
Self-Interpretable Model with Transformation Equivariant Interpretation

Yipei Wang, Xiaoqian Wang *

Department of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907
wang4865@purdue.edu, joywang@purdue.edu

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 interpretations through transformation equivariant regularization in a self-interpretable model. The resulting model is capable of capturing valid interpretations that are 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) models have 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 access, 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 (Shorten and Khoshgoftaar, 2019; Qi et al., 2020). Transformation equivariance refers to the property that data representations learned from the model capture the intrinsic coordinates of the entities (Hinton et al., 2011), i.e., transformations on the data will result in the same transformations to the model representations. Building transformation-equivariant DL models is desired in many kinds of applications, such as medical image analysis (Chidester et al., 2019), reinforcement learning (Mondal et al., 2020), etc.

Although DL models can exert excellent performance in various tasks, 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 (Doshi-Velez and Kim, 2017). In low-risk tasks 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 (Kim and Canny, 2017), the applied model needs to be more convincing than a black box.

*This paper is accepted by 2021 Conference on Neural Information Processing Systems (NeurIPS). Correspondence to: Xiaoqian Wang.

On the one hand, by faithfully explaining the model behavior, it can ensure the end user intuitively understands and trusts the DL model. On the other hand, the explanation of black-box models can provide insights into the relationship between input and output, thereby improving model design. However, with the rapid growth in computational power, DL models are designed to be more and more complex to meet the performance (Rahwan et al., 2019), and the most advanced DL models can have billions of trainable parameters (Ramesh et al., 2021). High complexity leads to the complete black box for human beings, which results in a lack of trust in the model. The demand for building more reliable and easy-to-understand DL models 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 (Murdoch et al., 2019a). Unlike post-hoc models, which generate interpretations to 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 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 (Adebayo et al., 2018; Kindermans et al., 2019; Teso, 2019). In contrast, self-interpretable models make interpretations at the same time as predictions, thus revealing the intrinsic mechanism of the models, and are thereby preferred by users in high-stakes tasks (Rudin, 2019). 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 mechanism. This correspondence between transformation equivariance and faithfulness suggests that self-interpretable models may perform better than post-hoc models in transformation awareness given their 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 humans. This process usually involves prior domain knowledge provided by human experts (Lage et al., 2018; Rieger et al., 2020). 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 the performance of prediction results (Du et al., 2019; Murdoch et al., 2019b). 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 of 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 regularizes the interpretations to be meaningful and comprehensible for humans, while the transformation regularizer constrains the interpretations to be transformation equivariant. We validate that SITE presents understandable and faithful interpretations without requiring additional domain knowledge, and 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.

- 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 (Lundberg and Lee, 2017), interpretable clustering (Monnier et al., 2020), interpretable reinforcement learning (Mott et al., 2019), disentanglement (Shen et al., 2020), 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 are 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 (Zhou et al., 2016; Selvaraju et al., 2017; Wang et al., 2020a; Shrikumar et al., 2017; Rebuffi et al., 2020; Bach et al., 2015), 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) (Zhou et al., 2016) visualizes the feature importance in convolutional neural networks by mapping the weights in the 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 (Wang et al., 2020a). While for approximation methods (Ribeiro et al., 2016), the interpretation is obtained by fitting an interpretable model to the black-box prediction around the target sample. Deconvolution methods (Zeiler and Fergus, 2014) interpret a convolutional neural network via image deconvolution. For perturbation-based interpretation (Petsiuk et al., 2018; Fong et al., 2019), the methods interpret the feature importance by imposing perturbation to certain feature and checking the changes in the output. Moreover, Shapley values (Lundberg and Lee, 2017) 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 (Adebayo et al., 2018; Kindermans et al., 2019).

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 self-interpretation constraints can limit the expressive power, thus sacrificing prediction performance. For example, in order to build an interpretable decision set (Lakkaraju et al., 2016), there is a constraint on the number of rules for the sake of interpretation, which restricts its application to complex tasks. Recent models propose to build self-interpretable models with neural network (Agarwal et al., 2020; Alvarez-Melis and Jaakkola, 2018; Chen et al., 2018a; Jain et al., 2020; Koh et al., 2020; Wang et al., 2020b, 2018) and kernel methods (Chen et al., 2017). FRESH (Jain et al., 2020) focuses on the interpretability for natural language processing tasks. SENN (Alvarez-Melis and Jaakkola, 2018), Concept Bottleneck Models (Koh et al., 2020) generate interpretations in high-level spaces instead of the raw pixel space. ProtoPNet (Chen et al., 2018a) provides interpretations in the pixel space, but it focuses more on the local patches corresponding to the local areas of the image instead of the global interpretation. NAM (Agarwal et al., 2020) provides the same kind of interpretations as SITE. It combines neural networks with additive models to facilitate the self-interpretation via component function. But it decouples all pixels, which results in low expressiveness. Moreover, attention models have been widely used to build interpretable predictions (Mohankumar et al., 2020). However, recent works find that the interpretation via attention weights can fail to identify the important representations (Serrano and Smith, 2019).

Different from the related works, our goal is to build a self-interpretable model that learns faithful interpretation and has high expressive power. Previously the transformation equivariance property

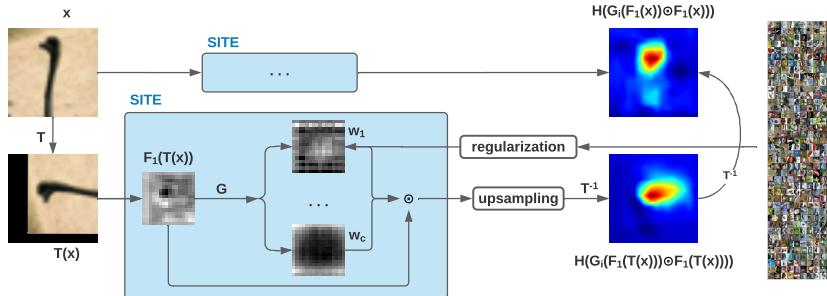


Figure 1: An illustration of our SITE model. SITE can take both original image x and transformed image $T(x)$ as input. The input is first fed to the feature extractor F_1 , then SITE generates c prototypes w_1, \dots, w_c through generator G . Finally, both the prediction and interpretation come from the Hadamard product between the latent representation $F_1(T(x))$ 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.

has been studied in prediction via deep neural networks. Many recent studies integrate the transformation equivariance in object detection, with the goal of building convolutional neural networks that are equivariant to image translations. The models learn features equivariant to translation and rotation (Worrall et al., 2017; Weiler et al., 2018; Cheng et al., 2018), 3D symmetries (Thomas et al., 2018), and build sets with symmetric elements for general equivariant tasks (Maron et al., 2020). Despite transformation equivariance in the prediction, these methods may not guarantee the transformation equivariance in the interpretation, i.e., the prediction mechanisms of transformed and untransformed inputs may be inconsistent. We thus introduce the interpretation equivariance to complement the prediction equivariance. To the best of our knowledge, we are the first to learn transformation-equivariant interpretations to ensure faithful and robust interpretation.

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 image 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 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

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

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 $G = [G_1, \dots, G_c]$ 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,

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

where σ is softmax activation. The prediction \hat{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

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

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{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, $\hat{w}_i^j, i = 1, \dots, c, j = 1, \dots, d$ 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 (Sundararajan et al., 2017), and also known as the **local accuracy** in (Lundberg and Lee, 2017). This property assures that the interpretation is related to the corresponding prediction in the numerical sense.

