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

Samuel Lerman Charles Venuto Henry Kautz Chenliang Xu
University of Rochester
{slerman@ur.,charles.venuto@chet.,kautz@cs.,chenliang.xu@}rochester.edu

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 for CNNs, to the higher order. While Grad-CAM can only explain the importance of individual objects in images, our method, which we call Taylor-CAM, can explain a neural network's relational reasoning across multiple objects. We show the success of our explanations both qualitatively and quantitatively, including with a user study. We will release all code as a tool package to facilitate explainable deep learning.

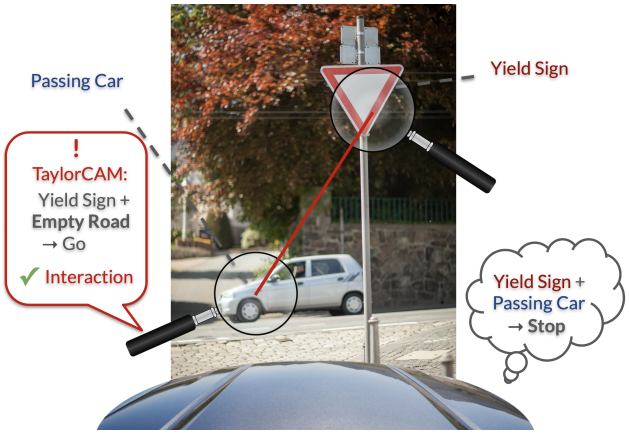


Figure 1: An automated driver decides whether to "stop" or "go." Here, the decision cannot be explained by individual factors alone, but by the interaction between the yield sign and the passing car. Taylor-CAM identifies interactions by considering how changing one object affects the significance of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

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. 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 extrapolated 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. However, interactional strategies are not always summarizable by heatmaps [6, 24, 25, 39, 40] or ordered rankings [10, 21, 29]; and they often require an understanding of many

dependencies — complex dependencies, such as those between higher-level concepts (*e.g.* vector representations in deep neural networks [3, 22, 23, 38]) — not just single-dimensional features as typically explored in the statistical interaction effects literature [9, 13, 31, 32]. In light of all of this, we propose a number of contributions towards explaining interactions in deep learning:

T-NID, an algorithm for statistical interaction effects that outperforms recent state-of-the-art baselines with both pairwise and higher-order interactions. Interaction effects are a fundamental notion in statistics [36]. We make this computation tractable by translating local interaction effects into global interaction effects via representative samples and employing a simple subsampling heuristic.

Taylor-CAM, an explanatory tool that extends Grad-CAM [24], which assigns importances to feature vectors based on input gradients, by generalizing it to the 2-way and higher-order setting using the same formalism of interaction effects as for T-NID. This method is demonstrated on multi-object detection and relational reasoning in visual question-answering (VQA).

Visualizations of Taylor-CAM's explanations that enable a human cohort to reverse engineer questions in relational VQA without knowing the answers and interpret relational reasoning better than with existing explanatory tools like Grad-CAM and GLIDER [31] from just a convolutional neural network's (CNN) feature maps.

2. Related Work

In Deep Learning Recently, there have been several attempts to compute statistical interactions with deep learning. Neural Interaction Detection (NID) [32] 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, [8] 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. [9] used 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. [27] relied on self attention [34] 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 [2, 5, 7, 16, 37].

Cui et al. [8] 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.

[13] and [31], like our substitution of ReLU with GELU, substitute ReLU with Softplus in order to induce differentiability. 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. While our Taylor-CAM formulation is expressly adapted from Grad-CAM for intuitively explaining feature vectors in CNNs, [13] derive their formulation from integrated gradients and [31, 33] directly use cross partials.

Individual Importances [10, 21, 29] used input gradients to explain the reasoning of a neural network. [40] did so with class activation maps. Grad-CAM [24] and Grad-CAM++ [6] combined both approaches to localize important feature vectors in computer vision with class activation maps and gradients, visualized by heatmaps. Similar to us, [18] used Taylor decomposition to explain neural network decisions, but only for main effects, not interactions.

