

# Fast Axiomatic Attribution for Neural Networks

Paper Authors: Robin Hesse, Simone Schaub-Meyer, Stefan Roth

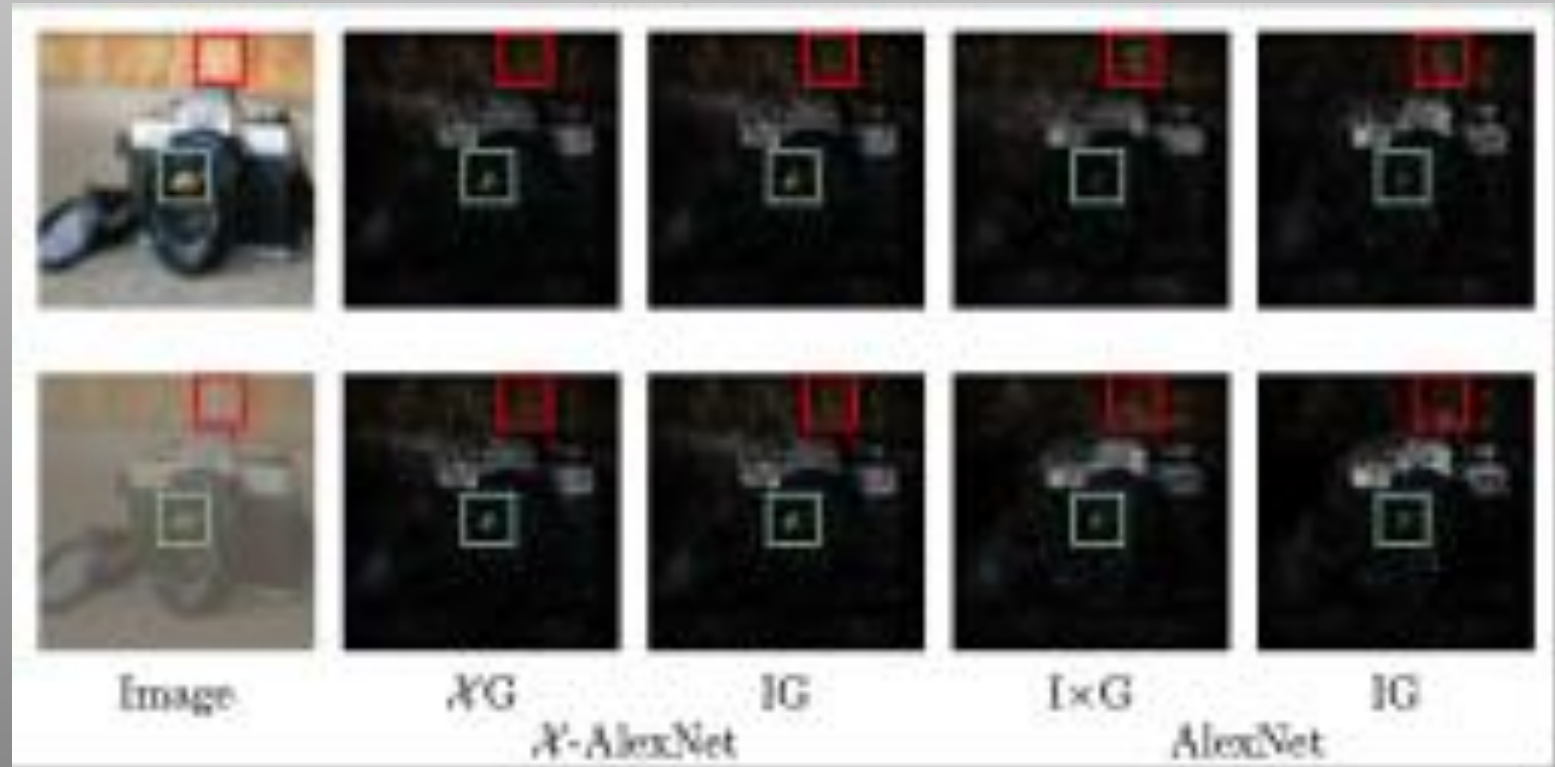


Image reproduced under fair use from <https://arxiv.org/pdf/2111.07668.pdf>

# Motivation for $X$ -DNNs

- Trade off
  - high-quality attributions
    - satisfying axioms
  - computational time/cost
- Goal: Obviate this trade-off
- Search for a class of efficiently axiomatically attributable DNNs
  - only a single forward/backward pass for computing attributions.
- *nonnegatively homogeneous DNNs or  $X$ -DNNs*
  - Constructed from DNNs by removing the bias term.

