Explaining Local, Global, And Higher-Order Interactions In Deep Learning

Sam Lerman,[1] **Chenliang Xu,** [1] **Charles Venuto** [2] **Henry Kautz** [1]
[1] University of Rochester
[2] University of Rochester Medical Center
slerman@ur.rochester.edu, chenliang.xu@rochester.edu, Charles.Venuto@chet.rochester.edu, henry.kautz@gmail.com

Abstract

We present a simple yet highly generalizable method for explaining interacting parts within a neural network's reasoning process. First, we design an algorithm based on cross derivatives for computing statistical interaction effects between individual features, which is generalized to both 2-way and higher-order (3-way or more) interactions. We present results side by side with a weight-based attribution technique, corroborating that cross derivatives are a superior metric for both 2-way and higher-order interaction detection. Moreover, we extend the use of cross derivatives as an explanatory device in neural networks to the computer vision setting by expanding Grad-CAM, a popular gradient-based explanatory tool in computer vision, to the higher order. While Grad-CAM can only explain the importance of individual objects in images, our method, which we call TaylorCAM, can explain a neural network's relational reasoning across multiple objects. We show the success of our explanations both qualitatively and quantitatively with a human study. Code for all experiments, fully reproducible, may be found at https://www.github.com/slerman12/ExplainingInteractions.

1 Introduction

The universe is made up of myriad interacting parts. To truly understand complex systems and processes, it is not enough to view their functions as an amalgamation of independent contributors. Rather, they are a complex web of inter-operating influences (Battaglia et al. 2018). For much of the past, explainable deep learning has concerned itself with identifying important features, feature vectors, and isolated concepts. However, in the real world, humans intuitively understand that decisions are consequences of complex relations, not merely extrapolations from rankings of singular phenomena. For example, upon seeing a yield sign, it is natural to look to see if there are also passing cars. If not, the yield sign may be safely dismissed and one could keep driving without stopping. If there is a passing car, the law is to yield to the other car. If an intelligent agent made the decision to stop upon approaching a yield sign and a passing car, explaining their actions with precision would require an explanation of this interaction. As far as individual factors go, perhaps a nearby pedestrian is also present, but without an interactional interpretation, one would not be able to distinguish the independence of the yield sign and passing car from the pedestrian, and one would not be privy to the knowledge of the salient interaction. Furthermore, a naive observer might think that yield signs always indicate "stop" without realizing that the agent's response to the yield sign would depend on the presence of a passing car. Similarly, explaining an agent's strategies in any task — be it computer vision, natural language processing, biomedicine, reinforcement learning, or future forecasting — is imprecise without an interactional approach. In chess, good strategies are derived from different interactions of pieces; a strategy may not be wholly inferred from just seeing what individual pieces the agent prioritized. In the economy, crashes are not easily summarized and if one is forecasted, preventing it requires an understanding of many dependencies.

In light of all of this, we propose a number of contributions towards explaining interactions in deep learning. To begin, we design a method for extracting interaction effects based on input cross derivatives that we call T-NID. Inter-

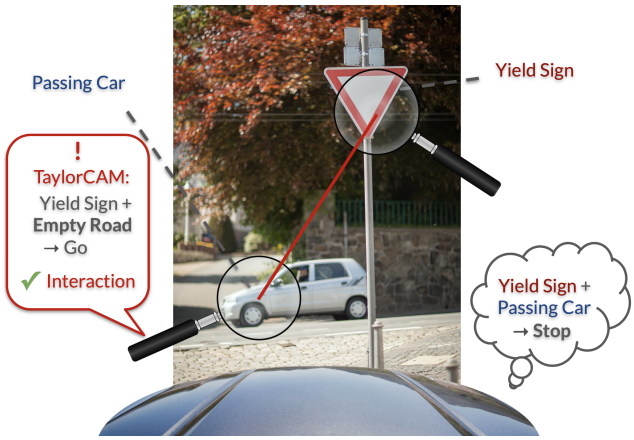


Figure 1: An automated driver decides whether to "stop" or "go." Here, the decision cannot be explained by individual factors alone, but by the interaction between the yield sign and the passing car. TaylorCAM identifies interactions by considering how changing one object affects the significance of another, such as how changing a passing car into an empty road would change the meaning of the yield sign from "stop" to "go."

action effects are a fundamental notion in statistics (Wonnacott and Wonnacott 1977). Our method generalizes existing formalisms and surpasses baselines with both pairwise and higher-order interactions. We make this computation tractable by translating local interaction effects into global interaction effects via representative samples and employing a simple subsampling heuristic.

Then, we generalize Grad-CAM (Selvaraju et al. 2017), an input gradient-based method for explaining feature vector importances, to the two-way and higher-order setting using our interaction effects formalism, and in doing so, we enable the explanation of interactions of multidimensional representations in arbitrary deep neural networks. This method, which we call TaylorCAM, is demonstrated on explaining a neural network's relational reasoning, and we verify the quality of our explanations quantitatively with a human study.

Finally, we conduct a real-world application of these techniques on a many-dimensional biomedical dataset with which we explain the interacting factors behind the progression of Parkinson's disease. In the real world, if one asks a clinician about a single variable such as *age* — "How does age affect disease progression?" — the answer is usually "it depends." The natural question, which we attempt to answer, is "it depends on what?" For example, what is the individual's gender? What medications are they taking? How severe is their current disease status? In order to reflect reality and the true complexity of disease progression, such higher-order interactions must be understood in biomedicine.