Relational Reasoning We also connect interaction effects with relational reasoning, which has received increased attention in deep learning [3, 22, 23, 38], and use Taylor-CAM to interpret the reasoning process of Relation Networks [23]. 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 [34] could be interpreted as relational explanations for neural networks that include MHDPA in their architecture [27]. In contrast, Taylor-CAM is architecture agnostic and can explain decisions unique to each output dimension directly from gradient information.

Unlike other works, we expressly derived Taylor-CAM for the purpose of explaining interactions between higher level representations, such as feature maps from a CNN, which standardly represent objects in computer vision (rather than using raw RGB pixels). 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 define statistical interaction effect analogous to [1]:

Definition 3.1. Interaction Effect An interaction effect $\boldsymbol{IE}_{1,\dots,\ell}$ between variables $\boldsymbol{x}_1, \dots, \boldsymbol{x}_\ell \in \boldsymbol{x}$ on a function $F(\boldsymbol{x})$ with inputs $\boldsymbol{x}$ is measured as:

$$\boldsymbol{IE}_{1,\dots,\ell} = \frac{\partial^\ell F(\boldsymbol{x})}{\partial \boldsymbol{x}_1 \cdots \partial \boldsymbol{x}_\ell}. \tag{1}$$

In plain English, an interaction effect is how much the

meaning of one variable changes for a unit change in another variable. 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 $F(\boldsymbol{x}) = \boldsymbol{x}_1 sin(\boldsymbol{x}_2) + cos(\boldsymbol{x}_3)$. F consists of an interaction between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ for some $\boldsymbol{x}$ since $\partial^2 F(\boldsymbol{x})/(\partial \boldsymbol{x}_1 \partial \boldsymbol{x}_2)$ is nonzero. However, $\boldsymbol{x}_3$ does not belong to an interaction since any cross derivative w.r.t. $\boldsymbol{x}_3$ is zero.

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 comparable in performance [11]. Otherwise, Definition 3.1 affords the computation of interaction effects for arbitrary neural network architectures.

Translate Local Effects to Global Effects Often in statistics, there is greater interest in computing global interaction effects, statistics that generalize across all datapoints. Similarly, this need may be found in analyzing scene graphs, object co-currency, and contextual information [20, 26, 35]. In tandem with our work, [8] 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 ones 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, 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 aggregates to find which combination performed best on a wide array of synthetic datasets distinct from those we trained on selected from prior works [12, 17, 28, 32], 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* [28]. 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, and use its squared cross partial derivatives.

4. Taylor-CAM

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 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.

4.1. Intuition

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:

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

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 [6, 24, 40], which we use to derive our method. We use the term *interaction salience* due to deviation from interaction effects in Definition 3.1.

4.2. 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 $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ are just feature vectors produced by a CNN and each one is associated with an image region. F is the portion of the network downstream of these feature vectors.

Quantify Importance To fill IMP in Equation 2, we turn to class activation maps (CAMs) [40]. However, as observed by the solution of [24], 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 [24]:

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

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 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 3.1:

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

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 Hessian-CAM. Hessian-CAM 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 Taylor-CAM, a higher-order generalization of Grad-CAM, where Grad-CAM (or a close variant) is the special case $\ell = 1$ and Hessian-CAM 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. Lastly, we zero out the diagonals and redundant grid cells of the resulting interaction saliences to only consider interactions between non-redundant feature vectors.

4.3. Limitations

One limitation of Taylor-CAM, 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 [14], 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 pink square?" If the furthest circle is blue, and the second furthest is also blue, then changing the furthest into a square does not meaningfully impact the pink square's contribution to the output, as determined by Grad-CAM, since the answer to the question would be unchanged (blue). Grad-CAM++ [6] may hold an insight as to how to address this, via even-higher order derivatives.

Another limitation is that "change" is measured locally, as derivatives do not account for non-local rates of change. This means that Taylor-CAM, like other deep learning explanatory tools, depends on 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 [4] and libraries like the new Pytorch functional autograd API [19], 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.