The interpretation obtained from optimizing L_{cls} ensures the faithfulness (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 $G_i(\mathbf{x}), i = 1, \dots, c$ 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 $F_1(\mathbf{x}_i)$:

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

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 (Jaderberg et al., 2015). We want SITE to learn interpretations that are equivariant to the transformations. Here $\beta \sim \mathcal{B}$ denotes the randomly sampled parameters from a pre-defined parameter distribution \mathcal{B} . This is because an affine transformation operator T can be parameterized by an 3×3 matrix β . During the 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 randomly 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 $T_\beta^{-1}(G_i(F_1(T_\beta(\mathbf{x}))))$, $i = 1, \dots, c$ and the latent representations of $\mathbf{x}_i \in \mathcal{X}_i, i = 1, \dots, c$, respectively:

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

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:

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

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

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

where λ is a hyper-parameter that balances the training paces between the classification loss and the transformation loss. The first term in 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 Monte 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, the 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 $H : \mathbb{R}^d \rightarrow \mathbb{R}^p$ 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 an interpretation method properly, we propose a numerical metric, namely the *self-consistency score*. It measures the self-consistency (Tai et al., 2019) 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}_β is approximated by the 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

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

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. Please refer to the Appendix for more details about the experimental setup.

4.1 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 clarify that SITE does not sacrifice prediction power for interpretability. Please refer to Sec. 4.3 for more details.

The interpretations of SITE on MNIST are shown in Fig. 2. The Hadamard product decides that the interpretations are 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 preserve 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.

4.2 Experiments on CIFAR-10

Interpretations of SITE

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

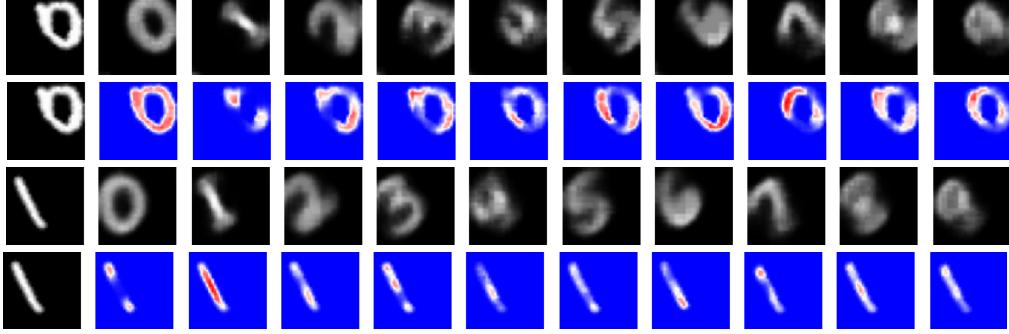


Figure 2: Interpretations of SITE on MNIST dataset. For each digit, the first column shows the randomly transformed images. The following $c = 10$ columns are the prototypes $\{\mathbf{w}_i\}_{i=1}^c$ (the first and third rows), and the interpretation heatmaps $\{\mathbf{w}_i \odot \mathbf{x}\}_{i=1}^c$ (the second and fourth rows).

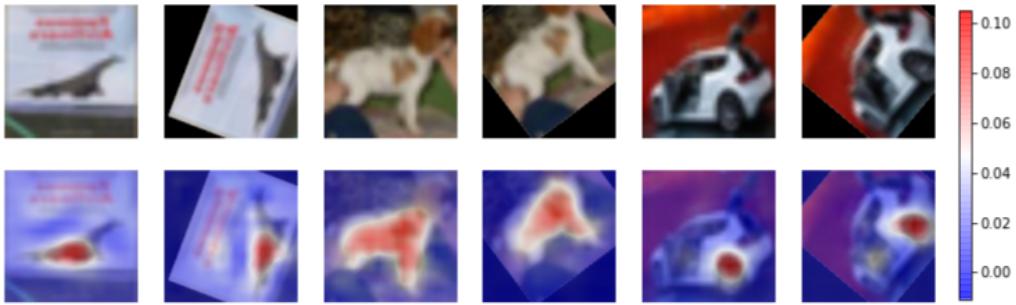


Figure 3: Interpretations of SITE on CIFAR-10 dataset. The first row shows the original images (odd columns) and their random affine transformed version (even columns). The second row shows the interpretation heatmaps overlaid on corresponding images.

The interpretation results are of SITE on CIFAR-10 are shown in Fig. 3. Here we sample three images of a plane, a dog, and a car for demonstrations. Each image is transformed by a constrained affine transformation $T_\beta \in \mathcal{T}_B$ that is 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 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 (Tai et al., 2019). 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 sacrifice expressive power for interpretations. We take the ResNet-18 classifier as the benchmark since SITE takes the same structure as the feature extractor. Given the same transformation family \mathcal{T}_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. 4.3 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 (Selvaraju et al., 2017), excitation back-propagation (Zhang et al., 2018), guided back-propagation (Springenberg et al., 2014), gradient (Simonyan et al., 2013), DeConvNet (Zeiler and Fergus, 2014), and linear approximation. And also there are perturbation methods such as randomized input sampling (RISE) (Petsiuk et al., 2018) and extremal perturbation (EP) (Fong et al., 2019). 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

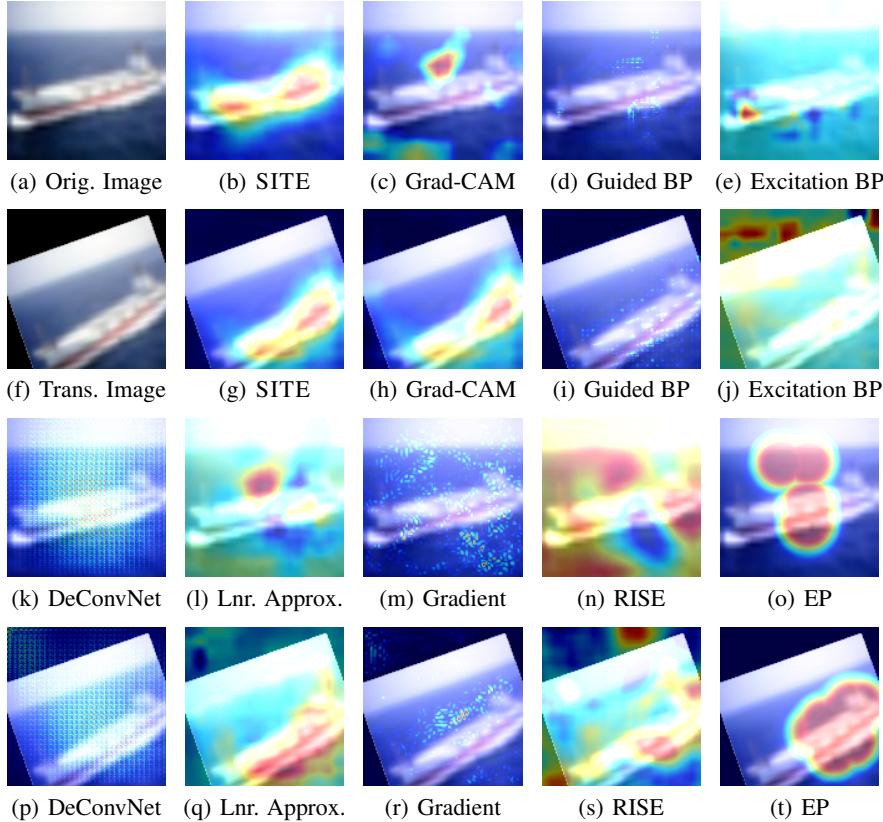


Figure 4: Interpretation comparison on CIFAR-10 dataset. The odd rows show the interpretations on the original image, while the even rows show the interpretations on the transformed image. Note that the model is trained on **transformed images**, thus all interpretations on the transformed images are relatively reasonable. However, most interpretations are highly disturbed on untransformed image, while SITE preserves the most transformation equivariant interpretations.

