Influence-Directed Explanations for Deep Convolutional Networks

Klas Leino
Shayak Sen
Anupam Datta
and Matt Fredrikson
Carnegie Mellon University

Linyi Li
Tsinghua University

***Abstract*—We study the problem of explaining a rich class of behavioral properties of deep neural networks. Distinctively, our *influence-directed explanations* approach this problem by peering inside the network to identify neurons with high *influence* on a quantity and distribution of interest, using an axiomatically-justified influence measure, and then providing an *interpretation* for the concepts these neurons represent. We evaluate our approach by demonstrating a number of its unique capabilities on convolutional neural networks trained on ImageNet. Our evaluation demonstrates that influence-directed explanations *(1)* identify influential concepts that generalize across instances, *(2)* can be used to extract the "essence" of what the network learned about a class, and *(3)* isolate individual features the network uses to make decisions and distinguish related classes.**

I. Introduction

We study the problem of explaining a class of behavioral properties of deep neural networks, with a focus on convolutional neural networks. This problem has received significant attention in recent years with the rise of deep networks and associated concerns about their opacity [1].

A growing body of work on explaining deep convolutional network behavior is based on mapping models' prediction outputs back to relevant regions in an input image. This is accomplished in various ways, such as by visualizing gradients [2], [3], [4], or by backpropagation [5], [6], [4]. An appealing feature of these approaches is that they capture *input influence*. However, because these approaches relate instance-specific features to instance-specific predictions, the explanations that they produce do not generalize beyond a single test point (see Section III-A, Figure 2).

An orthogonal approach is to visualize the features learned by networks by identifying input instances that maximally activate an internal neuron, by either optimizing the activation in the input space [2], [7], [8], or searching for instances in a dataset [9]. Importantly, this type of explanation gives insight into the higher-level concepts learnt by the network, and naturally generalizes across instances and classes. However, this approach does not relate these higher-level concepts to the predictions that they cause. Indeed, examining activations alone is not sufficient to do so (see Section III-B).

This paper introduces *influence-directed explanations* for deep networks to combine the positive attributes of these two lines of work. Our approach peers inside the network to identify neurons with high *influence* on the model's behavior, and then uses existing techniques (e.g., visualization) to provide an *interpretation* for the concepts they represent. We introduce a novel *distributional influence* measure that allows us to identify which neurons are most influential in determining the model's behavior on a given distribution of instances. From this we are able to identify the learned concepts that cause the network to behave characteristically, for example, on the distribution of instances that share a particular label.

(a) (b)

Fig. 1: **(a)** Images of cars labeled 'sports car' by the VGG16 network, and **(b)** receptive fields of the most influential feature map on a comparative quantity that characterizes the model's tendency to predict 'sports car' over 'convertible.'

Figure 1 demonstrates the capability of influence-directed explanations to extract meaningful insight about the network's inner workings. We measure the influence of feature maps at the `conv4_1` layer on the network's tendency to predict 'sports car' over 'convertible.' The images in Figure 1b are computed by rendering the receptive field of the *most* influential map in the original feature space for the corresponding image in Figure 1a. The results coincide with an intuitive understanding of the distinction between these classes: in most instances, the depicted interpretation highlights the portion of the image depicting the car's top.

Our empirical evaluation demonstrates that influence-directed explanations *(1)* extract influential concepts that generalize across instances, whereas those computed using input influence fail to do so (Section III-A), *(2)* reveal the

"essence" of how the network views a class and distinguishes it from others (Section III-B), and *(3)* isolate high-level features that the network uses to make predictions (Section IV-A, IV-B). In each case, our influence-directed explanations leverage the ability to measure internal influence to produce useful explanations that would not have been possible otherwise.

II. Influence

In this section, we propose *distributional influence*, an axiomatically-justified family of influence measures. Distributional influence is parameterized by a *slice* s of the network (e.g. a particular layer), a *quantity of interest* f and a *distribution of interest* P. Given these elements, we measure influence as the partial derivative of f at the slice s averaged over P. We describe the measure and its parameters in more detail below. In Section V, we justify this family of measures by proving that these are the only measures that satisfy some natural properties.

The slice parameter exposes the internals of a network, allowing us to measure influence with respect to intermediate neurons. This is a significant departure from prior work, and is key to our goal of identifying high-level concepts that are learned by a network. As we show in Section III, influence measurements on internal units lead to explanations that generalize across instances. This is usually not possible by measuring input features (i.e., pixels), as learned concepts can manifest themselves in many different ways in the input space, with a high degree of variance among the influence of particular input features across instances.

The distribution and quantity of interest together capture aspects of network behavior that we are interested in explaining. Examples of distributions of interest are: *(1)* a single instance (i.e., the influence measure just reduces to the gradient at the point); *(2)* the distribution of 'cat' images, or *(3)* the distribution of all images in a dataset. While the first distribution of interest focuses on why a single instance was classified a particular way, the second explains the "essence" of a class, and the third identifies generally-influential neurons over the entire population. Another example is the uniform distribution on the line segment of scaled instances between an instance and a baseline, which yields a measure similar to one called Integrated Gradients [3].