5. Experiments

5.1. Statistical Interaction Effects

We evaluate T-NID's ability to rank interactions on the suite of synthetic functions proposed by [12, 17, 28, 32], which were "designed to have a mixture of pairwise and higher-order interactions, with varying order, strength, nonlinearity, and overlap" [32]. These are available to see in the *Appendix* and in Table 1 of [32].

Pairwise Interactions For pairwise interaction effects (see Table 1), we report or reproduce the experiments of [32] 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 [17, 28]. Included for comparison are benchmarks from various statistical and machine learning methods [28, 30, 31, 32, 36]. NID [32] 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. GLIDER [31] is a recent cross-partial method that induces higher-order differentiability with Softplus.

In contrast, T-NID uses only a standard MLP and GELU activations. GELU demonstrably performs better. Unlike NID, we found no 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 and GLIDER appears to struggle with this as well, T-NID recognizes it as an interaction. All around, T-NID performs on par or better than NID and GLIDER at computing pairwise statistical interaction effects on these synthetic tasks.

Higher-Order Interactions 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 [8], we have shown that cross derivatives are a promising metric for interaction attribution in DNNs.

5.2. Object Detection

We ran two qualitative assessments of Taylor-CAM in multi-object detection. In both, the task was to identify whether a pair of objects were present in tandem. We tested the objects "car" and "person" in the COCO annotated-image dataset [15], and we designed our own toy dataset consisting of cars (rectangles), signs (triangles), and a yield sign (red triangle) with labels "go" or "stop." The COCO task suffered from model overfitting and lower test accuracy due to the limited pairwise data, but we still observed sensible explanations. Figure 2a) shows such interactions assigned the highest interaction salience by Taylor-CAM.

In the Yield-or-Go task, Taylor-CAM revealed two prediction strategies. The first is expected: the model interacts the yield sign (red triangle) with a car (rectangle), as seen in Figure 2b), then predicts "stop" accordingly. In the second, the model interacts one car with all of the other cars. One would expect it to relate the car and the yield sign, but the

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

	ANOVA	HierLasso	RuleFit	AG	NID [32]	NID MLP-M [32]	GLIDER [31]	T-NID
$F_1(\mathbf{x})$	0.992	**1.00**	0.754	**1**	0.970	$0.995 \pm 4.4e-3$	0.973 ± 0.01	0.962 ± 0.022
$F_2(\mathbf{x})$	0.468	0.636	0.698	0.88	0.79	$0.85 \pm 3.9e-2$	0.84 ± 0.097	$\mathbf{0.885} \pm 0.039$
$F_3(\mathbf{x})$	0.657	0.556	0.815	**1**	0.999	$\mathbf{1} \pm 0.0$	0.919 ± 0.075	0.999 ± 0.001
$F_4(\mathbf{x})$	0.563	0.634	0.689	**0.999**	0.85	$0.996 \pm 4.7e-3$	0.951 ± 0.073	0.998 ± 0.003
$F_5(\mathbf{x})$	0.544	0.625	0.797	0.67	**1**	$\mathbf{1} \pm 0.0$	0.997 ± 0.008	0.991 ± 0.016
$F_6(\mathbf{x})$	0.780	0.730	0.811	0.64	**0.98**	$0.70 \pm 4.8e-2$	0.767 ± 0.033	0.954 ± 0.026
$F_7(\mathbf{x})$	0.726	0.571	0.666	0.81	0.84	$0.82 \pm 2.2e-2$	0.751 ± 0.207	$\mathbf{0.98} \pm 0.021$
$F_8(\mathbf{x})$	0.929	0.958	0.946	0.937	0.989	$0.989 \pm 4.5e-3$	0.998 ± 0.005	$\mathbf{1.0} \pm 0.0$
$F_9(\mathbf{x})$	0.783	0.681	0.584	0.808	0.83	$0.83 \pm 3.7e-2$	0.754 ± 0.098	$\mathbf{0.98} \pm 0.023$
$F_{10}(\mathbf{x})$	0.765	0.583	0.876	**1**	0.995	$0.99 \pm 2.1e-2$	0.974 ± 0.027	$\mathbf{1.0} \pm 0.0$
Average	0.721	0.698	0.764	0.87	0.92	$0.92 \pm 1.8e-2$	0.892 ± 0.063	$\mathbf{0.975} \pm 0.015$

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

	3-Way Interactions		4-Way Interactions		5-Way Interactions	
	NID [32]	T-NID	NID [32]	T-NID	NID [32]	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

model discovered that the problem can be solved by checking if (1) a car is present, and (2) a red car is not present. Since each object has a different color, (2) implies that a yield sign is present and thus to "stop." Demystifying such reasoning strategies is a unique benefit of Taylor-CAM.

