On the Evaluation Consistency of Attribution-based Explanations

– Supplementary Material –

A Related Work on Attribution Methods

To comprehend decisions made by deep neural networks (DNNs), a variety of attribution methods offer interpretations from diverse perspectives. Existing attribution methods can be roughly categorized into three groups: gradient-based attribution, perturbation-based attribution, and CAM-based attribution.

Gradient-based methods [2, 15, 16, 20, 21] rely on computing the gradient of a model's output with respect to its input. Input⊙Gradient [16] expresses the contribution as the element-wise product of the gradient and the input features. Integrated Gradients [21] computes the integral of the gradient along the path from a baseline input to the input of interest. It then obtains the attribution value by taking the element-wise product of the path integral and the input. Backpropagation-based methods [2, 15, 20] estimate the feature attribution values of intermediate layers and propagate backward layer by layer to obtain the attribution value of the input.

Perturbation-based methods [10–12, 24] infer attribution values by studying the impact of masking inputs on model outputs. RISE [11] highlights the importance of input by observing the values of outputs after input perturbation. Occlusion [24] assigns the attribution value of a patch to the change in output values between its state of masking and un-masking. When the masking state of other variables changes, the Shapley value method [10] estimates the attribution value by averaging these output changes.

CAM-based methods [4, 14] employ the output (*i.e.* feature maps) of the model's global average pooling (GAP) layer to calculate the weighted sum and generate a class activation mapping (CAM) [25] for each target class. Grad-Cam [14] calculates a gradient of the loss function on the feature channel, which is then weighted with the target class feature map to obtain the final class activation map. Guided Grad-CAM [14] combines Grad-Cam with Guided Backpropagation to reduce redundant information and preserve details of the image. In addition to the base approaches, there are also some ensembling strategies, such as SmoothGrad (SG) [19], SmoothGrad^2 (SQ) [8] and VarGrad (Var) [1].

In this paper, we evaluate eight attribution methods, which cover most of the mainstream attribution methods. It is worth emphasizing that our evaluation system is expandable, allowing other methods to be added at any time.

B Evaluated Attribution Methods

In this section, we further elaborate on the methods evaluated in this study. Prior to that, we first formalize some fundamental concepts to facilitate subse-

quent usage. Attribution methods aim to determine the contribution scores of individual pixels by measuring the variation of the output $f(\mathbf{x})$ with respect to the input $\mathbf{x}$. The linkage between layers in a model $f(\cdot)$ is defined as:

$$\mathbf{a}^{(l+1)} = \sigma(W \cdot \mathbf{a}^{(l)} + b). \tag{S1}$$

Here, $\mathbf{a}^{(l)}$ denotes the feature in the l-th layer, where $l \in (0, 1, 2, \ldots, L-1)$ and L is the total number of layers in the network. W denotes the weight, and b denotes the bias. $\sigma(\cdot)$ serves as an activation function, such as the sigmoid function.

Saliency [17]. It is one of the most primitive attribution methods, also known as gradient attribution. It determines the contribution of each pixel in the input to the final classification result by calculating the gradient of the output with respect to the input. It can be described as follows:

$$\texttt{Attr}_i = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i}, \tag{S2}$$

where $\mathbf{x}_i$ represents the i-th feature of the input image and $\texttt{Attr}_i$ represents the contribution of the i-th feature in the input to the output.

Input⊙Gradient [16] computes the gradient of the output in relation to the input features. It then multiplies the gradient by the value of the corresponding input feature to determine the importance of each pixel's contribution. Its formula is as follows:

$$\texttt{Attr}_i = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \odot \mathbf{x}_i. \tag{S3}$$

Integrated Gradients [21]. This method employs a baseline input $\mathbf{x}'_i$ as a reference point, determines the contribution of different input features to the output result by calculating the gradient integration between the real input and the baseline, and finally generates pixel-level interpretable images. Its formula is as follows:

$$\texttt{Attr}_i = (\mathbf{x}_i - \mathbf{x}'_i) \int_{\alpha=0}^{1} \frac{\partial f(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial \mathbf{x}_i} d\alpha, \tag{S4}$$

where α is an integration variable ranging from 0 to 1, which denotes the distance between point $\mathbf{x}_i$ and the reference $\mathbf{x}'_i$.