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 is 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 of 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. The comparison of self-consistency scores is 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 carry out the mask- k -pixels experiments (Chen et al., 2018b) 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 model, 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,

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)	0.8860	0.8817	0.7830	0.1159	0.7174	0.8485	0.8591
$v_{\mathcal{X}}$ (backbone)	-	0.8416	0.8168	0.2460	0.6926	0.4183	0.7721

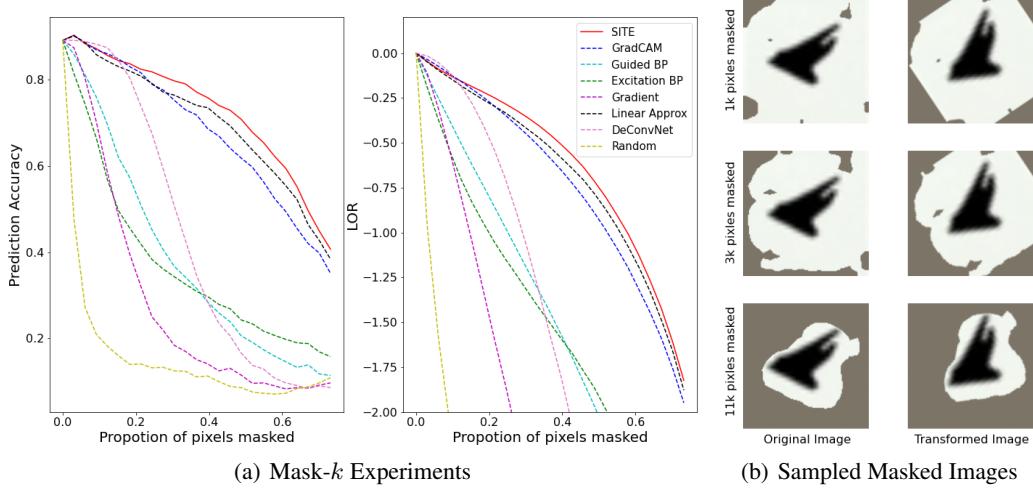


Figure 5: (a) The trend of accuracy (left) and log-odds ratio (right) with various proportions of pixels masked; (b) Images where 1k (top), 3k (middle) and 11k (bottom) pixels (out of 16.4k pixels) are masked. The left and right columns are for the original and the transformed images, respectively.

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 parenthesis.

	Decision Tree	Random Forest	Logistic Regression	XGBoost	NAM	SITE-CNN (Blackbox CNN)	SITE-ResNet (Blackbox ResNet)
MNIST	0.886	0.970	0.929	0.975	0.935	0.988 (0.981)	-
CIFAR-10	0.229	0.396	0.357	0.450	0.370	0.840 (0.828)	0.892 (0.862)

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 alleviated, 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 (Yang and Kim, 2019). Please refer to the Appendix for details.

4.3 Expressiveness

Since there is an inevitable trade-off between expressiveness and interpretability, most existing self-interpretable models have relatively low accuracy on image datasets such as MNIST and CIFAR. Here we compare the expressiveness of SITE and existing self-interpretable models including simple models (trained using sklearn) like Decision Tree, Random Forest, Logistic Regression,

and complex models like XGBoost (Chen and Guestrin, 2016) (trained using `xgboost`), Neural Additive Model (NAM) (Agarwal et al., 2020). The results of XGBoost are reported in (Ponomareva et al., 2017). We include SITE with backbones of different levels of complexity to demonstrate the scalability of SITE. The CNN backbone contains 235k parameters for MNIST and 1.2m parameters for CIFAR-10, while the ResNet backbone is the ResNet-18 used in all previous experiments. The backbone models share the same structures and the same (transformed) training data as the corresponding SITE in feature extraction. 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-interpretation (Boopathy et al., 2020).

5 Conclusions

In this paper, we propose a self-interpretable model SITE with transformation-equivariant interpretations. We focus on the robustness and self-consistency of the interpretations of 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. It is worth noticing that although applied in most of the CNN visualization methods, the bilinear upsampling approximation is a rough approximation, which 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 MNIST experiments). Besides, we consider the translation and rotation transformations in our model. In future work, we will explore the robust interpretations under more complex transformations such as scaling and distortion. Moreover, we clarify that SITE is not limited to geometric transformation (that we used in the computer vision domain), and will explore SITE in other domains in future work.

Acknowledgement

This work was partially supported by NSF IIS #1955890, Purdue’s Elmore ECE Emerging Frontiers Center. The applications of various post-hoc methods are implemented through the TorchRay toolkit. We are grateful to the anonymous NeurIPS reviewers for the insightful comments.

References

- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. (2018). Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*.
- Agarwal, R., Frosst, N., Zhang, X., Caruana, R., and Hinton, G. E. (2020). Neural additive models: Interpretable machine learning with neural nets. *arXiv preprint arXiv:2004.13912*.
- Alvarez-Melis, D. and Jaakkola, T. S. (2018). Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7786–7795.
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140.
- Boopathy, A., Liu, S., Zhang, G., Liu, C., Chen, P.-Y., Chang, S., and Daniel, L. (2020). Proper network interpretability helps adversarial robustness in classification. In *International Conference on Machine Learning*, pages 1014–1023. PMLR.
- Bossard, L., Guillaumin, M., and Van Gool, L. (2014). Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*.
- Chen, C., Li, O., Tao, C., Barnett, A. J., Su, J., and Rudin, C. (2018a). This looks like that: Deep learning for interpretable image recognition. *arXiv preprint arXiv:1806.10574*.
- Chen, H., Wang, X., Deng, C., and Huang, H. (2017). Group sparse additive machine. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 197–207.

- Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. (2018b). L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*.
- Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.
- Cheng, X., Qiu, Q., Calderbank, R., and Sapiro, G. (2018). RotDCF: Decomposition of convolutional filters for rotation-equivariant deep networks. In *International Conference on Learning Representations*.
- Chidester, B., Zhou, T., Do, M. N., and Ma, J. (2019). Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, 35(14):i530–i537.
- Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.
- Du, M., Liu, N., and Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77.
- Fong, R., Patrick, M., and Vedaldi, A. (2019). Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2950–2958.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.
- Hinton, G. E., Krizhevsky, A., and Wang, S. D. (2011). Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer.
- Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. *arXiv preprint arXiv:1506.02025*.
- Jain, S., Wiegreffe, S., Pinter, Y., and Wallace, B. C. (2020). Learning to faithfully rationalize by construction. *arXiv preprint arXiv:2005.00115*.
- Kim, J. and Canny, J. (2017). Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2942–2950.
- Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D., and Kim, B. (2019). The (un)reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer.
- Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. (2020). Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR.
- Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.
- Lage, I., Ross, A. S., Kim, B., Gershman, S. J., and Doshi-Velez, F. (2018). Human-in-the-loop interpretability prior. *arXiv preprint arXiv:1805.11571*.
- Lakkaraju, H., Bach, S. H., and Leskovec, J. (2016). 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.
- LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer.

- Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4768–4777.
- Maron, H., Litany, O., Chechik, G., and Fetaya, E. (2020). On learning sets of symmetric elements. In *International Conference on Machine Learning*, pages 6734–6744. PMLR.
- Mohankumar, A. K., Nema, P., Narasimhan, S., Khapra, M. M., Srinivasan, B. V., and Ravindran, B. (2020). Towards transparent and explainable attention models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4206–4216.
- Mondal, A. K., Nair, P., and Siddiqi, K. (2020). Group equivariant deep reinforcement learning. *arXiv preprint arXiv:2007.03437*.
- Monnier, T., Groueix, T., and Aubry, M. (2020). Deep transformation-invariant clustering. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 7945–7955.
- Mott, A., Zoran, D., Chrzanowski, M., Wierstra, D., and Rezende, D. J. (2019). Towards interpretable reinforcement learning using attention augmented agents. *arXiv preprint arXiv:1906.02500*.
- Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019a). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080.
- Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019b). Interpretable machine learning: Definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*.
- Petsiuk, V., Das, A., and Saenko, K. (2018). Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*.
- Ponomareva, N., Colthurst, T., Hendry, G., Haykal, S., and Radpour, S. (2017). Compact multi-class boosted trees. In *2017 IEEE International Conference on Big Data*, pages 47–56. IEEE.
- Qi, G.-J., Zhang, L., Lin, F., and Wang, X. (2020). Learning generalized transformation equivariant representations via autoencoding transformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.
- Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J.-F., Breazeal, C., Crandall, J. W., Christakis, N. A., Couzin, I. D., Jackson, M. O., et al. (2019). Machine behaviour. *Nature*, 568(7753):477–486.
- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*.
- Rebuffi, S.-A., Fong, R., Ji, X., and Vedaldi, A. (2020). There and back again: Revisiting backpropagation saliency methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8848.
- Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). “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.
- Rieger, L., Singh, C., Murdoch, W., and Yu, B. (2020). Interpretations are useful: Penalizing explanations to align neural networks with prior knowledge. In *International Conference on Machine Learning*, pages 8116–8126. PMLR.
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626.

