Causal Analysis for Robust Interpretability of Neural Networks - A Supplementary Material

Ola Ahmad[1] Nicolas Béreux[2*] Loïc Baret[1] Vahid Hashemi[3] Freddy Lecue[4*]
[1] Thales Digital Solutions, CortAIx, Montreal, Canada
[2] Paris-Saclay University, Paris, France
[3] AUDI AG, Ingolstadt, Germany
[4] Inria, Sophia Antipolis, France
ola.ahmad@thalesdigital.io

1. Extended Related Work

Interpretability using single-neuron techniques. Understanding deep neural networks (DNNs) through individual neurons is commonly done by visualizing the response of individual neurons such as [8–10, 14, 16], regardless of the interactions between individual neurons within connected layers. Methods like [6] try to identify important neurons while accounting for such interactions. However, their method is based on discovering critical neurons using a multi-armed bandit technique, which suffers high computational complexity and often leads to sub-optimal performance [7]. In contrast, we rely on causal inference and develop a path intervention technique to discover critical neurons.

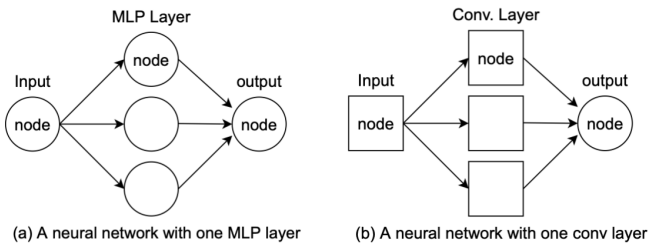
2. Background on Causality

In this section we provide relevant information from causality [12] which our paper has relied on.

2.1. Causal Models

Definition 1 *A causal model (M) is a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the nodes $\mathcal{V}$ represent the set of variables or signals $\mathcal{A}$ (i.e., each node v_i encodes a signal a_i). The edges $\mathcal{E}$ represent a set of causal mechanisms or functions $\mathcal{F}$ that describe the associations between signals. Precisely, each function determines the value of a signal (a_i) based on its parent nodes $pa_i \in \mathcal{V}$.*

The inner representation of neural networks has been first observed as a DAG model in [5] (particularly, RNNs). In this paper, the nodes of the causal graph are channels in case of convolution layers and neurons in MLP or fully connected layers (see Fig. 1). To unify our definitions, we use interchangeably nodes and neurons regardless of the structure of the neural layers. An activated signal a_i outgoing



(a) A neural network with one MLP layer (b) A neural network with one conv layer

Figure 1. **Directed acyclic graphs in neural networks.**

from node i is modulated by the weights (or the response of the filters) that control its transmission to the next layer.

2.2. Interventions

An Intervention is defined through a mathematical operator called $do(z)$ [12]. It implies deleting certain functions from the causal model, and replacing them with constant value $Z = z$, while keeping the rest of the model unchanged. The manipulated model is then denoted by M_z. The outcome from the action $do(Z = z)$ is given by $P_M(y|do(z)) = P_{M_z}(y)$. The variables that we could intervene on in a DAG model are nodes, edges or paths, while the role of intervention $(do(.))$ meets one of three rules: 1) Insertion/deletion of observations; 2) action/observation exchange; or 3) Insertion/deletion of actions [13].

2.3. Edge Interventions

Node interventions are very popular in causal inference because many applications require discovering the effect of treatments applied on input variables or features encoded in the nodes. Path (or edge) intervention is not common, however, it can be very helpful in settings where only some components of the signals may have direct consequences [17].

*Work done while at Thales Digital Solutions

In our case, these components are the weights of a trained neural network that control the flow of information between the nodes.

Considering the subgraphs between the hidden and output layers of the two DAG examples shown in Fig. 1. Let nodes v_1, v_2, v_3 be the set of variables in the hidden layer and v_4 is the output node. We define the set of edges from the parent variables by $e = \{(v_1v_4)_{\rightarrow}, (v_2v_4)_{\rightarrow}, (v_3v_4)_{\rightarrow}\} \subset \mathcal{E}$. An edge intervention is a forced assignment to instances or a subset of the variables in e. For example, assign the value $z = \beta w_1$ to the edge $(v_1v_4)_{\rightarrow}$ in the last layer, where w_1 is the original weight and β is a value identified to represent the intervention action.