Guided Backpropagation [20] is similar to gradient attribution, but it modifies negative gradients to accurately determine the forward and backward impact of pixels on the classification result. Specifically, it relies on backpropagation to generate the contribution. In the process of backpropagation, the negative values of the gradient are modified according to the correction function (such as the ReLU function). Its formula is as follows:

$$\texttt{Attr}_i = \mathrm{ReLU}(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i}). \tag{S5}$$

DeepLift [15] proposes using a reference input $\tilde{\mathbf{a}}$ to compute the contribution of individual input features to the output. The influence of input features on the

model output is determined by comparing the disparities between each input sample $\mathbf{a}$ and the reference sample $\tilde{\mathbf{a}}$. The formula for disparity is expressed as follows:

$$\mathtt{DIF}_{ij}^{(l)}(\mathbf{a}, \tilde{\mathbf{a}}) = W_{ij} \cdot \mathbf{a}_i^{(l)} - W_{ij} \cdot \tilde{\mathbf{a}}_i^{(l)}. \tag{S6}$$

In this equation, $\mathtt{DIF}(\mathbf{a}, \tilde{\mathbf{a}})$ is a difference function that estimates the similarity between the feature $\mathbf{a}$ and the reference point $\tilde{\mathbf{a}}$. Besides, W_{ij} denotes the weight between the i-th feature in layer l and the j-th feature in layer $l+1$. Thus, the attribution map is defined as:

$$\mathtt{Attr}_i^{(l)} = \sum_j \frac{\mathtt{DIF}_{ij}^{(l)}(\mathbf{a}, \tilde{\mathbf{a}})}{\sum_i \mathtt{DIF}_{ij}^{(l)}(\mathbf{a}, \tilde{\mathbf{a}})} \cdot \mathtt{Attr}_j^{(l+1)}, \tag{S7}$$

where j represents the j-th neuron in the layer l+1. $\mathtt{Attr}_i^{(l)}$ denotes the attibution value of the i-th pixel in the l-th layer.

Deconvolution [24]. This method first applies the forward propagation algorithm to calculate the output feature $f(\mathbf{x})$. Then, it employs deconvolution to map the output feature map back to the input feature map. Thus, we can obtain the importance score of each input feature for the output result. Explicitly, the formula for this method is as follows:

$$\mathtt{Attr}_i^{(l)} = \sum_j W_{ij} \cdot \mathtt{Attr}_j^{(l+1)} \odot \mathbf{a}_i, \tag{S8}$$

where W_{ij} represents the weight of the convolution kernel between layer l and layer $l+1$. For the last layer, the attribution value is:

$$\mathtt{Attr}_i^{(L)} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{a}_i^{(L)}}. \tag{S9}$$

LRP [2]. By analyzing the activation maps and weights of each pixel using the backpropagation mechanism, LRP computes the significance of input features. As a result, the contribution of any layer feature map with respect to the output can be derived, ultimately determining the contribution of the original input to the output. The corresponding formula is as follows:

$$\mathtt{Attr}_i^{(l)} = \sum_j \frac{W_{ij} \cdot \mathbf{a}_i}{(\sum_{i'} W_{i'j} \cdot \mathbf{a}_{i'} + b_{i'}) + \epsilon} \mathtt{Attr}_j^{(l+1)}. \tag{S10}$$

In the equation, W_{ij} represents the weight from the input feature $\mathbf{a}_i^l$ to the j-th neuron in the layer $l+1$. $\mathbf{a}_i$ represents the activation value. ϵ is a minuscule value used in numerical computations to prevent division-by-zero errors.