2 Related Work

Recently, there have been several attempts to compute statistical interactions with deep learning. Neural Interaction Detection (NID) (Tsang, Cheng, and Liu 2018) used neural network weights to interpret interactions, observing that interactions occur at nonlinear activations in the first hidden layer of an MLP. Like our approach T-NID, (Cui, Marttinen, and Kaski 2019) used gradient information to compute statistical interaction effects. However, they relied on Bayesian neural networks, required averaging a high number of hessians, and only computed global interaction effects, not focusing on local or higher-order interactions. (Eberle et al. 2020) use cross derivatives between single features to explain interactions in deep similarity models, whereas we use an adaptation of Grad-CAM to demonstrate explainability in a more general computer vision setting. (Song et al. 2019) relied on self attention (Vaswani et al. 2017) to compute a measure analogous to non-emergent interaction effects and apply this to an analysis in the biomedical domain. Higher-order interactions have been considered throughout biomedicine, particularly for understanding gene interactions (Yi 2010; Aschard 2016; Liu, Zeng, and Gifford 2019; Chen and Thomas 2010; Caruana et al. 2015).

(Cui, Marttinen, and Kaski 2019) applied their approach to a toy MNIST dataset consisting of a fixed set of feature vectors such that they could compute global interaction effects, but they mapped those feature vectors to single neurons and computed standard interaction effects between those mapped neurons. The limitation of this approach is that it cannot be used to explain local phenomena, which is traditionally what is of interest in computer vision, NLP, and other areas where multidimensional feature vectors are used.

(Ross, Hughes, and Doshi-Velez; Sundararajan, Taly, and Yan 2017; Hechtlinger 2016) used input gradients to explain the reasoning of a neural network. (Zhou et al. 2016) did so with class activation maps. Grad-CAM (Selvaraju et al. 2017) and Grad-CAM++ (Chattopadhay et al. 2018) combined both approaches to localize important feature vectors in computer vision with class activation maps and gradients. Similar to us, (Montavon et al. 2017) use Taylor decomposition to explain neural network decisions, but only for main effects, not interactions.

We also connect the notion of interaction effects with relational reasoning, which has received increased attention in deep learning (Battaglia et al. 2018; Santoro et al. 2017; Zambaldi et al. 2019; Santoro et al. 2018), and use our method of TaylorCAM to interpret the reasoning process of Relation Networks (Santoro et al. 2017). While most past works have mainly focused on explaining individual factors of a neural network's predictions, the weights in multi-head dot product attention (Vaswani et al. 2017) could be interpreted as interactional explanations for neural networks that include MHDPA in their architecture (Song et al. 2019). The interactions identified in this manner may not necessarily be emergent or naively extrapolated to higher orders. In contrast, TaylorCAM is applicable to explaining any sufficiently differentiable neural network directly from its gradient information.

Contemporaneously, both (Janizek, Sturmfels, and Lee 2020) and (Tsang, Rambhatla, and Liu 2020), like our substitution of ReLU with GELU, substitute ReLU with Sofplus in order to induce differentiability. While our TaylorCAM formulation is expressly adapted from Grad-CAM for intuitively explaining feature vectors in CNNs, (Janizek, Sturmfels, and Lee 2020) derive their formulation from integrated gradients and (Tsang, Rambhatla, and Liu 2020) directly use cross partials. The latter, like our work, translate local interaction effects to global interaction effects by aggregating across representative samples. While they use a random batch, we use a small subset of common aggregates.

Unlike other works, we expressly derived TaylorCAM for the purpose of explaining interactions between higher level representations projected by neural networks, such as feature maps from a CNN, which standardly represent objects in computer vision. Thus, we project an arbitrary-size grid of features rather than using raw RGB pixels. This is what we mean when we discuss interactions between "multidimensional features." In our case, we used a 5x5 grid with regions the size of the bounding boxes in Figure 2. As Grad-CAM is built on projected feature vectors in addition to gradients, so is our higher-order extension w.r.t. cross derivatives to explain interactions rather than isolated phenomena.

3 Statistical Interaction Effects

We will discuss three kinds of interaction effects: local, global, and higher-order. Local interactions occur within individual datapoints and vary across the dataset. The automated driving example with an interaction between the yield

sign and oncoming car indicating "stop" illustrates this idea. In computer vision, objects — typically represented by feature vectors projected by a Convolutional Neural Network (CNN) — interact differently from point to point. Global interactions come in the form of universal features across the whole dataset. These are summarized not for one point, but for general points in the entire domain. An example of this may be the various interactions of biomedical features that hold across patients, *e.g.*, how two medications, when administered separately, may generally be beneficial, but when administered together, may instead be harmful.

We will begin formalizing this notion by defining statistical interaction as follows:

Definition 3.1. Statistical Interaction An interaction of order ℓ is a set of unique variables $x_1, ..., x_\ell$ which have a nonzero *interaction effect*.

