Fast Axiomatic Feature Attribution for Training Neural Networks with Explanations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Mitigating the dependence on spurious correlations present in the training dataset is a quickly emerging and important topic of deep learning. Recent approaches include priors on the explanation, specifically the feature attribution, of a deep neural network (DNN) into the training process to reduce the dependence on unwanted features. However, until now we faced a trade-off between high-quality explanations satisfying desirable axioms and the time required to compute them. This in turn led to long training times or ineffective attribution priors. In this work, we break this trade-off by considering a special class of DNNs that we call *nonnegatively homogeneous DNNs* ($\mathcal{X}$-DNNs). They can be effortlessly constructed from a wide range of regular DNNs by simply removing the bias term of each layer. We show that this nonnegative homogeneity is a desirable property and formally prove that it implies the DNN being *efficiently explainable*, *i.e.* the existence of a closed-form solution for axiomatic feature attribution that can be computed efficiently. Various experiments demonstrate the advantages of $\mathcal{X}$-DNNs, beating state-of-the-art generic attribution methods for learning with attribution priors.

1 Introduction

Many traditional machine learning (ML) approaches, such as linear models or decision trees, are inherently explainable [4]. Therefore, an ML practitioner can comprehend why a method yields a particular prediction and correct the method if the explanation for the result is flawed. The prevailing ML architectures in use today [23], namely deep neural networks (DNNs), unfortunately, do not come with this inherent explainability. This can cause models to depend on dataset biases and spurious correlations. For real-world applications, *e.g.* credit score and insurance risk assignment, this can be fatal and potentially lead to models discriminating against certain demographic groups [3, 19]. To mitigate the dependence on spurious correlations in DNNs, attribution priors have been recently proposed [7, 20, 21]. By enforcing priors on the explanation of a DNN at training time, they allow actively controlling its behavior. As it turns out, attribution priors are a very flexible tool, allowing even complex model interventions such as making an object recognition model focus on shape [20] or less sensitive to high-frequency noise [7]. However, their use brings new challenges over regular training. First, computing the attribution of a DNN is a nontrivial task. It is critical to use an explanation method that faithfully reflects the true behavior of the model and ideally satisfies the axioms proposed by Sundararajan *et al.* [31]. Otherwise, spurious correlations may go undetected. Second, since the explanation is used in each training step, it needs to be efficiently computable. Existing work incurs a trade-off between high-quality explanations for which formal axioms hold and the time required to compute them. Prior work on attribution priors thus had to choose whether to rely on high-quality explanations *or* allow for efficient training. In this work, we obviate this trade-off.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Specifically, we make the following contributions: *(i)* We propose to consider a special class of DNNs, termed *efficiently explainable DNNs*, for which an efficiently computable axiomatic feature attribution method exists.[1] *(ii)* We formally prove that *nonnegatively homogeneous DNNs* ($\mathcal{X}$-DNNs) are efficiently explainable DNNs in that there exists a closed-form solution for an axiomatic feature attribution method that fulfills the axioms of Sundararajan *et al.* [31], requiring only one gradient evaluation. This makes the computation of axiomatic explanations for nonnegatively homogeneous DNNs two orders of magnitude more efficient than for regular DNNs, which require a costly numerical approximation of an integral. *(iii)* We show how $\mathcal{X}$-DNNs can be instantiated from a wide range of regular DNNs by simply removing the bias term of each layer. While this may seem like a significant restriction, we show that the impact on the predictive accuracy in two different application domains is surprisingly minor. In a variety of experiments, we demonstrate the advantages of $\mathcal{X}$-DNNs, showing that they *(iv)* admit accurate axiomatic feature attributions at a fraction of the computational cost and *(v)* beat state-of-the-art generic attribution methods for training with an attribution prior.

2 Related work

Attribution methods can roughly be divided into perturbation-based [12, 34, 36, 37] and backpropagation-based [1, 5, 26, 28, 31] methods. The former repeatably perturb individual inputs or neurons to measure their impact on the final prediction. Since each perturbation requires a separate forward pass through the DNN, those methods can be computationally inefficient [26] and consequently inappropriate for inclusion into the training process. We thus consider *backpropagation-based methods* or, more precisely, gradient-based and rule-based attribution methods. They propagate an importance signal from the DNN output to its input using either the gradient or predefined rules, making them particularly efficient [26], and thus, well suited for inclusion into the training process. Gradient-based methods have the advantage of scaling to high-dimensional inputs, can be efficiently implemented using GPUs, and directly applied to any differentiable model without changing it [2].

The saliency method [28], defined as the absolute input gradient, is an early gradient-based attribution method for DNNs. Shrikumar *et al.* [25] proposed the Input$\times$Gradient method, *i.e.* weighting the (signed) input gradient with the input features, to improve sharpness of the attributions for images. Bach *et al.* [5] introduced the rule-based Layerwise Relevance Propagation (LRP), with predefined backpropagation rules for each neural network component. As it turns out, LRP without modifications to deal with numerical instability can be reduced to Input$\times$Gradient for DNNs with ReLU [18] activation functions [1, 25], hence can be expressed in terms of gradients as well. DeepLIFT [26] is another rule-based approach similar to LRP, relying on a neutral baseline input to assign contribution scores relative to the difference of the normal activation and reference activation of each neuron. Generally, rule-based approaches have the disadvantage that each DNN component requires custom modules that may have no GPU support and require a re-implementation of the model.