Guided Grad-CAM [14] combines Grad-Cam [14] and Guided Backpropagation [20]. Through the process of backpropagation of gradients, weight calculation, and result combination, the contribution score of input to output is

Table S1: Statistical information of the four datasets. For Food-101 and ImageNet-1k, the data in "Resolution" is the average value. And for NWPU-RESISC45 and Places-365, it represents the general value.

Datasets	Images	Training Set	Test Set	Resolution	Classes	Granularity
NWPU-RESISC45 [5]	31,500	25,200	6,300	256×256	45	coarse
Food-101 [3]	101,000	75,750	25,250	512×384	101	fine
ImageNet-1k [6]	1,331,167	1,281,167	50,000	224×224	1,000	coarse
Places-365 [26]	1,839,960	1,803,460	36,500	256×256	365	semi-coarse

finally obtained. Initially, we formalize the weight β_i used in Guided Grad-CAM as follows:

$$\beta_i = \frac{\partial f(\mathbf{x})}{\partial \mathbf{a}_i}, \tag{S11}$$

where β_i represents the weight of the feature map, and $\mathbf{a}_i$ represents the activation value of the feature map at the i-th position. Eventually, the attribution value is represented by:

$$\texttt{Attr}_i = \mathrm{ReLU}(\mathrm{ReLU}(\beta_i) \cdot \mathbf{a}_i), \tag{S12}$$

where $\mathrm{ReLU}(\beta \cdot \mathbf{a}_i)$ is the essence of the Grad-Cam attribution method.

C Datasets and Models

C.1 Datasets

- **NWPU-RESISC45** [5] is a medium-scale dataset for remote sensing image classification, containing 45 categories of images covering various scenes such as streets, airports, factories, forests, grasslands, and deserts. We manually divide the dataset and randomly select 80% of the images as the training set and 20% of the images as the test set based on classes.
- **Food-101** [3] is a medium-scale dataset for food image classification, encompassing a total of 101 different categories. Specifically, it includes staple foods, desserts, fruits, vegetables, seafood, and meats. The training images are intentionally left unprocessed, thus still containing a degree of noise.
- **ImageNet-1k** [6] is a large-scale dataset widely employed for image classification, comprising $1,000$ diverse categories that encompass a variety of objects, including animals, plants, vehicles, characters, natural landscapes, and indoor scenes. Currently, ImageNet-1k has been extensively used in numerous studies and serves as a benchmark for evaluating the performance of various computer vision models.
- **Places-365** [26] is a large-scale dataset designed for scene comprehension and image classification. It encompasses an extensive collection of 365 diverse scene categories, including cities, beaches, mountains, offices, indoor spaces, and many more.

Here, we have compiled detailed statistical data for the four datasets, which is recorded in Table S1.

Table S2: Accuracy results of {ResNet-18, Inception-v4, VGG-19} × {NWPU-RESISC45, Food-101, ImageNet-1k}.

Models	**NWPU-RESISC45** [5]	**Food-101** [3]	**ImageNet-1k** [6]
ResNet-18 [7]	89.17	67.48	66.48
Inception-v4 [22]	88.60	70.19	73.43
VGG-19 [18]	87.24	66.56	65.64

Table S3: Accuracy results of {ResNet-18, ResNet-50} × {Places-365}.

Models	**Places-365** [26]
ResNet-18 [7]	53.73
ResNet-50 [7]	54.84

Table S4: Accuracy results of {EfficientNet, DenseNet121} × {NWPU-RESISC45}.