2.4. Causal Effects

Definition 2 *(Average Treatment Effect (ATE) [11]) The average treatment (or causal) effect is one measure of the efficacy of an intervention which compares different aspects of the distribution $P_M(y|do(z))$ at different levels (or types) of treatments z_1, z_0. Formally, it is defined by*

$$ATE = E[y|do(z_1)] - E[y|do(z_0)], \tag{1}$$

where $E[y|do(z_*)]$ is the expected value. In our work, the action $do(z_*)$ refers to edge intervention. The average of the treatment effect is obtained over the input samples of certain class. To find out how significant the difference of outcome distributions under the treatments: 1) z_1 remove an edge and 2) z_0 do nothing, we use a hypothesis testing process taking into account the mean and variance over all possible changes in one subgraph, controlling thereby for the error type I.

3. Background on Evaluation Metrics for Explanations

Different evaluation metrics have been proposed to measure explanations' correctness and compare different attribution methods. We focused on measures that reflect how explanations are reliable and trustworthy concerning model predictions. More specifically, Lipschitz estimates [1] and IROF [15] among many developed alternatives [2–4, 18].

Definition 3 *(Local Lipschitz Estimate [1]) Given $s : \boldsymbol{X} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ an explanation generation function which is locally difference-bounded by $h : \boldsymbol{X} \subset \mathbb{R}^n \rightarrow \mathbb{R}^k$, where $h(.)$ are functions mapping raw inputs to a space of interpretable basis concepts, then stability of the explanation generation function is defined by estimating for a given input x and neighbourhood size ϵ*

$$\hat{L}(x_i) = \underset{x_j \in B_\epsilon(x_i)}{\operatorname{argmax}} \frac{\|s(x_i) - s(x_j)\|_2}{\|h(x_i) - h(x_j)\|_2}$$

For all attribution methods mentioned in this paper, we replaced h with the corresponding raw input x, i.e., $h(x) = x$, as suggested in [1]. In our method, the explanation generation function s is given by equation eq. (3) in the main paper. Note that we have changed some notations in [1] for consistency with the content of our paper. In our implementations, the ϵ-neighbourhood of each example $\boldsymbol{x}$ is obtained by perturbing the inputs with white-noise $B_\epsilon = \mathcal{N}(\mu, \epsilon)$, where μ and ϵ were both set to 0.1, respectively. The Lipschitz Estimate was computed over 10 random runs for each example $\boldsymbol{x} \in \boldsymbol{X}$.

Definition 4 *(IROF [15]) Assuming a neural network $f : \boldsymbol{X} \rightarrow y$, an explanation generation function $s : \boldsymbol{X} \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$, and a segmentation method providing a set of binary segments $\{m^k\}_{k=0}^K$ for an input $\boldsymbol{x}$. An importance measure, with respect to s, is defined for each segment by $\|s.m^k\|_1 / \|m^k\|_1$. Given $\tilde{\boldsymbol{x}}^k$ a perturbed version of the input $\boldsymbol{x}$, such that all the pixels in a local region defined by the segment m^k were replaced by their corresponding mean value, then the IROF metric of an explanation method s is defined by*

$$IROF(s) = \frac{1}{N} \sum_{n=1}^{N} AOC \left(\frac{f(\tilde{\boldsymbol{x}}_n^k)_y}{f(\tilde{\boldsymbol{x}}_n^0)_y} \right)_{k=0}^{K}$$

where the average is obtained over all inputs of the same class label y. The AOC is computed using the Trapezoidal rule. As we mentioned in the paper, to compute the IROF, we first sorted the regions based on their importance scores.

