the heatmaps for different values of k shows how influence changes as we restrict ourselves to smaller and smaller subsets.

Our method is compared against existing measures of attribution on a pretrained ResNet-18 [27] for ImageNet[28]. The first comparisons are made against commonly used Gradient-based methods, such as Input x Gradients [5], and Integrated Gradients. Given the ResNet architecture, we could not compare to approaches like DeepLift[4], since it has compatibility issues with the ReLU activations. We wanted to compare with Shapley value approaches (since alternative approaches to higher-order decomposition use them [15]), but every implementation we tried was too computationally expensive (even for a single ImageNet sample under maximally restrictive assumptions). As a substitute, we show attributions for GradientShap[29], which is a gradient-based approximation of the Shapley values. Finally, we compare against Occlusion [8] for different sized $k \times k$ blocks of ablated input pixels. Many of the attribution methods we used were based on implementations from the Captum library [30]. Occlusion is the approach that is the most conceptually similar to ours for a number of reasons. To start, occlusion samples contiguous blocks of the input space. Although our approach is technically not restricted to contiguous blocks as a sampling strategy, we found it to be the best choice for evaluating the input pixels of an image. Another similarity is that occlusion applies a perturbation to each selected subset of pixels. The difference is that in Occlusion, the perturbations are always to the zero vector. As discussed in previous sections, this is to contrast the presence of signal in a subset with a global change capturing its absence. In addition to the flaws pointed out in Section 3.3, large

-	$\mathcal{I}_{(180)}$	$\mathcal{I}_{(140)}$	$\mathcal{I}_{(100)}$	$\mathcal{I}_{(60)}$	$\mathcal{I}_{(20)}$
Layer 1	4.7	5.4	12.6	**29.9**	**35.7**
Layer 2	7.9	11.6	**23.6**	**29.9**	**19.5**
Layer 3	**32.9**	**18.6**	**21.4**	15.7	7.5

Table 2: Influence Decomposition for 3-layer MLP

global perturbations to many variables may lead to a changes in classification despite the variables not being important. This idea is brought up in [11], where they argue that such large changes bring the sample out-of-distribution. In this setting, it's impossible to tell whether the attribution is due to the out-of-distribution errors, or feature importance. Since our approach only makes imperceptible adversarial changes relative to the input sample, the likelihood of distribution shift is greatly reduced.

In the visualizations of Figure 3, we use adversarial perturbations based on the $\mathcal{L}_2$-FG (FG stands for Fast Gradient [25]) attack. This approach applies an ϵ change independently to each variable of the subset in the direction opposite the gradient. The global change is then bounded based on the $\mathcal{L}_2$ norm. This metric was chosen for computational efficiency and ease of use. In order to compare with existing attribution methods and properly visualize the scores, we restrict ourselves to evaluation on a single input image. Note that this differs from previous experiments that average over multiple samples. Given an input sample, we construct a $(k \times k)$ sliding window with stride 4, which is used for both our approach and the occlusion baseline. If perturbations to pixels within this window lead to misclassification, each perturbed pixel is assigned a uniform score of $1/(k \times k)$. The pixel-level scores of all misclassifying windows are averaged and displayed as a heatmap. Across multiple values of k, our approach produces cleaner qualitative results. Additionally, different values of k seem to focus on different parts of the image, potentially hinting at relevant features at that order. Approaches based on gradients do not show intuitive results for this image, and Occlusion demonstrates its tendency to over-attribute many irrelevant parts of the image (such as the background). Although we used this particular sample as a representative for the paper, it was by no means an outlier. We show additional examples of qualitative evaluation in the Appendix. The main computational cost of our approach comes from the number of times the network is evaluated, scaling roughly $O(Mn)$ for M the number of subsets (sliding windows) and n the number of samples. Each heatmap took less than 20 seconds to generate on a single GTX 2070 GPU with a batch size of 250. While this is longer than the gradient methods (such as Integrated Gradients, which took 2 seconds per plot and used a single pass of the network), it is still a reasonable amount of time for a higher-order approach. In contrast, all approaches to estimate the Shapley values (other than GradientShap) were unable to run on ImageNet due to memory issues and intractable computational complexity.

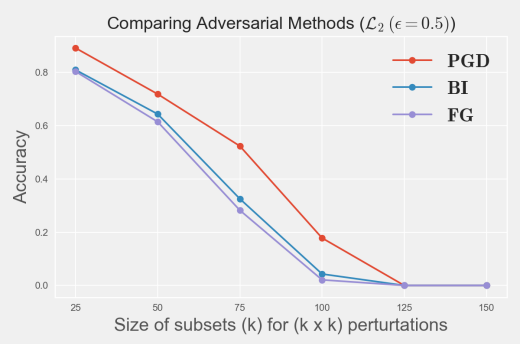

Figure 5: Plot showing the effect of attack strategy on adversary strength for fixed norm and ϵ.

