Self-Interpretable Model with Transformation Equivariant Interpretation

Anonymous Author(s)
Affiliation
Address
email

Abstract

With the proliferation of machine learning applications in the real world, the demand for explaining machine learning predictions continues to grow especially in high-stakes fields. Recent studies have found that interpretation methods can be sensitive and unreliable, where the interpretations can be disturbed by perturbations or transformations of input data. To address this issue, we propose to learn robust interpretation through transformation equivariant regularization in a self-interpretable model. The resulting model is capable of capturing valid interpretation that is equivariant to geometric transformations. Moreover, since our model is self-interpretable, it enables faithful interpretations that reflect the true predictive mechanism. Unlike existing self-interpretable models, which usually sacrifice expressive power for the sake of interpretation quality, our model preserves the high expressive capability comparable to the state-of-the-art deep learning models in complex tasks, while providing visualizable and faithful high-quality interpretation. We compare with various related methods and validate the interpretation quality and consistency of our model.

1 Introduction

Deep learning (DL) model has been a great success in various domains of applications, including object detection, image classification, etc. However, many applications suffer from the overfitting problem, which is usually due to the lack of various training data. For scenarios with limited data accessible, data augmentation is usually applied to alleviate the overfitting problem. As one of the most simple but effective data augmentation methods, geometric transformation plays an important role in exploring the intrinsic visual structures of image data [42, 33]. Transformation equivariance refers to the property that data representations learned from the model capture the intrinsic coordinates of the entities [15], i.e., transformations on the data will result in the same transformations to the model representations. Building transformation-equivariant DL model is desired in many kinds of applications, such as medical image analysis [10], reinforcement learning [26], etc.

At the same time, with the rapid development of Deep Neural Network (DNN), the performance of DL models has been improved greatly. DL models are even able to surpass human experts in tasks like image recognition [14] and game playing [24]. Although DL models can exert excellent performance in various tasks, the nature of DNN determines that DL models are usually expressed as black boxes. Therefore, DL models can have great performance in complex tasks but lack an explanation of the results [11]. In tasks where the risks of tasks are low, such as adaptive email filtering, the direct deployment of black-box models without reasoning might be acceptable. However, for high-risk decision-making tasks such as disease diagnosis and autonomous vehicles [17], the applied model needs to be more convincing than a black box. On the one hand, by faithfully explaining the model behavior, it can ensure that the end user intuitively understands and trusts the DL model. On the other

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

hand, the explanation of black-box model can provide insights in the relationship between input and output, thereby improving the model design. However, with the rapid growth in computational power, DL models are designed to be more and more complex to meet the performance requirements from in tasks [34], and the most advanced DL models can have billions of trainable parameters [2]. High complexity leads to a complete black box to human beings, which results in a lack of trust in the model. The demand for building more reliable and easy-to-understand DL model is growing rapidly.

Depending on the stages where predictions and interpretations are conducted, the methods can be divided into two opposing categories: self-interpretable models and post-hoc models [29]. Unlike post-hoc models, which generate interpretations to a pre-trained black-box models, self-interpretable models aim to build models that are intrinsically interpretable themselves. The main difference between these two categories is that for post-hoc models, the interpretation and prediction are obtained in two different stages. The interpretation is separately obtained after the the black-box models are trained. Therefore, interpretations obtained from post-hoc models are considered to be more fragile, sensitive, and less faithful to the predictive mechanism [1, 18, 48]. In contrast, self-interpretable models make interpretations at the same time of predictions, thus reveal the intrinsic mechanism of the models, and are thereby preferred by users in high-stakes tasks [38]. Besides, considering how powerful and common transformation can be in data augmentations, it is reasonable to take the robustness of interpretation to transformations into consideration when designing and evaluating the interpretations. Robust interpretation towards transformation implies two requirements: 1) the predictive mechanism indicated by the interpretation should remain the same after transformation (e.g., the highlighted region should remain the same despite the transformation); 2) the location of interpretation should change according to the transformation. These two requirements naturally lead to *transformation equivariance on interpretation*. The transformation-equivariance property will enhance robust and faithful interpretation, where the interpretation is aware of the transformations and preserves the predictive meachanism. This correspondence between transformation equivariance and faithfulness suggests that self-interpretable models may perform better than post-hoc models in transformation awareness given its higher faithfulness. And the experiments also demonstrate this.

Although self-interpretable models surpass post-hoc models in faithfulness and stability, there are non-negligible challenges in building self-interpretable models. First, the interpretations may need additional regularization to be in forms that are rational to human. This process usually involves prior domain knowledge provided by human experts [19, 37]. Besides, since the interpretability is intrinsic, specific constraints are required in the models to ensure the interpretability. The prediction power of such models will be damaged since it is essentially adding constraints to optimization problems. It is acknowledged that the increase of the interpretation quality is likely to decrease performance of prediction results [12, 30]. As a consequence, self-interpretable models are usually less expressive compared with black-box models, which can be interpreted by post-hoc models.

Our Model: In this paper, we develop a transformation-equivariant self-interpretable model for classification tasks. As a self-interpretable model, our method makes predictions and generates interpretations to the predictions at the same stage. In other words, the interpretations are directly involved in the feed-forward prediction process, and are therefore faithful to the final results. We name our method as SITE (Self-Interpretable model with Transformation Equivariant Interpretation). In SITE, we generate data-dependent prototypes for each class and formulate the prediction as the inner product between each prototype and the extracted features. The interpretations can be easily visualized by upsampling from the prototype space to the input data space. Besides, we introduce transformation regularization and reconstruction regularization to the prototypes. The reconstruction regularizer is to regularize the interpretations to be meaningful and comprehensible to human, while the transformation regularizer is to constrain the interpretations to be transformation equivariant. We validate that SITE can present understandable and faithful interpretations, without additional requirement for domain knowledge, and SITE preserves high expressive power in prediction. We summarize the main contributions through this work as:

- To our best knowledge, we are the first to learn transformation equivariant interpretations.
- We build a self-interpretable model SITE with high-quality faithful and robust interpretation.
- Our SITE preserves the high expressive power with comparable or better accuracy than related black-box models.
- We propose *self-consistency score*, a new quantitative metric for interpretation methods. It quantifies the robustness of interpretation by measuring the consistency of interpretations to geometric transformations.