- Serrano, S. and Smith, N. A. (2019). Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951.
- Shen, Y., Yang, C., Tang, X., and Zhou, B. (2020). Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.
- Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48.
- Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.
- Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.
- Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.
- Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.
- Tai, K. S., Bailis, P., and Valiant, G. (2019). Equivariant transformer networks. In *International Conference on Machine Learning*, pages 6086–6095. PMLR.
- Teso, S. (2019). Toward faithful explanatory active learning with self-explainable neural nets. In *Proceedings of the Workshop on Interactive Adaptive Learning*, pages 4–16. CEUR Workshop Proceedings.
- Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. (2018). Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*.
- Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., and Hu, X. (2020a). 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.
- Wang, R., Wang, X., and Inouye, D. I. (2020b). Shapley explanation networks. In *International Conference on Learning Representations*.
- Wang, X., Chen, H., Yan, J., Nho, K., Risacher, S. L., Saykin, A. J., Shen, L., Huang, H., and ADNI (2018). Quantitative trait loci identification for brain endophenotypes via new additive model with random networks. *Bioinformatics*, 34(17):i866–i874.
- Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. (2018). 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.
- Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. (2017). Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5028–5037.
- Yang, M. and Kim, B. (2019). Benchmarking attribution methods with relative feature importance. *arXiv preprint arXiv:1907.09701*.
- Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer.
- Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., and Sclaroff, S. (2018). Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. (2017). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464.

A Experimental Setup

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.

First, we define the family of geometric transformations to which we want SITE to be equivariant as constrained parametric affine transformations (Jaderberg et al., 2015): $\mathcal{T}_{\mathcal{B}} = \{T_{\beta} : \beta \sim \mathcal{B}\}$, where $\beta \in \mathbb{R}^6$. And \mathcal{B} is defined to constrain the affine transformations to be compositions of rotations in $[-\pi/2, \pi/2]$ and translation in $[-h/2, h/2]$, 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 the dataset. For MNIST (LeCun et al., 2010), 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 $G_i(\mathbf{x}), i = 1, \dots, c$. Hence the structure of SITE is built to be an autoencoder-based structure, where there are c parallel decoders. As for CIFAR-10 (Krizhevsky, 2009), due to the need for upsampling in visualization, the image data are resized to 128×128 . The feature extractor F_1 is built based on ResNet-18 (He et al., 2016). Here $F_1 : \mathbb{R}^{3 \times 128 \times 128} \rightarrow \mathbb{R}^{10 \times 16 \times 16}$. And for the generator G , it consists of $c = 10$ (number of categories) parallel autoencoders, such that $G_i : \mathbb{R}^{10 \times 16 \times 16} \rightarrow \mathbb{R}^{10 \times 16 \times 16}$. 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. We also test on more complex datasets like Food-101 (Bossard et al., 2014) to demonstrate the scalability of SITE. Please refer to the Appendix E for details. Besides, in order to balance the classification loss and the transformation loss we set the scalar factor to be $\lambda = 5$ throughout the training phase.

B More Examples on CIFAR-10

In this section, we present more results of SITE on the CIFAR-10 dataset as a supplement to *Fig. 3 in the main body of the paper*. Here we only present correctly classified examples. We sample 3 images for each class from the default validation set of CIFAR-10. The interpretations of SITE are shown in Fig. 6-15. Here the first rows are the untransformed and the transformed images, while the second rows are the corresponding interpretations. And two adjacent columns are a pair of untransformed and transformed images. The results of ten classes are listed alphabetically, that is, Fig. 6 for airplanes, Fig. 7 for birds, Fig. 8 for cars, Fig. 9 for cats, Fig. 10 for deer, Fig. 11 for dogs, Fig. 12 for frogs, Fig. 13 for horses, Fig. 14 for ships, and finally Fig. 15 for trucks. It can be clearly found that SITE can accurately highlight the important features of both untransformed and transformed images in classification. Besides, SITE also demonstrates great self-consistency, as the highlighted areas preserve very similar shapes between the untransformed and transformed images.

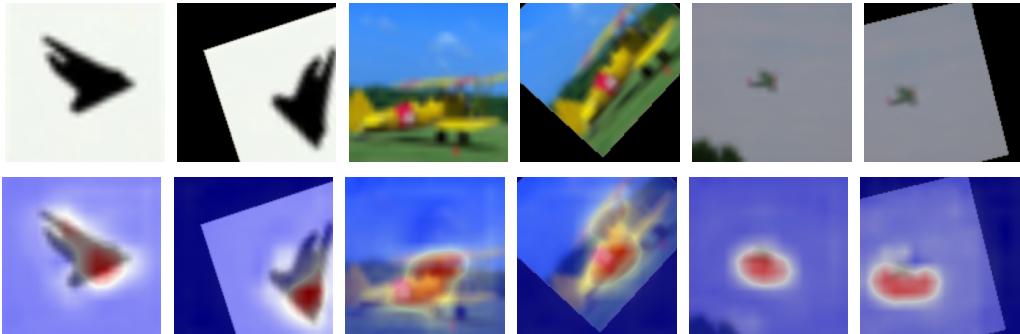


Figure 6: Additional examples of class ‘‘Airplane’’. The first row shows the original images, and the second row shows the heatmaps learned from SITE.

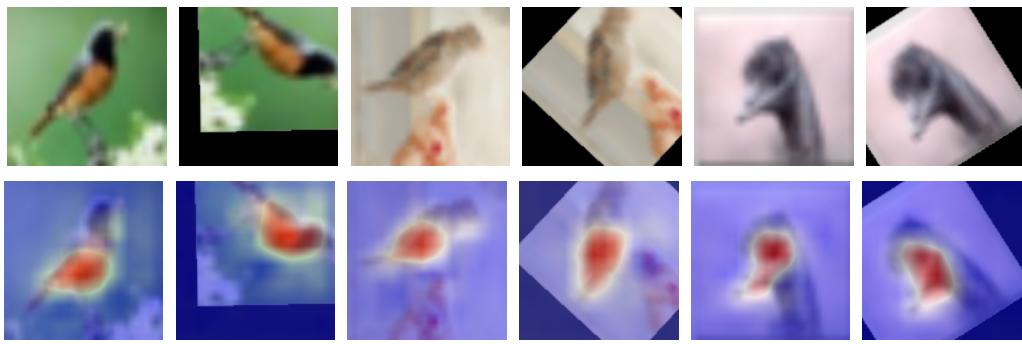


Figure 7: Additional examples of class “Bird”.



Figure 8: Additional examples of class “Car”.



Figure 9: Additional examples of class “Cat”.

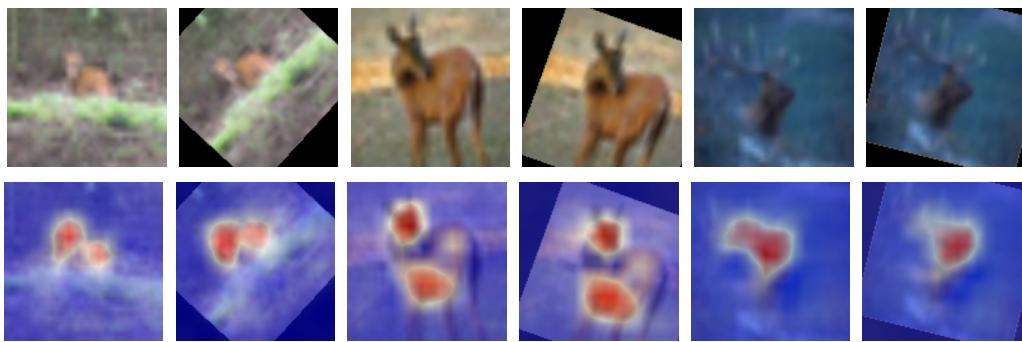


Figure 10: Additional examples of class “Deer”.