Whereas the distribution of interest identifies the subjects of an explanation, the quantity of interest identifies the question that is being addressed. For example, the quantity of interest may correspond to the network's outcome for the 'cat' class, or its *comparative* outcome towards 'cat' versus 'dog' (i.e., the difference in the network scores for cat and dog classes). The first quantity addresses the question of why a particular input is classified as 'cat', whereas the second addresses how the network distinguishes 'cat' instances from 'dog' instances.

We represent quantities of interest of networks as continuous and differentiable functions $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subseteq \mathbb{R}^n$ and n is the number of inputs to f. A distributional influence measure, denoted by $\chi_i(f, P)$, measures the influence of input i for a quantity of interest f, and a distribution of interest P where P is a distribution over $\mathcal{X}$.

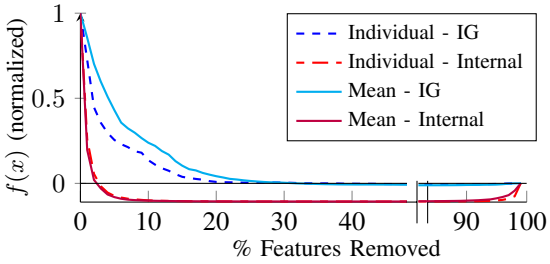


Fig. 2: Plot of the decrease in the function value, $f(x)$, as features are removed from the input (**IG**) or first fully-connected layer (**Int**) in order of influence, using the VGG16 [10] network. The vertical axis was normalized so that the average value of $f(x)$ is 1, and the average value of $f(0)$ is 0. The dashed curves depict the average quantity when influence is measured for each instance individually, and the solid curves when the mean influence over the respective class is used for each instance. Plots are averaged across 5 randomly-selected ImageNet classes.

Next, we define a slice of a network. A particular layer in the network can be viewed as a slice. More generally, a slice is any partitioning of the network into two parts that exposes its internals. Formally, a slice s of a network f is a tuple of functions $\langle g, h \rangle$ such that $h : \mathcal{X} \to \mathcal{Z}$, $g : \mathcal{Z} \to \mathbb{R}$ and $f = g \circ h$. The internal representation for an instance $\mathbf{x}$ is given by $\mathbf{z} = h(\mathbf{x})$. In our setting, elements of $\mathbf{z}$ can be viewed as the activations of neurons at a particular layer.

Definition 1. *The influence of an element j in the internal representation defined by $s = \langle g, h \rangle$ is*

$$\chi_j^s(f, P) = \int_{\mathcal{X}} \frac{\partial g}{\partial z_j}\bigg|_{h(\mathbf{x})} P(\mathbf{x}) d\mathbf{x} \tag{1}$$

III. Identifying Influential Concepts

The influence measure defined in Section II is parameterized by a distribution of interest P (Equation 1) over which the measure is taken. By selecting P to be a point mass, the resulting measurements characterize the importance of features for the models behavior on a single instance. Defining the distribution of interest with support over a larger set of instances yields explanations that capture the factors common to network behaviors across the corresponding population of instances. In this section, we demonstrate that when taken at a high internal layer, distributional influence identifies concepts that generalize well across instances. Furthermore, we show this measure often lets us identify a relatively small set of concepts that characterize the "essence" of the class, and are sufficient for distinguishing instances of that class from others.

A. Effectiveness of Internal Influence

One of our central claims is that the ability to measure internal influence across an appropriately chosen distribution lets us identify learned concepts that are relevant to classification predictions. Figure 2 quantifies the degree to which internal units identified using internal influence measurements correspond

to relevant general concepts, compared against the influence measurements obtained using integrated gradients (IG) [3]. The curves report the network's output at the coordinate of the predicted class, normalized to begin at 1, as input features (IG) or internal units at the lowest fully-connected layer are "turned off" in decreasing order of influence. We adapted this approach from Samek et al. [11] for internal units by setting their activation to 0. The vertical axis depicts the dropoff of the networks output against the percentage of features that have been removed.

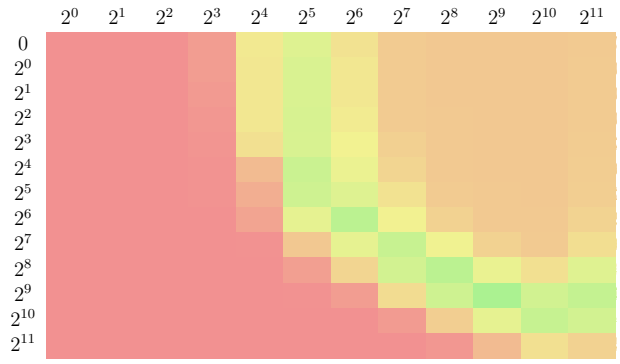
We evaluated this measure on instances of five randomly-selected ImageNet classes on VGG16 [10], and display the averaged results. We selected integrated gradients as our point of comparison because we found that it outperformed comparable methods discussed in the related work. Influence is calculated in two ways to characterize the difference between instance-specific and general measurements. In the cases labeled "Individual", we measure influence for each instance individually and rank features and units accordingly, whereas those labeled "Mean" rank features and units by influence measured over the distribution of instances in the appropriate class.