Next, we will define interaction effect as follows:

Definition 3.2. Interaction Effect An interaction effect $\boldsymbol{IE}_{1,...,\ell}$ between variables $x_1, ..., x_\ell \in x$ on a function $F(x)$ with inputs x is measured as:

$$\boldsymbol{IE}_{1,...,\ell} = \frac{\partial^\ell F(x)}{\partial x_1 \cdots \partial x_\ell}. \tag{1}$$

This definition is inspired by the theory suggested by (Ai and Norton 2003). In plain English, an interaction effect is how much the meaning of one variable changes for a unit change in another variable. Naturally, this change is reflected by the cross partial derivative. "Change" is an intuitive measure for interaction. From the earlier example, given a representation of a yield sign and an oncoming car, *changing* the representation of the oncoming car into a representation of an empty road also changes the meaning of the yield sign from "stop" to "go." For a more formal example, consider $F(x) = x_1 sin(x_2) + cos(x_3)$. F consists of an interaction between x_1 and x_2 for some x since $\partial F(x)/(\partial x_1 \partial x_2)$ is nonzero. However, x_3 does not belong to an interaction since any cross derivative w.r.t. x_3 is zero. We discuss our definition above with the colloquial understanding of "interaction" as well as the mathematical meaning of relation in the *Appendix*.

Adapt to Neural Networks Substituting F with a trained neural network, we can compute the local interaction effects for a datapoint up to order ℓ as long as the neural network F is ℓ-times differentiable. In classification, softmax ensures this to be the case. In regression, we substitute ReLUs with Gaussian-error rectified linear units (GELUs), which have been shown to be comparable in performance (Hendrycks and Gimpel 2016). Otherwise, this formalism affords the computation of interaction effects for arbitrary neural network architectures.

Translate Local Effects to Global Effects While computing local interaction effects is relevant to two of our application domains — computer vision and relational reasoning — typically in statistics, there is greater interest in computing global interaction effects. In tandem with our work, (Cui, Marttinen, and Kaski 2019) converted local pairwise interaction effects to global pairwise interaction effects by averaging a set of representative samples retrieved via k-means clustering, in effect dividing the dataset by Euclidean distance and computing the global average from the centroids. We will similarly average representative local interaction effects in order to compute a global summary, but we will use a simpler and more efficient technique. In our case, efficiency is of more concern because computing higher-order interaction effects requires the computation of higher-order derivatives, which for many samples can become intractable. To translate local interaction effects into global interaction effects at any order, we sample representative samples that have a wide range over the dataset and that are potentially meaningful. We choose the samples that are closest to a subset of common aggregates, including mean, median, min, max, and mode. As well as a random sample for good measure. Likewise, we used L2 distance to measure closeness. In addition to this, we considered different ways to aggregate the interaction effects of these samples. Again, namely mean, median, min, max, or mode. We ran a wide sweep of the complete power set of these potential samples and aggegates to find which combination performed best on a wide array of synthetic datasets distinct from those we trained on selected from prior works (Tsang, Cheng, and Liu 2018; Sorokina et al.; Lou et al.; Hooker), chosen to test for various types of interactions. Results of this power sweep are reported in the *Appendix*. We ended up using the mean interaction effect of the samples closest to the mean, minimum, and mode of all samples, as well as a random sample.

Improve Efficiency Another heuristic for efficiency that we employed was subsampling the interactions that would be computed. Naturally, testing for every combination up to order ℓ would be very expensive. Every double, every triple, every quadruple, etc. — the problem grows combinatorially. We were able to mitigate this to a degree by taking advantage of the property of statistical interaction effects that *an ℓ-way interaction can only exist if all its corresponding (ℓ - 1)-interactions exist* (Sorokina et al.). In turn, we were able to reduce the search space by only selecting non-redundant combinations of the k interactions from the previous order whose interaction effects were highest, beginning with using every combination up to order o and then subsampling the top k for every order thereafter.

Our complete algorithm, which we call Taylor-Neural Interaction Detection (T-NID) due to the higher-order derivatives, is described in pseudocode in the *Appendix*.

Finally, we need to make a point about the sign of the resulting cross partial derivatives. A positive value indicates change in the positive direction; negative, negative. Since in regression we are interested in the overall effect of an interaction and are agnostic to the direction, we take the squared value of the cross-partial as our measure of interaction effect. In contrast, for classification, we use the sign — positive or negative — corresponding to the class of interest. And for multi-class classification, we take F to be the network corresponding to the class output of interest, usually sampling the class with the highest estimated probability, and use its squared cross partial derivatives.

4 TaylorCAM

To this point, we have generalized our computation of interaction effects to the local, global, and higher-order setting, but we have not yet considered the case where features are multidimensional, as is the case in higher-level deep neural network representations.

Explaining the influence of feature vectors is common in computer vision and is a mainstay of interpreting CNNs. However, we have illustrated with multiple examples why a precise explanation of a model's decisions requires an explanation of its interacting components, not just singular entities.

Intuition

Unfortunately, for arbitrary objects in the computer vision setting, a cross derivative alone is not sufficient. Besides the obvious reason that such objects are not represented by singular features but by multidimensional feature vectors learned by a CNN, it is also because fundamentally a cross derivative measures changes of changes. More formally, a cross derivative $\frac{\partial^2 F}{\partial x \partial y}$ measures the effect of a unit change of x on the effect on F of a unit change of y. When reasoning about visual relations, it is convenient to think of dependencies between objects that inform a decision, such as the dependency between a yield sign and a passing car in informing an automated driver's decision to "stop" or "go." *Changing* the passing car into another object, such as merely an empty road, would on its own change the neural network's interpretation of the yield sign from meaning "stop" to meaning "go," even while keeping the yield sign fixed and unchanged — yet a cross derivative only measures the effect of changing both. To account for this, instead of naively using cross derivatives, we measure how much changing one object would change the *importance* of another object to a neural network's decision, *e.g.*, how changing the yield sign into a speed limit sign would change the passing car's importance or how changing the passing car into a gush of leaves would change the yield sign's importance with regards to the decision of whether to "stop" or "go" — even when not necessarily both are changed.