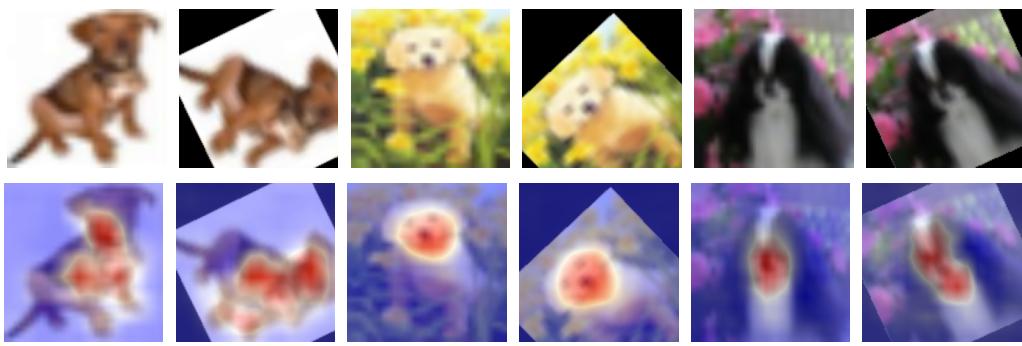


Figure 11: Additional examples of class “Dog”.

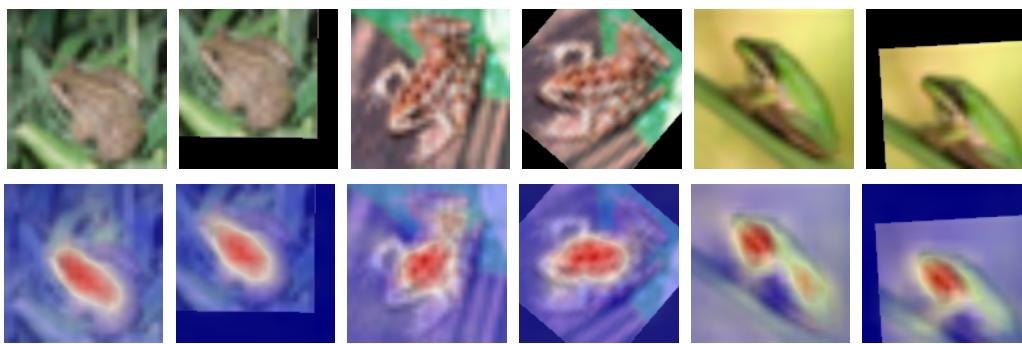


Figure 12: Additional examples of class “Frog”.

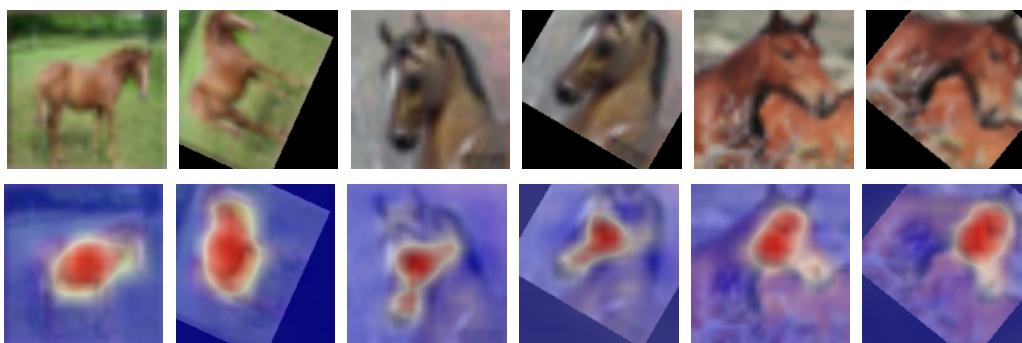


Figure 13: Additional examples of class “Horse”.

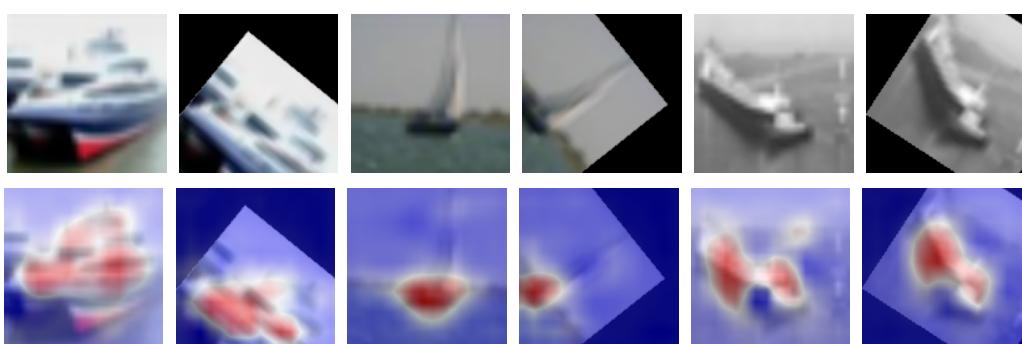


Figure 14: Additional examples of class “Ship”.



Figure 15: Additional examples of class “Truck”.

C More Examples in Comparison

In this section, we present more comparison results among SITE and the post-hoc methods mentioned above. This is a supplement to *Fig. 4 in the main body of the paper*. The companions include: back-propagation methods such as Grad-CAM (Selvaraju et al., 2017), excitation back-propagation (Zhang et al., 2018), guided back-propagation (Springenberg et al., 2014), gradient (Simonyan et al., 2013), DeConvNet (Zeiler and Fergus, 2014), and linear approximation. And also there are perturbation methods such as randomized input sampling (RISE) (Petsiuk et al., 2018) and extremal perturbation (EP) (Fong et al., 2019). The various comparing methods are implemented through the TorchRay toolkit (Fong et al., 2019). 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 results are presented in Fig. 16. In Fig. 16, we demonstrate a similar result as shown in the main body. Although all methods present good results for the transformed images, post-hoc methods show less faithful interpretations when dealing with untransformed images. This is because the model is trained on transformed images. And therefore, we can claim that post-hoc methods are not faithful to the predictions.

D Benchmarking Attribution Methods

The Benchmarking Attribution Methods (BAM) (Yang and Kim, 2019) are evaluations to the correctness of attribution interpretation methods. The BAM dataset consists of artificial images, which are combinations of the scene dataset MiniPlaces (Zhou et al., 2017) and objects dataset MSCOCO (Lin et al., 2014). The dataset is constructed by overlaying the scaled objects on the scenes. Since there are 10 classes for both objects and scenes dataset, there are 100 classes of BAM dataset after the composition. And each class has 1000 images. Using the same dataset, with different labels, the BAM dataset can be denoted by \mathcal{X}_o and \mathcal{X}_s , where \mathcal{X}_o has labels for the objects, while \mathcal{X}_s has labels for the scenes. An attribution interpretation method is considered to be reasonable only when it can highlight correct areas – if trained on \mathcal{X}_o , the highlighted area should be the objects, and vice versa. We train SITE on both datasets, and present the attribution interpretations by comparing them with other post-hoc methods mentioned above. The results are shown in Fig. 17 and 18. Here Fig. 17 (a)(f) are repeated just for an aligned illustration. In Fig. 17, we present the results on an (correctly classified) untransformed image from the validation set of BAM. It can be found that most methods present reasonable interpretations of the prediction result. That is, they highlight correct areas for corresponding models (trained on \mathcal{X}_s and \mathcal{X}_o). However, from Fig. 18, where the image is transformed, SITE shows more self-consistent and faithful interpretations than the comparing methods. Here the two images Fig. 18 (a)(f) are for object and scene datasets, respectively, and are thereby randomly transformed separately. From Fig. 18(b)(c) we can find that SITE outperforms Grad-CAM in highlighting the object area.

