Computer Science and Information Systems 20(4):1419–1437 https://doi.org/10.2298/CSIS230310047L

Explaining Deep Residual Networks Predictions with Symplectic Adjoint Method

Xia Lei[1], Jia-Jiang Lin[1], Xiong-Lin Luo[1]*, and Yongkai Fan[2]

[1] Department of Automation, China University of Petroleum Bejing
102249 Beijing, China
leixia2008530059@163.com
lawliet3@qq.com
luoxl@cup.edu.cn
[2] State Key Laboratory of Media Convergence and Communication
Communication University of China
100024 Beijing, China
fanyongkai@gmail.com

Abstract. Understanding deep residual networks (ResNets) decisions are receiving much attention as a way to ensure their security and reliability. Recent research, however, lacks theoretical analysis to guarantee the faithfulness of explanations and could produce an unreliable explanation. In order to explain ResNets predictions, we suggest a provably faithful explanation for ResNet using a surrogate explainable model, a neural ordinary differential equation network (Neural ODE). First, ResNets are proved to converge to a Neural ODE and the Neural ODE is regarded as a surrogate model to explain the decision-making attribution of the ResNets. And then the decision feature and the explanation map of inputs belonging to the target class for Neural ODE are generated via the symplectic adjoint method. Finally, we prove that the explanations of Neural ODE can be sufficiently approximate to ResNet. Experiments show that the proposed explanation method has higher faithfulness with lower computational cost than other explanation approaches and it is effective for troubleshooting and optimizing a model by the explanation.

Keywords: Deep residual networks, Explanation, Neural ODE, Symplectic adjoint method.

1. Introduction

Deep neural networks, such as deep residual networks (ResNets) [9], have been widely applied due to their superior performance. However, they are usually regarded as black box models driven by data, and it is difficult to explain how the models work and predict their decisions. Therefore, the result of ResNets may be out of control. For instance, the networks are easily tricked into incorrect classification results by adversarial instances, which are produced by artificially created hostile perturbations that are invisible to humans [19], [28],[25]. The uninterpretability of ResNets makes them potentially dangerous for applications in safety-critical tasks such as intelligent medical diagnosis and autonomous driving [12],[18],[10].

* Corresponding author

How to make black box models transparent is a significant and intriguing subject given the requirements for trusted and safe AI. Finding the ResNets' decision-making attribution is a focus of several researchers. By identifying the attribution and creating an explanation why ResNets' predictions are going, gradient-based backpropagation methods such as Guided-BP [27], IntegratedGrad [29] and SmoothGrad [26], and the combined gradient class activation map (CAM) methods such as Grad-CAM [23] and Grad-CAM++ [3] have been proposed. Although these gradient-based methods make it easy to understand the contribution of the input features and can locate meaningful image regions with good semantic visual performance for humans, the gradient vanishing and saturation issues may result in inaccurate explanations. Moreover, they require the gradients layer-by-layer backpropagation, which leads to low computational efficiency. On the other hand, the gradient-free methods such as Score-CAM [31] and Group-CAM [33] use feature maps as masks on the original image to output the forward passing score as the weights instead of gradients. Although the methods outperform current state-of-the-art methods on both visual performance and localization tasks, the weights are obtained entirely based on training, the lack of theoretical analysis cannot demonstrate the faithfulness of the explanation, and retraining greatly reduces the computational efficiency.

Considering the problem of the naive gradient and the lack of theoretical analysis to guarantee the faithfulness of the explanation, this paper proposes a novel explanation method for ResNet with the symplectic adjoint method. An explainable neural ordinary differential equation network (Neural ODE), which is proved to be the convergent model of ResNets, as shown in Fig. 1, is built as a surrogate model to explain the decision features of ResNets. Thus, inspired by [17], the symplectic adjoint method is used to get over the naive gradient problem. Our method takes advantage of gradient-based methods, and it makes up for the lack of theoretical analysis to demonstrate the faithfulness of the explanation. The contributions of this paper are as follows.

1. ResNets are proved to converge to a Neural ODE which is obtained as a surrogate model to explain ResNet predictions. And then the symplectic adjoint method instead of the naive backpropagation algorithm is used to obtain a more accurate calculation of the decision features with less memory and lower computational cost than other explanation approaches.

2. The proposed method has class sensitivity not only in the deep layer but also in the input layer. The gradient decomposition produces an explanation map for the Neural ODE in the input layer that is sufficiently close to the explanation of ResNets under certain conditions, based on a trade-off between faithfulness and intelligibility.

3. The faithfulness of the proposed explanation method is quantitatively evaluated through the evaluation indicators of the deletion and insertion metrics. Experiments results show that the proposed approach has higher faithfulness with lower computational cost than others, and it is effective for troubleshooting and optimizing a model by the explanation.

The remainder of the paper is structured as follows: In Section 2, the related work is presented, while the definitions of ResNet and Neural ODE, and the relationship between ResNet and Neural ODE are introduced in Section 3. Section 4 proposes the symplectic adjoint method. The faithfulness of the explanation between ResNet and the surrogate model - Neural ODE is demonstrated in Section 5. In Section 6, the experimental results

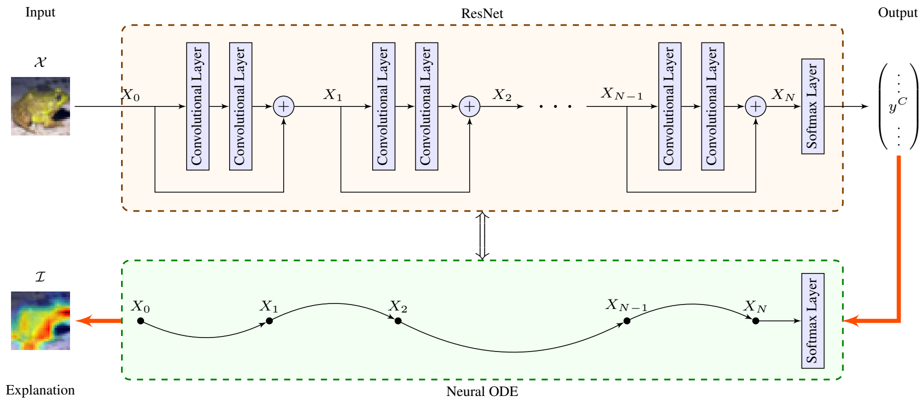

Fig. 1. Overview of our proposed method

are discussed, and finally, the concluding remarks and future directions are depicted in Section 7.

2. Related works

With the widespread application of ResNets, making them interpretable is the impelling priority to be solved to make them trustworthy. Recently, many researches on explicitly explaining the decision attribution of models have emerged. Gradient-based backpropagation methods, CAM-based methods, and Surrogate-based methods are the three major methods.

Gradient-based backpropagation methods. These approaches compute and visualize the gradient of the output concerning the input features for the target class to interpret the contribution of the input features for the prediction result. Simonyan et al. [24] proposed an explanation method to compute the gradient by the backpropagation algorithm to visualize the saliency maps, highlighting the importance of the decision features of the target class. Considering that the saliency maps obtained by the approach are noisy, Guided-BP [27] only backpropagates positive error signals during the process of gradient backpropagation, which is helpful to explain the positive influence of each neuron in the deep network on the input image. Instead of only computing the gradient of the output to the current input, IntegratedGrad [29] calculates the integral of the gradient of the input scaled up from some starting value to the current value. In addition, SmoothGrad [26] adds Gaussian noise to the sample to be interpreted to obtain similar samples, and then the backpropagation algorithm is implemented to generate a saliency map for each input sample. Finally, the average of all the saliency maps of the similarity-generated samples is an explanation of ResNet's decisions with less visual noise. Although the gradient-based explanation methods have theoretical analysis to compute the gradient to make it easy to understand the contribution of the input features, the explanations lack semantic intelligibility and the gradient obtained by the backpropagation algorithm has vanishing and saturation issues, which are the main reason to generate inaccurate explanations.