Axiomatic attributions. As it is hard to empirically evaluate the quality of attributions, Sundararajan *et al.* [31] proposed several axioms that high-quality attribution methods should satisfy:

Sensitivity (a) is satisfied if for every input and baseline that differ in one feature but have different predictions, the differing feature should be given a non-zero attribution.

Sensitivity (b) is satisfied if the function implemented by the deep network does not depend (mathematically) on some variable, then the attribution to that variable is always zero.

Implementation invariance is satisfied if the attributions for two functionally equivalent networks are always identical.

Completeness is satisfied if the attributions add up to the difference between the output of the network for the input and for the baseline.

Linearity is satisfied if the attribution of a linearly composed deep network $aF_1 + bF_2$ is equal to the weighted sum of the attributions for F_1 and F_2 with weights a and b, respectively.

Symmetry preservation is satisfied if for all inputs and baselines that have identical values for symmetric variables, the symmetric variables receive identical attributions.

[31] shows that none of the above methods satisfies all axioms, *e.g.* the saliency method and Input$\times$Gradient suffer from the well-known problem of gradient saturation, which means that even

[1] Informally, the attribution of a DNN for a given input yields information about how important an input feature is for the prediction. As a consequence, it can be used to explain why a certain decision was made. The terms attributions and explanations will thus be used interchangeably throughout the remainder of this work.

important features can have zero attribution. To overcome this, [31] introduced Integrated Gradients, a gradient-based backpropagation method that provably satisfies these axioms; it is considered a high-quality attribution method to date. Its crucial disadvantage over previous methods is that an integral has to be solved, which generally requires an approximation based on $\sim$20–300 gradient calculations, making it correspondingly computationally more expensive than, *e.g.*, Input$\times$Gradient.

Attribution priors. The above attribution methods can not only be used for explaining a model's behavior but also to actively control a model's behavior. To that end, the training objective can be formulated as

$$\theta^* = \arg\min_{\theta} \frac{1}{|X|} \sum_{(x,y)\in X} \mathcal{L}(F_\theta; x, y) + \lambda\Omega(\mathcal{A}(F_\theta, x)), \tag{1}$$

where a model F_θ with parameters θ is trained on the dataset X. $\mathcal{L}$ denotes the regular task loss, and Ω is a scalar-valued loss of the feature attribution $\mathcal{A}$, which is called the attribution prior [7]; λ controls the relative weighting. For example, by forcing certain values of the attribution to be zero, we can mitigate the dependence on unwanted features [21]. But also more complex model interventions like making an object recognition model focus on shape [20] or less sensitive to high-frequency noise [7] can be formulated using attribution priors.

An early instance of this concept is the Right for the Right Reasons (RRR) approach of Ross *et al.* [21], which uses the input gradient of the *log* prediction to mitigate the dependence on unwanted features. While this is more stable than simply using the input gradient, it still suffers from the problem of saturation. RRR may thus not reflect the true behavior of the model and, therefore, miss relevant features. Subsequent work addressed this using axiomatic feature attribution methods, specifically Integrated Gradients [6, 13, 31], which however incur significant computational overhead, rendering them impractical for many scenarios. Rieger *et al.* [20] proposed an alternative attribution prior based on a rule-based contextual decomposition [17, 29] (CD) as attribution method. This allows to consider clusters of features [7] instead of individual features and define attribution priors working on feature groups. However, computing the attribution for individual features becomes computationally inefficient [7]. Additionally, since CD is a rule-based attribution method, it requires custom modules and cannot be applied to all types of DNNs [7]. The very recently proposed Expected Gradients [7] method reformulates Integrated Gradients as an expectation, allowing a sampling-based approximation of the attribution. Erion *et al.* argue that similar to batch gradient descent, where the true gradient of the loss function is approximated over many training steps, the sampling-based approximation allows to approximate the attribution over many training steps. This results in better attributions while using fewer approximation steps. Even using as little as one reference sample, *i.e.* only one gradient must be computed, can yield advantages over the regular input gradient. However, we show that using only one reference sample still does not yield the same attribution quality as an axiomatic feature attribution method. Schramowski *et al.* [24] proposed a human-in-the-loop strategy to define appropriate attribution priors while training. Our attribution method is complementary and could be used within their framework.

3 Efficiently explainable DNNs

Formally, given a function $F\colon \mathbb{R}^n \mapsto \mathbb{R}$ representing a single output of a DNN and an input $x \in \mathbb{R}^n$, the attribution for the prediction at input x relative to a baseline input x' is a vector $\mathcal{A}(F, x, x') \in \mathbb{R}^n$, where each entry a_i is the contribution of feature x_i to the prediction $F(x)$ [31]. Ideally, we want an attribution method that satisfies the axioms proposed by Sundararajan *et al.* [31], while being as efficiently computable as a single input gradient. In general, however, this is not possible for arbitrary DNNs. In this work, we consider a special class of DNNs, termed *efficiently explainable DNNs*, that require only a single gradient evaluation to compute Integrated Gradients. We show that nonnegatively homogeneous DNNs belong to this class and use this insight to guide the design of a concrete instantiation of efficiently explainable DNNs. While there may be several such instantiations, we chose this particular one as it can be easily constructed from a wide range of regular DNNs by simply removing the bias term of each layer. This ensures comparability to prior work and allows for an easy adaptation of existing network architectures.