Comparing the individual and mean results tells us how well the components identified as relevant by the influence measurements generalize across the class. If the individual cases significantly outperform their respective mean cases, then we might conclude that the distributional influences failed to identify concepts that are relevant across the class. The results in Figure 2 show a very small gap in performance between the individual and mean cases for internal influence, but, unsurprisingly, this was not the case for input influence. This suggests that units deemed relevant to the class on-average also tend to contribute consistently across instances in that class. Moreover, the steeper dropoff for the internal influence measurements indicates that the identified units correspond to highly-relevant concepts with a greater degree of class-specificity.

B. Validating the "Essence" of a Class

The steep dropoff for internal influences in Figure 2 suggests that it is often the case that relatively few units are highly influential towards a particular class. Combined with the fact that these units tend to be relevant *across* the class suggests the existence of a consistent, relatively small set of units that are sufficient to predict and explain the class. We refer to this set as the "essence" of the class, and validate our hypothesis by isolating these units from the rest of the model to extract a binary classifier for membership in the corresponding class.

We show that these classifiers, which we call *experts*, are often more proficient than the original model at distinguishing instances of the class from other classes in the distribution, despite comprising fewer units than the original model. Furthermore, the performance of the original model can be achieved by experts using as few as 1% of the available internal units. Finally, we show that experts derived by using activation levels rather than influence measurements to identify the "essence" are not as effective for a fixed number of units, demonstrating that explanations based on activations are not as effective at identifying and isolating learned concepts.



(a)

Class	Orig.	Infl.	Act.
Chainsaw (491)	.14	.71	.21
Bonnet (452)	.62	.92	.77
Park Bench (703)	.52	.71	.63
Sloth Bear (297)	.36	.75	.44
Pelican (144)	.65	.95	.79

(b)

Fig. 3: **(a)** F_1 score for experts derived from the first fully-connected layer of the VGG16 network on a randomly-selected ImageNet class. The rows and columns correspond to β and α respectively. The layer contains 4096 neurons, so the bottom right corner corresponds to the entire network. High F_1 scores are shown in green, and low scores in red. **(b)** Model compression recall for five randomly-selected ImageNet classes. Columns marked Orig. correspond to the original model, Infl. to experts computed using influence measures, and Act. to experts computed using activation levels. Precision in all cases was 1.0.

1) Class-specific experts: Given a model f with softmax output, and slice $\langle g, h\rangle$ where $g : \mathcal{Z} \to \mathcal{Y}$, let $M_h \in \mathcal{Z}$ be a 0-1 vector. Intuitively, M_h masks the set of units at layer h that we wish to retain, and thus is 1 at all locations corresponding to such units and 0 everywhere else. Then the *slice compression* $f_{M_h}(X) = g(h(X) * M_h)$ corresponds to the original model after discarding all units at h not selected by M_h. Given a model f, we obtain a binary classifier, f^i, for class L_i (corresponding to output i) by taking the argmax over outputs and combining all classes $j \neq i$ into one class, $\neg i$; i.e., f^i predicts i when f predicts i, and $\neg i$ when f predicts $j \neq i$.

A class-wise *expert* for L_i is a slice compression f_{M_h} whose corresponding binary classifier $f^i_{M_h}$ achieves better recall on L_i than the binary classifier f^i, while achieving comparable or better precision. To derive an expert, we compute M_h by measuring the slice influence (Equation 1) over P_i using the quantity of interest $g|_i$. We then select α units at layer h with the greatest positive influence, and β units with the lowest negative influence (i.e., greatest magnitude among those with negative influence). M_h is then defined to be zero at all positions except those corresponding to these $\alpha + \beta$ units. In our experiments, we obtain concrete values for

α and β by a parameter sweep, ultimately selecting parameter values that yield the best experts by recall rate.

Figure 3a shows the F_1 score obtained on a randomly-selected class as a function of α and β. Figure 3b shows the recall of experts found in this way for five randomly selected ImageNet classes. Notably, the α and β yielding the best performance correspond to less than a quarter of the units available, and the resulting expert achieves significantly better performance than the original model. Additionally, the performance of the original model can be matched using a tiny fraction of the available neurons (as few as 1%), supporting the claim that the network's behavior on the class can be effectively summarized by identifying a small number of the most influential units for that class.