2 Related Work

Machine learning interpretation can have different goals, such as attribution [22], interpretable clustering [27], interpretable reinforcement learning [28], disentanglement [41], etc. Our method lies in the attribution category, thus we mainly review the related interpretation methods for attribution.

Attribution methods target at identifying the contribution of different elements in the prediction. Based on if the prediction and interpretation is obtained in the same stage, the methods can be divided into post-hoc interpretation and self-interpretable methods.

For post-hoc interpretation, the prediction results are obtained by a black-box model while the interpretation is obtained separately to explain the predictive mechanism of the black box. Among the different post-hoc interpretation techniques, backpropagation methods [56, 39, 50, 43, 35, 5], trace from the output back to the input to determine how the different elements in the input contribute to the prediction result. Class Activation Mapping (CAM) [56] visualizes the feature importance in convolutional neural network by mapping the weights in last fully connected layer to the input layer via upsampling. Score-CAM learns weighting scores for the activation maps by integrating the increase in confidence for an improved CAM visualization [50]. While for approximation methods [36], the interpretation is obtained by fitting an interpretable model to the black-box prediction around the target sample. Deconvolution methods [54] interpret a convolutional neural network via image deconvolution. For perturbation-based interpretation [31, 13], the methods interpret the feature importance by imposing perturbation to certain feature and checking the changes in the output. Moreover, Shapley values [22] have been used to calculate the feature importance due to the nice properties preserved by Shapley values. For post-hoc interpretation, since the prediction and interpretation are separated, the prediction can be obtained by a highly expressive black-box model to handle complex tasks. However, the post-hoc interpretation may not capture the true predictive mechanism of the black box and is less reliable [1, 18].

Different from post-hoc interpretation, self-interpretable models target at building white boxes that are intrinsically interpretable, which are able to conduct prediction and interpretation at the same time. A self-interpretable model preserves faithful interpretation since the model itself is a white box. However, the constraints on self-interpretation can cause limitation in the expressive power thus make sacrifice in prediction performance. For example, in order to build an interpretable decision sets [20], there is constraint on the number of rules for the sake of interpretation, which restrict its application to complex tasks. Recent models propose to build self-interpretable neural network [3, 4]. NAM [3] combines neural network with additive model for facilitate the self-interpretation via component function. Moreover, attention models have been widely used to build interpretable predictions [25]. However, recent works find that the interpretation via attention weights can fail to identify the important representations [40].

Different from the related works, our goal is to build a self-interpretable model that learns faithful interpretation and has high expressive power. To the best of our knowledge, we are the first to learn transformation-equivariant interpretations to ensure faithful and robust interpretation. Previously the transformation equivariance property has been studied in prediction via deep neural network. Many recent studies integrate the transformation equivariance in object detection, with the goal of building convolutional neural network that is equivariant to image translations. The models learn features equivariant to translation and rotation [52, 51, 9], 3D symmetries [49], and build sets with symmetric elements for general equivariant tasks [23].

3 Building Transformation Equivariant Interpretation in SITE

In this section, we introduce the structure of SITE. For notations, all normal lowercase letters stand for numbers; all bold lowercase letters stand for tensors; all normal uppercase letters stand for operations (including functions, networks, etc); and all curly uppercase letters denote sets and families. Additionally, all Greek letters will be explained when they are introduced in context.

3.1 Formulation

For images classification tasks, suppose that $\mathbf{x} \in \mathbb{R}^p$ denotes the input image, one-hot vector $\mathbf{y} \in \{0,1\}^c$ denotes the label, and $\hat{\mathbf{y}} \in [0,1]^c$ denotes the predicted class probabilities. We clarify that p is the product of the number of channels, the width, and the height of the image $\mathbf{x}$, while c

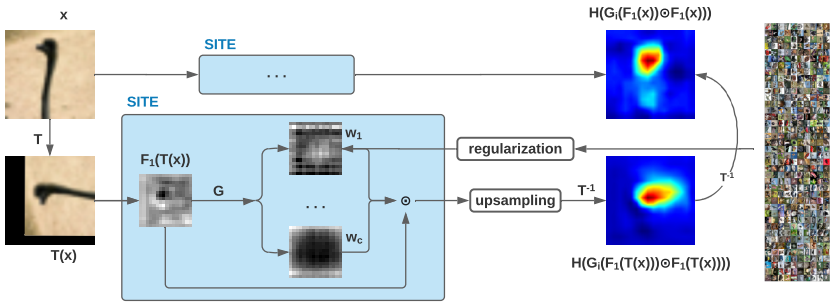


Figure 1: An illustration of our SITE model. SITE can take both original image $\mathbf{x}$ and transformed image $T(\mathbf{x})$ as input. The input is first fed to the feature extractor F_1, then SITE generate c prototypes $\mathbf{w}_1, \cdots, \mathbf{w}_c$ through generator G. Finally, both the prediction and interpretation come from the Hadamard product between the latent representation $F_1(T(\mathbf{x}))$ and each prototype. The interpretation is obtained by upsampling the Hadamard product, and the prediction is obtained by the element-wise summation of it. SITE ensures transformation equivariant interpretation by constraining on the interpretations before and after transformation.