Definition 3.1. We call a DNN $F\colon \mathbb{R}^n \mapsto \mathbb{R}$ *efficiently explainable w.r.t.* a baseline x', if there exists a closed form solution of the axiomatic feature attribution method Integrated Gradients $\mathrm{IG}_i(F, x, x')$ along the i^{th} dimension of x, requiring only one gradient evaluation.

Note that all differentiable models are efficiently explainable *w.r.t.* the trivial baseline $x' = x$. However, using such a baseline is not helpful. Instead, commonly chosen baselines are some kind of averaged input features or baselines such that $F(x') = 0$, which allow an interpretation of the attributions that amounts to distributing the output to the individual input features [31].

Proposition 3.2. *For a DNN $F\colon \mathbb{R}^n \mapsto \mathbb{R}$ there exists a closed form solution of $IG_i(F, x, \mathbf{0})$* w.r.t. *the zero baseline $\mathbf{0}$ requiring only one gradient evaluation, if F is strictly positive homogeneous of degree $k \in \mathbb{R}_{\geq 1}$,* i.e. *$F(\alpha x) = \alpha^k F(x)$ for $\alpha \in \mathbb{R}_{>0}$.*

Proof. We assume $k \geq 1$. Sundararajan *et al.* [31] define the axiomatic feature attribution method Integrated Gradients (IG) along the i^{th} dimension for a given model F, input x, baseline $\mathbf{0}$, and straightline path $\gamma(\alpha) = \alpha x$ as

$$\text{IG}_i(F, x, \mathbf{0}) = \int_0^1 \frac{\partial F\left(\gamma(\alpha)\right)}{\partial \gamma_i(\alpha)} \frac{\partial \gamma_i(\alpha)}{\partial \alpha}\, d\alpha \;= \int_0^1 \frac{\partial F\left(\alpha x\right)}{\partial \alpha x_i} \frac{\partial \alpha x_i}{\partial \alpha}\, d\alpha\,. \tag{2}$$

Assuming F is strictly positive homogeneous of degree k, we can write Integrated Gradients in Eq. (2) as

$$\text{IG}_i(F, x, \mathbf{0}) = \lim_{\beta \to 0} \int_\beta^1 \frac{\partial F\left(\alpha x\right)}{\partial \alpha x_i}\, x_i\, d\alpha = \lim_{\beta \to 0} \int_\beta^1 \alpha^{k-1} \frac{\partial F\left(x\right)}{\partial x_i}\, x_i\, d\alpha = \frac{1}{k} x_i \frac{\partial F\left(x\right)}{\partial x_i}\,. \tag{3}$$

□

While Ancona *et al.* [1] already found that Input×Gradient equals Integrated Gradients with the zero baseline for linear models or models that behave linearly for a selected task, our Proposition 3.2 is more general: We only require strict positive homogeneity of an arbitrary order $k \geq 1$. This allows us to consider a larger class of models including nonnegatively homogeneous DNNs, which generally are not linear.

Definition 3.3. We call a DNN $F\colon \mathbb{R}^n \mapsto \mathbb{R}$ *nonnegatively homogeneous*, if $F(\alpha x) = \alpha F(x)$ for all $\alpha \in \mathbb{R}_{\geq 0}$.

Corollary 3.4. *Any nonnegatively homogeneous DNN is efficiently explainable* w.r.t. *the zero baseline $\mathbf{0}$ and a closed form solution of the axiomatic feature attribution method Integrated Gradients requiring only one gradient evaluation exists.*

Proof. Corollary 3.4 follows directly from Proposition 3.2 and Definitions 3.1 and 3.3. □

Definition 3.5. We let *$\mathcal{X}$-DNN* denote a nonnegatively homogeneous DNN. Further, for any $\mathcal{X}$-DNN $F\colon \mathbb{R}^n \mapsto \mathbb{R}$, we let *$\mathcal{X}$-Gradient* be an axiomatic feature attribution method relative to the $\mathbf{0}$ baseline defined as

$$\mathcal{X}\text{G}_i(F, x) = \text{IG}_i(F, x, \mathbf{0}) = x_i \frac{\partial F(x)}{\partial x_i}\,. \tag{4}$$

Note that while the formulas for the existing attribution method Input×Gradient and our novel $\mathcal{X}$-Gradient are equal, $\mathcal{X}$-Gradient is only defined for $\mathcal{X}$-DNNs and provably satisfies axioms that are generally not satisfied by Input×Gradient. Additionally, from the nonnegative homogeneity of $\mathcal{X}$-DNNs it follows that $\mathcal{X}$-Gradient attributions are also nonnegatively homogeneous. This allows us to define another desirable axiom that is in line with intuition about how attribution should work and that is satisfied by $\mathcal{X}$-Gradient.

Definition 3.6. An attribution method $\mathcal{A}$ satisfies *nonnegative homogeneity* if $\mathcal{A}(F, \alpha x, \alpha x') = \alpha\, \mathcal{A}(F, x, x')$ for all $\alpha \in \mathbb{R}_{\geq 0}$.

For an overview of the axioms that are satisfied by popular gradient-based attribution methods, see Table 1. The right-hand side methods use only one gradient evaluation, and therefore, have similar computational expense. The left-hand side methods generally require multiple gradient evaluations until convergence, making them correspondingly more computationally expensive. Note that $\mathcal{X}$-Gradient satisfies all the axioms satisfied by Integrated Gradients and Expected Gradients [7], assuming convergence, while requiring only a fraction of the computational cost. Existing methods that have similar computational expense as $\mathcal{X}$-Gradient generally do not satisfy all of the axioms.