Models	**NWPU-RESISC45** [5]
EfficientNet [23]	90.62
DenseNet121 [9]	91.44

C.2 Models

- **ResNet-18** [7] is a deep residual network composed of 18 convolutional layers (including BasicBlock with [2, 2, 2, 2]), one global average pooling layer, and one fully connected layer. It is the smallest model in the ResNet series and effectively solves the problems of gradient vanishing and gradient explosion in previous model training.
- **Inception-v4** [22] is the fourth version of the Inception series, which improves the structure of convolutional networks by using convolution and pooling in deep neural networks to enhance model performance. Inception-v4 can better control the depth and computing power of the model.
- **VGG-19** [18] has a deeper depth, composed of 16 convolutional layers and 3 fully connected layers. All convolutional layers use 3×3 convolutional kernels, allowing the network to learn more abstract features during the training phase and have better representative and expressive power.
- **ResNet-50** [7] is a prominent network among the ResNet series, comprising a total of 50 convolutional layers. Bottleneck units are stacked in the following configuration: [3, 4, 6, 3]. Each Bottleneck unit comprises three convolutional layers. By diminishing the number of channels, Bottleneck units assist in reducing the number of parameters and computations in the network. This leads to enhanced model speed while improving overall performance.
- **EfficientNet** [23] introduces a compound scaling method to achieve balance by continuously increasing the scaling coefficients of depth, width, and resolution. The basic module of EfficientNet is MBConv, which is repeatedly stacked together to form the main part of the network. Specificially, EfficientNet_b2 has a width scaling coefficient of 1.1 and a depth scaling coefficient of 1.2.
- **DenseNet** [9] introduces dense connections to improve information transmission in the network. Each layer is called a Dense Block, and each Dense Block consists of multiple convolutional layers, where the input of each convolutional layer is the concatenation of the output of all previous layers within the Dense Block. In order to facilitate information flow between different layers, the output of each Dense Block is further added with the outputs of all

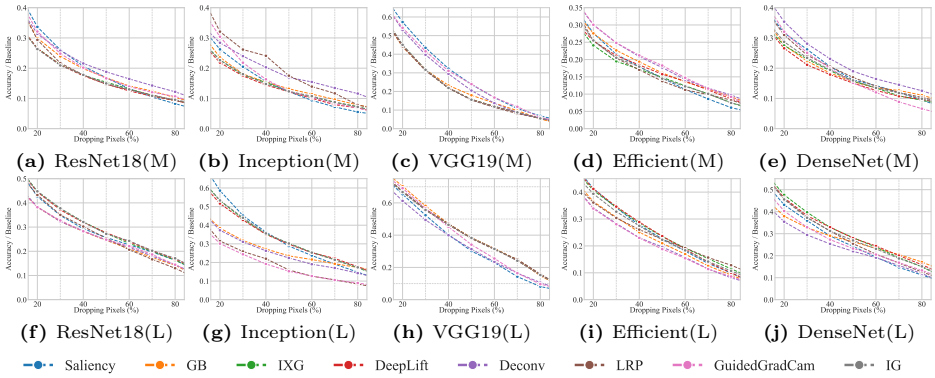


Fig. S1: Evaluation results on the NWPU-RESISC45 dataset. The settings in this figure are the same as those in the main text.

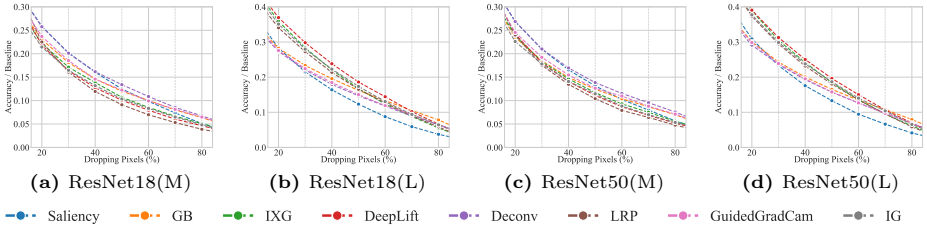


Fig. S2: Evaluation results on the Places-365 dataset. The settings in this figure are the same as those in the main text.

preceding layers, resulting in the final output of the Dense Block. DenseNet also employs transition layers to further reduce the number of parameters and computations in the network. A transition layer typically consists of a convolutional layer followed by a pooling layer, used to decrease the size of the feature map and number of channels, thus facilitating subsequent extraction and fusion by Dense Blocks. DenseNet enables all layers to directly access the feature information of previous layers, effectively improving feature reuse and information transmission efficiency. DenseNet-121 includes Dense Blocks with [6, 12, 24, 16].