4. Implementation details of Algorithm 1

For estimating the causal graphs of a DNN architecture $N(L)$, the algorithm works in a top-down manner. Starting from the penultimate layer $l = L-1$, we select target edges $w_j^{L-1 \rightarrow L}$ directed from parent node (j) to the target class (output). We apply interventions on each edge individually and compute eq. (2) (in the main paper) for each input $\boldsymbol{x}$. We chose as default a binary intervention $\beta \in \{0, 1\}$. Then, by solving eq. (1) (in main paper), we discover the critical nodes in $L-1$, and chose them to identify the edges that we will intervene on in the subsequent lower layer. The process is repeated until it reaches the input. For complex architectures, such as MobileNetV2, ResNet50V2, and the tiny ConvNext, which include layers with a large number of neurons, we neglect 10% of the weights with norm values close to the mean in each layer. We observe that the weights distributed near the mean value are most likely to be similar, and intervening on all of them adds unnecessary computational cost.

4.1. Time complexity of Algorithm 1

We measured the time needed for Algorithm 1 to compute a causal graph of the target model using 100 samples

from a class of the dataset it was trained on. The experiments were carried on an AMD Ryzen Threadripper PRO 3955WX 16-Cores (4.4Ghz) coupled to an Nvidia Geforce RTX 3090. The results are displayed in Tab. 1

4.2. Visualization of LeNet's Causal Graphs

We present an abstraction of the causal graphs generated for labels 3 and 8 in Fig. 2 and 3, respectively. To the right are irrelevant and noisy nodes which are not part of the graphs. In these figures, critical nodes are shown in red and green.

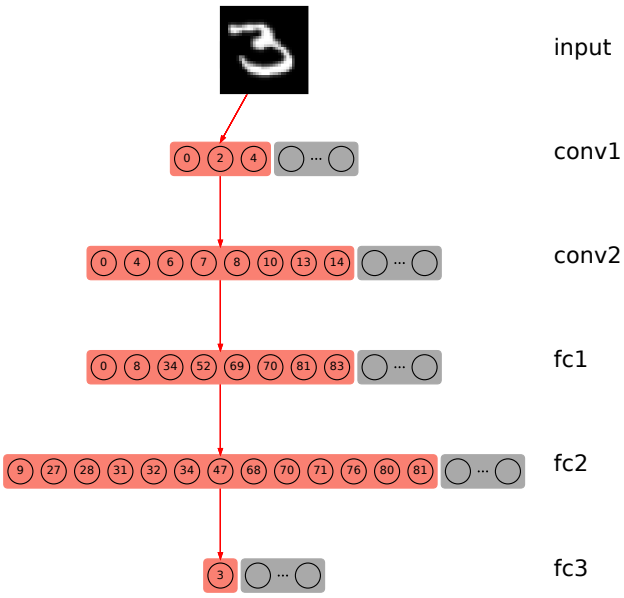


Figure 2. **LeNet causal graph on MNIST for the class 3.** Red nodes refer to critical channels and neurons in conv and fully connected layers, respectively.

4.3. Visualizing Explanations on Hard and Easy Examples

In Fig. 4 and Fig. 5, we compare explanations on hard and easy examples of MNIST and ImageNet. Hard examples refer to images containing distracting objects or context. We use the causal graphs of the actual labels to generate the explanations (from the last conv layers) and qualitatively show why the models have made mistaks. As can be seen in these figures, the responses of causal filters are localized at different semantic parts of the objects in the case of easy examples. For hard examples, attributions are localized in the same region, which belongs to the distracting class.