CAM-based methods. For better visual performance than gradient-based backpropagation methods, CAM-based methods calculate and visualize the linearly weighted combination of the gradient of the target class and feature maps of the last convolutional layer to locate meaningful image regions to explain important decision features. In [34], the fully connected layer is replaced by a global average pooling layer to project the weight of the output for the target class to the weighted combination of the last convolution layer. And the class activation map is obtained to explain the region of the input image that the model mainly depends on when making classification decisions. Without modifying the architecture of the network, Grad-CAM [23] calculates the average gradients of feature maps in the last convolutional layer for the target class as the weight of feature maps to obtain gradient-weighted class activation maps. To obtain more exact explanation results, Grad-CAM++ [3] uses higher-order gradients instead of the average gradient in Grad-CAM as the weight of feature maps. But the gradient vanishing and saturation issue during the backpropagation may also result in a trustless explanation, sometimes image regions with higher weight values contribute less to the target class. To get rid of the gradient, Score-CAM [31] applies feature maps as a mask on the original image as the input image to obtain the forward passing score as the weight of each feature map on the target class instead of the gradients, finally computes the linear combination of the weights and feature maps. Furthermore, for less visual noise, Group-CAM [33], Augmented Score-CAM [11] and FIMF Score-CAM [15] performs meaningful perturbations on initial masks, which are then fed into the network to obtain new weights. However, these weights are obtained entirely based on retraining, which requires high computational costs, and the faithfulness of the explanation cannot be demonstrated due to the lack of theoretical analysis.

Surrogate-based methods. Surrogate-based methods attempt to explain the local or global decision-making basis of the initial network by using a simple interpretable model as a surrogate model. By perturbing the input samples and constructing a local linear model as a model-independent local explanation method LIME to show the image region that is highly sensitive for the target score. While LIME cannot accurately explain neural networks such as Recurrent Neural Networks (RNN), Guo, et al. [5] proposed a high-fidelity explanation method LEMNA suitable for security applications, and a simple regression model is trained to approximate the local decision boundaries of RNN. Then, by introducing the fused lasso regularization to deal with the feature-dependent problem in RNN, LEMNA effectively improves the approaches such as LIME in the fidelity of the explanation. Although LIME and LEMNA are simple to understand, random perturbation and feature selection methods lead to unstable explanations. Bramhall et al. [2] proposed the quadratic approximation framework QLIME redefine the linear representation by LIME as a quadratic relationship, extending the flexibility in non-linear cases and improving the accuracy of explanations. However, there is no theoretical analysis to guarantee the consistency of the explanation between the initial network and the surrogate model.

All these methods visualize the decision features of ResNets and help explain the mechanism of ResNets, but there is no theoretical analysis to guarantee the faithfulness of the explanation. In this paper, the advantages of gradient-based methods and CAM-based methods are integrated, on a trade-off between explanation faithfulness and intelligibility, an explainable Neural ODE, which is proved to be the convergent model of ResNets, is built as a surrogate model to explain ResNets to improve the faithfulness of the explanation and reduce computational cost.

3. ResNets and Neural ODE

In this section, we first introduce the definitions of ResNets, and a neural ordinary differential equation network, namely Neural ODE, is obtained to be a convergent model of ResNets, i.e. ResNets is sufficiently close to the Neural ODE. Thus, the Neural ODE can be regarded as a surrogate model to explain ResNets predictions.

3.1. Definitions of ResNets

First, we generalize the mathematical definitions of ResNets $\mathcal{R}_N$. Suppose that an input $\boldsymbol{x} \in \mathbb{R}^d$ and its corresponding class label $\boldsymbol{y} \in \mathbb{R}^s$ are given for a ResNet. The input layer of ResNet is usually composed of $\mathcal{C}(\boldsymbol{x}) \triangleq \boldsymbol{X}_0$ which is obtained by the convolution operation $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^m$, and then it is composed of a series of residual blocks. One residual block is composed of the identity mapping part and the residual part, which can be described as

$$\boldsymbol{X}_{l+1} = \boldsymbol{X}_l + \lambda f\left(\boldsymbol{X}_l, \boldsymbol{\theta}_l^{(N)}\right), l = 0, 1, \ldots, N-1 \tag{1}$$

where $\boldsymbol{\theta}_l^{(N)}$ is the model parameters of the l-th residual block, N represents the number of residual blocks, $\lambda = T/N$, T is a given constant. $\boldsymbol{X}_l = \left(\boldsymbol{X}_l^{(i)}\right)_{m\times 1} \in \mathbb{R}^m$. And the residual part is generally composed of two convolution operations. To simplify the discussion, let

$$f\left(\boldsymbol{X}_l, \boldsymbol{\theta}_l^{(N)}\right) = \sigma\left(\boldsymbol{\omega}_l^{(N)} \boldsymbol{X}_l + \boldsymbol{b}_l^{(N)}\right), l = 0, 1, \ldots, N-1 \tag{2}$$

where $\sigma(\cdot)$ indicates the ReLU activation function, $\boldsymbol{\omega}_l^{(N)} \in \mathbb{R}^{m\times m}$ is the weight matrix and $\boldsymbol{b}_l^{(N)} \in \mathbb{R}^m$ is the bias.

Finally, let $\boldsymbol{\theta}^{(N)} = \left\{\boldsymbol{\theta}_l^{(N)}\right\}_{l=0}^{N-1}$, $\boldsymbol{X}_N$ is the output of the last residual block and it is operated by a fully connected layer and Softmax normalization operation ψ to obtain the classification prediction probability

$$\boldsymbol{y}_N = \boldsymbol{y}_N\left(\boldsymbol{\theta}^{(N)}\right) = \psi\left(\boldsymbol{\omega}\boldsymbol{X}_N + \boldsymbol{b}\right) \tag{3}$$

where $\boldsymbol{\omega} \in \mathbb{R}^{s\times m}$ is the weight matrix and $\boldsymbol{b} \in \mathbb{R}^s$ is the bias.

Consequently, if the input and label pairs $\left\{\boldsymbol{x}^i, \boldsymbol{y}^i\right\}_{i=1}^n$ of n samples are given, the learning process of ResNets $\mathcal{R}_N$ can be described as the following optimization problem

$$\begin{aligned} \min_{\theta^{(N)}} \quad & \varepsilon_N = \frac{1}{n}\sum_{i=1}^{n} L\left(\boldsymbol{y}^i, \boldsymbol{y}_N^i\left(\theta^{(N)}\right)\right) + R\left(\boldsymbol{\theta}^{(N)}\right) \\ & \text{s.t. } \boldsymbol{X}_{l+1}^i = \boldsymbol{X}_l^i + \lambda f\left(\boldsymbol{X}_l^i, \boldsymbol{\theta}_l^{(N)}\right) \\ & \boldsymbol{X}_0^i = C\left(\boldsymbol{x}^i\right) \\ & l = 0, 1, \ldots, N-1, i = 1, \ldots, n \end{aligned} \tag{4}$$

where $L\left(\boldsymbol{y}^i, \boldsymbol{y}_N^i\right)$ is the loss function between $\boldsymbol{y}^i$ and $\boldsymbol{y}_N^i$, R is the regularization term.