However, when the correct label is "go," *i.e.*, a car and yield sign are not present together, Taylor-CAM finds that the model rarely interacts anything, but rather either all interaction saliences are zero or objects interact with themselves (immediately adjacent regions) (Figure 2c)). This self-interacting is an intuitive and convenient interpretation that Taylor-CAM provides in the lack of salient interactions.

5.3. Relational Reasoning

Sort-Of-CLEVR is a toy dataset for relational reasoning proposed by [23]. It is a less-computationally expensive 2D form of the CLEVR VQA dataset [14] with a focus on relational questions. In our setup, these questions include distance and compare-&-count tasks. To test Taylor-CAM's capacity to explain a neural network's relational reasoning, we train a Relation Network [23] on Sort-Of-CLEVR and visualize its top interactions in Figure 3. Relation Networks are simple modules augmented to CNNs that enable relational reasoning between image regions.

In Figure 3, interacting regions are indicated by two bounding boxes, and the top 4 interactions discovered by Taylor-CAM 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, which we hope Taylor-CAM can be suited to explain. We invite

Table 3: Quantitative analysis on Sort-Of-CLEVR (%)

	Taylor-CAM	Grad-CAM* [24]	GLIDER [31]
Ques 1	**90**%	35%	60%
Ques 2	**55**%	50%	35%
Ques 3	**60**%	40%	45%

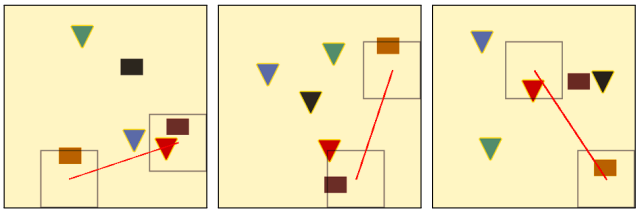
the reader to use the discovered interactions in Figure 3 (as visualized by the bounding boxes) to try to deduce the objects of interest and questions for themselves before looking at the captions. For example, if the top 4 interactions each consist of objects that are close to each other and if each interaction includes the pink square, one might guess that the question is "Which shape is closest to the pink square?"

The 6 objects are "blue", "purple", "pink", "yellow", "orange", and "green" and the 3 questions are (1) "Which shape is closest to the object of interest?", (2) "Which shape is furthest from the object of interest?", and (3) "How many objects have the same shape as the object of interest?"

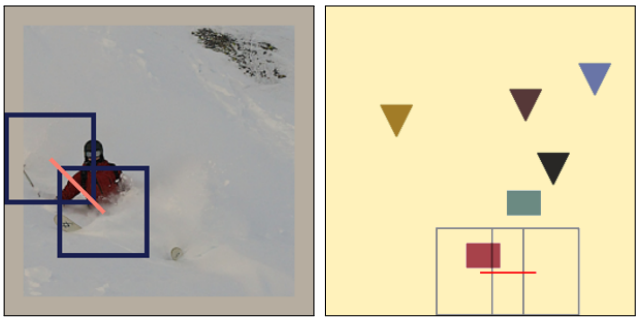
While decisions are frequently relational [3], Grad-CAM is only designed to explain the importance of individual objects in isolation. We observed that Taylor-CAM affords much clearer explanations when decisions are relational.