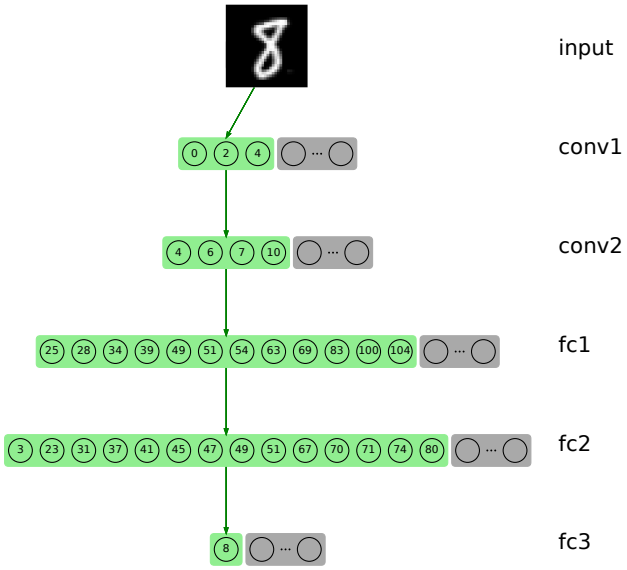


Figure 3. **LeNet causal graph on MNIST for the class 8.** Green nodes refer to critical channels and neurons in conv and fully connected layers, respectively.

4.4. Qualitative Comparison with Attribution Methods

In Fig. 6, we compare explanations between different attribution methods and ours on test examples from ImageNet. For our method, we show the aggregated attributions of all critical (or relevant) nodes at the last convolution layer. As can be seen, our method provides clean and accurate explanations. They are localized on the different parts of the objects.

References

[1] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2

[2] Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques. *arXiv e-prints*, 2019. 2

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On Pixel-Wise Explanations for Non-Linear Classi-

Table 1. Time complexity of our algorithm to generate the causal graphs of different DNN models for one task/class.

Dataset	Model	Time
ImageNet	tiny-ConvNeXt	0hrs 19mn 03s
ImageNet	MobilenetV2	0hrs 44mn 06s
ImageNet	ResNet50V2	2hrs 54mn 01s
ImageNet	ResNet18	0hrs 34mn 37s
CIFAR10	ResNet18	0hrs 32mn 13s
CIFAR10	LeNet	0hrs 00mn 27s
MNIST	LeNet	0hrs 00mn 23s

fier Decisions by Layer-Wise Relevance Propagation. *PLoS ONE*, 10(7):e0130140, July 2015. 2

[4] Umang Bhatt, Adrian Weller, and José M. F. Moura. Evaluating and aggregating feature-based model explanations. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3016–3022. ijcai.org, 2020. 2

[5] Aditya Chattopadhyay, Piyushi Manupriya, Anirban Sarkar, and Vineeth N Balasubramanian. Neural network attributions: A causal perspective. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 981–990, 2019. 1

[6] Amirata Ghorbani and James Zou. Neuron shapley: discovering the responsible neurons. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 5922–5932. Curran Associates Inc., Dec. 2020. 1

[7] T. Lattimore and Cs. Szepesvári. *Bandit Algorithms*. Cambridge University Press, August 2020. 1

[8] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *Visualization for Deep Learning workshop, International Conference in Machine Learning*, 2016. arXiv preprint arXiv:1602.03616. 1

[9] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. https://distill.pub/2020/circuits/zoom-in. 1

[10] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature Visualization. *Distill*, 2(11):10.23915/distill.00007, Nov. 2017. 1

[11] Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3(none):96–146, Jan. 2009. 2

[12] Judea Pearl. *Causality*. Cambridge University Press, 2009. 1

[13] Judea Pearl. The Do-Calculus Revisited. *arXiv:1210.4852 [cs, stat]*, Oct. 2012. 1

[14] Ivet Rafegas, Maria Vanrell, Luís A. Alexandre, and Guillem Arias. Understanding trained cnns by indexing neuron selectivity. *Pattern Recognition Letters*, 136:318–325, 2020. 1

[15] Laura Rieger and Lars Kai Hansen. IROF: a low resource evaluation metric for explanation methods. *CoRR*, abs/2003.08747, 2020. 2

[16] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models, 2017. 1

[17] Ilya Shpitser and Eric Tchetgen Tchetgen. Causal inference with a graphical hierarchy of interventions. *The Annals of Statistics*, 44(6), Dec. 2016. 1

[18] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. On the (In)fidelity and Sensitivity for Explanations. *arXiv:1901.09392 [cs, stat]*, Nov. 2019. 2