Table 1: *Overview of different gradient-based DNN attribution methods and the axioms that they provably satisfy.* The left-hand side methods (Integrated Gradients, Expected Gradients) induce one to two orders of magnitude computational overhead compared to the methods on the right-hand side, which require only one gradient evaluation (indicated by (1) for Expected Gradients with one sample). Note how $\mathcal{X}$-Gradient satisfies all axioms while requiring as little computational cost as a simple gradient evaluation, however being only defined for $\mathcal{X}$-DNNs.

Axiom	Integrated Gradients	Expected Gradients	Expected Gradients(1)	(Input $\times$) Gradient	$\mathcal{X}$-Gradient
Sensitivity (a)	✓	✓	✗	✗	✓
Sensitivity (b)	✓	✓	✓	✓	✓
Implementation invariance	✓	✓	✗	✓	✓
Completeness	✓	✓	✗	✗	✓
Linearity	✓	✓	✗	✓	✓
Symmetry-preserving	✓	✓	✗	✗	✓
Nonnegative homogeneity	✗	✗	✗	✗	✓

With this motivation in mind, we will now study concrete instantiations of nonnegatively homogeneous DNNs. We define the output of a regular feedforward DNN $F\colon \mathbb{R}^n \mapsto \mathbb{R}^o$, for an input $x \in \mathbb{R}^n$, as a recursive sequence of layers i that are applied to the output of previous layers:

$$F_i(x) = \begin{cases} \psi_i\left(\phi_i\left(W_i F_{i-1}(x) + b_i\right)\right) & \text{if } i > 1 \\ x & \text{if } i = 0, \end{cases} \tag{5}$$

with W_i and b_i being the weight matrix and bias term for layer i, ϕ_i being the corresponding activation function, and ψ_i being the corresponding pooling function. Both ϕ_i and ψ_i are optional, in which case they are the identity function. For simplicity, we assume that the last task-specific layer, *e.g.* the softmax function for classification tasks, is part of the loss function. Further, for a cleaner notation that aligns with [31], we assume that we are only considering one output logit at a time, *e.g.* the logit of the target class for classification tasks. This yields the DNN $F\colon \mathbb{R}^n \mapsto \mathbb{R}$ we consider and allows us to directly compute the derivative of the model *w.r.t.* an input feature x_i. Importantly, the above formalization comprises many popular layer types and architectures. For example, fully connected and convolutional layers are essentially matrix multiplications [33], and therefore, can be expressed by Eq. (5). Skip connections can also be expressed as matrix multiplication by appending the identity matrix to the weight matrix so that the input is propagated to later layers [33]. This allows us to describe even complex architectures such as the ResNet [9] architecture proposed by [35]. As the above definition of a DNN includes models that are generally not nonnegatively homogeneous, we have to make some assumptions.

Assumption 3.7. *The activation functions ϕ_i and pooling functions ψ_i in the model are nonnegatively homogenous. Formally, for all $\alpha \in \mathbb{R}_{\geq 0}$:*

$$\alpha\phi_i(z) = \phi_i(\alpha z) \quad \textit{and} \quad \alpha\psi_i(z) = \psi_i(\alpha z). \tag{6}$$

Proposition 3.8. *Piecewise linear activation functions with two intervals separated by zero satisfy Assumption 3.7. For $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$, these activation functions $\phi_i\colon \mathbb{R}^n \mapsto \mathbb{R}^n$ are defined as*

$$\phi_i(z) = (\phi_i'(z_1), \ldots, \phi_i'(z_n)) \quad \textit{with} \quad \phi_i'(z_j) = \begin{cases} a_{i,1} z_j & \textit{if } z_j > 0 \\ a_{i,2} z_j & \textit{if } z_j \leq 0 \end{cases}. \tag{7}$$

Proposition 3.9. *Linear pooling functions or pooling functions selecting values based on their relative ordering satisfy Assumption 3.7. For $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$, these pooling functions $\psi_i\colon \mathbb{R}^n \mapsto \mathbb{R}^m$ are defined as*

$$\psi_i(z) = (\psi_i'(z_1'), \ldots, \psi_i'(z_m')), \tag{8}$$

with z_j' being a grouping of entries in z based on their spatial location and $\psi_i'\colon \mathbb{R}^m \mapsto \mathbb{R}$ being linear or a selection of a value based on its relative ordering, e.g. *the maximum or minimum value.*

For proofs of Propositions 3.8 and 3.9, please refer to the supplemental material. Activation functions in Proposition 3.8 include ReLU [18], Leaky ReLU [15], and PReLU [8]. Linear pooling functions

in Proposition 3.9 include average pooling, global average pooling, and strided convolution. Other pooling functions in Proposition 3.9 include max pooling and min pooling [30], where the largest or smallest value is selected. Therefore, DNN architectures satisfying Assumption 3.7 include, inter alia, AlexNet [11], VGGNet [27], ResNet [9] as introduced in [35], and MLPs with ReLU activations. They alone have been cited well over one hundred thousand times, showing that we are considering a substantial fraction of commonly used DNN architectures. However, these architectures are generally still not nonnegatively homogeneous. It is easy to see that even for a simple linear model $F(x) = ax + b$ that can be expressed by Eq. (5) and that satisfies Assumption 3.7, nonnegative homogeneity does not hold, because $0F(x) = 0 \neq b = F(0x)$. Therefore, in a final step we set the bias term of each layer to zero. As this may seem like a significant restriction, we show in Sec. 4 that the impact on the predictive accuracy in two different application domains is surprisingly minor.