denotes the number of classes. Generally, a traditional classifier $F : \mathbf{x} \mapsto \hat{\mathbf{y}}$ can be decomposed into $F = F_2 \circ F_1$ with a feature extractor F_1 and a simple classifier F_2, where $F_1 : \mathbf{x} \mapsto \mathbf{z}$ and $F_2 : \mathbf{z} \mapsto \hat{\mathbf{y}}$. Here $\mathbf{z} \in \mathbb{R}^d$ denotes the extracted latent representations of $\mathbf{x}$, and usually has a lower dimension ($d < p$). The extractor F_1 usually consists of convolutional neural networks or ResNet structures, and F_2 consists of fully connected layers. The goal is to minimize the classification loss

$$\min_{F=F_2\circ F_1} \mathbb{E}_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}} L_{ce}(F(\mathbf{x}), \mathbf{y}) \,, \tag{1}$$

where $\mathcal{X}, \mathcal{Y}$ are the input data set and the target set, and L_{ce} denotes the cross-entropy loss function.

Traditional methods in (1) is not intrinsically interpretable w.r.t. the contribution of features in $\mathbf{x}$ to the prediction $\hat{\mathbf{y}}$. In order to address this, in SITE we build a generative model $G = [G_1, \cdots, G_c]$ that maps the latent representation $\mathbf{z}$ to c *prototypes* $\{\mathbf{w}_i\}_{i=1}^c \subset \mathbb{R}^d$, where $\mathbf{w}_i = G_i(\mathbf{z})$. Each prototype corresponds to a specific class. We formulate the final prediction as the inner product of the latent representation $\mathbf{z}$ and each prototype $\{G_i(\mathbf{z})\}_{i=1}^c$. That is,

$$\hat{\mathbf{y}} = \sigma(G(\mathbf{z})^\top \mathbf{z}) = \sigma\left(\left[G_1(\mathbf{z})^\top \mathbf{z}, G_2(\mathbf{z})^\top \mathbf{z}, \ldots, G_c(\mathbf{z})^\top \mathbf{z}\right]^\top\right) \,, \tag{2}$$

where σ is softmax activation. The prediction $\hat{\mathbf{y}}$ is the similarity between the latent representation $\mathbf{z}$ and the generated prototype $G_i(\mathbf{z})$. Thus we have the modified classification loss

$$L_{cls} = \mathbb{E}_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}} L_{ce}\left(\sigma\big(G(F_1(\mathbf{x}))^\top F_1(\mathbf{x})\big), \mathbf{y}\right) \,. \tag{3}$$

Note that we formulate the prediction result for class i as $\hat{y}_i = \sigma(G_i(\mathbf{z})^\top \mathbf{z}) = \sigma(\mathbf{w}_i^\top \mathbf{z})$. According to our formulation of $\hat{\mathbf{y}}$ in (2), we can explicitly capture the contribution of elements in $\mathbf{z}$ to the final prediction by the Hadamard product between $\mathbf{w}_i$ and $\mathbf{z}$ as $\hat{\mathbf{w}}_i = \mathbf{w}_i \odot \mathbf{z}$. Naturally we take $\hat{\mathbf{w}}_i$ as the *interpretation* of the i-th prediction result, such that the contribution of different elements to each prediction result is clear from the interpretation. For instance, $\hat{w}_i^j, i = 1, \cdots, c, j = 1 \cdots, d$ denotes the contribution of the j-th element of $\mathbf{z}$ to the i-th class. Based on our formulation of $\hat{\mathbf{w}}$, the interpretation from our model preserves the **completeness** property. That is, the summation of the importance scores of all features equals the prediction result. This is introduced as Proposition. 1 in [46], and also known as **local accuracy** in [22]. This property assures that the interpretation is related to the corresponding prediction in the numerical sense.

The interpretation obtained from optimizing L_{cls} ensures faithful interpretations (i.e., which shows the true predictive mechanism of the model), but may not ensure that the interpretation is human-understandable. In order to build high-quality interpretation, we propose to regularize the prototypes $G_i(\mathbf{x}), i = 1, \cdots, c$ with the following. For an input image $\mathbf{x}$, we enforce each generated prototype $G_i(F_1(\mathbf{x}))$ to be similar to its corresponding class's latent representation $F_1(\mathbf{x}_i)$:

$$L_1 = \sum_{i=1}^{c} \mathbb{E}_{\mathbf{x}\in\mathcal{X},\mathbf{x}_i\in\mathcal{X}_i} L_{bce}\left(G_i(F_1(\mathbf{x})), F_1(\mathbf{x}_i)\right) \,, \tag{4}$$

where L_{bce} denotes the binary cross-entropy loss, and $\mathcal{X}_i \subset \mathcal{X}$ denotes the set of input data that belongs to the i-th class.

In addition, we propose to regularize on the *transformation equivariance* property of interpretation from our SITE model. Let T_β denote pre-defined parametric transformations as described in [16]. We want SITE to learn interpretations that are equivariant to the transformations. Here $\beta \sim \mathcal{B}$ denotes the random sampled parameters from a pre-defined parameter distribution $\mathcal{B}$. During training process, we suppose that the random transformation T_β is known and we can have access to its inverse T_β^{-1}. In the feed-forward process of training, we first transform the input image $\mathbf{x}$ by random sampled transformations $T_\beta(\mathbf{x})$, $\beta \sim \mathcal{B}$, then feed it to the model $G \circ F_1$. So the prediction result on the transformed image is $G(F_1(T_\beta(\mathbf{x})))^\top F_1(T_\beta(\mathbf{x}))$. The prototypes of the transformed input image $G(F_1(T_\beta(\mathbf{x})))$ can be transformed back by the inverse transformation T_β^{-1}. We build the reconstruction loss between the transformed prototypes $T_\beta^{-1}\Big(G_i\big(F_1(T_\beta(\mathbf{x}))\big)\Big), i = 1, \cdots, c$ and the latent representations of $\mathbf{x}_i \in \mathcal{X}_i, i = 1, \cdots, c$, respectively:

$$L_2 = \sum_{i=1}^{c} \mathbb{E}_{\mathbf{x}\in\mathcal{X}} L_{bce}\left(T_\beta^{-1}(G_i(F_1(T_\beta(\mathbf{x})))), G_i(F_1(\mathbf{x}))\right) . \tag{5}$$