3.2. Relationship between ResNet and Neural ODE

As observed in [32,6,16], Eq. (1) is the Euler discretization of the ordinary differential equation

$$\frac{dX(t)}{dt} = f(X(t), \theta(t)), t \in [0, T] \tag{5}$$

when the initial condition $X(0) = \boldsymbol{X}_0$ is satisfied and the time step λ is selected.

Let $t_l = Tl/N$, the network obtained by replacing the residual blocks $\boldsymbol{X}_{l+1} = \boldsymbol{X}_l + \lambda f\left(\boldsymbol{X}_l, \boldsymbol{\theta}_l^{N)}\right)$ in the residual network $\mathcal{R}_N$ with the ordinary differential equation $\dot{X}(t) = f(X(t), \theta(t)), t \in [t_l, t_{l+1}]$ is called the neural ordinary differential equation network (Neural ODE) $\mathcal{D}$. Suppose that the output of the Neural ODE $\mathcal{D}$ is

$$\boldsymbol{y}_T = \boldsymbol{y}_T(\theta) = \psi(\boldsymbol{\omega} X(T) + \boldsymbol{b}) \tag{6}$$

Therefore, the learning process of the Neural ODE $\mathcal{D}$ can be described as the following optimization problem

$$\begin{aligned} \min_{\theta} \quad & \varepsilon = \frac{1}{n} \sum_{i=1}^{n} L\left(y^i, y_T^i(\theta)\right) + R(\theta) \\ & \text{s.t. } \dot{X}^i(t) = f\left(X^i(t), \theta(t)\right), t \in [0, T] \\ & \quad X^i(0) = X_0^i, i = 1, \ldots, n \end{aligned} \tag{7}$$

Remark 1. As observed in [8,30], If the functions ψ and L are continuous, σ is Lipschitz continuous and $\sigma(0) = 0$, then for $N \to \infty$, the optimal solution of Eq. (4) converges to the optimal solution of Eq. (7). Moreover, assume that $f(t, X)$ be continuous, and $|f| \leq p$ satisfy the Lipschitz condition

$$|f(t, \boldsymbol{X}) - f(t, \boldsymbol{Y})| \leq k|\boldsymbol{X} - \boldsymbol{Y}| \tag{8}$$

on $D = \{(t, \boldsymbol{X}) | 0 \leq t \leq T, |\boldsymbol{X} - \boldsymbol{X}_0 | \leq \eta\}$. Suppose that $f(t, \boldsymbol{X})$ is differentiable with respect to t and $\boldsymbol{X}$, and $|\partial f / \partial \boldsymbol{X}| \leq k, |\partial f / \partial t| \leq q$, If $T \leq \eta / p$, then it can be easily proved that when $N \to \infty, \boldsymbol{X}_N$ converges to the value at the solution $X(T)$ of Eq. (5) when $t = T$, and the error estimate is

$$|X(T) - \boldsymbol{X}_N| \leq \left(\frac{q}{k} + p\right) \frac{T}{N} \left(e^{kT} - 1\right) \tag{9}$$

In addition, the output $\boldsymbol{y}_N = \psi\left(\omega \boldsymbol{X}_N + b\right)$ of the residual network $\mathcal{R}_N$ and the output $y_T = \psi(\omega X(T) + b)$ of the Neural ODE $\mathcal{D}$ satisfy that

$$\|y_T - y_N\| = o\left(\frac{1}{N}\right) \tag{10}$$

Thus, we call that ResNet $\mathcal{R}_{\mathrm{N}}$ converges to the Neural ODE $\mathcal{D}$, that is, the Neural ODE $\mathcal{D}$ can be sufficiently approximated with ResNet $\mathcal{R}_{\mathrm{N}}$ when N is large enough. Therefore, in the following, the Neural ODE is applied to explain the decision attribution of the ResNet $\mathcal{R}_{\mathrm{N}}$.

4. Symplectic Adjoint Method

Since the Neural ODE can be regarded as a surrogate model to explain the influence of the input features for the ResNets' prediction, this section presents the symplectic adjoint method, which can be used to calculate the gradient that exactly characterizes the contribution of input features for the target class.

First, we discuss the influence of the perturbation for a given input on the prediction of Neural ODE. For the differential system Eq. (5), when the initial value $X(0) = \boldsymbol{X}_0$ is perturbed by τ to be $X'(0) = \boldsymbol{X}_0 + \tau$, the corresponding perturbed solution $X'(t)$ satisfies

$$X'(t) = X(t) + \delta(t) + o(\tau), \tau \to 0 \tag{11}$$

where $\delta(t)$ is the variational variable which solves the variational equation

$$\frac{d\delta(t)}{dt} = \frac{\partial f(X(t), \theta(t))}{\partial X(t)}\delta(t) \tag{12}$$

for $\delta(0) = I$.

Remark 2. As observed in [8], suppose that $f(X(t), \theta(t))$ is continuous and differentiable with respect to X, then there exists $\partial X(t)/\partial \boldsymbol{X}_0$ and $\partial X(t)/\partial \boldsymbol{X}_0 = \delta(t)$, where the variational variable $\delta(t)$ satisfies the variational equation Eq. (12).

According to Remark 2,, the gradient of $X(T)$ with respect to initial values $\boldsymbol{X}_0$ is $\partial X(T)/\partial \boldsymbol{X}_0 = \delta(T)$. Furthermore, we consider the adjoint equation of Eq. (12)

$$\frac{d\alpha(t)}{dt} = -\left(\frac{\partial f(X(t), \theta(t))}{\partial X(t)}\right)^T \alpha(t) \tag{13}$$

Where the adjoint variable $\alpha(t)$ is the gradient of $L(X(T))$ with respect to $X(t)$ and $\alpha(T) = \partial L(X(T))/\partial X(T)$, and the conservation property Eq. (14) for the solutions of variational equation Eq. (12) and adjoint equation Eq. (13) below holds

$$\alpha(T)^{\mathrm{T}}\delta(T) = \alpha(0)^{\mathrm{T}}\delta(0) \tag{14}$$

Aim at computing the gradients of $L(X(T))$ with respect to the initial condition $\boldsymbol{X}_0$ and the parameters $\boldsymbol{\theta}$ for the Neural ODE $\mathcal{D}$, [4] proposed the adjoint method to solve the gradient, but this approach cannot obtain the accurate gradient and requires high computational costs to suppress numerical errors, we present the symplectic adjoint Method, which solves the adjoint equation by the Symplectic Runge-Kutta method to obtain the exact gradient and the final trained network is denoted as the Neural ODE $\mathcal{D}_N$. The symplestic adjoint method consumes much less memory consumption, but performs faster computation and more robust to rounding errors than the naive backpropagation algorithm.