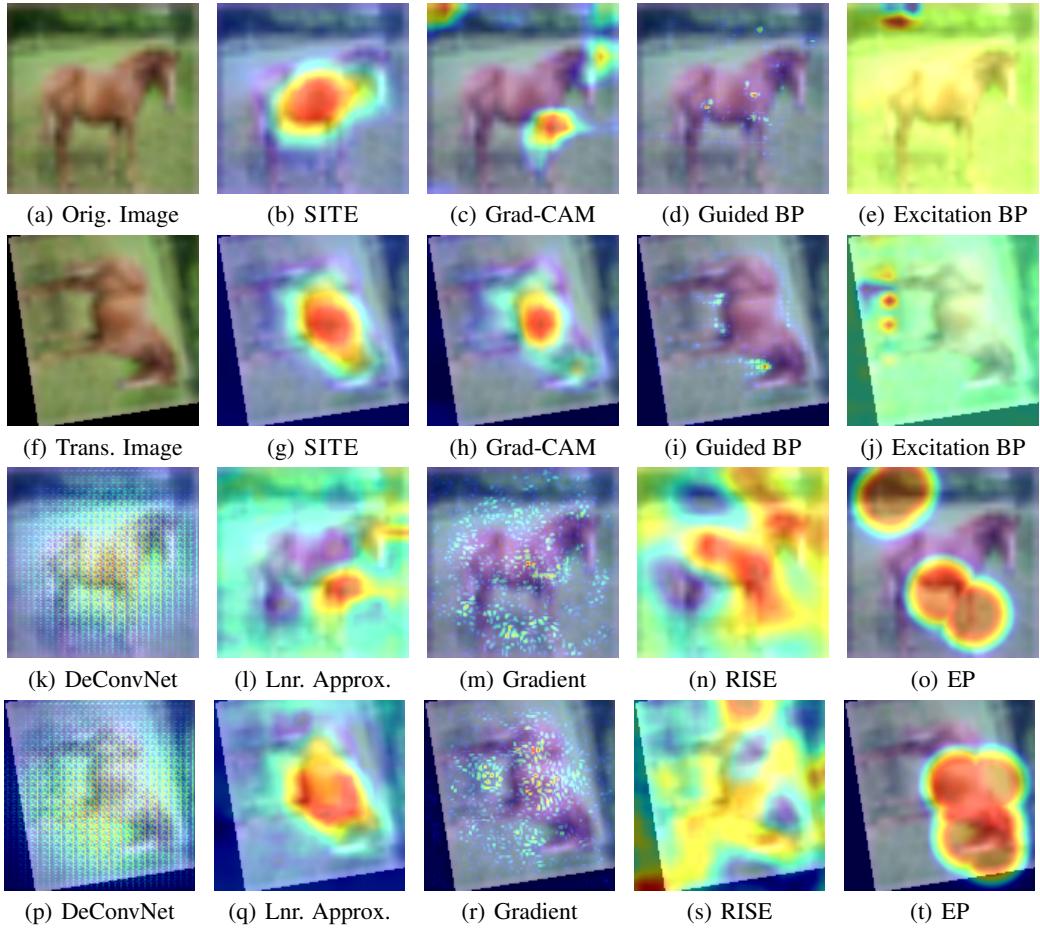


Figure 16: Additional comparisons of interpretations from SITE and post-hoc methods. Here SITE accurately captures the important features in both transformed and untransformed images. However, most of the post-hoc methods can only have comparable results on transformed images (where the model is trained), but fail on the untransformed image.

E Results on Food-101 Dataset

Food-101 (Bossard et al., 2014) is a fine-grained food image dataset. It is much more complicated than CIFAR-10 and MNIST. We use this to demonstrate the scalability of SITE. The dataset contains 101 categories and 1000 images per category. The 1000 images of each category are split into the training (750) and validation (250) subsets. Images are randomly transformed during both the training and the validation phases. We resize all images to 128×128 , and use the same structure F_1, G as CIFAR-10. The difference is that the number of categories is $c = 101$ instead of 10. Under this setting, F_1 contains around 11.2m parameters, while G contains around 3.4m parameters. As a result, SITE only increases the number of parameters by 30% for such a complex dataset. The accuracy results of both SITE and the backbone model (ResNet-18) are shown in Table 3. And the interpretations are illustrated in Fig. 19. SITE obviously preserves great self-consistency between transformed and untransformed images.

Model	Accuracy
SITE	63.91%
black-box backbone (ResNet-18)	64.04%

Table 3: The expressiveness results of SITE and the corresponding black-box model on Food-101.

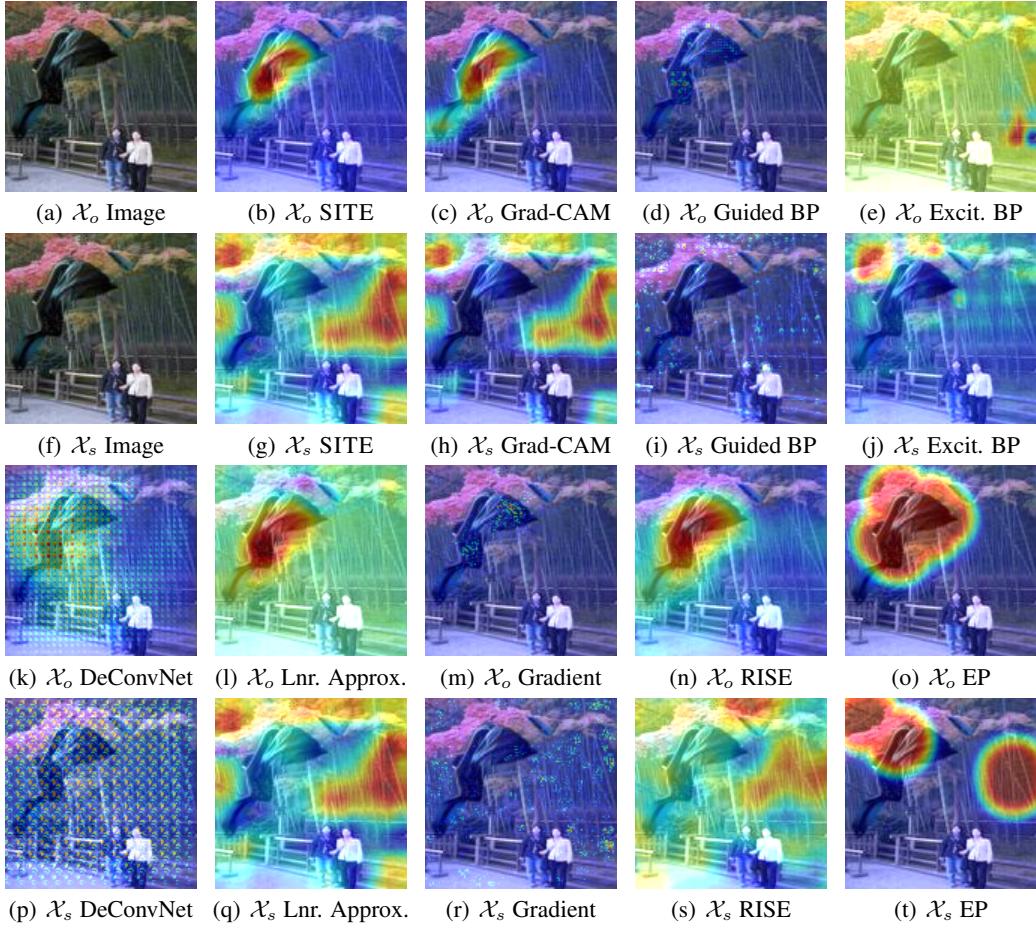


Figure 17: The comparison among different interpretation methods on BAM dataset. The sample is from the *untransformed* validation set. The interpretations in the first and the third rows ((a)-(e), (k)-(o)) are to the model that is trained on \mathcal{X}_o (with labels for objects). And the interpretations in the second and the last rows ((g)-(h), (p)-(t)) are to the model that is trained on \mathcal{X}_s (with labels for scenes).