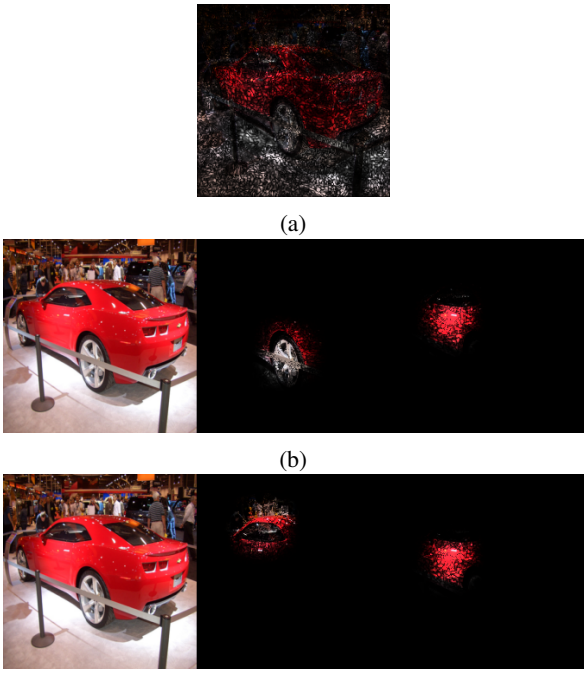
2) Inadequacy of activation levels: Some recent prior work [12], [5] uses unit activation levels to determine relevance when identifying concepts. Here we consider an alternative approach for deriving experts by measuring the average activation at h across the distribution of interest to compute M_h, and ranking units by average activation level. Figure 3b shows the best recall of the resulting activation-based experts, and we see that activations are considerably less effective than influences for finding good experts. Moreover, experts derived from activations are unable to match the original model performance without using *at least half* of the available units, and those with small α, β (close to 1%) achieve zero recall in every case we evaluated. This appeals to the intuition that a unit may be highly active on an instance without necessarily contributing to the prediction outcome, and suggests that activation levels are not a consistent proxy for the relevance of a neuron.

IV. Explaining Instances

In this section we demonstrate that our influence measure is also useful when explaining model behavior by instantiating general information measured across a distribution of interest to an *individual* instance. We begin by noting that our measure generalizes previous gradient-based influence measures, so it can be parameterized to produce the same sorts of explanations shown in prior work. We then introduce two parameterizations that yield new sorts of explanations, showing the broader potential for our work in practical settings. In particular, we show that *(1)* internal influence can be leveraged to gain a more complete understanding of a model's decision on an instance by breaking the influential features into high-level components recognized by the model, and *(2)* changing the quantity of interest yields explanations specific to how the model distinguishes between related classes on specific instances.

A. Focused Explanations from Slices

Slice influence (Equation 1) characterizes the extent to which neurons in an intermediate layer are relevant to a particular network behavior. We can construct explanations by using existing visualization techniques [3] to interpret the concepts represented by internal units that are distinguished by high slice influence on an appropriate quantity of interest.



(a)

(b)

(c)

Fig. 4: **(a)** Input influence in the VGG16 [10] network parameterized to match saliency maps [2]. **(b)** Interpretation of the two most influential units from the convolutional layer `conv4_1`. **(c)** Comparative explanation visualizing the top two units at the `conv4_1` layer that distinguish 'sports car' from 'convertible.'

These explanations allow us to decompose the influential input features into high-level concepts recognized by the model.

Figure 4b shows the results of interpreting the influences taken on a slice of the VGG16 [10] network corresponding to an intermediate convolutional layer (`conv4_1`). In this example we visualize the two most influential units for the quantity of interest characterizing correct classification of the image shown on the left of Figure 4b (sports car). More precisely, the quantity of interest used in this example is $f|_L$, i.e., the projection of the model's softmax output to the coordinate corresponding to the correct label L of this instance. The interpretation for each of these units was then obtained by measuring the influence of the input pixels on these units along each color channel, and scaling the pixels in the original image accordingly [3].

Because convolutional units have a limited receptive field, the resulting interpretation shows distinct regions in the original image, in this case corresponding to the wheel and side of the car, that were most relevant to the model's predicted classification. When compared to the explanation provided by input influence, e.g., as shown in Figure 4a, it is evident that the explanation based on the network's internal units more effectively localizes the features used by the network in its prediction.

B. Comparative Explanations

Influence-directed explanations are parameterized by a quantity of interest, corresponding to the function f in

Equation 1. Changing the quantity of interest gives additional flexibility in the characteristic explained by the influence measurements and interpretation. One class of quantities that is particularly useful in answering counterfactual questions such as, "Why was this instance classified as L_1 rather than L_2?", is given by the *comparative quantity*. Namely, if f is a softmax classification model that predicts classes $L_1, \ldots, L_n$, then the comparative quantity of interest between classes L_i and L_j is $f|_i - f|_j$. When used in Equation 1, this quantity captures the tendency of the model to classify instances as L_i over L_j.

Figure 4c shows an example of a comparative explanation. The original instance shown on the left of Figure 4c is labeled as 'sports car.' We measured influence using a comparative quantity against the leaf class 'convertible,' using a slice at the `conv4_1` convolutional layer. The interpretation was computed on the top two most influential units at this layer in the same way as discussed in Section IV-A.