Firstly, the ordinary differential equation Eq. (5) is discretized by the Runge-Kutta method

$$\boldsymbol{X}'_{l+1} = \boldsymbol{X}'_l + \lambda \sum_{i=1}^{s} \boldsymbol{b}_i f_{l,i}, l = 0, 1, \ldots, N-1 \tag{15}$$

Where

$$f_{l,1} = f\left(t_l, \boldsymbol{X}_l'\right), f_{l,i} = f\left(t_l + c_i\lambda, \boldsymbol{X}_l' + \lambda \sum_{j=1}^{i-1} a_{i,j} f_{i,j}\right) \tag{16}$$

$i = 2, \ldots, s$, $a_{i,j}, b_i, c_i$ are summarized as the Butcher tableau [8,7,22], and there is $\left\|X(T) - \boldsymbol{X}_N'\right\| = o(1/N)$ for $N \to \infty$. Suppose that the output of the Neural ODE $\mathcal{D}_N$ is $\boldsymbol{y}_N' = \psi\left(\boldsymbol{\omega}\boldsymbol{X}_N' + \boldsymbol{b}\right)$.

Remark 3. ([7,1]) When the ordinary differential equation Eq. (5) is discretized by the Runge-Kutta method in Eq. (15), the variational equation Eq. (12) is discretized by the same Runge-Kutta method as follows

$$\boldsymbol{\delta}_{i+1} = \boldsymbol{\delta}_i + \lambda \sum_{i=1}^{s} b_i p_{l,i}, l = 0, 1, \ldots, N-1 \tag{17}$$

Where δ_i is the discretization of the variational variable $\delta(t)$ in Eq.(12) and

$$\begin{aligned} p_{l,1} &= \frac{\partial f}{\partial \boldsymbol{X}}\left(t_l, \boldsymbol{X}_l\right) \\ p_{l,i} &= \frac{\partial f}{\partial \boldsymbol{X}}\left(t_l + c_i\lambda_l, \delta_l + \lambda \sum_{j=1}^{i-1} a_{i,j} p_{i,j}\right), i = 2, \ldots, s \end{aligned} \tag{18}$$

Moreover, assume the adjoint variable $\boldsymbol{\alpha}_N = \partial L\left(\boldsymbol{X}_N'\right)/\partial \boldsymbol{X}_N'$, to obtain the exact backpropagation gradient for the Neural ODE $\mathcal{D}_N$ efficiently, the Symplectic Runge-Kutta method solves the adjoint equation by the Symplectic Runge-Kutta method as follows

$$\begin{aligned} \boldsymbol{\alpha}_{l+1} &= \boldsymbol{\alpha}_l + \lambda_l \sum_{i=1}^{s} \tilde{b}_i \phi_{l,i}, l = 0, 1, \ldots, N-1 \\ \phi_{l,i} &= -\left(\frac{\partial f}{\partial \boldsymbol{X}}\left(t_l + c_i\lambda, \boldsymbol{X}_l' + \lambda \sum_{j=1}^{i-1} a_{i,j} f_{l,j}\right)\right)^T \xi_{l,i} \\ \xi_{l,i} &= \begin{cases} \boldsymbol{\alpha}_l + \lambda \sum_{j=1}^{s} \tilde{b}_j \left(1 - \frac{a_{j,i}}{b_i}\right) \phi_{l,j}, & \text{if } i \notin \mathrm{I}_0 \\ -\sum_{j=1}^{s} \tilde{b}_j a_{j,i} \phi_{l,j}, & \text{if } i \in \mathrm{I}_0 \end{cases} \end{aligned} \tag{19}$$

where

$$\tilde{b}_i = \begin{cases} b_i, \text{ if } i \notin \mathrm{I}_0 \\ \lambda, \text{ if } i \in \mathrm{I}_0, \end{cases} \mathrm{I}_0 = \{i \mid i = 1, \ldots, s, b_i = 0\} \tag{20}$$

Finally, the gradient that exactly characterizes the contribution of input features for the output $\boldsymbol{\alpha}_0 = \partial L\left(\boldsymbol{X}_N'\right)/\partial \boldsymbol{X}_0$ can be obtained by the backpropagation iteration.

5. Explaining ResNet Predictions

In this section, we show that our proposed method guarantees the faithfulness of the explanation between ResNet and the surrogate model - Neural ODE. First, the decision feature and the explanation map of an input to the target class for Neural ODE are generated via the symplectic adjoint method. Then, we prove that the explanations of Neural ODE can be sufficiently approximate to the true behavior of ResNet when N is large enough.

5.1. Understanding Neural ODE Predictions via symplectic adjoint method

Since it is well known that the deep convolution layer has high-level semantics, the previous CAM-based explanation methods focused on the analysis of the feature maps in the last convolution layer, while the feature maps in the shallow layer is noisy and lacks semantic intelligibility. However, in theory, to explain the importance of the input features for the output prediction score of the target class, the gradient of the target class should be back-propagated to the input layer to highlight the image region that has a great influence on the prediction. In this paper only positive influence on the prediction is considered, as observed in [14], the definition of the explanation map based on the gradient decomposition of the input layer is given as follows.

Definition 1. *Assume that any instance $\boldsymbol{x}$ and the trained Neural ODE $\mathcal{D}_N$ is given, the input layer of $\mathcal{D}_N$ is $\boldsymbol{X}_0$ and the output prediction vector is y, the i-th row vector $\partial y_i/\partial \boldsymbol{X}_0$ of $\partial y/\partial \boldsymbol{X}_0$ is called the decision feature about $\boldsymbol{X}_0$ that $\boldsymbol{x}$ belongs to class i. If the derivative of $\boldsymbol{y}$ with respect to the input $\partial y/\partial \boldsymbol{X}_0$ is obtained, then $\mathcal{D}_N$ is called explainable and the explanation map that x belongs to class i for $\mathcal{D}_N$ is*

$$\boldsymbol{I} = \operatorname{ReLU}\left(\frac{\partial y_i}{\partial \boldsymbol{X}_0}\right)^{\mathrm{T}} \circ \boldsymbol{X}_0 \tag{21}$$

where $\circ$ denotes Hadamard product.

Considering the problem of the gradient $\partial y_i/\partial \boldsymbol{X}_0$ computed by previous research, we first introduce a novel gradient solution method to ensure the accuracy of the explanation. The symplectic adjoint method for Neural ODE which consumes less memory than the naive backpropagation algorithm and has faster computational efficiency, is used to compute the gradient to explain the Neural ODE $\mathcal{D}_N$. By section 4, according to the symplectic adjoint Method, the initial $\boldsymbol{\alpha}_0 = \partial L\left(\boldsymbol{X}'_N\right)/\partial \boldsymbol{X}_0$ has been obtained by the iteration of the adjoint variable $\boldsymbol{\alpha}_N = \partial L\left(\boldsymbol{X}'_N\right)/\partial \boldsymbol{X}'_N$ into Eq. (19). On the other hand, as observed in [22], if α_0, δ_N are respectively the solutions for Eq. (19) and Eq. (17), then we have

$$\boldsymbol{\alpha}_{\mathrm{N}}{}^{\mathrm{T}}\boldsymbol{\delta}_{\mathrm{N}} = \boldsymbol{\alpha}_0{}^{\mathrm{T}}\boldsymbol{\delta}_0 \tag{22}$$

where $\delta(0) = \boldsymbol{I}$. Therefore, it is not necessary to spend much more time on iterative calculations in Eq. (12), the variational variable $\delta_N = \partial \boldsymbol{X}'_N/\partial \boldsymbol{X}_0$ can be solved from the Eq. (22) and

$$\frac{\partial \boldsymbol{y}'_N}{\partial \boldsymbol{X}_0} = \frac{\partial \boldsymbol{y}'_N}{\partial \boldsymbol{X}'_N}\boldsymbol{\delta}_N = \psi'\boldsymbol{\omega}\boldsymbol{\delta}_N \tag{23}$$

Thus the i - th row vector $\left(\psi'\boldsymbol{\omega}\boldsymbol{\delta}_N\right)_i$ of $\psi'\boldsymbol{\omega}\boldsymbol{\delta}_N$ is called the decision feature about $\boldsymbol{X}_0$ that $\boldsymbol{x}$ belongs to class i for Neural ODE $\mathcal{D}_N$.