By integrating the equivariance property (5) with transformation $T_\beta, \beta \in \mathcal{B}$ in the interpretation regularization in (4), we propose the transformation loss as:

$$L_{trans} = \sum_{i=1}^{c} \mathbb{E}_{\mathbf{x}\in\mathcal{X},\mathbf{x}_i\in\mathcal{X}_i} L_{bce}\left(T_\beta^{-1}(G_i(F_1(T_\beta(\mathbf{x})))), F_1(\mathbf{x}_i)\right) . \tag{6}$$

Hence, we propose the objective of SITE with classification loss and transformation loss as follows:

$$\begin{aligned}\min_{G,F_1} \quad & \mathbb{E}_{\beta\sim\mathcal{B}}\Big[\mathbb{E}_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathcal{Y}} L_{ce}\Big(\sigma\big(G(F_1(T_\beta(\mathbf{x})))^\top F_1(T_\beta(\mathbf{x}))\big), \mathbf{y}\Big) + \\ & \lambda \sum_{i=1}^{c} \mathbb{E}_{\mathbf{x}\in\mathcal{X},\mathbf{x}_i\in\mathcal{X}_i} L_{bce}\big(T_\beta^{-1}(G_i(F_1(T_\beta(\mathbf{x})))), F_1(\mathbf{x}_i)\big)\Big] ,\end{aligned} \tag{7}$$

where λ is a hyper-parameter which balances the training paces between the classification loss and the transformation loss. The first term in the objective (7) ensures a transformation-aware classifier, while the second term ensures transformation-equivariant interpretations. In practice, the expectation over $\mathbf{x}_i \in \mathcal{X}_i$ and the expectation over $\mathcal{B}$ can be properly approximated by Mont Carlo sampling.

3.2 Visualization Methods

In the previous subsection, we obtain the self-interpretable model $G \circ F_1$, and the corresponding interpretation $\hat{\mathbf{w}}_i$ for input $\mathbf{x}$. However, since the interpretations $\hat{\mathbf{w}}_i \in \mathbb{R}^d$ are not in the original image space, direct visualization of $\hat{\mathbf{w}}_i$ will be less meaningful.

Notice that the interpretations $\hat{\mathbf{w}}_i$ are approximations of the output space of the feature extractor F_1, it is natural to visualize it by visualizing $H(\hat{\mathbf{w}}_i)$, where $H : \mathbb{R}^d \to \mathbb{R}^p$ is an approximated inverse of F_1. And since F_1 is based on convolutional neural networks, a simple but judicious choice for H would be the bilinear upsampling function. On the one hand, the output space of F_1 will preserve the relative relationship between features. And on the other hand, the Lipschitz continuity of H can preserve all the intrinsic properties in $\hat{\mathbf{w}}_i$. Finally the interpretation $\hat{\mathbf{w}}_i$ is visualized in the original space of the input images by overlaying on the input $\mathbf{x}$ as a heatmap.

3.3 Transformation Self-Consistency Scores

In order to measure the transformation equivariance of a interpretation method properly, we propose a numerical metric, namely the *self-consistency score*. It measures the self-consistency [47] of an attribution interpretation method. For a given input data $\mathbf{x}$ and a parameterized transformation T_β, let $I(\mathbf{x})$ denote the interpretation of $\mathbf{x}$, then the self-consistency score $v_\mathcal{X}(I)$ is defined as the cosine similarity between the transformation of the interpretation to $\mathbf{x}$ and the interpretation to the transformed images $T_\beta(\mathbf{x})$ as $v_\mathcal{X}(I) = \mathbb{E}_{\beta\sim\mathcal{B}}\mathbb{E}_{\mathbf{x}\in\mathcal{X}} S(T_\beta(I(\mathbf{x})), I(T_\beta(\mathbf{x})))$, where $S(\cdot,\cdot)$ is the cosine similarity. The expectation on the transformation family $\mathcal{T}_\mathcal{B}$ is approximated by Monte Carlo sampling method. However, note that in practice $T_\beta(I(\mathbf{x}))$ transforms the interpretation directly and will introduce zero padding in the corner of interpretation heatmaps. $I(T_\beta(\mathbf{x}))$ transforms the input data before the prediction so that the interpretation is not padded. To eliminate the influence from the padded area, we introduce a transformation mask $\mathbf{m}_\beta \in \{0,1\}^p$, where $\mathbf{m}_\beta^i = 0$ for the padding area of $T_\beta(\mathbf{x})$, and $\mathbf{m}_\beta^i = 1$ otherwise. Thus the self-consistency score is calculated by

$$\hat{v}_\mathcal{X}(I) = \mathbb{E}_{\beta\sim\mathcal{B}}\mathbb{E}_{\mathbf{x}\in\mathcal{X}} S\big(\mathbf{m}_\beta \odot T_\beta(I(\mathbf{x})), \mathbf{m}_\beta \odot I(T_\beta(\mathbf{x}))\big) . \tag{8}$$