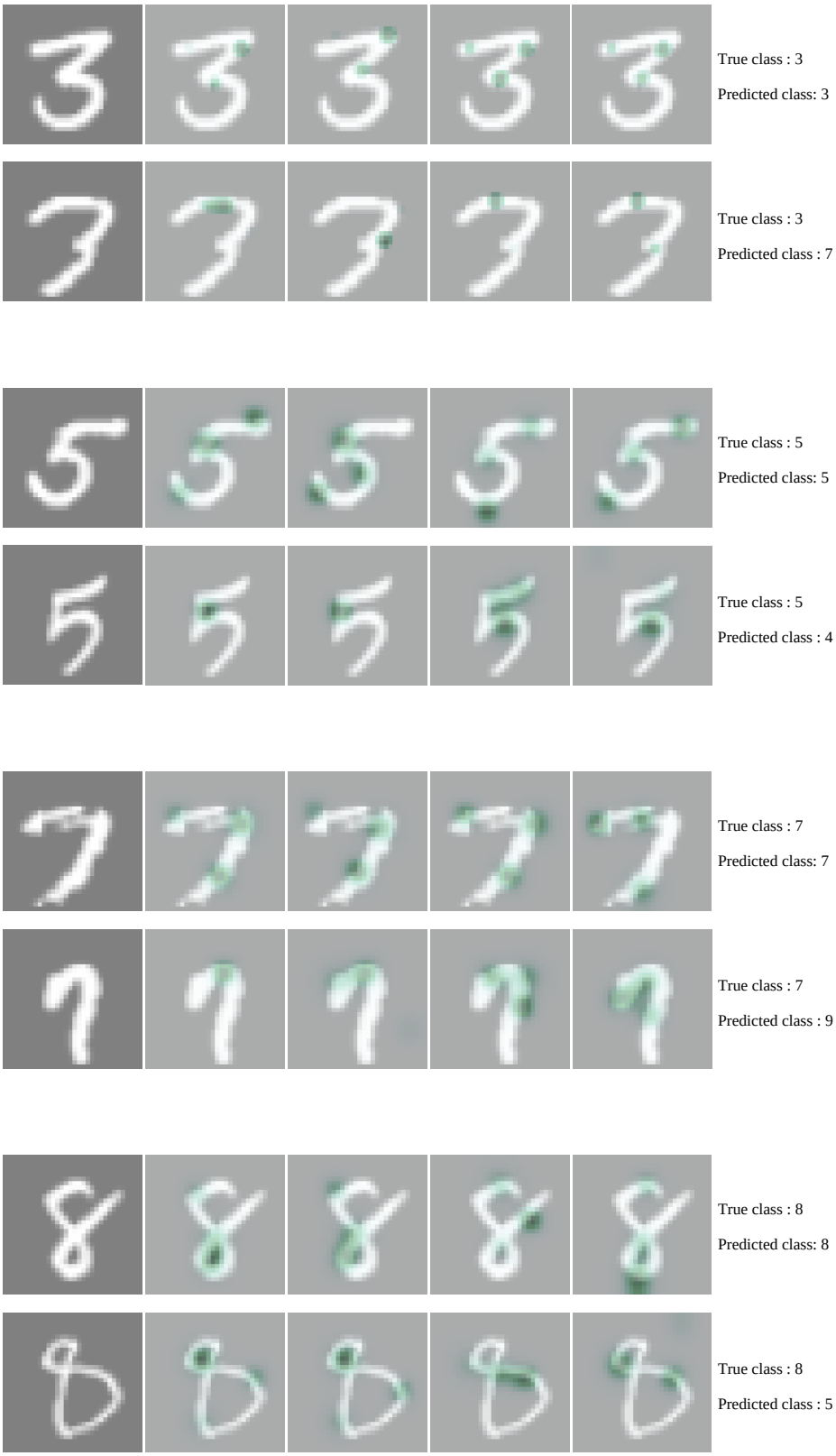


Figure 4. **Attributions for LeNet on easy and hard examples of the MNIST dataset.**