F Pointing Game

In order to further demonstrate the superiority of SITE in the interpretation quality compared with post-hoc models, we also carry out the pointing game experiment (Zhang et al., 2018) on the annotated MNIST dataset. In the pointing game, an interpretation method calculates an interpretation map \hat{w} of the input x w.r.t. the class c . The method scores a hit every time the largest value of s falls in the image region Ω within the tolerance τ . Ω is the region containing the object for the objects dataset (or excluding the object for the scenes dataset). The ratios $\frac{\text{hit}}{\text{hit} + \text{miss}}$ are shown in Table 4. As the baseline, we also include the post-hoc interpretations to the backbone models. It can be found that SITE outperforms all post-hoc models in the pointing game experiment. Similar to the experiments we conduct, all methods are trained with the same set of randomly transformed images (which includes the identity mapping, i.e., original images) and explain the same model. Thus, the significantly higher hit ratio by SITE in pointing game evaluation validated the improvement of SITE in transformation equivariance.

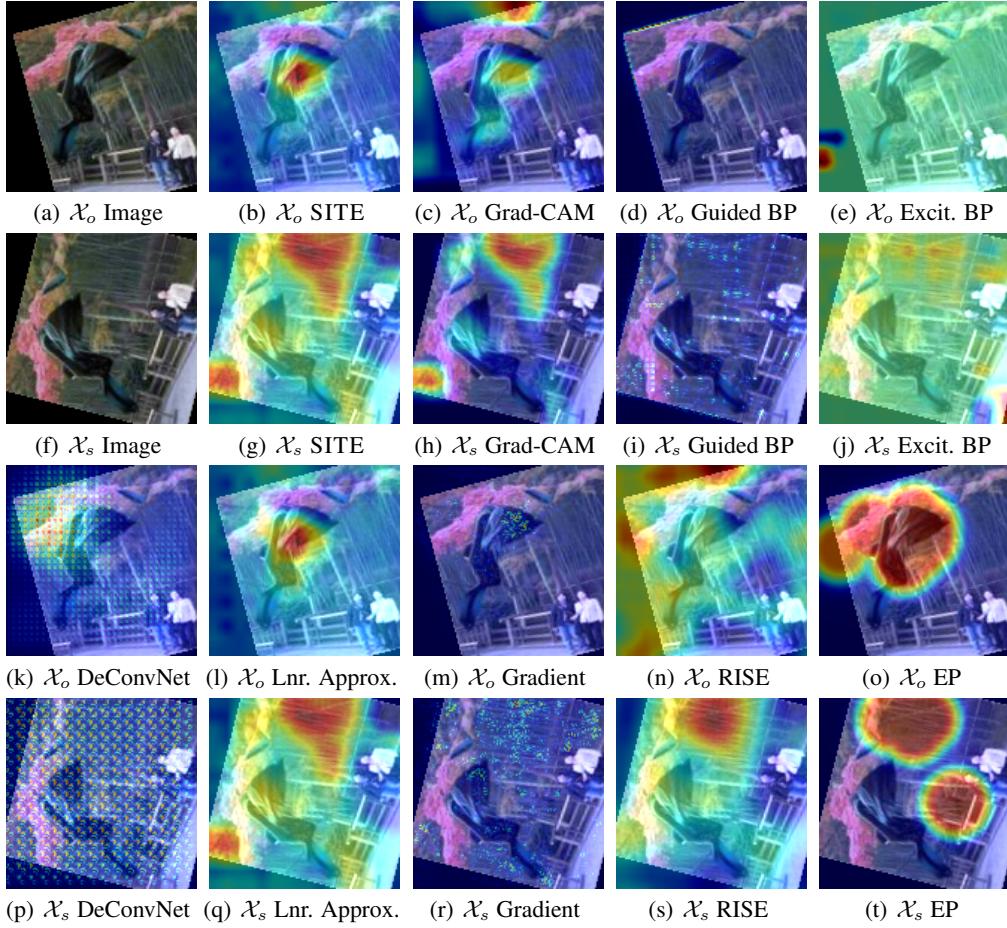


Figure 18: The comparison among different interpretation methods on BAM dataset. The sample is from the *transformed* validation set. The interpretations in the first and the third rows ((a)-(e), (k)-(o)) are to the model that is trained on \mathcal{X}_o (with labels for objects). And the interpretations in the second and the last rows ((g)-(h), (p)-(t)) are to the model that is trained on \mathcal{X}_s (with labels for scenes).

Table 4: The pointing game results on annotated MNIST dataset.

Interpreter	Classifier	Transformed		Untransformed	
		SITE	Backbone	SITE	Backbone
SITE	0.9993	-	0.9996	-	
Gradient	0.5423	0.8992	0.7741	0.9540	
GradCAM	0.7726	0.7730	0.8138	0.8062	
Linear Approx.	0.6540	0.9577	0.8381	0.9856	
DeconvNet	0.2546	0.6895	0.3128	0.6553	
Excitation BP	0.3588	0.9892	0.4716	0.9979	
Guided BP	0.4664	0.9974	0.8825	0.9990	

G Failure Case Analysis

There's no perfect model that does not make any mistake. Therefore, the interpretations of the misclassified data, i.e. the failure cases, are very important. On the one hand, it can give users comprehensible feedback to the mistake by revealing its reasoning. On the other hand, it can help model designers to better understand and debug it, and it can also provide insights into the training