As in the examples from Figure 1, the receptive field of the most influential unit corresponds to the region containing the hard top of the vehicle, which is understood to be its most distinctive feature according to this comparative quantity. While both the explanations from Figure 4b and Figure 4c capture features common to cars, only the comparative explanation isolates the elements of the feature space distinctive to the type of car.

V. Axiomatic Justification of Measures

In this section we justify the family of measures presented in Section II by defining a set of natural axioms for influence measures in this setting, and then proving a tight characterization. We first address the case where the influence is measured with respect to inputs, i.e. when a slice is f paired with the identity function, and then generalize to internal layers. This approach is inspired, in part, by axiomatic justification for power indices in cooperative game theory [13], [14]–an approach that has been previously employed for explaining predictions of machine learning models [15], [3]. An important difference, as we elaborate below, is that we carefully account for distributional faithfulness in this work.

A. Input Influence

We first characterize a measure $\chi_i(f, P)$ that measures the influence of input i for a quantity of interest f, and distribution of interest P. The first axiom, *linear agreement* states that for linear systems, the coefficient of an input is its influence. Measuring influence in linear models is straightforward since a unit change in an input corresponds to a change in the output given by the coefficient.

Axiom 1 (Linear Agreement)**.** *For linear models of the form* $f(\mathbf{x}) = \sum_i \alpha_i x_i$, $\chi_i(f, P) = \alpha_i$.

The second axiom, *distributional marginality* is inspired by the *marginality principle* [16] in prior work on cooperative game theory. The marginality principle states that an input's importance only depends on its own contribution to the output. Formally, if the partial derivatives with respect to an input of two functions are identical at all input instances, then that input is equally important for both functions.

Our axiom of distributional marginality (DM) is a weaker form of this requirement that only requires equality of importance when partial derivatives are same for points in the support of the distribution. This axiom ensures that the influence measure only depends on the behavior of the model on points within the manifold containing the input distribution. Such a property is important for deep learning systems since the behavior of the model outside of this manifold is unpredictable.

Axiom 2 (Distributional marginality (DM))**.** *If*

$$P\left(\left.\frac{\partial f_1}{\partial x_i}\right|_X = \left.\frac{\partial f_2}{\partial x_i}\right|_X\right) = 1,$$

where X is the random variable over instances from $\mathcal{X}$, then $\chi_i(f_1, P) = \chi_i(f_2, P)$.

The third axiom, *distribution linearity* states that the influence measure is linear in the distribution of interest. This ensures that influence measures are properly weighted over the input space, i.e., influence on infrequent regions of the input space receive lesser weight in the influence measure as compared to more frequent regions.

Axiom 3 (Distribution linearity (DL))**.** *For a family of distributions indexed by some $a \in \mathcal{A}$, $P(x) = \int_{\mathcal{A}} g(a) P_a(x) da$, then $\chi_i(f, P) = \int_{\mathcal{A}} g(a) \chi_i(f, P_a) da$.*

Theorem 1. *The only measure that satisfies linear agreement, distributional marginality and distribution linearity is given by*

$$\chi_i(f, P) = \int_{\mathcal{X}} \left.\frac{\partial f}{\partial x_i}\right|_{\mathbf{x}} P(\mathbf{x}) d\mathbf{x}.$$

Proof. Choose any function f and $P_{\mathbf{a}}(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{a})$, where δ is the Dirac delta function on $\mathcal{X}$. Now, choose $f'(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}_i}|_{\mathbf{a}} x_i$. By linearity agreement, it must be the case that, $\chi(f', P_{\mathbf{a}}(\mathbf{x})) = \frac{\partial f}{\partial x_i}|_{\mathbf{a}}$. By distributional marginality, we therefore have that $\chi_i(f, P_{\mathbf{a}}) = \chi_i(f', P_{\mathbf{a}}) = \frac{\partial f}{\partial x_i}|_a$. Any distribution P can be written as $P(\mathbf{x}) = \int_{\mathcal{X}} P(\mathbf{a}) P_{\mathbf{a}}(\mathbf{x}) d\mathbf{a}$. Therefore, by the distribution linearity axiom, we have that $\chi(f, P) = \int_X P(\mathbf{a}) \chi(f, P_a) da = \int_{\mathcal{X}} P(\mathbf{a}) \frac{\partial f}{\partial x_i}|_{\mathbf{a}} d\mathbf{a}$. □

B. Internal influence

In this section, we generalize the above measure of input influence to a measure that can be used to measure the influence of an internal neuron. We again take an axiomatic approach, with two natural invariance properties on the structure of the network.

The first axiom states that the influence measure is agnostic to how a network is sliced, as long as the neuron with respect to which influence is measured is unchanged. Below, the notation $\mathbf{x}_{-i}$ refers to the vector $\mathbf{x}$ with element i removed and $\mathbf{x}_{-i} y_i$ is the vector $\mathbf{x}$ with the i^{th} element replaced with y_i.