4.4 Effect of Adversarial Attack on the Influence Decomposition

Lastly, we look at the effect of hyperparameters associated with adversarial perturbations on terms of the influence decomposition. In particular, we will look at the perturbation radius ϵ that bounds the search space, the adversarial attack strategy, and the norm ($\mathcal{L}_2$ or $\mathcal{L}_\infty$) that ϵ is measured in. We use the same ResNet-18 architecture introduced in the previous section, and average over 20 randomly selected images. Within a given bounded region specified by the algorithm, some attack strategies may be better equipped to find a perturbation which leads to a change in classification. Employing powerful attack strategies that are able to discover these changes is a critical part of accurate reporting for our decomposition. However, there is always a trade-off between how powerful an attack is, and how computationally intensive it is. For each experiment, we measured the output accuracy after applying perturbations to $k \times k$ sliding windows of various size. The first set of results appear in Figure 4, which fixes the attack strategy and the norm, but varies ϵ. Both plots reveal that increasing ϵ generally leads to an increase in successful attacks. When using $\mathcal{L}_2$ as the norm (leftmost plot), there is a clear drop in accuracy every time ϵ increases. Additionally the shape of each curve is different, implying the influence decompositions will change slightly. For example, when using $\epsilon = 0.5$, less influence will be concentrated around lower-order perturbations (relative to the other values of ϵ). In contrast, the $\mathcal{L}_\infty$ norm (rightmost plot) is less sensitive to changes in ϵ. A saturation point is reached around $\epsilon = 0.5$, and larger values don't increase the misclassification. $\mathcal{L}_\infty$ leads to stronger adversaries than $\mathcal{L}_2$, as $\mathcal{L}_\infty(\epsilon = 0.1)$ is almost as strong as $\mathcal{L}_2(\epsilon = 10.0)$. We did not include $\mathcal{L}_2(\epsilon = 0.1)$ because it did not lead to any misclassification.

Figure 5 shows results for accuracy as a function of attack strategy. Here, PGD stands for projected gradi-

ent descent [31], and BI stands for the basic iterative method [32]. We attempted to compare against iterative noise attacks [33] – where an $\mathcal{L}_p$ ball is explored by iteratively sampling random noise from some distribution (Uniform or Gaussian) – but these approaches resulted in very weak adversaries. Additionally, we tried to use stronger adversaries (such as the Boundary attack [34]), but these approaches were too computationally intensive. A large table with all hyperparameter results is included in the Appendix. Although each method does not yield the same accuracy, overall we found very little difference between methods when a norm and ϵ were fixed. This result lends support to the hypothesis that reported results are close to the ground truth, given fixed hyperparameters.

5 Conclusion

In this work, we introduced a method for attributing different orders of influence to the output of a neural network. The method was inspired by influence in Boolean functions, and its ability to elucidate properties of higher-order coordination from perturbations alone. Using these measures, we demonstrated simple cases where existing attribution methods fail to explain the underlying logic of a function. Additionally, we compared against existing approaches to attribution in modern networks, and demonstrated how higher-order influence might shine a new light on interpretability in neural networks.

References

[1] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," 2017.

[2] J. D. Janizek, P. Sturmfels, and S.-I. Lee, "Explaining explanations: Axiomatic feature interactions for deep networks," 2020.

[3] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," 2017.

[4] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," 2019.

[5] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," 2017.

[6] H. Tanaka, A. Nayebi, N. Maheswaranathan, L. McIntosh, S. A. Baccus, and S. Ganguli, "From deep learning to mechanistic understanding in neuroscience: the structure of retinal prediction," 2019.

[7] A. Chattopadhyay, P. Manupriya, A. Sarkar, and V. N. Balasubramanian, "Neural network attributions: A causal perspective," 2019.

[8] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," 2013.

[9] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, "The (un)reliability of saliency methods," 2017.

[10] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," 2017.

[11] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," 2019.

[12] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," 2018.

[13] C. S. C. S. J.C. Helton, J.D. Johnson, "Survey of sampling-based methods for uncertainty and sensitivity analysis," vol. 91, pp. 1175–1209, 2006.

[14] O. Goldreich, S. Goldwasser, and D. Ron, "Property testing and its connection to learning and approximation," *J. ACM*, vol. 45, p. 653–750, July 1998.

[15] K. Dhamdhere, A. Agarwal, and M. Sundararajan, "The shapley taylor interaction index," 2020.

[16] P. Sturmfels, S. Lundberg, and S.-I. Lee, "Visualizing the impact of feature attribution baselines," *Distill*, 2020. https://distill.pub/2020/attribution-baselines.

[17] Z. C. Lipton, "The mythos of model interpretability," 2017.

[18] R. O'Donnell, *Analysis of Boolean Functions*. USA: Cambridge University Press, 2014.

[19] A. Saltelli, *Global sensitivity analysis: the primer*. John Wiley, 2008.

[20] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, p. 828–841, Oct 2019.

[21] S. Singla, E. Wallace, S. Feng, and S. Feizi, "Understanding impacts of high-order loss approximations and features in deep learning interpretation," 2019.

[22] M. Chapman-Rounds, M.-A. Schulz, E. Pazos, and K. Georgatzis, "Emap: Explanation by minimal adversarial perturbation," 2019.

[23] R. J. AUMANN and L. S. SHAPLEY, *Values of Non-Atomic Games*. Princeton University Press, 1974.

[24] M. Ancona, C. Öztireli, and M. Gross, "Explaining deep neural networks with a polynomial time algorithm for shapley values approximation," 2019.

[25] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[26] A. Shrikumar, J. Su, and A. Kundaje, "Computationally efficient measures of internal neuron importance," 2018.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[29] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017.

[30] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, J. Reynolds, A. Melnikov, N. Lunova, and O. Reblitz-Richardson, "Pytorch captum." `https://github.com/pytorch/captum`, 2019.

[31] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019.

[32] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2017.

[33] J. Rauber and M. Bethge, "Fast differentiable clipping-aware normalization and rescaling," 2020.

[34] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," 2018.