# Related work

- Two types of attribution methods
- Perturbation-based
  - repeatably perturb individual inputs or neurons to study impact on outcome
  - each perturbation requires a forward pass
    - Computationally inefficient
- Backpropagation-based
  - Back-propagate importance from output to input using gradients or rules
  - Gradient-based e.g. saliency, Input  $\times$  Gradient, IG
    - scale high-dimensional inputs
    - implemented on GPUs
    - applied to any differentiable model
  - Rule-based
    - Layerwise Relevance Propagation (LRP)
      - predefined backpropagation rules for every NN component
    - DeepLIFT relies on a neutral baseline input
      - uses the difference of the normal activation and reference activation of each neuron.

# Axiomatic attributions

- **Sensitivity (a)**
  - for every input and baseline that differ in 1 feature with different predictions,
  - the differing feature should be given a non-zero attribution.
- **Sensitivity (b)**
  - If a DNN does not depend (mathematically) on some variable  $v$ ,
  - then the attribution for  $v$  is 0.
- **Implementation invariance**
  - attributions for 2 functionally equivalent networks are always identical.
- **Completeness**
  - attributions add up to the difference between the DNN output for
    - the input and
    - the baseline.
- **Linearity**
  - attribution of a linearly composed deep network  $a F1 + b F2$
  - is same as the weighted sum of the attributions for  $F1$  and  $F2$  with weights  $a$  and  $b$ .
- **Symmetry preservation**
  - Symmetric variables with identical values receive identical attributions.

# Training using attribution priors

- Training objective formulated as

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

- Here,
  - a model  $F_{\theta}$  with parameters  $\theta$
  - trained on the dataset  $X$ .
  - $L$  is the task loss,
  - $\Omega$  is a scalar-valued loss of the feature attribution  $A$  (the attribution prior)
  - $\lambda$  controls the relative weighting
- IG can be used for  $A$ 
  - but it may involve  $\sim 20$ – $300$  gradient calculations
  - Liu and Avci report 30X increase in training time

# Efficiently axiomatically attributable DNNs

- Given a single DNN output  $F : \mathbb{R}^n \rightarrow \mathbb{R}$ ,
- an input  $x \in \mathbb{R}^n$ ,
- $A(F, x, x') \in \mathbb{R}^n$  is the feature attribution
  - for the prediction at input  $x$  relative to a baseline input  $x'$
  - each element  $a_i$  is the contribution of feature  $x_i$  to the prediction  $F(x)$ .
- Efficiently axiomatically attributable DNNs,
  - only a single forward/backward pass to compute IG
- A DNN  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  is efficiently axiomatically attributable
  - w.r.t. a baseline  $x' \in \mathbb{R}^n$ ,
  - if there exists a closed-form solution of Integrated Gradients  $IG_i(F, x, x')$
  - along the  $i^{\text{th}}$  dimension of  $x \in \mathbb{R}^n$
  - requiring only one forward/backward pass.

# Key Result - I

- For a DNN  $F : \mathbb{R}^n \rightarrow \mathbb{R}$ , there exists closed-form solution of  $\text{IG}_i(F, x, \mathbf{0})$ 
  - w.r.t. the zero baseline  $\mathbf{0} \in \mathbb{R}^n$
  - requiring only one forward/backward pass,
  - 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.** Definition of Integrated Gradients (IG) with baseline  $\mathbf{0}$ :

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



$$F(\alpha x) = \alpha^k F(x)$$

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

# Key Result - II

- *Nonnegatively homogeneous DNN*  $F : \mathbb{R}^n \rightarrow \mathbb{R}$

$$F(\alpha x) = \alpha F(x) \text{ for all } \alpha \in \mathbb{R}_{>0}.$$

- *Any nonnegatively homogeneous DNN is efficiently axiomatically attributable w.r.t. the zero baseline*  $\mathbf{0} \in \mathbb{R}^n$ .
- Proof Sketch: Last slide
- For any  $X$ -DNN  $F : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $X$ -Gradient ( $XG$ ) relative to the zero baseline  $\mathbf{0} \in \mathbb{R}^n$  is defined as

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



# Axioms re-visited

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

Image reproduced under fair use from  
<https://arxiv.org/pdf/2111.07668.pdf>

# Constructing X-DNNs

- Define a regular feedforward *DNN*  $F : \mathbb{R}^n \rightarrow \mathbb{R}^o$ , for an input  $x \in \mathbb{R}^n$ ,
- as a recursive sequence of layers  $l$  that are applied to the output of the respective previous layer:

$$F_l(x) = \begin{cases} \psi_l(\phi_l(W_l F_{l-1}(x) + b_l)) & \text{if } l \geq 1 \\ x & \text{if } l = 0, \end{cases}$$

- with  $W_l$  and  $b_l$  being the weight matrix and bias term for layer  $l$ ,
- $\phi_l$  being the corresponding activation function, and
- $\psi_l$  being the corresponding pooling function.