Given car C, yield sign Y, and binary decision "go" G, this intuition may be summarized mathematically as follows:

$$S_{Y,C} = \partial IMP(Y,G) / \partial C, \tag{2}$$

where $S_{Y,C}$ represents the interaction salience between the yield sign and passing car, and $IMP(Y,G)$ represents the importance of the yield sign to the neural network's decision to go or stop. Fortunately, the importance of individual objects in computer vision is the characteristic problem of the explanatory tool Grad-CAM (Selvaraju et al. 2017; Zhou et al. 2016; Chattopadhay et al. 2018), which we use to derive our method. We use the term *interaction salience* due to the deviation from interaction effects in Definition 3.2.

Methodology

Suppose we have an ℓ-times differentiable function $F : \mathbb{R}^{n,d} \to \mathbb{R}$, which will stand for our neural network, where $\ell \geq 2$. F takes in matrix $\boldsymbol{x}$ consisting of n feature vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_n \in \mathbb{R}^d$ of dimension d. So F is the portion of the network downstream of a set of feature vectors such as those projected by a CNN, which we flatten along the height and width dimension to produce $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$.

Quantify Importance To fill IMP in Equation 2, we turn to class activation maps (CAMs) (Zhou et al. 2016). However, as observed by the solution of (Selvaraju et al. 2017), to find out how a class activation map increases the class's likelihood, we would like to know how its features contribute to the output, which we can do with their gradients. We can estimate the global effect by summing the gradient of each feature vector $\boldsymbol{x}_k$ and weighing the sum to each CAM. This amounts exactly to Grad-CAM (Selvaraju et al. 2017):

$$\begin{aligned} IMP(\boldsymbol{x}_i, F(\boldsymbol{x})) &= GradCAM(\boldsymbol{x}_i, F(\boldsymbol{x})) \\ &= \sum_p \boldsymbol{x}_{ip} \sum_k \frac{\partial F(\boldsymbol{x})}{\partial \boldsymbol{x}_{kp}} \end{aligned} . \tag{3}$$

Generalize Grad-CAM to Compute Interactions Now that we have the importance of a feature vector (via essentially Grad-CAM), we can formulate $\boldsymbol{S}_{ij}$, the interaction salience between feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, by substituting Equation 3 into Equation 2 and summing the dimensions as follows:

$$\boldsymbol{S}_{ij} = \sum_m \partial \left[\sum_p \boldsymbol{x}_{ip} \sum_k \frac{\partial F(\boldsymbol{x})}{\partial \boldsymbol{x}_{kp}} \right] \Big/ \partial \boldsymbol{x}_{jm}. \tag{4}$$

Merge with Statistical Interaction Effects Finally, we bring this to an easy-to-compute form by realizing that the partial derivative in the denominator $\partial \boldsymbol{x}_j$ can be computed together with the partial derivative in the numerator. We also square the salience because a change of importance in either direction would be significant. We note that the following is a generalization of Grad-CAM that reduces elegantly to a modified interaction effects Definition 3.2:

$$\begin{aligned} \boldsymbol{S}_{ij}^2 &= \left(\sum_m \sum_p \boldsymbol{x}_{ip} \sum_k \frac{\partial^2 F(\boldsymbol{x})}{\partial \boldsymbol{x}_{kp} \partial \boldsymbol{x}_{jm}} \right)^2 \\ &= \left(\sum_{m,p,k} \boldsymbol{x}_{ip} \boldsymbol{IE}_{kp,jm} \right)^2 \end{aligned} . \tag{5}$$

In tests, we found setting $k = i$ in Equations 3 - 5 without the global sum over k to perform just as well and often better, perhaps because the local gradients in Equation 3 more precisely correspond to features. We call Equation 5 HessianCAM. HessianCAM may be further differentiated with respect to a cross partial $\partial \boldsymbol{x}_q$ to get a 3-way interaction salience, and that can be further differentiated up to any order ℓ. Thus, we name this TaylorCAM, a higher-order generalization of Grad-CAM, where Grad-CAM (or a close variant) is the special case $\ell = 1$ and HessianCAM is the special case $\ell = 2$.

Note that interaction saliences are conditional. The interaction salience of feature $\boldsymbol{x}_i$ on feature $\boldsymbol{x}_j$ is not necessarily the same as that of $\boldsymbol{x}_j$ on $\boldsymbol{x}_i$. Interaction salience $\boldsymbol{S}_{ij}$

represents the influence of $\boldsymbol{x}_i$ on the importance of $\boldsymbol{x}_j$. Interaction salience $\boldsymbol{S}_{ijk...}$ represents the influence of $\boldsymbol{x}_i$ on the interaction salience of interaction $\boldsymbol{x}_j, \boldsymbol{x}_k, ...$ To address this, we sum the mutual pairs, *e.g.*, $\boldsymbol{S}_{ij} + \boldsymbol{S}_{ji}$, although we note that we did so only to make the presentation clearer and not because it is required. For many interpretation tasks, understanding that the meaning of the yield sign depends on the car, but the meaning of the car does not depend on the yield sign is crucial to getting the most precise understanding. Computing the mutual pairs does not require recomputation of any derivatives, and can be achieved easily by permuting the resulting interaction saliences and summing them, as demonstrated in our public code. Lastly, we zero out the diagonals and redundant grid cells of the resulting interaction saliences to only consider interactions between non-redundant feature vectors.