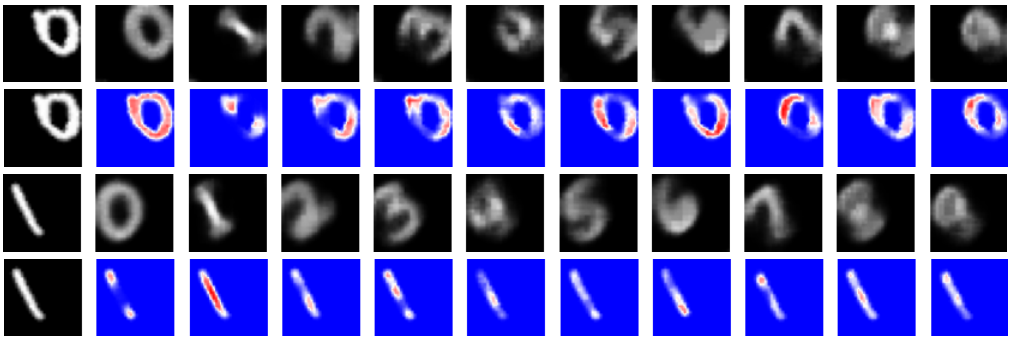


Figure 2: Interpretation results from SITE on digit 0 and 1 on MNIST. For each digit, the first column shows the randomly transformed images. The following $c = 10$ columns are the prototypes $\{\mathbf{w}_i\}_{i=1}^{c}$ (in the first and third rows), and interpretation heatmaps $\{\mathbf{w}_i \odot \mathbf{x}\}_{i=1}^{c}$ (the second and fourth rows) that indicate the contribution of pixels in the image to the prediction of a certain class.

4 Experiments

In this section, we conduct experiments on image classification tasks with and without transformations. The experiment results demonstrate the high-quality interpretations and the validity of SITE. All experiments are conducted @ NVIDIA Dual RTX5000 GPUs with the Intel Xeon W-2145 CPU and NVIDIA Dual RTX6000 GPUs with the Intel i9-9960X CPU.

4.1 Experimental Setup

First, we define the family of geometric transformations to which we want SITE to be equivariant as constrained parametric affine transformations [16]: $\mathcal{T}_{\mathcal{B}} = \{T_\beta : \beta \sim \mathcal{B}\}$, where $\beta \in \mathbb{R}^6$. And $\mathcal{B}$ is defined to constrain the affine transformations to be compositions of rotations in range $[-\pi/2, \pi/2]$ and translation in range $[-h/2, h/2]$, where h denotes the width and the length of input data.

During experiments, the model configurations can be divided into two different settings according to the complexity of dataset. For MNIST dataset, as it is simple, and the intrinsic structures are distinct by pixels among classes, the structure of SITE degenerates from $G \circ F_1$ to G by setting F_1 to be the identical operator. That is, $\mathbf{z} = F_1(\mathbf{x}) = \mathbf{x}$. Correspondingly, the generator G instead maps input $\mathbf{x}$ to its prototypes $G_i(\mathbf{x}), i = 1, \cdots, c$. Hence the structure of SITE is built to be an autoencoder-based structure, where there are c parallel decoders. As for CIFAR-10 dataset, due to the needs for upsampling in visualization, the CIFAR-10 data is resized to 128×128. The feature extractor F_1 is built based on ResNet-18 [14]. Here $F_1 : \mathbb{R}^{3\times128\times128} \to \mathbb{R}^{10\times16\times16}$. And for the generator G, it consists of $c = 10$ parallel autoencoders, such that $G_i : \mathbb{R}^{10\times16\times16} \to \mathbb{R}^{10\times16\times16}$. Both MNIST and CIFAR-10 datasets are split into the training and validation sets by default. And all presented examples are from the validation sets. Besides, in order to balance the classification loss and the transformation loss we set the scalar factor to be $\lambda = 5$ throughout the training.

4.2 Experiments on MNIST

First, we implement SITE on MNIST dataset. Since SITE $G \circ F_1$ shares the same backbone structure F_1 with the traditional classifier $F = F_2 \circ F_1$, we would like to clarify that SITE does not sacrifice prediction power for interpretability. Please refer to Sec. 4.4 for more details.

The interpretations of SITE on MNIST dataset are shown in Fig. 2. Here we present the interpretation for two digits, 0 and 1. It is decided by the Hadamard product that the interpretations is essentially the pixel-wise similarities between the input digit $\mathbf{x}$ and prototype $\mathbf{w}_i$. The interpretation to each prototype can be treated as how and where do the prototype $\mathbf{w}_i$ and $\mathbf{x}$ look similar. Therefore, $\mathbf{x}$ will be classified to the class where the prototype is the most similar to the input data. Besides, we can observe that the interpretations of SITE preserves good transformation equivariance property thanks to the transformation regularization. The interpretations are transformed automatically with the input data, while preserving the shape of the highlighted region.

4.3 Experiments on CIFAR-10

Interpretations of SITE

For CIFAR-10 dataset, input $\mathbf{x}$ is fed to the feature extractor F_1. Then the generator G takes the latent representation $\mathbf{z}$ as input and generate the $c = 10$ data-dependent prototypes $\{\mathbf{w}_i\}_{i=1}^{c}$. Then

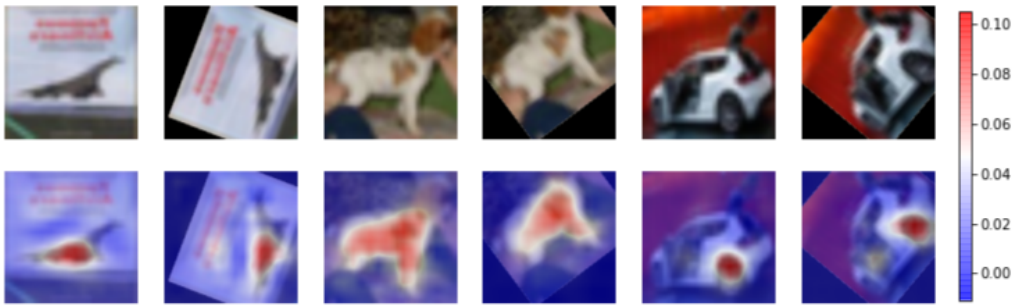


Figure 3: Illustration of the interpretations of SITE on CIFAR-10 dataset. In the first row, there are totally 3 sampled images including plane, dog, and car (odd columns), along with their random affine transformed versions (even columns). In the second row, interpretations plotted in Red White Blue (RWB) heatmaps are overlaid on corresponding original images.