Two slices, $s_1 = \langle g_1, h_1 \rangle$ and $s_2 = \langle g_2, h_2 \rangle$, are j-equivalent if for all $\mathbf{x} \in \mathcal{X}$, and $z \in \mathcal{Z}$, $h_1(\mathbf{x})_j = h_2(\mathbf{x})_j$, and $g_1(h_1(\mathbf{x})_{-j} z_j) = g_2(h_2(\mathbf{x})_{-j} z_j)$. Informally, two slices are j-equivalent as long as they have the same function for

representing z_j, and the causal dependence of the outcome on z is identical.

Axiom 4 (Slice Invariance)**.** *For all j-equivalent slices s_1 and s_2, $\chi_j^{s_1}(f,P) = \chi_j^{s_2}(f,P)$.*

The second axiom equates the input influence of an input with the internal influence of a perfect predictor of that input. Essentially, this encodes a consistency requirement between inputs and internal neurons that if an internal neuron has exactly the same behavior as an input, then the internal neuron should have the same influence as the input.

Axiom 5 (Preprocessing)**.** *Consider h_i such that $P(X_i = h_i(X_{-i})) = 1$. Let $s = \langle f_1, h\rangle$, be such that $h(\mathbf{x}_{-i}) = \mathbf{x}_{-i}h_i(\mathbf{x}_{-i})$, which is a slice of $f_2(\mathbf{x}_{-i}) = f_1(\mathbf{x}_{-i}h_i(\mathbf{x}_{-i}))$, then $\chi_i(f_1,P) = \chi_i^s(f_2,P)$.*

We now show that the only measure that satisfies these two properties is the one presented above in Equation 1. First, we prove the following lemma that shows that expected gradient computed at a slice can be computed with either the probability distribution at the input or the slice.

Lemma 1. *Let $s = \langle g, h\rangle$ be a slice for f. Given distribution $P_{\mathcal{X}}(\mathbf{x})$ on $\mathcal{X}$, let $P_{\mathcal{Z}}(\mathbf{z})$ be the probability distribution induced by applying h on $\mathbf{x}$, given by:*

$$P_{\mathcal{Z}}(\mathbf{z}) = \int_{\mathcal{X}} P_{\mathcal{X}}(\mathbf{x})\delta(h(\mathbf{x}) - \mathbf{z})d\mathbf{x}.$$

Then $\chi_j(g, P_{\mathcal{Z}}) = \chi_j^s(f, P_{\mathcal{X}})$.

Proof.

$$\chi_j(g,P_{\mathcal{Z}}) = \int_{\mathcal{Z}} \left.\frac{\partial g}{\partial z_j}\right|_{\mathbf{z}} P_{\mathcal{Z}}(\mathbf{z})d\mathbf{z} \tag{2}$$

$$= \int_{\mathcal{Z}} \left.\frac{\partial g}{\partial z_j}\right|_{\mathbf{z}} \int_{\mathcal{X}} P_{\mathcal{X}}(\mathbf{x})\delta(h(\mathbf{x}) - \mathbf{z})d\mathbf{x}d\mathbf{z} \tag{3}$$

$$= \int_{\mathcal{X}} P_{\mathcal{X}}(\mathbf{x}) \int_{\mathcal{Z}} \left.\frac{\partial g}{\partial z_j}\right|_{\mathbf{z}} \delta(h(\mathbf{x}) - \mathbf{z})d\mathbf{z}d\mathbf{x} \tag{4}$$

$$= \int_{\mathcal{X}} \left.\frac{\partial g}{\partial z_j}\right|_{h(\mathbf{x})} P_{\mathcal{X}}(\mathbf{x})d\mathbf{x} \tag{5}$$

$$= \chi_j^s(f, P_{\mathcal{X}}) \tag{6}$$

□

Theorem 2. *The only measure that satisfies slice invariance and preprocessing is Equation 1.*

Proof. Assume that two slices $s_1 = \langle g_1, h_1\rangle$ and $s_2 = \langle g_2, h_2\rangle$ are j-equivalent. Therefore, $g_1(h_1(\mathbf{x})_{-j}z_j) = g_2(h_2(\mathbf{x})_{-j}z_j)$. Taking partial derivatives with respect to z_j, we have that:

$$\left.\frac{\partial g_1}{\partial z_j}\right|_{h_1(\mathbf{x})_{-j}z_j} = \left.\frac{\partial g_2}{\partial z_j}\right|_{h_2(\mathbf{x})_{-j}z_j}$$

Now, since $h_1(\mathbf{x})_j = h_2(\mathbf{x})_j$, we have that

$$\left.\frac{\partial g_1}{\partial z_j}\right|_{h_1(\mathbf{x})} = \left.\frac{\partial g_2}{\partial z_j}\right|_{h_2(\mathbf{x})}$$

Plugging the derivatives into 1, we get that $\chi_j^{s_1}(f,P) = \chi_j^{s_2}(f,P)$, and that the measure satisfies slice invariance.