Limitations

One limitation of TaylorCAM, much like Grad-CAM, is that "importance" is based on contribution to the output, so if two different objects have the same contribution to the output, then changing one into the other would be considered meaningless, and so the interactions might not be identified. Suppose we have the setup from Sort-Of-CLEVR (Johnson et al. 2017), a relational reasoning task. Here, we have an image with an assortment of shapes of different colors and a relational question related to that image. An example of this limitation is when an agent is asked, "What is the color of the circle furthest from the red square?" If the furthest circle is blue, and the second furthest is also blue, then changing the furthest into a square does not have a meaningful impact on the red square's contribution to the output, as determined by Grad-CAM, since the answer to the question would be unchanged (blue). Grad-CAM++ (Chattopadhay et al. 2018) may hold an insight as to how to address this, via even-higher order derivatives. Another limitation is that "change" is being measured locally, as derivatives do not account for non-local rates of change. This means that TaylorCAM, like other deep learning explanatory tools, depends on the local regions of representations. Lastly, of course, is the time complexity of computing higher-order derivatives. Higher-order differentiation has become increasingly more accessible with Taylor-mode autograd methods like JAX (Bettencourt, Johnson, and Duvenaud 2019) and libraries like the new Pytorch functional autograd API (Paszke et al. 2017), yet remains a challenge as the order grows. For Hessian-CAM, we had no trouble computing 2nd-order derivatives of Relation Networks using Pytorch and CPU memory. None of our individual explanations required more than a few minutes to compute on a CPU, excluding neural network training times.

5 Experiments

Statistical Interaction Effects

We evaluate T-NID's ability to rank interactions on the suite of synthetic functions proposed by (Tsang, Cheng, and Liu 2018; Sorokina et al.; Lou et al.; Hooker), which were "designed to have a mixture of pairwise and higher-order interactions, with varying order, strength, nonlinearity, and overlap" (Tsang, Cheng, and Liu 2018). These are available to see in the *Appendix* and in Table 1 of (Tsang, Cheng, and Liu 2018).

For pairwise interaction effects (see Table 1), we report or reproduce the experiments of (Tsang, Cheng, and Liu 2018) verbatim, measuring AUC scores between predicted interaction rankings and ground truths. A pair x_i, x_j is considered an interaction either by itself or when it is a subset of a higher-order interaction, as in (Sorokina et al.; Lou et al.). Included for comparison are benchmarks from various statistical and machine learning methods (Wonnacott and Wonnacott 1977; Tibshirani 2011; Sorokina et al.; Tsang, Cheng, and Liu 2018), as reported by (Tsang, Cheng, and Liu 2018). NID (Tsang, Cheng, and Liu 2018) uses an interpretation of the weights from a standard MLP to detect interactions, whereas NID + MLP-M uses an MLP with additional univariate networks summed at the output to discourage modeling of main effects and false spurious interactions. In contrast, our T-NID uses only a standard MLP with GELU activations. Unlike NID, we found no significant benefit from MLP-M or sparsity regularization. Despite the simpler architecture, T-NID is immune to some of the deficits of NID and NID + MLP-M. T-NID is able to distinguish main effects and spurious interactions in F_2 and F_4, and while NID + MLP-M modeled spurious main effects in the $\{8, 9, 10\}$ interaction of F_6, T-NID recognizes it as an interaction, as the cross derivative is nonzero across the domain of x_8, x_9, x_{10}. All around, T-NID performs on par or better than NID at computing pairwise statistical interaction effects on these synthetic tasks. For higher-order interactions, we do not report AUC scores against the full ground truth, as that would grow combinatorially more expensive with higher orders. Since NID also extracts interactions one order at a time, we compare the AUC scores of NID and T-NID one order at a time and use ground truths from the union of their discovered interactions. That way, they can be assessed relative to one another, albeit not universally. In addition to the results reported in Table 2, we tested many variants of architectures and report results with NID + MLP-M in the *Appendix*. In all cases, the relative results were largely the same, with T-NID achieving the highest scores, except less so at 4-way interactions when equipped with its own main effects network (MLP-M). Since any-order NID tends to find supersets much better than subsets, at 3-way interactions, NID misses nearly all present interactions, whereas T-NID fares relatively well. Along with recent works (Cui, Marttinen, and Kaski 2019), we have shown that cross derivatives are a promising metric for interaction attribution in neural networks.

Relational Reasoning

Sort-Of-CLEVR is a toy dataset for relational reasoning proposed by (Santoro et al. 2017). It is a less-computationally expensive 2D form of the CLEVR VQA dataset (Johnson et al. 2017) with a focus on relational questions. In our setup, these questions include distance relationships and compare-and-count tasks. To demonstrate TaylorCAM's capability for explaining a neural network's relational reasoning, we train a Relation Network (RN) (Santoro et al. 2017) on Sort-