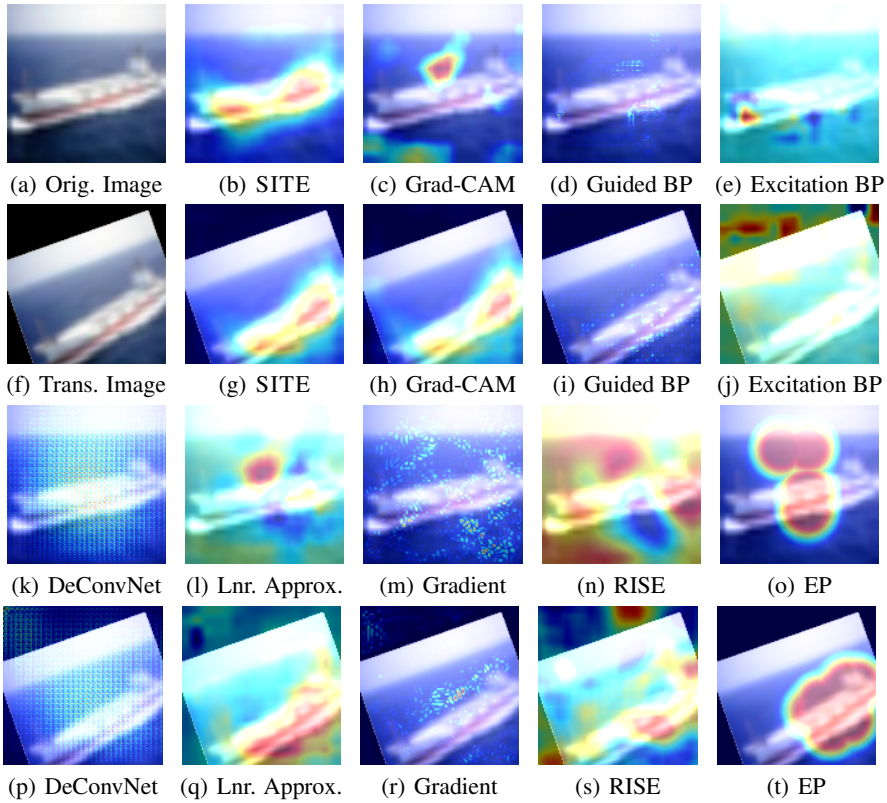
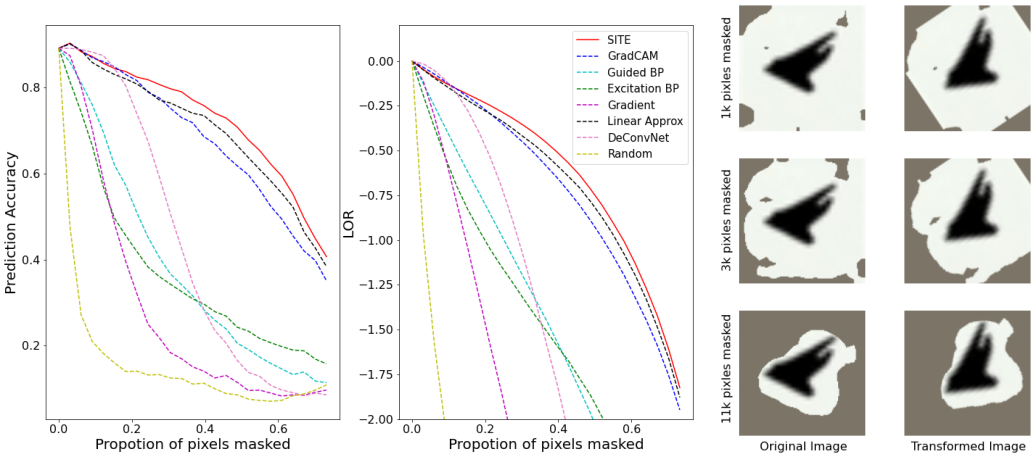


Figure 4: CIFAR-10 comparisons among SITE and eight different post-hoc interpretation methods. The first and the third rows (a)-(e), (k)-(o) show the interpretation results on the original image, while the second and the last rows (f)-(j), (p)-(t) show the interpretation results on the transformed image. Note that the model is trained on **transformed images**, thus all interpretations on the transformed images are relatively reasonable. However, most interpretations are highly disturbed on untransformed image, while SITE preserves the most transformation equivariant interpretations.

the visualizable interpretation of the input $\mathbf{x}$ is defined by $H(\mathbf{w}_{i'} \odot \mathbf{z})$, where H is the bilinear upsampling, and $i' = \arg\max_{1 \leq i \leq c} \mathbf{w}_i^\top \mathbf{z}$ is the predicted class for input $\mathbf{x}$.

The interpretation results are of SITE on CIFAR-10 are shown in Fig. 3. Here we sample three images of plane, dog, and car for demonstrations. Each image is transformed by a constrained affine transformation $T_\beta \in \mathcal{T}_\mathcal{B}$ that sampled independently. And in the bottom row, we overlay the interpretations $H(\mathbf{w}_{i'} \odot \mathbf{z})$ on corresponding images. It can be found clearly that SITE successfully highlights the main parts of objects on sampled images in a comprehensible way to humans. For



(a) Mask-k Experiments (b) Masked images

Figure 5: (a) The trend of accuracy (left) and the trend of log-odds ratio (right) with various proportions of pixels masked; (b) Sampled masked images where 1k (top), 3k (middle), and 11k (bottom) pixels (out of 16.4k pixels) are masked. The left column are the masked original images, and the right column are the transformed masked images.