# Constructing X-DNNs

- Define a regular feedforward *DNN*  $F : \mathbb{R}^n \rightarrow \mathbb{R}^o$ , for an input  $x \in \mathbb{R}^n$ ,

$$F_l(x) = \begin{cases} \psi_l(\phi_l(W_l F_{l-1}(x) + b_l)) & \text{if } l \geq 1 \\ x & \text{if } l = 0, \end{cases}$$

- Can capture VGG, AlexNet, ResNet–type architectures
- fully connected and convolutional layers = Matrix multiplications
- Skip connections = Identity matrix at future layers
- with  $W_l$  and  $b_l$  being the weight matrix and bias term for layer  $l$ ,
- $\phi_l$  being the corresponding activation function, and
- $\psi_l$  being the corresponding pooling function.
  - Both optional i.e. identity matrices
  - Softmax subsumed in loss function

# Constructing X-DNNs - II

- *Assumption for X-DNNs: the activation functions  $\phi_l$  and pooling functions  $\psi_l$  in the model are nonnegatively homogeneous.*
- *Formally, for all  $\alpha \in \mathbb{R}_{\geq 0}$ :  $\alpha\phi_l(z) = \phi_l(\alpha z)$  and  $\alpha\psi_l(z) = \psi_l(\alpha z)$ .*
- *Piecewise linear activation functions with two intervals separated by zero satisfy the above.*
  - ReLU, Leaky ReLU, and PReLU.
- *For  $z = (z_1, \dots, z_n) \in \mathbb{R}^n$ , these activation functions  $\phi_l : \mathbb{R}^n \rightarrow \mathbb{R}^n$  are defined as*

$$\phi_l(z) = (\phi'_l(z_1), \dots, \phi'_l(z_n)) \quad \text{with} \quad \phi'_l(z_i) = \begin{cases} a_{l,1}z_i & \text{if } z_i > 0 \\ a_{l,2}z_i & \text{if } z_i \leq 0. \end{cases}$$

# Constructing X-DNNs - III

- *Linear pooling functions or pooling functions selecting values based on their relative ordering are non-neg homogenous.*
  - Max/min/average pooling, global average pooling, strided convolutions
- *For  $z = (z_1, \dots, z_n) \in \mathbb{R}^n$ , these pooling functions  $\psi_l : \mathbb{R}^n \rightarrow \mathbb{R}^m$  are defined as*

$$\psi_l(z) = (\psi'_l(z'_1), \dots, \psi'_l(z'_m)),$$

- *with*
  - $z'_i$  *being a grouping of entries in  $z$  based on their spatial location*
  - $\psi'_l : \mathbb{R}^m \rightarrow \mathbb{R}$  *being*
    - *linear or*
    - *a selection of a value based on its relative ordering,*
      - *e.g., the maximum or minimum value.*

# Constructing X-DNNs - IV

- Final step: set the bias term of each layer to zero.

Model	Top-5 accuracy (% , $\uparrow$ )			Mean absolute relative difference (% , $\downarrow$ )		
	AlexNet	VGG16	ResNet-50	AlexNet	VGG16	ResNet-50
Regular DNN	<b>79.21</b>	<b>90.44</b>	<b>92.56</b>	79.0	97.8	93.8
$\mathcal{X}$ -DNN	78.54	90.25	91.12	<b>1.2</b>	<b>0.4</b>	<b>0.0</b>

Image reproduced under fair use from  
<https://arxiv.org/pdf/2111.07668.pdf>

# Constructing X-DNNs - V

- *Any DNN satisfying the non-neg homogenous can be transformed into an X – DNN*
  - *by removing the bias term of each layer.*
- *Proof.*
- *A DNN  $F$  with  $L$  layers with all biases  $b_i$  set to 0 can be written as*
- *$F(x) = \psi_L(\phi_L(WL(\dots(\psi_1(\phi_1(W_1x))))))$ .*
- *As all*
  - *pooling functions  $\psi_i$ ,*
  - *activation functions  $\phi_i$ , and*
  - *matrix multiplications  $W_i$  in  $F$*
- *are nonnegatively homogeneous, it follows that*  
$$F(\alpha x) = \alpha F(x) \text{ for all } \alpha \in \mathbb{R}_{\geq 0}.$$

# Contrast-invariant DNNs are X-DNNs

- If a DNN  $F : \mathbb{R}^n \rightarrow \mathbb{R}$ , taking an image  $x \in \mathbb{R}^n$  as input,
- is equivariant *w.r.t.* to the image contrast,
- it is efficiently axiomatically attributable.
  