Corollary 3.10. *Any regular DNN given by Eq.* (5) *satisfying Assumption 3.7 can be transformed into a $\mathcal{X}$-DNN by removing the bias term of each layer.*

Proof. A DNN F with m layers given by Eq. (5) with all biases b_i set to 0 can be rewritten as $F(x) = \psi_m(\phi_m(W_m(...(\psi_1(\phi_1(W_1 x))))))$. As all the pooling functions ψ_i, activation functions ϕ_i, and matrix multiplications W_i in F are nonnegatively homogeneous, it follows that $F(\alpha x) = \alpha F(x)$ for all $\alpha \in \mathbb{R}_{\geq 0}$. □

Further discussion. We additionally note that our results have interesting consequences for DNNs in certain application domains, *e.g.* in computer vision, as they allow to relate efficient explainability to desirable properties of DNNs:

Remark 3.11. If a DNN $F\colon \mathbb{R}^n \mapsto \mathbb{R}$ taking an image $x \in \mathbb{R}^n$ as input is equivariant *w.r.t.* to the image contrast, it is efficiently explainable.

This observation follows directly from the fact that contrast equivariance implies nonnegative homogeneity. Consequentially, contrast-equivariant DNNs for regression tasks, such as image restoration or image super-resolution, are automatically efficiently explainable. For classification tasks, such as image classification or semantic segmentation, contrast equivariance of the logits at the output implies efficient explainability. If the classification is done using a softmax, then this also implies contrast invariance of the classifier output. In other words, there is a close relation between efficient explainability and the desirable property of contrast equi-/invariance. We further illustrate this experimentally in Sec. 4.4.

Limitations. So far, we have discussed the advantages of $\mathcal{X}$-DNNs such as being able to efficiently compute high-quality attributions. However, we also want to mention the limitations of our method. First, our method can only be applied to certain DNNs satisfying Assumption 3.7. Although this is a large class of models, our method is not completely model agnostic as other gradient-based attribution methods. Second, removing the bias might be disadvantageous in certain scenarios. However, as we show in Sec. 4, removing the bias may have less of a negative impact than expected. Third, our method uses implicitly the zero baseline $\mathbf{0}$. As $F(\mathbf{0}) = 0$ this is a reasonable baseline because it can be interpreted as being neutral [31]. Nevertheless, other baselines could produce attributions that are better suited for certain tasks. Whether the advantages outweigh the disadvantages must be decided for each application, individually. In Sec. 4 we demonstrate the advantages of $\mathcal{X}$-DNNs, beating state-of-the-art attribution methods for learning with attribution priors.

4 Experiments

In the following we evaluate our proposed method and show its benefits on two data domains.

Experimental setup. For our experiments on models for image classification, *i.e.* Section 4.1, 4.2 and 4.4, we use the ImageNet [22] dataset, containing about 1.2 million images of 1000 different categories. We train on the training split and report numbers for the validation split. In Sec. 4.2 we quantify the quality of attributions for image classification models by adapting the metrics proposed by Lundberg *et al.* [14] to work with image data. The metrics reflect how well an attribution method captures the relative importance of features by masking out a progressively increasing fraction of the features based on their relative importance. As a mask, we use a Gaussian blur of the original image. For a detailed description of the metrics, please refer to [14] and the supplemental material. If not indicated otherwise, we assume numerical convergence for Integrated Gradients and Expected Gradients, which we found to occur after ~ 128 approximation steps (see supplemental material).

Table 2: *Top-5 accuracy* on the ImageNet [22] validation split and relative distance of an $\mathcal{X}$-Gradient attribution to Integrated Gradients. Note how removing the bias ($\mathcal{X}$-DNN) impairs the accuracy only marginally while reducing the relative attribution distance to Integrated Gradients significantly.

	Top-5 accuracy (%, ↑)			Relative distance (%, ↓)		
Model	AlexNet	VGG16	ResNet-50	AlexNet	VGG16	ResNet-50
Regular DNN	**79.2**	**90.4**	**92.6**	79.0	97.8	93.8
$\mathcal{X}$-DNN	78.5	90.2	91.1	**1.2**	**0.4**	**0.0**

Table 3: *Metrics [14] to measure the attribution quality* of different attribution methods. We evaluate Integrated Gradients (IG) [31], random attributions (Random), input gradient attributions (Grad), Expected Gradients (EG) [7], and our novel $\mathcal{X}$-Gradient ($\mathcal{X}$G) attribution on a regular AlexNet [35] and the corresponding $\mathcal{X}$-AlexNet. The number in parentheses indicates the required gradient calls. Our method is on par with IG while requiring two orders of magnitude less computational power.