Therefore, by Definition 1, we have that the Neural ODE $\mathcal{D}_N$ is explainable, and the explanation map that $\boldsymbol{x}$ belongs to class i for $\mathcal{D}_N$ is

$$\boldsymbol{I} = \operatorname{ReLU}\left(\left(\psi'\boldsymbol{\omega}\boldsymbol{\delta}_N\right)_i\right)^{\mathrm{T}} \circ \boldsymbol{X}_0 \tag{24}$$

where $\circ$ denotes Hadamard product. Since we are only interested in features that have a positive impact on the target class and are good for visualization, the function is to filter the negative impact. The explanation map I is can explain the image region where the Neural ODE $\mathcal{D}_N$ is for image recognition has a high influence on the prediction result of the input image belonging to class i.

5.2. Relationship between explanations of ResNet and Neural ODE

To show the consistency of the explanation between ResNet and the Neural ODE, the metrics of the approximation of the two models is first introduced as follows [13].

Definition 2. *For a neural network $\mathcal{D}_n$, if there is a neural network $\mathcal{D}'_n$ which is explainable and the explanation map that $\boldsymbol{x}$ belongs to class i for $\mathcal{D}'_n$ is $\boldsymbol{I}$, and for $\forall \varepsilon > 0, \exists K \in \mathbb{N}^+$, when $n > K$, there is*

$$\kappa\left(\mathcal{D}_n, \mathcal{D}'_n\right) = \|\boldsymbol{y} - \boldsymbol{y}'\| < \varepsilon \tag{25}$$

for any instance $\boldsymbol{x}$, where $\boldsymbol{y}$ and $\boldsymbol{y}'$ are the prediction vectors obtained by the model $\mathcal{D}_n$ and $\mathcal{D}'_n$ for the input $\boldsymbol{x}$ respectively, then we define $\mathcal{D}'_n$ as an approximate interpretable model of $\mathcal{D}_n$ when n is large enough.

According to Definition 2, it is obvious that when $\kappa\left(\mathcal{D}_n, \mathcal{D}'_n\right)$ is smaller, the model approximation degree of the two models is higher and $\mathcal{D}'_n$ is considered to be a better explainable model of $\mathcal{D}_n$. In the following, we demonstrate that the Neural ODE $\mathcal{D}_N$ is the approximate explainable model of RessNet $\mathcal{R}_N$.

Theorem 1. *For the ResNet $\mathcal{R}_N$ and the corresponding Neural ODE $\mathcal{D}_N$, If the functions L and ψ are continuous, the function σ is Lipschitz continuous and $\sigma(0) = 0$, $f(X(t), \theta(t))$ is continuous and differentiable with respect to $\boldsymbol{X}$, then for $\forall \varepsilon > 0, \exists K \in \mathbb{N}^+$, when $N > K$, there is $\kappa\left(\mathcal{D}_N, \mathcal{R}_N\right) < \varepsilon$ for any $\boldsymbol{x}$.*

Proof. For any input $\boldsymbol{x}$, it is known that Eq. (1) in ResNet $\mathcal{R}_N$ is the Euler discretization of the ordinary differential equation Eq. (5) when the initial condition $X(0) = X_0$ is satisfied and the time step λ is selected. By Remark 1, if the functions ψ and L are continuous, σ is Lipschitz continuous and $\sigma(0) = 0$, then the numerical solution obtained from Eq. (1) converges to the exact solution of Eq. (5). So X_N converges to the solution $X(T)$ of Eq. (5) when $t = T$, that is, $\lim_{N\to\infty}\left(\boldsymbol{X}_N - X(T)\right) = 0$, then for $\forall \varepsilon' > 0, \exists M_1 \in \mathbb{N}^+$, when $N > M_1$, there is $\|\boldsymbol{X}_N - X(T)\| < \varepsilon'$.

Moreover, let

$$\boldsymbol{\omega}\boldsymbol{X}_N + \boldsymbol{b} = \left[x_N^{(1)}, x_N^{(2)}, \ldots, x_N^{(s)}\right] \in \mathbb{R}^s \tag{26}$$

$$\boldsymbol{\omega}\boldsymbol{X}(T) + \boldsymbol{b} = \left[x^{(1)}, x^{(2)}, \ldots, x^{(s)}\right] \in \mathbb{R}^s \tag{27}$$

then when $N > M_1$, there is $\left|x_N^{(i)} - x^{(i)}\right| < \boldsymbol{\omega}\varepsilon'$. Moreover, the output prediction vectors obtained by ResNet $\mathcal{R}_N$ and Neural ODE $\mathcal{D}$ are

$$y_N = \psi\left(\boldsymbol{\omega}\boldsymbol{X}_N + \boldsymbol{b}\right), y_T = \psi(\boldsymbol{\omega}X(T) + \boldsymbol{b}) \tag{28}$$

Let $\boldsymbol{y}_N = \left[y_N^{(1)}, y_N^{(2)}, \ldots, y_N^{(s)}\right], \boldsymbol{y}_T = \left[y_T^{(1)}, y_T^{(2)}, \ldots, y_T^{(s)}\right] \in \mathbb{R}^s$, then there exists $c_i \in \mathbb{R}$

$$\begin{aligned} \left|y_N^{(i)} - y_T^{(i)}\right| &= \left|\frac{e^{x_N^{(i)}}}{\sum_{j=1}^{s} e^{x_N^{(j)}}} - \frac{e^{x^{(i)}}}{\sum_{j=1}^{s} e^{x^{(j)}}}\right| \\ &\leq \left|e^{x_N^{(i)}} - e^{x^{(i)}}\right| \\ &\leq \omega c_i \varepsilon' \end{aligned} \tag{29}$$

Let $\varepsilon = 2\omega \sum_{i-1}^{s} c_i \varepsilon'$, so we have

$$\kappa\left(\mathcal{R}_N, \mathcal{D}\right) = \|\boldsymbol{y}_N - \boldsymbol{y}_T\| < \omega \sum_{i=1}^{s} c_i \varepsilon' = \varepsilon/2 \tag{30}$$

Therefore, $\forall \varepsilon > 0, \exists M_1 \in \mathbb{N}^+$, when $N > M_1$, there is $\kappa\left(\mathcal{R}_{\mathrm{N}}, \mathcal{D}\right) < \varepsilon/2$ for any $\boldsymbol{x}$.