Table 1: Self-consistency scores $\hat{v}_{\mathcal{X}}(I) \in [-1, 1]$ of interpretation methods. A higher score indicates better self-consistency. The first row is the scores for SITE, and the second row is the scores for the backbone model (ResNet-18). Both models are trained on transformed data. Perturbation methods RISE and EP are omitted in this experiment due to the inefficient computation.

I	SITE	Grad-CAM	Guided BP	Excitation BP	Gradient	Linear Approx.	DeConvNet
$v_{\mathcal{X}}$ (SITE)	**0.9102**	0.8929	0.7875	0.0047	0.7122	0.8732	0.8585
$v_{\mathcal{X}}$ (backbone)	-	0.8637	0.8171	0.0423	0.6852	0.4372	0.7682

instance, in the dog image, SITE highlights the silhouette of the dog in both transformed and untransformed images. And comparing the odd columns and the even columns, it's clear that the interpretability of SITE preserves great self-consistency during transformations [47]. This can be treated as the robustness to transformations. Please refer to the Appendix for more examples. Besides, we would like to clarify that SITE does not sacrifices expressive power for interpretations. We take ResNet-18 classifier as the benchmark, since SITE takes the same structure as feature extractor. Given the same transformation family $\mathcal{T}_{\mathcal{B}}$, SITE and ResNet-18 backbone model achieve comparable validation accuracy of 89% on randomly transformed images. And on untransformed images, SITE even demonstrate higher expressiveness. Please refer to Sec. 4.4 for more details.

Comparison with Post-Hoc Methods

We carry out comparison experiments with various attribution methods that interpret feature contributions to the prediction results. The comparing methods include: back-propagation methods such as Grad-CAM [39], excitation back-propagation [55], guided back-propagation [45], gradient [44], DeConvNet [54], and linear approximation. And also there are perturbation methods such as randomized input sampling (RISE) [31] and extremal perturbation (EP) [13]. The various comparing methods are implemented through TorchRay toolkit [13]. To illustrate the comparison results consistently, we use heatmaps of the same settings to visualize the interpretations of all methods. Since the interpretations of different models are obtained in very different ways, the visualizations of them are performed separately. Hence the heatmaps only demonstrate the relative importance of pixels within each interpretation itself. The comparison of the interpretation results are shown in Fig. 4, where we present the interpretations to the predictions of the given image, respectively. First, we clarify that for the sake of consistency, all post-hoc interpretations shown in Fig. 4 are obtained by applying the post-hoc interpretation models mentioned above to SITE. Since SITE is trained on the transformed dataset, all post-hoc interpretations are reasonable to the transformed image. However, their interpretations to the untransformed image are affected by the transformation. SITE does the best in capturing the main body of the ship in the untransformed image. It also preserves the best self-consistency. In fact, according to the self-consistency scores $\hat{v}_{\mathcal{X}}(I)$ over the whole validation set of CIFAR-10, SITE outperforms all post-hoc methods, and is thereby more robust to transformations.

Table 2: Accuracy comparison among self-interpretable models. We implement SITE on different backbone models and show the performance of the black-box backbone in parathensis.

	Decision Tree	Random Forest	Logistic Regression	XGBoost	NAM	SITE-CNN (Blackbox CNN)	SITE-ResNet (Blackbox ResNet)
MNIST	0.852	0.941	0.912	0.912	0.935	**0.988** (0.981)	-
CIFAR-10	0.362	0.467	0.373	0.405	0.370	**0.840** (0.828)	**0.892** (0.862)

The comparison of self-consistency scores are shown in Table 1. Due to the inefficient computation of perturbation methods, here we omit the calculations of RISE method and extremal perturbation method. For completeness, We also include the self-consistency scores of the post-hoc methods on the backbone model (ResNet-18). It is also trained on the transformed training set.

Finally, we also carry out the mask-k-pixels experiments [7] to demonstrate the equivariance of SITE as a self-interpretable model. This experiment is implemented by masking k pixels of the input data based on the interpretations provided. For each interpretation models, we obtain a series of masked subset of the original dataset based on the interpretations. Here we sample the first 1000 images from the validation set of CIFAR-10, and perform the mask-k-pixels experiments on all interpretation methods mentioned above except for the two perturbation methods. Besides, we also add the case where pixels are randomly masked. In order to demonstrate the transformation equivariance, here we mask the least k important pixels according to the interpretations to untransformed images, and feed the random transformations of the masked images to the classifier. We present the trend of the prediction accuracy and the log-odds ratio (LOR) of the predicted logits to the true classes in Fig. 5(a). It is expected that the more slowly a curve drops, the better equivariance the interpretation possesses. Hence, it can be found that SITE outperforms all other post-hoc interpretation methods. Besides, we observe that when very few pixels ($< 20\%$) are masked, the decrease of SITE is almost negligibly faster than some post-hoc methods. We explain this phenomenon by presenting a typical example in Fig. 5(b). Generally the least important pixels are located at the corners, therefore, those masks are eliminated when the corners are hidden after the transformation, as shown in the top two images in Fig. 5(b). This results in almost no mask at the beginning. As the proportion of masked pixels increase, this phenomenon is gradually alleviate, as shown in the other four images in Fig. 5(b). Furthermore, we validate the faithfulness of SITE compared with post-hoc methods on Benchmarking Attribution Methods (BAM) dataset [53]. Please refer to the Appendix for details.