Consider h_i such that $P(X_i = h_i(X_{-i})) = 1$. Let $s = \langle f_1, h\rangle$, be such that $h(\mathbf{x}_{-i}) = \mathbf{x}_{-i}h_i(\mathbf{x}_{-i})$, which is a slice of $f_2(\mathbf{x}_{-i}) = f_1(\mathbf{x}_{-i}h_i(\mathbf{x}_{-i}))$.

$$\chi_i^s(f_2,P) = \int_{\mathcal{X}} \left.\frac{\partial f_1}{\partial x_i}\right|_{\mathbf{x}_{-i}h(\mathbf{x}_{-i})} P(\mathbf{x})d\mathbf{x} \tag{7}$$

$$= \int_{\mathcal{X}} \left.\frac{\partial f_1}{\partial x_i}\right|_{\mathbf{x}} P(\mathbf{x})d\mathbf{x} \tag{8}$$

$$= \chi_i(f_1,P). \tag{9}$$

Therefore, the measure satisfies preprocessing.

For the opposite direction, consider any slice $s = \langle g, h\rangle$ of f. We wish to show that if $\chi_j^s(f, P_{\mathcal{X}})$ satisfies slice invariance and preprocessing, then $\chi_j^s(f, P_{\mathcal{X}}) = \int_{\mathcal{X}} \left.\frac{\partial g}{\partial z_j}\right|_{h(\mathbf{x})} P(\mathbf{x})d\mathbf{x}$. Consider the slice $s' = \langle g', h'\rangle$ such that $h'(\mathbf{x}) = (\mathbf{x}, h_j(\mathbf{x}))$, and $g'(\mathbf{x}, z_j) = g(h_{-j}(\mathbf{x})z_j)$. Essentially s' is a slice of f that only processes $h_j(\mathbf{x})$. By Lemma 1, $\chi_j(g', P_{\mathcal{X}}) = \int_{\mathcal{X}} \left.\frac{\partial g}{\partial z_j}\right|_{h(\mathbf{x})} P(\mathbf{x})d\mathbf{x}$. By preprocessing $\chi_j^{s'}(f, P_{\mathcal{X}}) = \chi_j(g', P_{\mathcal{X}})$. As s and s' are j-equivalent, $\chi_j^s(f, P_{\mathcal{X}}) = \chi_j^{s'}(f, P_{\mathcal{X}})$. □

VI. Related Work

We begin by pointing out some high-level differences between our work and other approaches as shown in Table I. We then discuss important specific differences in more detail below.

The leftmost three columns of Table I describe properties on which explanation techniques differ, and on which our approach is parameterized. First, our approach is parametric in a **Quantity** of interest that allows us to provide explanations for different behaviors of a system, as opposed to simply explaining absolute instance predictions. Second, we can specify a **Distribution** of interest, allowing explanations of network behavior across different groups of instances (e.g., an instance or a particular class). Cells marked ✓⁻ in these columns denote limited flexibility along this dimension through the choice of a baseline, as in integated gradients [3]. Finally, our approach can select which **Internal** neurons to measure, which, as we demonstrate in Section III, is key to identifying learned concepts. By contrast, integrated gradients [3], sensitivity analysis [2], and simple Taylor decomposition [4] assign importance solely to the input features. Deconvolution [5], guided backpropagation [6], and layer-wise relevance propagation[4], use internal influence in the course of computing input influence, but do not apply internal influence measurements to identifying learned concepts.

The rightmost two columns in Table I describe properties of the influence measure used to build explanations. **Marginality** requires that the influence of each feature depends only on its own marginal contribution, which is implied by distributional marginality. Measures not satisfying marginality may attribute behavior to the wrong features, giving misleading results. **Sensitivity** requires that if the instance and a *baseline* instance differ in one feature and yield different predictions, then that

	Explanation framework properties			*Influence properties*	
	Quantity	**Distribution**	**Internal**	**Marginality**	**Sensitivity**
Influence-Directed	✓	✓	✓	✓	✓*
Integrated Gradients [3]		✓⁻		✓	✓
Simple Taylor [4]		✓⁻		✓	
Sensitivity Analysis [2]				✓	
Deconvolution [5]			✓†		
Guided Backpropagation [6]			✓†	✓	
Relevance Propagation [4]		✓⁻	✓†	✓*	✓*

TABLE I: Comparison of the influence-directed explanations proposed here to prior related work. ✓⁻ denotes that the framework has limited flexibility for the feature, ✓* denotes that the framework may have the feature under certain parameterizations, and ✓† denotes that the framework measures internal influence only as an intermediary step to computing feature influence.

feature is assigned non-zero influence. Because sensitivity refers to a baseline, our explanations must specify the baseline via the distribution of interest to achieve this property. Measures failing to satisfy sensitivity may fail to identify features that are causally relevant to the explanation, leading to "blind spots" and misleading results.

1) Identifying influential regions: One approach to interpreting predictions for convolutional networks is to map activations of neurons back to regions in the input image that are the most relevant to the outcomes of the neurons. Possible approaches for localizing relevance include: *(1)* visualizing gradients [2], [3], [4], *(2)* propagating activations back using gradients [5], [6], [4], and *(3)* fitting a simpler interpretable model around a test point to predict relevant input regions [17]. Because these approaches relate instance-specific features to instance-specific predictions, their results do not generalize beyond a single input point, as demonstrated in Section III-A.