Similarly, $\forall \varepsilon > 0, \exists M_2 \in \mathbb{N}^+$, when $N > M_2$, there is $\kappa\left(\mathcal{D}, \mathcal{D}_N\right) < \varepsilon$ for any $\boldsymbol{x}$, assume that the output prediction vector obtained by the Neural ODE $\mathcal{D}_N$ is

$$\boldsymbol{y}_N' = \psi\left(\boldsymbol{\omega} \boldsymbol{X}_N' + \boldsymbol{b}\right) = \left[y_N^{(1)}, y_N^{(2)}, \ldots, y_N^{(s)}\right] \in \mathbb{R}^s \tag{31}$$

then let $K = \max\left\{M_1, M_2\right\} \in \mathbb{N}^+$, when $N > K$, we have

$$\kappa\left(\mathcal{R}_{\mathrm{N}}, \mathcal{D}_N\right) \leq \kappa\left(\mathcal{R}_{\mathrm{N}}, \mathcal{D}\right) + \kappa\left(\mathcal{D}, \mathcal{D}_N\right) < \varepsilon \tag{32}$$

Overall, the Neural ODE $\mathcal{D}_N$ is proved to be an explainable model sufficiently close to ResNet $\mathcal{R}_N$ when N is large enough, and the explanation map $\boldsymbol{I} = \mathrm{ReLU}\left(\left(\psi' \omega \delta_N\right)_i\right)^{\mathrm{T}} \circ \boldsymbol{X}_0$ can explain the image region where the ResNet $\mathcal{R}_N$ for image recognition has high influence on the prediction result of the input image belonging to class i. Theorem 1 ensures the faithfulness of the explanation between ResNet and the surrogate model, and since the symplectic adjoint method instead of the naive backpropagation algorithm is used, the proposed explanation method has lower computational cost than other explanation approaches.

6. Experimental Implementation and Evaluation

In this section, we evaluate the effectiveness of the explanation method for ResNet in this paper. First, we visualize the saliency maps obtained by the CAM-based explanation of ResNet with the gradient via the symplectic adjoint method to compare other CAM-based methods with the gradient obtained by the naive backpropagation algorithm. Second, we demonstrate the faithfulness of the proposed explanation method for ResNet by conducting deletion and insertion tests to compare with the ones in Grad-CAM and Group-CAM. Moreover, we present the proposed explanation method, which is more efficient than others and effective for users to find out the reason for the wrong decision on some samples in the prediction. Finally, by modifying the training set and retraining, the effectiveness of the explanation for debugging and optimizing a model can be validated.

In the following experiments, the model architecture we performed is a pre-trained ResNet with six standard residual blocks [4] and the publicly available classification dataset Cifar10 is used. For the input images, all images are resized to 3 × 32 × 32, and then transformed to tensors and normalized to the range [0, 1]. The performance of the proposed explanation method via the symplectic adjoint method and existing methods is evaluated by PyTorch 1.7.0 [20] and demonstrated by extending the adjoint method implemented in the package torchdiffeq 0.8.1 [34].

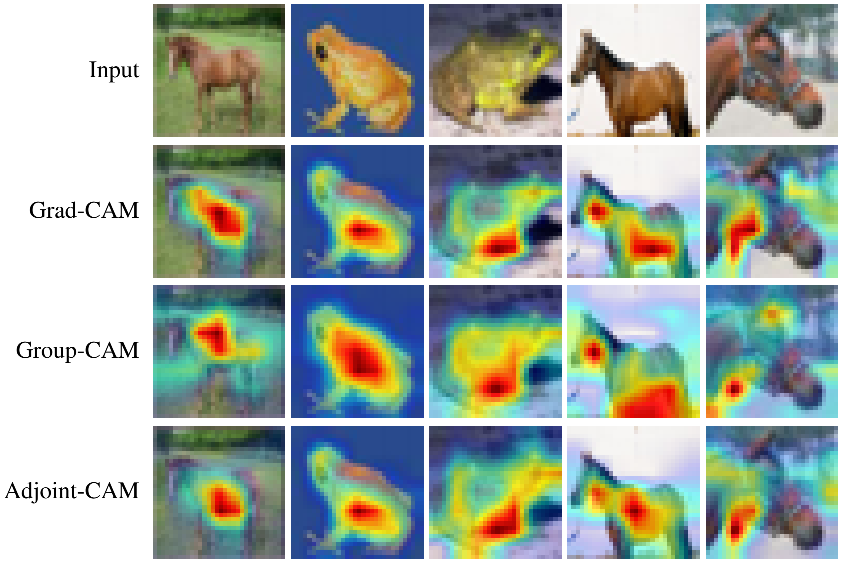

Fig. 2. Visualization results of Grad-CAM, Group-CAM, and Adjoint-CAM. The class activation maps are similar for the various explanation methods in the last convolutional layer, especially for Grad-CAM and Adjoint-CAM

6.1. CAM-based explanation of ResNet via symplectic adjoint method and naive backpropagation Algorithm

Since the CAM-based explanation methods have a good performance on class discriminative visualization and localization ability, they have been widely applied to visualize the task of locating objects in images. For instance, Grad-CAM calculates the average of the naive gradients of feature maps in the last convolutional layer for the target class as the weight of feature maps to obtain gradient-weighted class activation maps to highlight important features. In this section, the method that generates the weighted class activation maps by computing the average of gradients of feature maps in the last convolutional layer for the target class via the symplectic adjoint method as the weight of feature maps is called Adjoint-CAM, and the weighted class activation maps obtained by Adjoint-CAM can be demonstrated to be similar to those generated by Grad-CAM which computes the average of gradients via the naive backpropagation algorithm. Therefore, the proposed method also has a good performance on the visualization of class-conditional localization of objects in the last convolutional layer for the target class.

As shown in Remark 1, the output of the last residual block of ResNet is approximate to the output of the corresponding Neural ODE at $t = T$,which means the feature maps of ResNet and Neural ODE in the last convolutional layer are similar. On the other hand, the output of ResNet and Neural ODE is Eq. (3) and Eq. (6) respectively, then it is obvious that the gradients of feature maps in the last convolutional layer for the target class via the symplectic adjoint method for Neural ODE are the same as the gradients via the naive backpropagation algorithm for ResNet. Therefore, the weighted class activation maps obtained by Adjoint-CAM which computes the weight via the symplectic adjoint method approximate with that generated by Grad-CAM to locate the important region of given images. We visualize the depth-wise saliency maps through Grad-CAM, Group-CAM, and Adjoint-CAM in Fig. 2. As shown in Fig. 2, the class activation maps are similar for the various methods and localize the target object well in the last convolutional layer.

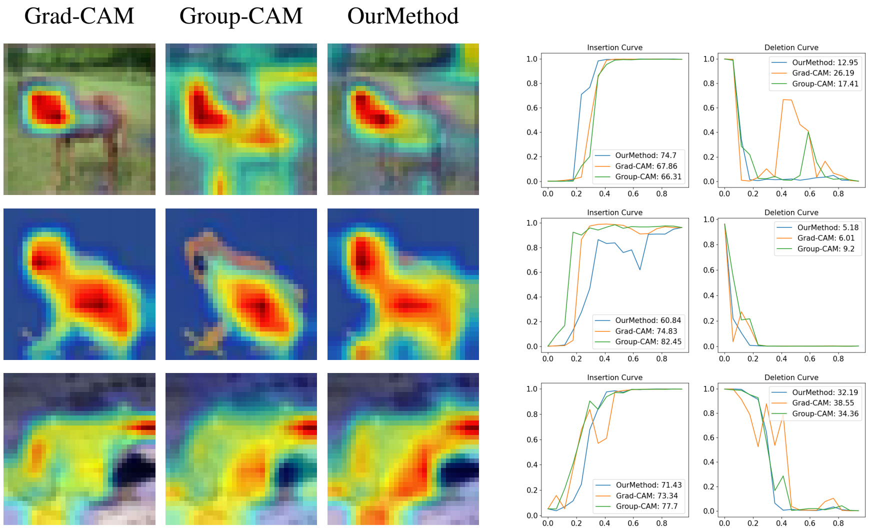

Fig. 3. The explanation maps generated by Grad-CAM, Group-CAM, and OurMethod for representative images with deletion and insertion curves. In the insertion curve, a more faithful explanation should increase faster, the area under the curve is expected to be large. While in the deletion curve, a more faithful explanation should drop faster, the area under the curve is expected to be small