	ANOVA	HierLasso	RuleFit	AG	NID	NID + MLP-M	T-NID
$F_1(\mathbf{x})$	0.992	**1.00**	0.754	**1**	0.970	$0.995 \pm 4.4\text{e}-3$	0.962 ± 0.022
$F_2(\mathbf{x})$	0.468	0.636	0.698	0.88	0.79	$0.85 \pm 3.9\text{e}-2$	$\mathbf{0.885} \pm 0.039$
$F_3(\mathbf{x})$	0.657	0.556	0.815	**1**	0.999	$\mathbf{1} \pm 0.0$	0.999 ± 0.001
$F_4(\mathbf{x})$	0.563	0.634	0.689	**0.999**	0.85	$0.996 \pm 4.7\text{e}-3$	0.998 ± 0.003
$F_5(\mathbf{x})$	0.544	0.625	0.797	0.67	**1**	$\mathbf{1} \pm 0.0$	0.991 ± 0.016
$F_6(\mathbf{x})$	0.780	0.730	0.811	0.64	**0.98**	$0.70 \pm 4.8\text{e}-2$	0.954 ± 0.026
$F_7(\mathbf{x})$	0.726	0.571	0.666	0.81	0.84	$0.82 \pm 2.2\text{e}-2$	$\mathbf{0.98} \pm 0.021$
$F_8(\mathbf{x})$	0.929	0.958	0.946	0.937	0.989	$0.989 \pm 4.5\text{e}-3$	$\mathbf{1.0} \pm 0.0$
$F_9(\mathbf{x})$	0.783	0.681	0.584	0.808	0.83	$0.83 \pm 3.7\text{e}-2$	$\mathbf{0.98} \pm 0.023$
$F_{10}(\mathbf{x})$	0.765	0.583	0.876	**1**	0.995	$0.99 \pm 2.1\text{e}-2$	$\mathbf{1.0} \pm 0.0$
Average	0.721	0.698	0.764	0.87	0.92	$0.92 \pm 1.8\text{e}-2$	$\mathbf{0.975} \pm 0.015$

Table 1: AUC scores for pairwise interaction effects. Top-1 scores are bolded.

	3-Way Interactions		4-Way Interactions		5-Way Interactions	
	NID	T-NID	NID	T-NID	NID	T-NID
Average	0.08 ± 0.013	**0.76** ± 0.07	0.75 ± 0.13	**0.78** ± 0.11	0.92 ± 0.06	**0.97** ± 0.05

Table 2: AUC scores for higher-order n-way interaction effects

	Objects	Questions
Grad-CAM	14.0%	29.3%
Random	16.7%	33.3%
TaylorCAM	**34.7%**	**59.3%**

Table 3: Human study

Top N-Way Interaction	Strength
np3rign, handed	2.92E-05
id_num, scau20, mcarec4	4.77E-06
scau13, np1slpn, np1cnst, nhy	6.00E-07
slplmbmv, np1dprs, np2walk, np3rigru, np3pstbl	1.23E-07

Table 4: Top MoCA interactions

Of-CLEVR and visualize its top consecutive interactions in Figure 2. Relation Networks are simple modules augmented to CNNs that enable relational reasoning.

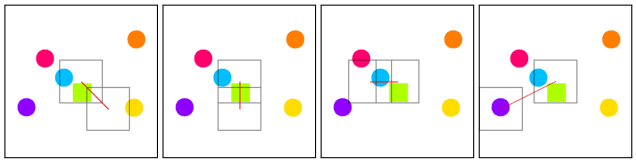
In Figure 2, interacting regions are indicated by two bounding boxes, and the top 4 interactions discovered by TaylorCAM are shown per image. The input is an image of objects and a question about a particular *object of interest* and its relation to another object, and the output is the answer to that question. Since these questions are relational in nature, this problem requires relational reasoning. Specifically, each question asks about an object of interest identified by one of 6 colors: red, green, purple, blue, yellow, or orange. The 3 questions include, "Which shape is closest to the object of interest?", "Which shape is furthest from the object of interest?", and "How many objects have the same shape as the object of interest?" We invite the reader to use the discovered interactions in Figure 2 (as visualized by the bounding boxes) to try to deduce the objects of interest and questions for themselves before looking at the captions. For example, if the top 4 interactions consist of two objects that are close to each other, one might guess that the question was, "Which shape is closest to the object of interest?" If each interaction includes the red square, then one might interpret that the object of interest was the red square and thus the question was, "Which shape is closest to the red square?" In this way, one could reverse engineer the question in a visual question-answering problem from just looking at TaylorCAM's visualized interactions.

While decisions are frequently relational (Battaglia et al. 2018), Grad-CAM is only designed to explain the importance of individual objects in isolation. We observed that TaylorCAM affords much more intuitive explanations of the neural network's reasoning for a question. The object of interest usually belongs to each top interaction, while its corresponding interactions are usually sensible for the question, focusing nearby or far away in proximity questions, and on the appropriate shapes in counting questions. To quantify, we selected a random batch of 15 samples and their ordered interaction saliences, and conducted a small human study ($n = 10$), asking each individual to guess (1) the object of interest and (2) the question being asked, from just looking at the ranked interaction visuals. We report the results in Table 3, demonstrating strong explainability with significantly higher guess-accuracy than using Grad-CAM or random guessing. A more in-depth breakdown of these performances is available in the *Appendix*.

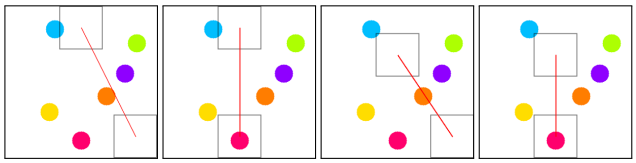
We used our explanations to devise a minor adjustment to the RN architecture, which we call Interactional Relation Network (IRN), that mitigated non-relational behavior and achieved better performance. These details, as well as all architectural details (hyperparameters, layer sizes, epochs), may be found in the *Appendix*.