- Examples:
  - contrast-equivariant DNNs for regression tasks
    - image restoration
    - image super-resolution
  - Assuming contrast equivariance of the logits at the output
    - image classification
    - semantic segmentation



# Experimental Results - I

- Methods
  - Integrated Gradients,
  - random attributions (Random),
  - input gradient attributions (Grad),
  - Expected Gradients (EG), and
  - the new X -Gradient (XG) attribution
- on a regular AlexNet [40] and the corresponding X -AlexNet.
- On par with IG in terms of quality
- Requires 100X less computation

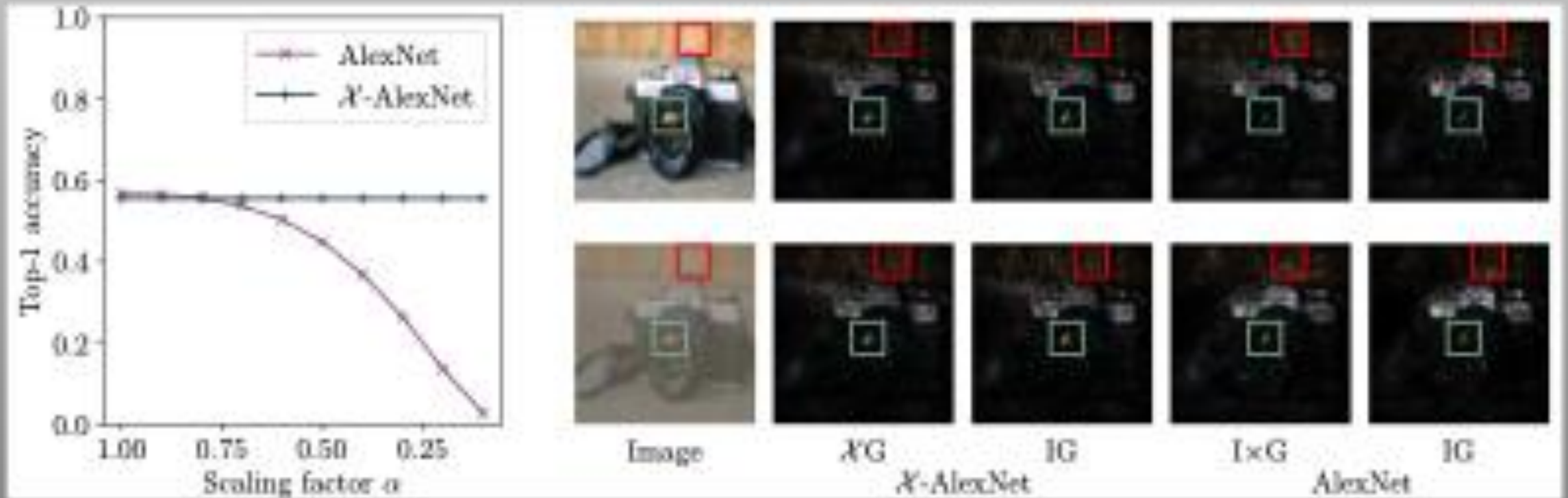
---

Method	AlexNet				$\mathcal{X}$ -AlexNet			
	KPM $\uparrow$	KNM $\downarrow$	KAM $\uparrow$	RAM $\downarrow$	KPM $\uparrow$	KNM $\downarrow$	KAM $\uparrow$	RAM $\downarrow$
IG (128)	<b>7.57</b>	<b>1.67</b>	<b>25.22</b>	<b>11.12</b>	<b>7.38</b>	<b>2.21</b>	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	<b>11.25</b>
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	<b>7.38</b>	<b>2.21</b>	<b>21.83</b>	11.68

---

# Scaling factor vs. Gradients

Image reproduced under fair use from <https://arxiv.org/pdf/2111.07668.pdf>



- (left) *Top-1 accuracy* for AlexNet on ImageNet with decreasing contrast ( $\alpha$ ).
- (right) *Qualitative examples* of normalized attributions AlexNet using
  - X -Gradient ( $\mathcal{X}$  G) resp.
  - Input $\times$ Gradient (I $\times$ G)
  - Integrated Gradients (IG).