Quantitative Performance To assess quantitatively, 20 images per question that were classified correctly by the model were randomly selected and annotated with their question's object of interest and answer-relevant objects. For example, for the question, "What is the shape of the object closest to the green square?" the green square and the object that is closest to it are annotated. If Taylor-CAM's top-1 interaction (a pair of bounding boxes) intersects with

a) Objects "person" and "car" are interacted to produce the output classification of whether both are present in the image in tandem.



b) Taylor-CAM interacts the yield sign (red triangle) with any present car (rectangle).

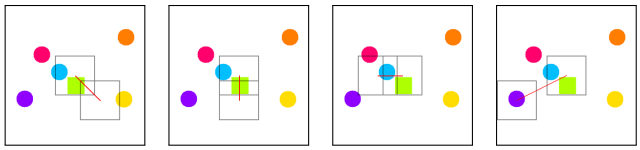


c) When no interactions present, Taylor-CAM's interactions intuitively are 0 or occur primarily between adjacent regions as above.

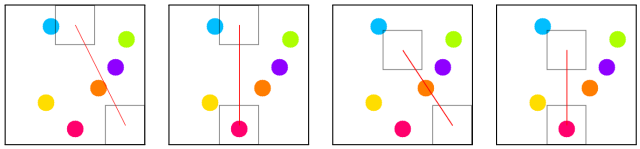
Figure 2: Top-1 bounding boxes generated by Taylor-CAM representing simple interactions in multi-object detection.

the annotated pair, then it is counted as accurate for that image. Same with GLIDER. If Grad-CAM's top-2 saliences include the annotated pair, then it is counted as accurate for that image. Since Grad-CAM does not provide relational interpretations, we refer to this relational interpretation of Grad-CAM's saliences as Grad-CAM*. The bounding boxes in Figure 4 exemplify what a single salience looks like for Taylor-CAM and Grad-CAM respectively. Results of the quantitative analysis are reported in Table 3.

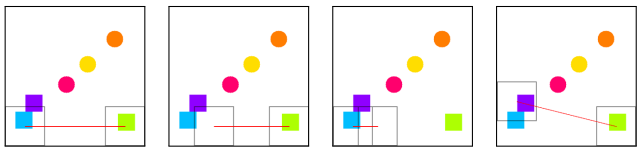
Qualitative Performance To measure Taylor-CAM's qualitative explainability, we selected a random batch of 15 samples and their ordered interaction saliences, and conducted a small user study ($n = 10$), asking each individual to guess (1) the object of interest and (2) the question being asked, from just looking at the top-4 ranked interaction visuals. Taylor-CAM achieves strong explainability with better guess-accuracy than Grad-CAM and the recent GLIDER [31]. With Taylor-CAM, participants were able to reverse



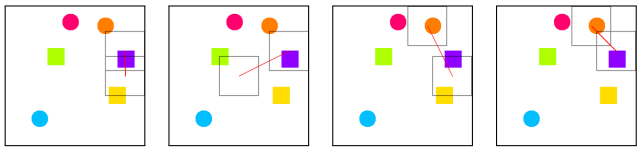
a) Q: "Which shape is closest to the green square?"



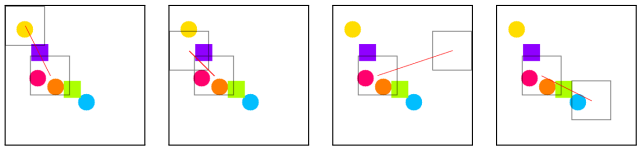
b) Q: "Which shape is furthest from the blue circle?"



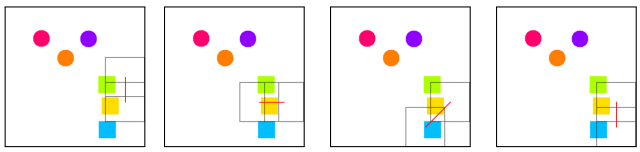
c) Q: "How many objects have shape of green object?"



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



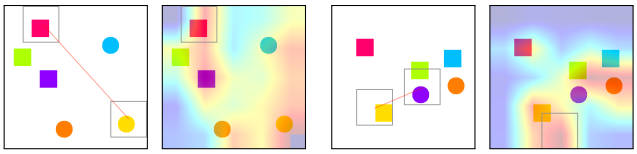
e) Q: "Which shape is furthest from the pink circle?"