4.4 Expressiveness

Since there's an inevitable trade-off between expressiveness and interpretability, most of the existing self-interpretable models have relatively low accuracy on image dataset such as MNIST and CIFAR. Here we compare the expressiveness of SITE and existing self-interpretable models including Decision Tree, Random Forest, Logistic Regression, XGBoost [8], and Neural Additive Model (NAM) [3]. The results of XGBoost are reported in [32]. We include SITE with backbones of different level of complexity to demonstrate the scalability of SITE. The CNN backbone has 3 layers for MNIST and 7 layers for CIFAR-10, while the ResNet backbone is the ResNet-18 used in all previous experiments. The backbone models included share the same structures and the same (transformed) training data as the corresponding SITE in feature extraction. We clarify that here the test is performed on the *untransformed* validation set. As shown in Table 2, SITE outperforms all other self-interpretable models by a large margin. It has even higher expressiveness than the backbone model. We give this credit to the regularization to the self-interpretations [6].

5 Conclusions

In this paper, we propose SITE model as a self-interpretable model with transformation-equivariant interpretations. We focus on the robustness and self-consistency of the interpretations to geometric transformations. Apart from the transformation equivariance, as a self-interpretable model, SITE has comparable expressive power as the benchmark black-box classifiers, while being able to present faithful and robust interpretations with high quality. However, it is worth noticing that although applied in most of the CNN visualization methods, the bilinear upsampling approximation is a rough approximation. It can only provide interpretations in the form of heatmaps (instead of pixel-wise). It remains an open question whether such interpretations can be direct to the input space (as shown in the experiments of the MNIST dataset). Besides, we consider the translation and rotation transformations in our model. In future work we will explore the robust interpretations to more complex transformations such as scaling and distortion.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[2] Gabriel Goh Scott Gray Aditya Ramesh, Mikhail Pavlov. Dall·e: Creating images from text, 2021.

[3] Rishabh Agarwal, Nicholas Frosst, Xuezhou Zhang, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *arXiv preprint arXiv:2004.13912*, 2020.

[4] David Alvarez-Melis and Tommi S Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7786–7795, 2018.

[5] 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, 2015.

[6] Akhilan Boopathy, Sijia Liu, Gaoyuan Zhang, Cynthia Liu, Pin-Yu Chen, Shiyu Chang, and Luca Daniel. Proper network interpretability helps adversarial robustness in classification. In *International Conference on Machine Learning*, pages 1014–1023. PMLR, 2020.

[7] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*, 2018.

[8] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[9] Xiuyuan Cheng, Qiang Qiu, Robert Calderbank, and Guillermo Sapiro. Rotdcf: Decomposition of convolutional filters for rotation-equivariant deep networks. In *International Conference on Learning Representations*, 2018.

[10] Benjamin Chidester, Tianming Zhou, Minh N Do, and Jian Ma. Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, 35(14):i530–i537, 2019.

[11] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[12] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019.

[13] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2950–2958, 2019.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, pages 44–51. Springer, 2011.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.

[17] Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, pages 2942–2950, 2017.

[18] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.

[19] Isaac Lage, Andrew Slavin Ross, Been Kim, Samuel J Gershman, and Finale Doshi-Velez. Human-in-the-loop interpretability prior. *arXiv preprint arXiv:1805.11571*, 2018.

[20] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1675–1684, 2016.

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

[22] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774, 2017.

[23] Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements. In *International Conference on Machine Learning*, pages 6734–6744. PMLR, 2020.

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[25] Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. Towards transparent and explainable attention models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4206–4216, 2020.

[26] Arnab Kumar Mondal, Pratheeksha Nair, and Kaleem Siddiqi. Group equivariant deep reinforcement learning. *arXiv preprint arXiv:2007.03437*, 2020.

[27] Tom Monnier, Thibault Groueix, and Mathieu Aubry. Deep transformation-invariant clustering. In *Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[28] Alex Mott, Daniel Zoran, Mike Chrzanowski, Daan Wierstra, and Danilo J Rezende. Towards interpretable reinforcement learning using attention augmented agents. *arXiv preprint arXiv:1906.02500*, 2019.

[29] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[30] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.

[31] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[32] Natalia Ponomareva, Thomas Colthurst, Gilbert Hendry, Salem Haykal, and Soroush Radpour. Compact multi-class boosted trees. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 47–56. IEEE, 2017.

[33] Guo-Jun Qi, Liheng Zhang, Feng Lin, and Xiao Wang. Learning generalized transformation equivariant representations via autoencoding transformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[34] Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W Crandall, Nicholas A Christakis, Iain D Couzin, Matthew O Jackson, et al. Machine behaviour. *Nature*, 568(7753):477–486, 2019.

[35] Sylvestre-Alvise Rebuffi, Ruth Fong, Xu Ji, and Andrea Vedaldi. There and back again: Revisiting backpropagation saliency methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8848, 2020.

[36] 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*, pages 1135–1144, 2016.

[37] Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International Conference on Machine Learning*, pages 8116–8126. PMLR, 2020.

[38] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[39] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[40] Sofia Serrano and Noah A Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, 2019.

[41] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[42] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[43] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

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

[45] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[46] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[47] Kai Sheng Tai, Peter Bailis, and Gregory Valiant. Equivariant transformer networks. In *International Conference on Machine Learning*, pages 6086–6095. PMLR, 2019.

[48] Stefano Teso. Toward faithful explanatory active learning with self-explainable neural nets. In *Proceedings of the Workshop on Interactive Adaptive Learning (IAL 2019)*, pages 4–16. CEUR Workshop Proceedings, 2019.

[49] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[50] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–25, 2020.

[51] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: learning rotationally equivariant features in volumetric data. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10402–10413, 2018.

[52] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5028–5037, 2017.

[53] Mengjiao Yang and Been Kim. Benchmarking attribution methods with relative feature importance. *arXiv preprint arXiv:1907.09701*, 2019.

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

[55] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

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

[57] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 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]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (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? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
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] We will provide the code and instructions at request. The data used in the experiments are public available. See Section 4.1.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (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 the beginning of Section 4.
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? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
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]