	AlexNet				$\mathcal{X}$-AlexNet			
Method	KPM ↑	KNM ↓	KAM ↑	RAM ↓	KPM ↑	KNM ↓	KAM ↑	RAM ↓
IG (128)	**7.57**	**1.67**	**25.22**	**11.12**	**7.38**	**2.21**	21.79	11.68
Random	3.68	3.68	14.12	14.10	3.81	3.81	13.52	13.50
Grad (1)	3.62	3.88	20.78	11.82	3.87	4.34	19.75	**11.25**
EG (1)	4.92	2.97	20.49	13.76	5.41	3.19	19.47	13.19
$\mathcal{X}$G (1)	N/A	N/A	N/A	N/A	**7.38**	**2.21**	**21.83**	11.68

4.1 Removing the bias term in DNNs

Historically, the bias term plays an important role and almost all DNN architectures use one. In this first experiment, we evaluate how much removing the bias to obtain a $\mathcal{X}$-DNN affects the accuracy of different DNNs. To this end, we train multiple popular image classification networks, AlexNet [11], VGG16 [27], and the ResNet-50 of [35], as well as their corresponding $\mathcal{X}$-variants obtained by removing the bias term, on the challenging ImageNet [22] dataset. The resulting top-5 accuracy on the validation split is given in Table 2. As we can observe, removing the bias decreases the accuracy of the models only marginally. This is a somewhat surprising result since prior work indicates that the bias term in DNNs plays an important role [33]. We hypothesize that when removing the bias term, the DNN learns some kind of layer averaging strategy that compensates for the missing bias. For an additional comparison between a DNN with bias and its corresponding $\mathcal{X}$-DNN in a non-vision domain, see Sec. 4.3, which mirrors our findings here. Additionally, to empirically validate our finding that $\mathcal{X}$-Gradient ($\mathcal{X}G$) equals Integrated Gradients for $\mathcal{X}$-DNNs, we report the mean relative distance between the attribution obtained from Integrated Gradients [31] and the attribution obtained from calculating Input$\times$Gradient for regular DNNs resp. $\mathcal{X}$-Gradient for $\mathcal{X}$-DNNs over the ImageNet validation split. For regular models with biases, Integrated Gradients produces a very different attribution compared to Input$\times$Gradient. For $\mathcal{X}$-DNNs on the other hand, the two attribution methods are virtually identical. The small deviation can be explained by the fact that the result of Integrated Gradients [31] is computed via numerical approximation, whereas our method computes the exact integral (of course only for $\mathcal{X}$-DNNs). The pre-trained $\mathcal{X}$-DNN models will be made publicly available to promote a wide application of efficiently explainable models.

4.2 Benchmarking gradient-based attribution methods

To demonstrate that our method not only satisfies several axioms [31] but also produces high-quality attributions, we benchmark our method against existing gradient-based attribution methods that are commonly used for training with attribution priors, using the metrics of [14]. Table 3 shows results for a regular AlexNet and our corresponding $\mathcal{X}$-AlexNet. Due to the axioms satisfied by Integrated Gradients, it produces the best attributions for the regular network. However, as it approximates an integral where each approximation step requires an additional gradient evaluation, it also introduces one to two orders of magnitude of computational overhead compared to the other methods (Sundararajan *et al.* [31] recommend 20–300 gradient evaluations to approximate

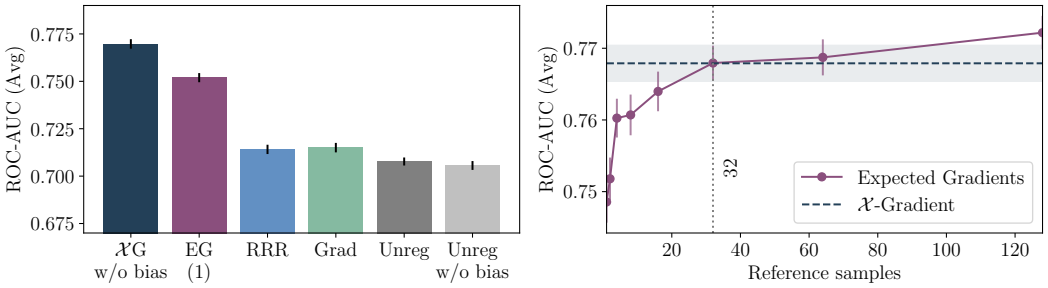

Figure 1: (left) *Average ROC-AUC* across 200 randomly subsampled datasets for the same attribution prior using different attribution methods. (right) *Average ROC-AUC* across 200 randomly subsampled datasets of Expected Gradients (EG) over the number of reference samples. The current state-of-the-art EG requires approximately 32 reference samples, and thus, 32 times more computational power to outmatch $\mathcal{X}$G. Confidence intervals indicate two times the standard error of the mean.

attributions). For the $\mathcal{X}$-AlexNet, however, our $\mathcal{X}$-Gradient method is on par with Integrated Gradients and produces the best attributions while requiring only one gradient evaluation, and therefore, a fraction of the compute power of Integrated Gradients. Since the input gradient and Expected Gradients [7] with only one reference sample do not satisfy many of the desirable axioms listed in Section 3, they produce clearly lower quality attributions as expected.

4.3 Training with attribution priors

To benchmark our method against other attribution methods when training with attribution priors, we replicate the sparsity experiment introduced in [7]. To that end, we employ the public NHANES I survey data [16] of the CDC of the United States, containing 118 one-hot encoded medical attributes, *e.g.* age, sex, and vital sign measurements, from 13,000 human subjects (no personally identifiable information). The objective of the binary classification task is to predict if a human subject will be dead (0) or alive (1) ten years after the data was measured. A simple MLP with ReLU activations is used as the model. Therefore, it can be transformed into a $\mathcal{X}$-DNN by simply removing the bias terms. To emulate a setting of scarce training data and average out variance, we randomly subsample 200 training and validation datasets containing 100 data points from the original dataset. Erion *et al.* [7] proposed a novel attribution prior that maximizes the Gini coefficient, *i.e.* minimizes the statistical dispersion, of the feature attributions. They show that this allows to learn sparser models, which have improved generalizability on small training datasets. The more faithfully the attribution reflects the true behavior of the model, the more effective the attribution prior should be.