Biomedical Application

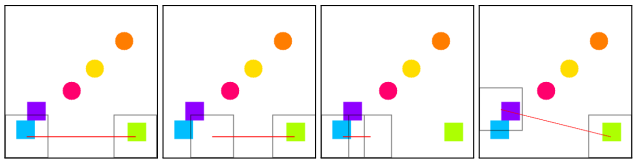
Parkinson's disease (PD) is a neurodegenerative disease characterized clinically by motor and non-motor symptoms that vary over time, progressing interdependently. We classi-



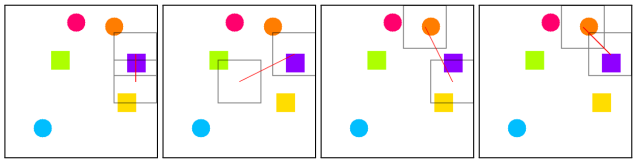
(a) Q: "Which shape is closest to the green square?"



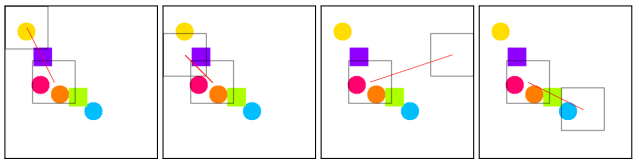
(b) Q: "Which shape is furthest from the blue circle?"



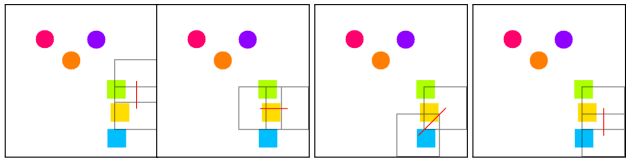
(c) Q: "How many objects have shape of green object?"



(d) Q: "Which shape is closest to the purple square?"



(e) Q: "Which shape is furthest from the pink circle?"



(f) Q: "How many objects have shape of yellow object?"

Figure 2: Shown are the top 4 interactions identified from a Relation Network's predictions on 6 visual question-answering samples. The boxes can be interpreted as saying, "the meaning of one region depends on the contents of the other region." We recommend testing yourself to see if you can guess (1) the object of interest and (2) the question being asked, without looking at the caption. The 6 objects are "blue", "purple", "red", "yellow", "orange", and "green" and the 3 questions are "Which shape is closest to the object of interest?", "Which shape is furthest from the object of interest?", and "How many objects have the same shape as the object of interest?"

fied patients from the PPMI study dataset (http://www.ppmi-info.org/) with more severe progression in decline of cognitive function, as measured by the Montreal Cognitive Assessment (MoCA) scale. Top interactions are displayed in Table 4. The top pairwise interaction was handedness and severity of rigidity in the neck. Handedness has been significantly associated with specific genetic loci implicated in the pathogenesis of neurologic disorders including PD (Wiberg et al. 2019). More severe rigidity symptoms in PD are also associated with faster cognitive decline (Rajput et al. 2009). Our analysis suggests that various measures previously thought to be unrelated should be considered together when predicting faster cognitive progression in PD. See *Appendix* for more details, analysis, and interpretations.

6 Conclusion

With T-NID and TaylorCAM, we have shown that input cross derivatives, combined with a few simple heuristics and intuitions, are a powerful tool for explaining interactions in deep learning. T-NID, using GELU activations, representative samples, and interaction subsampling, successfully ranks statistical interactions, outperforming NID. Meanwhile, TaylorCAM generalizes Grad-CAM to the higher order and effectively explains interactions in object detection and relational reasoning, affording a human cohort the insight to guess questions in VQA from only seeing the top discovered visual interactions. We also tied these metrics to relational reasoning and note that we used them to better customize the Relation Network architecture. To cap it off, we applied T-NID to the real-world problem of classifying rate of clinical progression in Parkinson's disease and made some expected as well as novel observations about potential underlying mechanisms of PD progression. By making our code publicly available, we hope that these simple explanatory tools can be used and built upon to better explain the complex interoperating factors underlying neural network reasoning and the world.

Ethics

A common critique of deep neural networks has been their apparent "black box" nature. Any field that benefits from understanding why a neural network predicts something, not just what it predicts, may benefit from an explanatory tool that affords more precise understandings of relations and dependencies underlying predictions, *e.g.*, biomedicine, economics, and areas where AI might have authority, like in the judicial system. However, it would be dangerous to trust these explanatory systems indiscriminately. If an explanation of clinical disease progression points to a beneficial interaction between two drugs, careful study is needed to determine if those drugs are indeed implicated before administering them as treatment. Although these explanations are useful tools, they are not perfect, and they are only as good as the model they are applied to. A racist model, due to racial biases in data, may explain that the cause of something is racial when in fact the real cause is something more complicated — as always, these technologies should not be trusted blindly.

References

Ai, C.; and Norton, E. C. 2003. Interaction terms in logit and probit models. *Economics letters* 80(1): 123–129.

Aschard, H. 2016. A perspective on interaction effects in genetic association studies. *Genetic epidemiology* 40(8): 678–688.

Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261* .

Bettencourt, J.; Johnson, M. J.; and Duvenaud, D. 2019. Taylor-Mode Automatic Differentiation for Higher-Order Derivatives in JAX. In *Advances in neural information processing systems, Workshop Program Transformations*.

Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; and Elhadad, N. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1721–1730.

Chattopadhay, A.; Sarkar, A.; Howlader, P.; and Balasubramanian, V. N. 2018. Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 839–847.

Chen, G. K.; and Thomas, D. C. 2010. Using biological knowledge to discover higher order interactions in genetic association studies. *Genetic epidemiology* 34(8): 863–878.

Cui, T.; Marttinen, P.; and Kaski, S. 2019. Recovering Pairwise Interactions Using Neural Networks. In *Advances in neural information processing systems, Bayesian Deep Learning workshop*.

Eberle, O.; Büttner, J.; Kräutli, F.; Müller, K.-R.; Valleriani, M.; and Montavon, G. 2020. Building and Interpreting Deep Similarity Models. *arXiv preprint arXiv:2003.05431* .

Hechtlinger, Y. 2016. Interpretation of Prediction Models Using the Input Gradient. *ArXiv* abs/1611.07634.

Hendrycks, D.; and Gimpel, K. 2016. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *CoRR* abs/1606.08415. URL http://arxiv.org/abs/1606.08415.

Hooker, G. ???? Discovering additive structure in black box functions. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, 575. ACM Press. doi:10.1145/1014052.1014122. URL http://portal.acm.org/citation.cfm?doid=1014052.1014122.

Janizek, J. D.; Sturmfels, P.; and Lee, S.-I. 2020. Explaining Explanations: Axiomatic Feature Interactions for Deep Networks. *arXiv preprint arXiv:2002.04138* .

Johnson, J.; Hariharan, B.; van der Maaten, L.; Fei-Fei, L.; Lawrence Zitnick, C.; and Girshick, R. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2901–2910.

Liu, G.; Zeng, H.; and Gifford, D. K. 2019. Visualizing complex feature interactions and feature sharing in genomic deep neural networks. *BMC bioinformatics* 20(1): 1–14.

Lou, Y.; Caruana, R.; Gehrke, J.; and Hooker, G. ???? Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, 623. ACM Press. ISBN 978-1-4503-2174-7. doi:10.1145/2487575.2487579. URL http://dl.acm.org/citation.cfm?doid=2487575.2487579.

Montavon, G.; Lapuschkin, S.; Binder, A.; Samek, W.; and Müller, K.-R. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition* 65: 211–222.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch. In *Advances in neural information processing systems*.

Rajput, A.; Voll, A.; Rajput, M.; Robinson, C.; and Rajput, A. 2009. Course in Parkinson disease subtypes: a 39-year clinicopathologic study. *Neurology* 73(3): 206–212.

Ross, A. S.; Hughes, M. C.; and Doshi-Velez, F. ???? Right for the right reasons: training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, 2662–2670. AAAI Press. ISBN 978-0-9992411-0-3.

Santoro, A.; Faulkner, R.; Raposo, D.; Rae, J.; Chrzanowski, M.; Weber, T.; Wierstra, D.; Vinyals, O.; Pascanu, R.; and Lillicrap, T. 2018. Relational recurrent neural networks. In *Advances in neural information processing systems*, 7299–7310.

Santoro, A.; Raposo, D.; Barrett, D. G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; and Lillicrap, T. 2017. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, 4967–4976.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 618–626.

Song, W.; Shi, C.; Xiao, Z.; Duan, Z.; Xu, Y.; Zhang, M.; and Tang, J. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1161–1170.

Sorokina, D.; Caruana, R.; Riedewald, M.; and Fink, D. ???? Detecting statistical interactions with additive groves of trees. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, 1000–1007. ACM Press. ISBN 978-1-60558-205-4. doi:10.1145/1390156.1390282. URL http://portal.acm.org/citation.cfm?doid=1390156.1390282.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th*

International Conference on Machine Learning - Volume 70, ICML'17, 3319–3328. JMLR.org.

Tibshirani, R. 2011. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73(3): 273–282.

Tsang, M.; Cheng, D.; and Liu, Y. 2018. Detecting statistical interactions from neural network weights. In *International Conference on Learning Representations*.

Tsang, M.; Rambhatla, S.; and Liu, Y. 2020. How does this interaction affect me? Interpretable attribution for feature interactions. *arXiv preprint arXiv:2006.10965* .

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wiberg, A.; Ng, M.; Al Omran, Y.; Alfaro-Almagro, F.; McCarthy, P.; Marchini, J.; Bennett, D. L.; Smith, S.; Douaud, G.; and Furniss, D. 2019. Handedness, language areas and neuropsychiatric diseases: insights from brain imaging and genetics. *Brain* 142(10): 2938–2947.

Wonnacott, T.; and Wonnacott, R. 1977. *Introductory statistics*. Wiley series in probability and mathematical statistics. Wiley. ISBN 9780471959823. URL https://books.google.com/books?id=XmNdAAAAIAAJ.

Yi, N. 2010. Statistical analysis of genetic interactions. *Genetics research* 92(5-6): 443–459.

Zambaldi, V.; Raposo, D.; Santoro, A.; Bapst, V.; Li, Y.; Babuschkin, I.; Tuyls, K.; Reichert, D.; Lillicrap, T.; Lockhart, E.; Shanahan, M.; Langston, V.; Pascanu, R.; Botvinick, M.; Vinyals, O.; and Battaglia, P. 2019. Deep reinforcement learning with relational inductive biases. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=HkxaFoC9KQ.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.