6.2. Evaluating Faithfulness of Explanations

Although the CAM-based explanation methods outperform other methods on visualization performance in the last convolutional layer, the gradient obtained by the naive backpropagation algorithm in the shallow layer may result in an inaccurate explanation for the gradient vanishing and saturation issue. Sometimes, image regions with higher weight values contribute less to the target class. Therefore, we propose a novel explanation method in which the shallow gradient is obtained exactly by the symplectic adjoint method. To demonstrate the faithfulness of the explanation of our proposed method in this paper is higher than that of other methods, the deletion and insertion metrics that are prepared in [21] are used to qualitatively evaluate the faithfulness of the interpretation methods.

Intuitively, if the pixels obtained from the explanation methods are more important for decisions, the removal of pixels will cause the predicted score for the target class to drop more significantly, which shows that the explanation method is more faithful. Therefore, the deletion metric is a qualitative evaluation metric to measure the decrease in the probability of the predicted class when more and more important pixels obtained by the explanation methods are removed from the original image. A lower area under the probability curve indicates a more faithful explanation. In addition, the insertion metric, which starts with a blurred image, evaluates the increase in the probability of the predicted score for the target class as more and more pixels are introduced, and a higher area under the probability curve means a more faithful explanation.

In detail, for the deletion test, there are several approaches to removing pixels from an image and all of these have different pros and cons. In our experiment, the deletion test gradually replaces 1% of pixels from the original image with a highly blurred one according to the importance of pixels for decisions obtained by the explanation methods until no pixels are left. On the other hand, for the insertion test, we gradually replace 1% pixels of the blurred image with the original version according to the importance of pixels until the image is well recovered. Some examples generated by Grad-CAM, Group-CAM, and our proposed method for the first layer of the ResNet block and the corresponding deletion and insertion curves are illustrated in Fig. 3. Furthermore, for a more general comparison, the average results calculated by area under the probability curve of deletion and insertion tests over 1000 images are demonstrated in Table 1. As shown in Table 1, our proposed method outperforms other CAM methods in terms of deletion AUC compared with gradient-based CAM methods.

Table 1. Comparative evaluation in terms of insertion (higher is better) and deletion (lower is better) scores

	Grad-CAM	Group-CAM	OurMethod
Insertion	0.614	**0.623**	0.616
Deletion	0.161	0.152	**0.132**

6.3. Comparative Evaluation in Terms of Memory and Computation Efficiency

In the experiment, the Neural ODE obtained by replacing residual blocks of ResNet with ODESolve modules is the surrogate explainable model of ResNet and the explanation map is generated by the gradient calculated by the symplectic adjoint method. Table 2 summarizes the test error and number of parameters for ResNet and the Neural ODE on Cifar10, respectively. As shown in Table 2, the Neural ODE has around the same performance as the ResNet with fewer parameters.

Furthermore, the memory consumption and running time for the explanation by Grad-CAM, Grad-CAM++, Group-CAM and the proposed method are shown in Table 3. L

Table 2. Comparative evaluation in terms of the test accuracy and number of parameters for ResNet and neural ode on Cifar10.

	Test Accuracy	Parameters
ResNet	83.89	0.58M
Neural ODE	83.57	0.21M

is the number of layers in the ResNet, and all the feature maps and neuron importance weights are splited into G groups. Grad-CAM and Grad-CAM++ which generate the shallow layer gradients via the naive backpropagation algorithm for the ResNet requires a memory of $O(L)$ for the backpropagation, and compute gradients by the chain rule step by step with roughly computational cost $O(L)$. Group-CAM needs to input the constructed mask back into the network for training to obtain new weights for G rounds, so the computational cost is $O(GL)$. While the symplectic adjoint method has superior performance than the naive backpropagation in terms of the memory consumption and running time. Our proposed method had no more need of storing any intermediate quantities of the backpropagation, only the first and last adjoint variable is required to obtain the explanation map, and then the memory consumption and running time are both $O(1)$. Therefore, as shown in Table 3, the proposed explanation approach outperforms Grad-CAM, Grad-CAM++ and Group-CAM in terms of the memory consumption and computational cost.

Table 3. Comparative evaluation in terms of the memory consumption and running time for Grad-CAM, Grad-CAM++, Group-CAM and our proposed method on cifar10.

	Grad-CAM	Grad-CAM++	Group-CAM	OurMethod
Memory	$O(L)$	$O(L)$	$O(L)$	$O(1)$
Running Time	$O(L)$	$O(L)$	$O(GL)$	$O(1)$

6.4. Effectiveness for Troubleshooting and Optimizing a Model by the Explanation

When we know how the deep learning model thinks, it provides us with the privilege to optimize it. This section illustrates that our proposed approach can explain why ResNet makes a wrong prediction, and then improve the performance of the model pertinently. Intuitively, if the pixels obtained from the explanation methods are more important for prediction, the modification of pixels will cause the probability of the predicted class to decrease more significantly. As shown in Fig.3, the deletion of less than 10% pixels from the original image can generate adversarial images to producing incorrect classification results. Furthermore, Fig.4 shows the explanation maps for two misclassified images with respect to their top-3 predicted classes by the proposed method. As shown in Fig. 4, the first image labeled “horse” is classified as an airplane, the second one labeled “horse” is classified as a frog, and the second column are the explanation maps generated by our method to show the image region has a high influence on the prediction results for the top-1 predicted classes of airplane and frog. It can be seen that ResNet makes a wrong prediction for the two images, mostly in terms of the background of the images. It seems reasonable that the sky is white and the frog is green in the training dataset, but the reason why the model makes a wrong prediction is that it ignores the shape of images.

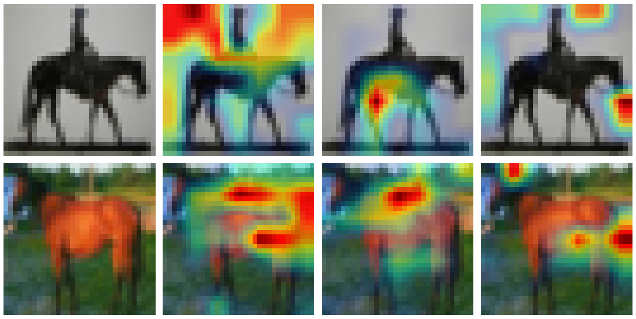

Fig. 4. The explanation maps with respect to top3 predicted class for mis-classified images. The first column is the input images. The prediction score for the first input image with respect to top3 predicted class is 0.6373 (airplane), 0.1715 (horse), 0.1623 (bird), the second is 0.6770 (frog), 0.1336 (automobile), 0.1306 (horse)

Table 4. Comparative evaluation in terms of the test accuracy before and after debugging

	No debugging	After debugging
Test Accuracy	83.89%	87.29%

Therefore, by diagnosing the reason why the model makes a wrong prediction, these mistakes can be avoided with sufficient training. Specifically, for a given misclassified sample, first we identify the corresponding regions in the model that were undertrained. Then some targeted training samples are generated by replacing the values of important pixels with random values to augment the original training data and retrain the model. Due to the addition of new samples, the impact of misleading features is reduced, and target errors are patched without reducing the previous accuracy of the model. We consider the proposed explanation method is practical for optimizing a model by modifying the training set and retraining. After adding 3000 artificially constructed images by the interpretation method to the original training set, ResNet is retrained by the reconstructed training set. As shown in Table 4, the test accuracy of the pre-trained model is 83.89% and the test accuracy of the model is improved to 87.29% after the training of the new training set. Moreover, after retraining, the image labeled ”horse” in Fig.4 can be correctly identified, and the corresponding predicted probabilities are increased to 0.6312 and 0.7065, respectively. Overall, the explanation map obtained by our method is helpful for overall diagnosing and debugging of the model.