Comparing attribution methods. We compare different attribution methods that have previously been used for training with attribution priors and require only one gradient evaluation; thus, they have comparable computational cost. The results in Fig. 1(left) show that our method ($\mathcal{X}$G w/o bias) outperforms all other competing methods. We can also see that for the unregularized model removing the bias (Unreg w/o bias) has almost no effect on the average ROC-AUC of the method, once again showing that our modification for making attributions efficient, *i.e.* removing the bias term, is feasible in many scenarios.

Since the attribution quality of Expected Gradients can be improved using more reference samples, since this yields a better approximation to the true integral, we plot the average ROC-AUC of Expected Gradients over the number of reference samples used in Fig. 1(right). We can clearly see that adding more samples improves the ROC-AUC when training with an EG attribution in the prior. However, we also find that approximately 32 reference samples are needed, and hence 32 times more computational power, to match the quality of our efficient $\mathcal{X}$-Gradient. When using more than 32 reference samples, Expected Gradients slightly outperform our method in terms of ROC-AUC, which is due to the limitations discussed in Sec. 3 (fixed baseline, no bias terms). We argue that it is often worth accepting this small accuracy disadvantage in light of the significant gain in efficiency of computing high-quality attributions. To put this improvement in efficiency into perspective, consider regular DNNs that require days of training using a single GPU, *e.g.* ResNet on the ImageNet dataset. The computational overhead introduced when using Expected Gradients with 32 reference samples would turn several days of training into several months of training.

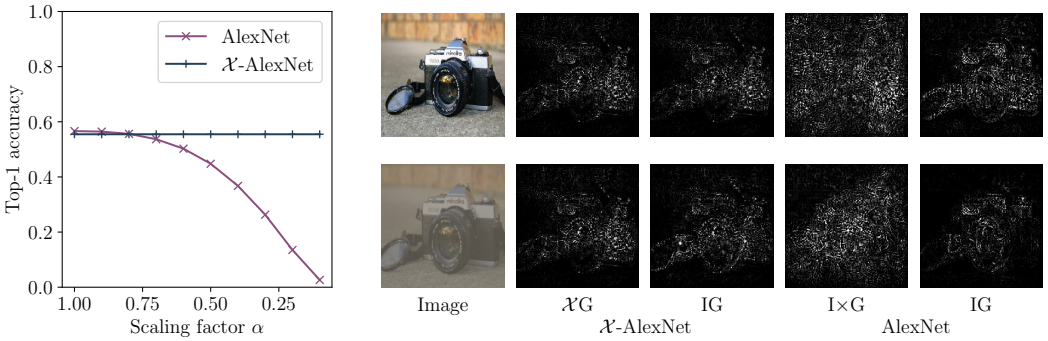

Figure 2: (left) *Top-1 accuracy* for AlexNet and $\mathcal{X}$-AlexNet on the ImageNet validation split with decreasing contrast (scaled by α). Due to the nonnegative homogeneity of $\mathcal{X}$-AlexNet, the accuracy does not drop when reducing the contrast. (right) *Qualitative examples* of normalized attributions for AlexNet and $\mathcal{X}$-AlexNet using the attribution methods $\mathcal{X}$-Gradient ($\mathcal{X}$G) and Integrated Gradients (IG). For $\mathcal{X}$-AlexNet the attributions from $\mathcal{X}$-Gradient and IG are almost identical. Additionally, the attribution of the $\mathcal{X}$-AlexNet does not change when scaling the input contrast.

4.4 Homogeneity of $\mathcal{X}$-DNNs

The fundamental difference between $\mathcal{X}$-DNNs and regular DNNs is the nonnegative homogeneity of the former. To show implications on the model and its attributions, we conduct the following experiment. Similarly to Hendrycks *et al.* [10], we reduce the contrast of the ImageNet [22] validation split by multiplying each image with varying factors α and report the top-1 accuracy of AlexNet and the corresponding $\mathcal{X}$-AlexNet. Results can be found in Fig. 2(left). We can observe that decreasing the contrast of the images leads to a strong drop in the accuracy of a regular AlexNet. On the other hand, due to the equivariance to contrast of $\mathcal{X}$-DNNs, the accuracy for $\mathcal{X}$-AlexNet is unaffected, showing improved robustness towards contrast changes. Additionally, to give some qualitative examples, in Fig. 2(right) we plot the attributions for a regular AlexNet and a $\mathcal{X}$-AlexNet for an original image and its corresponding low-contrast version obtained by multiplying the normalized image with $\alpha = 0.3$. Note how for the $\mathcal{X}$-AlexNet, our $\mathcal{X}$-Gradient and Integrated Gradients [31] are identical up to a small approximation error and how reducing the contrast of the images keeps the attribution unchanged up to a scaling factor (not visible due to normalization for display purposes). Additionally, it is noteworthy that the input gradient of $\mathcal{X}$-AlexNet seems to be visually more interpretable than that of the regular AlexNet. We argue that the above observations reflect generally desirable properties and show that $\mathcal{X}$-DNNs behave more predictably with contrast changes than regular DNNs.