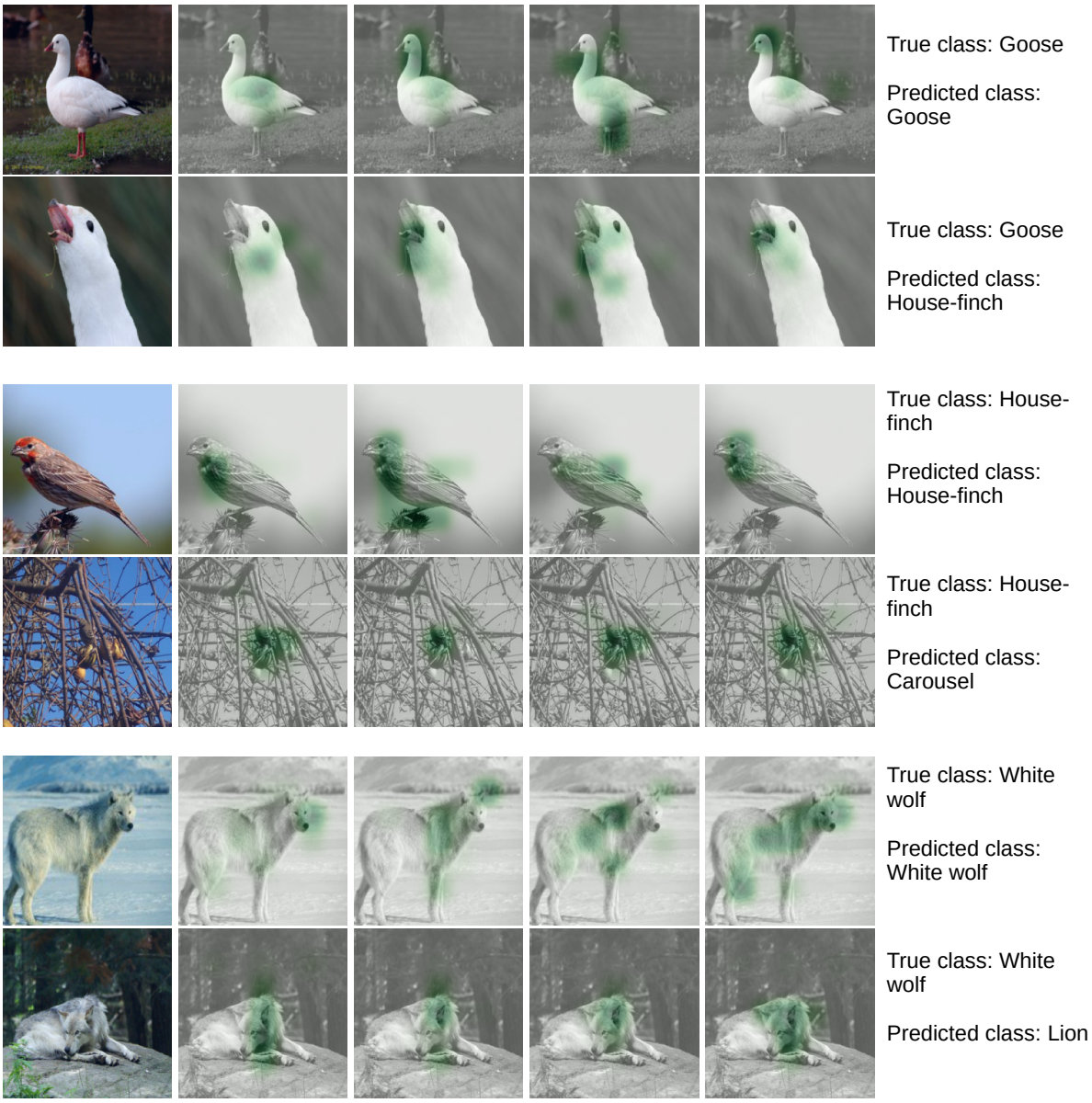


Figure 5. **Attributions for ResNet18 on easy and hard examples of ImageNet.**