7. Conclusion and Future Work

Using the Neural ODE as the surrogate model, we proposed a new way for a more faithful explanation of ResNets predictions. To get over the naive gradient problem and make sure that images regions with higher gradient values contribute more to the target class, the gradients with respect to the input layer are calculated using the symplectic adjoint method. Additionally, the gradient decomposition method yields an explanation map for the Neural ODE at the input layer that can be shown to be sufficiently near to the explanation of ResNets. Quantitative analyses are provided to demonstrate that our method outperforms than other cutting-edge methods in terms of effectiveness and explanation faithfulness, and it is effective for troubleshooting and debugging a model by the explanation. In the future work, we will analyze the stability and generalization ability of ResNet with symplectic adjoint method.

References

1. Bochev, P.B., Scovel, C.: On quadratic invariants and symplectic structure. BIT-Computer Science Numerical Mathematics 34(3), 337–345 (1994)
2. Bramhall, S., Horn, H., Tieu, M., Lohia, N.: Qlime-a quadratic local interpretable model-agnostic explanation approach. SMU Data Science Review 3(1), 4 (2020)
3. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE winter conference on applications of computer vision (WACV). pp. 839–847. IEEE (2018)
4. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. Advances in neural information processing systems 31 (2018)
5. Guo, W., Mu, D., Xu, J., Su, P., Wang, G., Xing, X.: Lemna: Explaining deep learning based security applications. In: proceedings of the 2018 ACM SIGSAC conference on computer and communications security. pp. 364–379 (2018)
6. Haber, E., Ruthotto, L.: Stable architectures for deep neural networks. Inverse problems 34(1), 014004 (2017)
7. Hairer, E., Lubich, C., Wanner, G.: Solving geometric numerical integration: Structure-preserving algorithms (2006)
8. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations. 1, Nonstiff problems. Springer-Vlg (1993)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
10. Hengstler, M., Enkel, E., Duelli, S.: Applied artificial intelligence and trust—the case of autonomous vehicles and medical assistance devices. Technological Forecasting and Social Change 105, 105–120 (2016)
11. Ibrahim, R., Shafiq, M.O.: Augmented score-cam: High resolution visual interpretations for deep neural networks. Knowledge-Based Systems 252, 109287 (2022)
12. Kleppe, A., Skrede, O.J., De Raedt, S., Liestøl, K., Kerr, D.J., Danielsen, H.E.: Designing deep learning studies in cancer diagnostics. Nature Reviews Cancer 21(3), 199–211 (2021)
13. Lei, X., Fan, Y., Li, K.C., Castiglione, A., Hu, Q.: High-precision linearized interpretation for fully connected neural network. Applied Soft Computing 109, 107572 (2021)
14. Lei, X., Fan, Y., Luo, X.L.: On fine-grained visual explanation in convolutional neural networks. Digital Communications and Networks (2022)
15. Li, J., Zhang, D., Meng, B., Li, Y., Luo, L.: Fimf score-cam: Fast score-cam based on local multi-feature integration for visual interpretation of cnns. IET Image Processing 17(3), 761–772 (2023)

16. Ma, C., Wu, L., et al.: Machine learning from a continuous viewpoint, i. Science China Mathematics 63(11), 2233–2266 (2020)
17. Matsubara, T., Miyatake, Y., Yaguchi, T.: Symplectic adjoint method for exact gradient of neural ode with minimal memory. Advances in Neural Information Processing Systems 34, 20772–20784 (2021)
18. Muhammad, K., Ullah, A., Lloret, J., Del Ser, J., de Albuquerque, V.H.C.: Deep learning for safe autonomous driving: Current challenges and future directions. IEEE Transactions on Intelligent Transportation Systems 22(7), 4316–4336 (2020)
19. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 427–436 (2015)
20. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
21. Petsiuk, V., Das, A., Saenko, K.: Rise: Randomized input sampling for explanation of blackbox models. arXiv preprint arXiv:1806.07421 (2018)
22. Sanz-Serna, J.M.: Symplectic runge–kutta schemes for adjoint equations, automatic differentiation, optimal control, and more. SIAM review 58(1), 3–33 (2016)
23. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
24. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
25. Slack, D., Hilgard, S., Jia, E., Singh, S., Lakkaraju, H.: Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. pp. 180–186 (2020)
26. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
27. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)
28. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation 23(5), 828–841 (2019)
29. Sundararajan, M., Taly, A., Yan, Q.: Gradients of counterfactuals. arXiv preprint arXiv:1611.02639 (2016)
30. Thorpe, M., van Gennip, Y.: Deep limits of residual neural networks. Research in the Mathematical Sciences 10(1), 6 (2023)
31. Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., Hu, X.: Score-cam: Score-weighted visual explanations for convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 24–25 (2020)
32. Weinan, E.: A proposal on machine learning via dynamical systems. Communications in Mathematics and Statistics 1(5), 1–11 (2017)
33. Zhang, Q., Rao, L., Yang, Y.: Group-cam: Group score-weighted visual explanations for deep convolutional networks. arXiv preprint arXiv:2103.13859 (2021)
34. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2921–2929 (2016)

Xia Lei received the B.S. degree in Mathematics and Applied Mathematics from Jimei University, China, in 2012; the M.S. degree in Basic Mathematics from Fuzhou University, China, in 2015, the Ph.D. degree of Computer Science and Technology in China

University of Petroleum (Beijing), in 2023. Her current research interests include theories of explainable AI.

Jia-Jiang Lin received Bachelor and Ph.D. degrees in automation from China University of Petroleum (Beijing), in 2014 and 2020, respectively. At present, he is an assistant professor at China University of Petroleum (Beijing). His main research direction is the theoretical solution and numerical solution of the hybrid system optimal control problem in the chemical process and machine learning.

Xiong-Lin Luo (Corresponding author: luoxl@cup.edu.cn) received the Ph.D. degree in China University of Petroleum (Beijing) in 1997. He is a professor at China University of Petroleum (Beijing). Main research directions: control theory and process control, chemical system engineering and machine learning.

Yongkai Fan received Bachelor, Master, and Ph.D. degrees from Jilin University, China, in 2001, 2003, and 2006, respectively. From 2006 to 2009, he was an assistant researcher at Tsinghua University, China. Currently, he is an associate professor at the Communication University of China, He was a visiting scholar in the Department of Computer science and Engineering at Lehigh University in the USA (2015) and a visiting scholar in the Department of Computer Science and Engineering at Penn State University in the USA (2016). He has published more than 50 journal/conference papers in journals and his research interests include theories of data security and software security.

Received: March 10, 2023; Accepted: June 25, 2023.