5 Conclusion and broader impact

In this work, we consider a special class of efficiently explainable DNNs, for which an axiomatic feature attribution can be computed with only one gradient evaluation. We show that nonnegatively homogeneous DNNs, termed $\mathcal{X}$-DNNs, are efficiently explainable. Moreover, we find that many commonly used architectures can be transformed into $\mathcal{X}$-DNNs by simply removing the bias term of each layer. Our empirical results indicate that this only marginally impairs accuracy. The resulting efficiently computable and axiomatic attributions are particularly well-suited for inclusion into the training process. For example, by enforcing priors on the attributions, we can mitigate dependence on unwanted features and biases induced by the training dataset, which is a major challenge in today's ML systems. Obermeyer *et al.* [19] found evidence that a widely used algorithm in the U.S. health care system contains racial biases that are attributable to biases in the dataset that was used to develop the algorithm. Using our method to generate high-quality explanations that reflect the true behavior of a $\mathcal{X}$-DNN and an appropriate attribution prior, such problems could potentially be resolved (though more research on attribution priors is necessary). However, allowing biases to be controlled by an ML practitioner can also introduce new risks. Just like datasets, humans are also not free of biases [32], which can potentially be reflected in such priors. We as a society need to be careful that this responsibility is not exploited and used for discriminatory or harmful purposes. One way to approach this problem for applications that affect the general public is to introduce an ethical review committee, which assesses whether the proposed priors are legitimate or reprehensible.

References

[1] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *ICLR*, 2018.

[2] M. Ancona, E. Ceolini, C. Öztireli, and M. H. Gross. Gradient-based attribution methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, pages 169–191. Springer, 2019.

[3] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica 2016*, 366(6464):447–453, May 2016. ISSN 0036-8075.

[4] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, 58:82–115, 2020.

[5] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015.

[6] J. Chen, X. Wu, V. Rastogi, Y. Liang, and S. Jha. Robust attribution regularization. In *NeurIPS*2019*.

[7] G. Erion, J. D. Janizek, P. Sturmfels, S. Lundberg, and S.-I. Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, to appear, 2021.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *ICCV*, pages 1026–1034, 2015.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[10] D. Hendrycks and T. G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*2012*, pages 1106–1114.

[12] J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure. *arXiv:1612.08220 [cs.CL]*, 2016.

[13] F. Liu and B. Avci. Incorporating priors with feature attribution on text classification. In *ACL*, pages 6274–6283, 2019.

[14] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, Jan. 2020. ISSN 2522-5839.

[15] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, volume 30, page 3, 2013.

[16] H. W. Miller. Plan and operation of the health and nutrition examination survey. United States–1971-1973. *Vital and Health Statistics. Ser. 1, Programs and Collection Procedures*, (10a):1–46, Feb. 1973. ISSN 0083-2014.

[17] W. J. Murdoch, P. J. Liu, and B. Yu. Beyond word importance: Contextual decomposition to extract interactions from lstms. In *ICLR*, 2018.

[18] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[19] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019. ISSN 0036-8075.

[20] L. Rieger, C. Singh, W. Murdoch, and B. Yu. Interpretations are useful: Penalizing explanations to align neural networks with prior knowledge. In *ICML*, volume 119, pages 8116–8126, 2020.

[21] A. S. Ross, M. C. Hughes, and F. Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *IJCAI*, pages 2662–2670, 2017.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(13):211–252, 2015.

[23] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks: The Official Journal of the International Neural Network Society*, 61:85–117, Jan. 2015. ISSN 1879-2782.

[24] P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H.-G. Luigs, A.-K. Mahlein, and K. Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, Aug. 2020. ISSN 2522-5839.

[25] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv:1605.01713*, 2016.

[26] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In D. Precup and Y. W. Teh, editors, *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR, 2017.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[28] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR*, 2014.

[29] C. Singh, W. J. Murdoch, and B. Yu. Hierarchical interpretations for neural network predictions. In *ICLR*, 2019.

[30] R. Socher, E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *NIPS*2011*, pages 801–809.

[31] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *ICML*, volume 70, pages 3319–3328, 2017.

[32] A. Tversky and D. Kahneman. Judgment under Uncertainty: Heuristics and Biases. *Science*, 185(4157): 1124–1131, 1974. ISSN 0036-8075.

[33] S. Wang, T. Zhou, and J. A. Bilmes. Bias also matters: Bias attribution for deep neural network explanation. In *ICML*, volume 97, pages 6659–6667, 2019.

[34] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, volume 1, pages 818–833, 2014.

[35] H. Zhang, Y. N. Dauphin, and T. Ma. Fixup initialization: Residual learning without normalization. In *ICLR*, 2019.

[36] J. Zhou and O. G. Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, 12(10):931–934, Oct. 2015. ISSN 1548-7105.

[37] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 3.
   (b) Did you describe the limitations of your work? [Yes] See Section 3 (Limitations).
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3.

(b) Did you include complete proofs of all theoretical results? [Yes] See Section 3 and supplemental material.

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code and instructions how to reproduce the main experimental results can be found in the code base in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All the training details are specified in Section 4 and the supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] in Section 4.3. [No] for the ImageNet experiments. As training ImageNet models requires a significant amount of computational power, which negatively affects our environment, and high variance is not expected, we decided to train only one model.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplemental material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See supplemental material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include our code base to reproduce our results.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] We are not curating data and use only publicly available datasets.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We are not curating data and use only publicly available datasets. We use the established ImageNet for training. The results shown in the paper do not contain any identifiable information or offensive content. The NHANES I survey contains no personally identifiable information.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]