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

Figure 3: Shown are the top 4 interactions identified from a Relation Network's predictions on 6 visual question-answering samples. The bounding boxes proposed by Taylor-CAM may be interpreted as indicating a relation. We recommend testing yourself to see if you can guess (1) the object of interest and (2) the question being asked (*closest*, *furthest*, or *same shape*), without looking at the caption.

engineer questions in relational VQA from just looking at the visualized interactions. We report a wide range of explainability across different colors and questions in Tables 4 and 5. Due to random sampling, none of the 15 sampled images for Grad-CAM included a purple object of interest, so it is marked "N/A" in Table 4.



a) Q: *"Which shape is furthest from the pink square?"* b) Q: *"Which shape is closest to the yellow square?"*

Figure 4: The bounding boxes show what a top-1 salience looks like for Taylor-CAM (on the left) and Grad-CAM (on the right) respectively. Taylor-CAM offers interpretable relational explanations from a single top-1 salience, whereas Grad-CAM depends on all saliences to produce a non-relational heatmap.

Table 4: User study **object of interest** accuracy (%)

	Grad-CAM [24]	GLIDER [31]	Taylor-CAM
Green	13.3%	33.3%	**40%**
Pink	30%	10%	**46.7%**
Blue	10%	22.2%	**40%**
Purple	N/A	**15%**	10%
Orange	3.3%	10%	**15%**
Yellow	25%	16.7%	**33.3%**

Table 5: User study **question** accuracy (%)

	Grad-CAM [24]	GLIDER [31]	Taylor-CAM
Ques 1	44%	38.9%	**76%**
Ques 2	14%	38.9%	**55%**
Ques 3	30%	23.8%	**48.3%**

While some Grad-CAM colors strongly outperform random guessing (pink and yellow), on average, people struggled guessing the object of interest with Grad-CAM. This is because Grad-CAM only explains which individual objects contribute to the output, which in relational VQA, is all of them with an equal importance assigned to the object of interest and any objects that are included in the question-answer, such as the furthest or nearest object. This results in uninterpretable and sometimes misleading visualizations, making it very hard to guess an object of interest from the visual only. Without knowing the object of interest, it is consequently much harder to guess the question asked.

Grad-CAM, GLIDER, and Taylor-CAM all did relatively well on question 1. Closeness is easier to interpret with all three explanatory tools, since it is usually more visually apparent. However, we found question 2 (furthest distance) to be harder to interpret for Grad-CAM, perhaps because it is unclear what the object of interest is, with multiple "far away" objects of different relative proximity being ranked highly. For example, two objects that are far away from the object of interest might be close to each other, creating the false impression that the question is asking about closeness. Thus, without confidence regarding the object of interest and the interacting parts, we found ranked importances alone to be unintuitive and even misleading.

5.4. Biomedical Application

We also applied T-NID to determine interactions in the PPMI study dataset (www.ppmi-info.org). Our analysis suggests that various measures previously thought to be unrelated should be considered together when predicting faster cognitive progression in Parkinson's disease. Please see Appendix for details in this domain.

6. Architecture Configurations

Please see Appendix.

7. Conclusion

With T-NID and Taylor-CAM, 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, Taylor-CAM generalizes Grad-CAM to the higher order and effectively explains interactions in object detection and relational reasoning, affording a user cohort the insight to guess questions in VQA from only seeing the top discovered visual interactions. Future work may explore localizing multi-modal interactions such as in audio-visual tasks, an agent's interactions in RL and robotics, and interactions between word embeddings in NLP. 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.

8. Acknowledgments

This work has been partially supported by the National Science Foundation (NSF) under Grants 1741472, 1813709, 1909912, and 1934962 and the National Institutes of Health (NIH - NINDS) under Grant P50NS108676. The article solely reflects the opinions and conclusions of its authors but not the funding agents.