Most prior approaches have not leveraged internal units. Two exceptions are class activation mapping (CAM, Grad-CAM) [18], [19], in which objects in an image are localized by measuring the influence of feature maps, and a recent technique proposed by Oramas et al. [12] in which internal neurons are interpreted to provide an explanation. CAM and Grad-CAM differ from our work in that internal influences are aggregated to represent the localization of a concept identifying an entire class in an input instance, whereas our approach is more granular and can isolate components that represent simpler concepts than an entire class. Oramas et al. [12] use unit activation levels to determine relevance, which, as we demonstrate in Section III-B, is less effective at identifying important concepts than our influence measure. Concurrent work [20] has also suggested a slightly different approach; namely, measuring the input attribution that "flows through" a particular internal neuron.

2) Visualization by maximizing activation: An orthogonal approach is to visualize learned features by identifying input instances that maximally activate a neuron, achieved by either optimizing the activation in the input space [2], [7], [8], or by searching for instances in a dataset [9]. These techniques can complement our work by providing a means to visualize the concept learned by a set of neurons that the influence measure identifies as important for a particular quantity and distribution of interest.

3) Attribution vs. Influence: Of the measures summarized in Table I, some, e.g., Integrated Gradients [3] and Relevance Propagation [4], measure *attribution*, while others, e.g., Sensitivity Analysis [2] and Influence-Directed explanations, measure *influence*. Here, attribution can be understood as the amount of the quantity of interest that can be attributed to a particular neuron. In contrast, influence addresses the sensitivity of the quantity of interest to a particular input or input distribution.

Dhamdhere et al. [20] claim that measures calculating influence, such as influence-directed explanations, lead to non-intuitive results in some cases [1]. This argument is predicated on the a priori insistence on an axiom called *completeness*, which states that the sum of the influences must equal the change in output relative to the baseline. We find this position difficult to get behind. First, it is unclear to us why this axiom should be demanded for influence measures given the nuanced difference between attribution and influence described above. Second, even for attribution measures (power indices) from co-operative game theory, the completeness axiom does not always hold. Straffin provides an analysis of two different power indices—one of which satisfies completeness and the other doesn't—proving that the degree of statistical independence between the inputs determines which index is appropriate for use (see [13], Chapter 5).

Another challenge with applying Integrated Gradients [3] is that it may not respect distributional faithfulness since the axioms used to arrive at that importance measure does not enforce such a constraint. Kindermans et al. [21] argue that measures calculating attribution may give undesirable explanations when the baseline is not appropriately selected to control for trends in the dataset.

We suspect that both attribution and influence may have complementary applications. Future work may help determine for which applications attribution or influence is more effective, and for which attribution and influence complement one another.

[1] Although this paper was released a few months after our paper on arXiv, we regard it as independent, concurrent work based on conversations with the authors.

VII. Future Work

We expect the distributional influence measure introduced in this paper to be applicable to a broad set of deep neural networks. One direction for future work is to couple this measure with appropriate interpretation methods to produce influence-directed explanations for other types of deep networks, such as recursive networks for text processing tasks. Another direction is to develop debugging tools for models using influence-directed explanations as a building block.

Acknowledgment.: This work was developed with the support of NSF grant CNS-1704845 as well as by DARPA and Air Force Research Laboratory under agreement number FA9550-17-1-0600. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright notation thereon. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of DARPA, the Air Force Research Laboratory, the National Science Foundation, or the U.S. Government.

References

[1] Will Knight. The Dark Secret at the Heart of AI. *MIT Technology review*, Apr 2017. Accessed: 2017-10-27.

[2] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *ArXiv e-prints*, 2014.

[3] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *ArXiv e-prints*, 2017.

[4] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140, 07 2015.

[5] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *ECCV*, 2014.

[6] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *ICLR*, 2015.

[7] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image repre tations by inverting them. *CoRR*, abs/1412.0035, 2014.

[8] Anh Mai Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *CoRR*, abs/1605.09304, 2016.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, 2014.

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[11] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. R. Mller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, Nov 2017.

[12] José Oramas M., Kaili Wang, and Tinne Tuytelaars. Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks. *CoRR*, abs/1712.06302, 2017.

[13] A.E. Roth. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[14] R. J. Aumann and L. S. Shapley. *Values of Non-Atomic Games*. Princeton University Press, 1974.

[15] Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 598–617, 2016.

[16] H. P. Young. Individual contribution and just compensation. In Alvin E. Roth, editor, *The Shapley Value*, chapter 17, pages 267–278. Cambridge University Press, 1988.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. ACM.

[18] 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)*, Oct 2017.

[19] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. How important is a neuron? *CoRR*, abs/1805.12233, 2018.

[21] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. The (Un)reliability of saliency methods. *ArXiv e-prints*, November 2017.