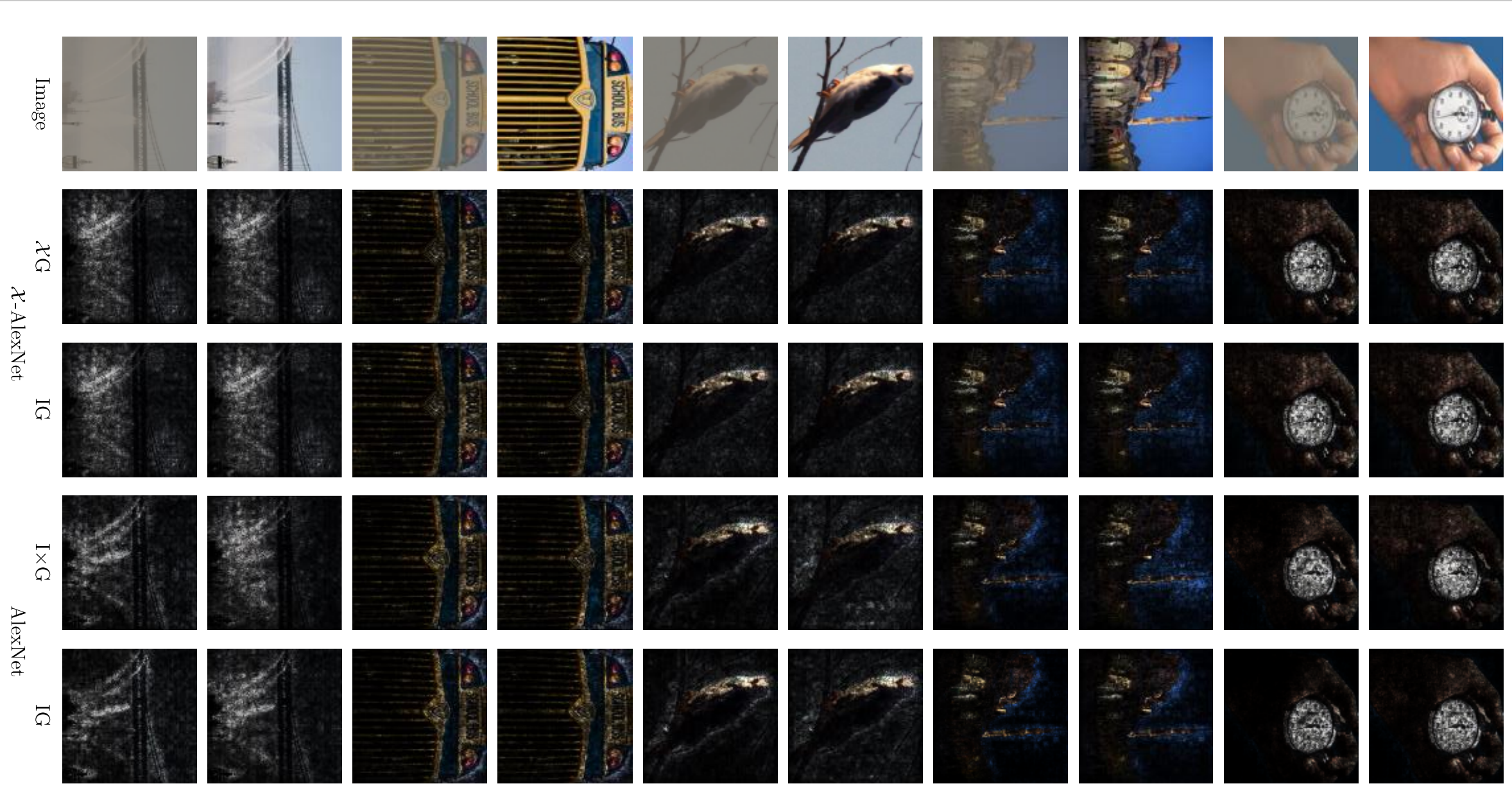


Image reproduced under fair use from

<https://arxiv.org/pdf/2111.07668.pdf>

# Conclusions

- Special class of efficiently axiomatically attributable DNNs
- A single forward/backward pass for axiomatic attributions.
- Nonnegatively homogeneous DNNs (X-DNNs) are efficiently axiomatically attributable
- ResNets, AlexNets, VGGs can be transformed into X -DNNs
  - by simply removing the bias term of each layer
  - a surprisingly minor impact on the accuracy
- Can be included into the training process
  - enable a wide application of IG