References

[1] Chunrong Ai and Edward C Norton. Interaction terms in logit and probit models. *Economics letters*, 80(1):123–129, 2003. 2

[2] Hugues Aschard. A perspective on interaction effects in genetic association studies. *Genetic epidemiology*, 40(8):678–688, 2016. 2

[3] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 2, 6

[4] Jesse Bettencourt, Matthew J. Johnson, and David Duvenaud. Taylor-mode automatic differentiation for higher-order derivatives in JAX. In *Advances in neural information processing systems, Workshop Program Transformations*, 2019. 5

[5] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 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*, pages 1721–1730, 2015. 2

[6] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018. 1, 2, 4, 5

[7] Gary K Chen and Duncan C Thomas. Using biological knowledge to discover higher order interactions in genetic association studies. *Genetic epidemiology*, 34(8):863–878, 2010. 2

[8] Tianyu Cui, Pekka Marttinen, and Samuel Kaski. Recovering pairwise interactions using neural networks. In *Advances in neural information processing systems, Bayesian Deep Learning workshop*, 2019. 2, 3, 5

[9] Oliver Eberle, Jochen Büttner, Florian Kräutli, Klaus-Robert Müller, Matteo Valleriani, and Grégoire Montavon. Building and interpreting deep similarity models. *arXiv preprint arXiv:2003.05431*, 2020. 2

[10] Yotam Hechtlinger. Interpretation of prediction models using the input gradient. *ArXiv*, abs/1611.07634, 2016. 1, 2

[11] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. 3

[12] Giles Hooker. Discovering additive structure in black box functions. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, page 575. ACM Press. 3, 5

[13] Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: Axiomatic feature interactions for deep networks. *arXiv preprint arXiv:2002.04138*, 2020. 2

[14] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 5, 6

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[16] Ge Liu, Haoyang Zeng, and David K Gifford. Visualizing complex feature interactions and feature sharing in genomic deep neural networks. *BMC bioinformatics*, 20(1):1–14, 2019. 2

[17] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, page 623. ACM Press. 3, 5

[18] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017. 2

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *Advances in neural information processing systems*, 2017. 5

[20] Amir Rosenfeld, Richard S. Zemel, and John K. Tsotsos. The elephant in the room. *CoRR*, abs/1808.03305, 2018. 3

[21] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 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, pages 2662–2670. AAAI Press. 1, 2

[22] Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. Relational recurrent neural networks. In *Advances in neural information processing systems*, pages 7299–7310, 2018. 2

[23] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017. 2, 6

[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. 1, 2, 4, 6, 8

[25] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1

[26] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: Learning to overcome contextual bias. *CoRR*, abs/2001.03152, 2020. 3

[27] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1161–1170, 2019. 2

[28] Daria Sorokina, Rich Caruana, Mirek Riedewald, and Daniel Fink. Detecting statistical interactions with additive groves of trees. In *Proceedings of the 25th international conference*

on Machine learning - ICML '08, pages 1000–1007. ACM Press. 3, 5

[29] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017. 1, 2

[30] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011. 5

[31] Michael Tsang, Dehua Cheng, Hanpeng Liu, Xue Feng, Eric Zhou, and Yan Liu. Feature interaction interpretability: A case for explaining ad-recommendation systems via neural interaction detection. In *International Conference on Learning Representations*, 2020. 2, 5, 6, 7, 8

[32] Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. In *International Conference on Learning Representations*, 2018. 2, 3, 5, 6

[33] Michael Tsang, Sirisha Rambhatla, and Yan Liu. How does this interaction affect me? interpretable attribution for feature interactions. *arXiv preprint arXiv:2006.10965*, 2020. 2

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[35] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Adversarial removal of gender from deep image representations. *CoRR*, abs/1811.08489, 2018. 3

[36] T.H. Wonnacott and R.J. Wonnacott. *Introductory statistics*. Wiley series in probability and mathematical statistics. Wiley, 1977. 2, 5

[37] Nengjun Yi. Statistical analysis of genetic interactions. *Genetics research*, 92(5-6):443–459, 2010. 2

[38] Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew Botvinick, Oriol Vinyals, and Peter Battaglia. Deep reinforcement learning with relational inductive biases. In *International Conference on Learning Representations*, 2019. 2

[39] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1

[40] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 1, 2, 4