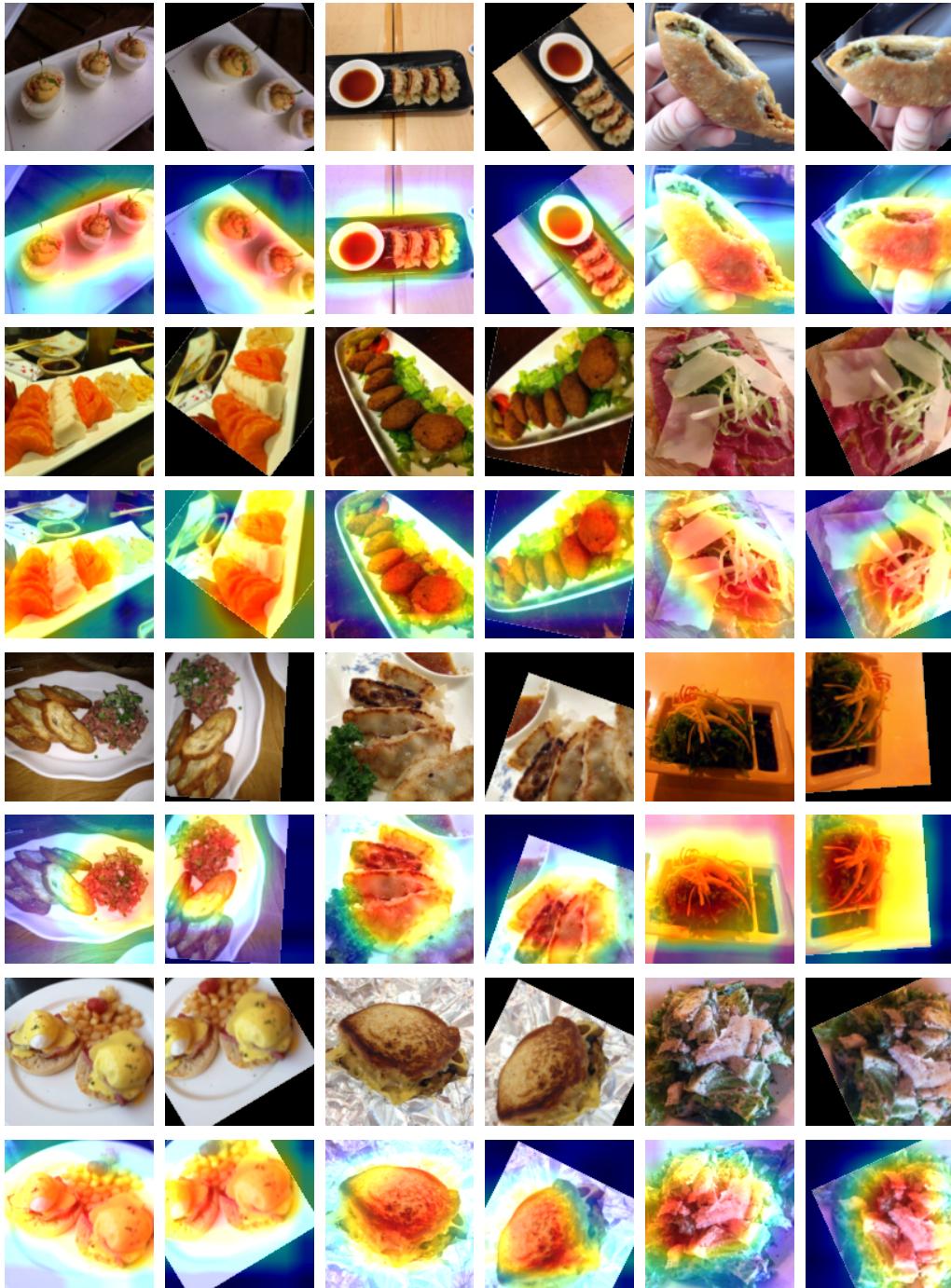


Figure 19: SITE Interpretations on Food-101

data. Based on the structure of SITE, we can easily present the interpretation to the predicted logits of arbitrary classes. Here in Fig. 20, we present five examples where SITE fails in predictions. For clarity concerns, we omit transformations here and use original CIFAR-10 data directly. The first row is the input images from the validation set of CIFAR-10. The second and the third rows are the interpretations of the true class, while the last two rows are the interpretations of the predicted (wrong) class.

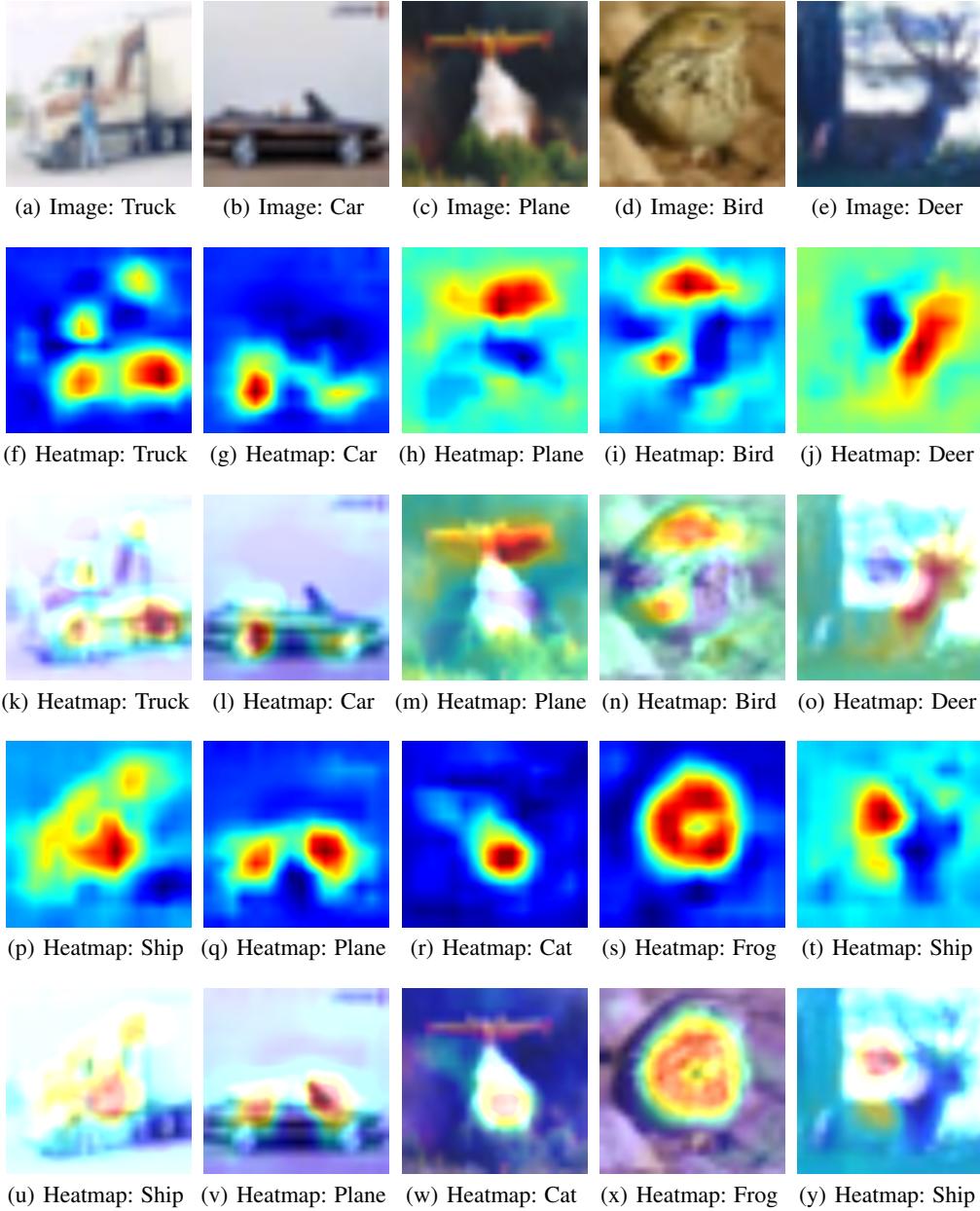


Figure 20: Failure cases of predictions on CIFAR-10. The first row images (a)(b)(c)(d)(e) are input images in the default validation set of CIFAR-10, where SITE makes wrong predictions. The second and the third rows (f)-(o) are the interpretations SITE makes to the true classes. The second row is the heatmaps themselves, while the third row is the heatmaps overlaid on the input images. The fourth and the last rows (p)-(y) are the interpretations SITE makes to the predicted (wrong) classes. The fourth row is the heatmaps themselves, while the last row is the heatmaps overlaid on the input images. The five columns are truck, car, plane, bird, and ship, respectively. And they are classified to be ship, plane, cat, frog, and ship, respectively.

From Fig. 20(a)(f)(k)(p)(u), we can see that a truck is predicted to be a ship by SITE. The interpretation of the prediction to the ship class is shown in (p)(u) and the interpretation to the truck class is shown in (f)(k). We can find that SITE focuses mainly on wheels for trucks but on the main body for ships.

Similarly, by comparing Fig. 20(b)(g)(l)(q)(v), we can deduce that SITE discriminates this car as a plane mainly because of the lateral view of the front windshield as shown in (q)(v). It is possible that SITE treats it as an airfoil. This can also provide an insight to the training data that there should be more lateral view of cars. And the prediction to the car class is due to the wheels of the object, as shown in (g)(l).

And in Fig. 20(c)(h)(m)(r)(w). We can see that SITE misclassifies a plane to a cat. With the interpretations shown in (r)(w), we can find this misclassification is because the white airflow fools SITE to treat it as a hairy cat.

In Fig. 20(d)(i)(n)(s)(x), a baby bird is classified to be a frog. We can find it is classified to be a frog mainly because of the main body as shown in (s)(x), while for the bird class it is mainly because of the head.

Finally, from Fig. 20(e)(j)(o)(t)(y), the image of a deer is classified to be a ship because of the horizontal ship-like structure that is behind the deer, as shown in (j)(o). And it captures the features of the deer for the deer class as shown in (t)(y).

From the above-mentioned examples, we can see that even when making wrong predictions, the faithfulness of SITE still lead to reasonable interpretations, which benefit human to understand and debug the model, and also to enhance the training dataset.