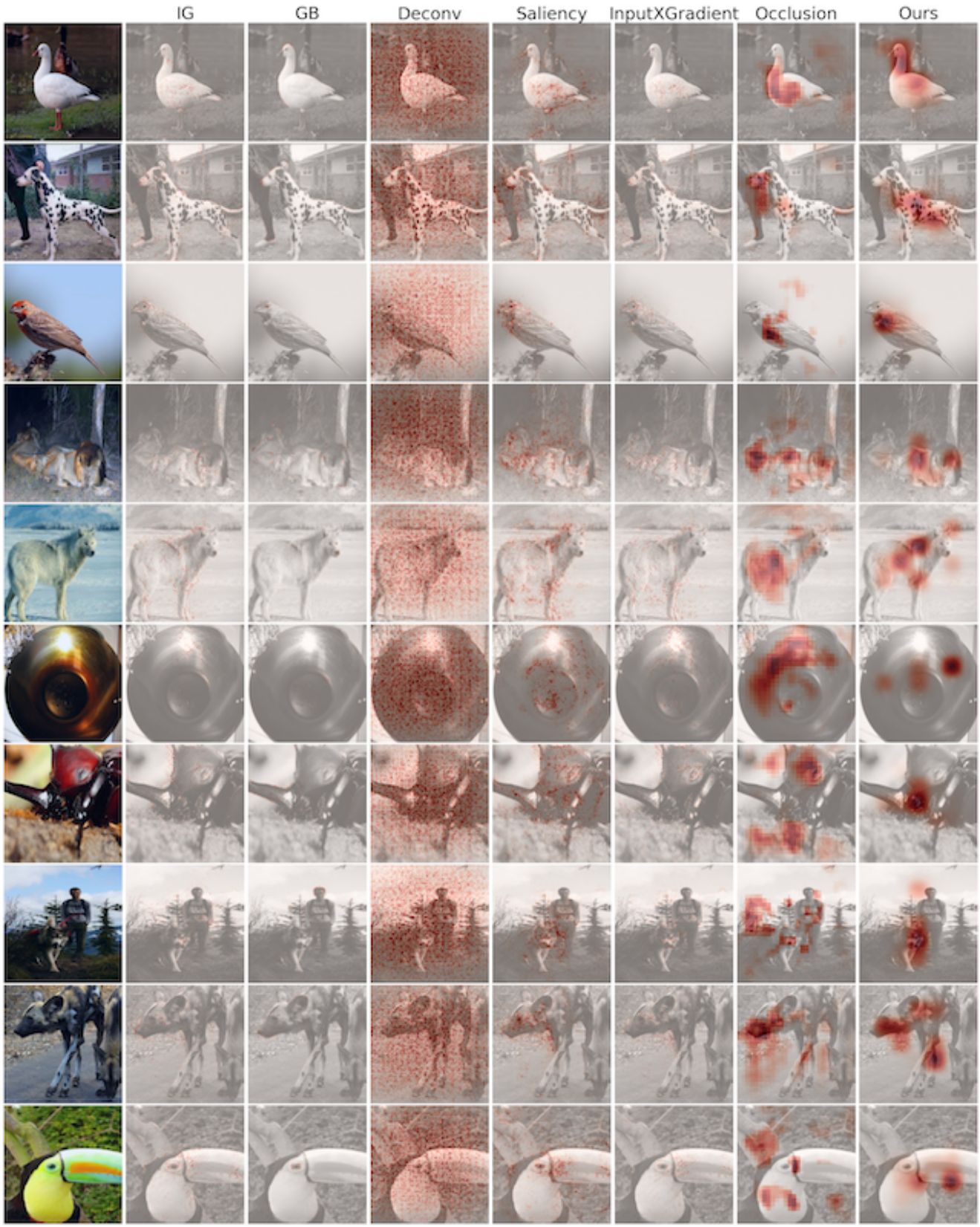


Figure 6. **Comparisons between explanation methods.** Baseline methods are applied on ResNet18 model trained on ImageNet.