Note: The accuracy results of all models are described in Table S2, Table S3, and Table S4.

D Additional Experimental Results

In this section, we present additional experimental results for Meta-Rank. These results comprise the following aspects: (1) evaluation results obtained from the NWPU-RESISC45 dataset and Places-365 dataset (in Figure S1 and Figure S2), which correspond to Figure 2 of RQ1 in the main text; (2) AUC results of the eight attribution methods among all cases (in Figure S3, Figure S4, Figure S5,

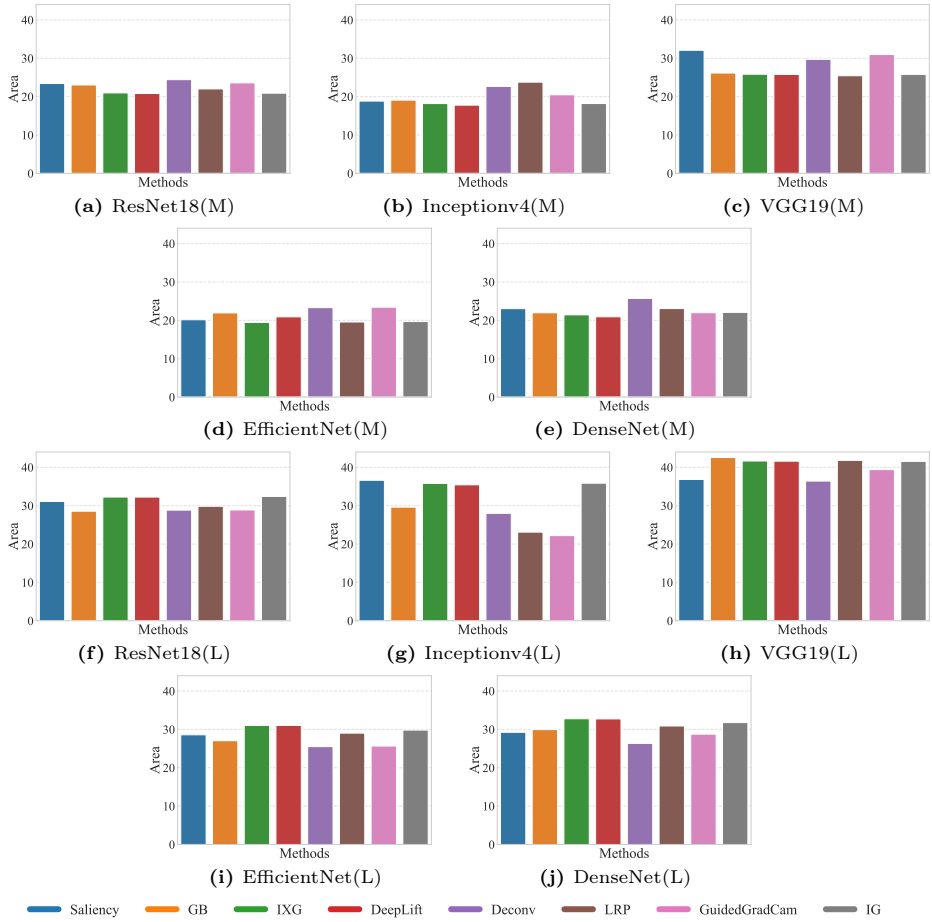


Fig. S3: AUC results on the NWPU-RESISC45 dataset. The method's performance exhibits quantifiable and substantial variations across different settings. "M" denotes the MoRF protocol and "L" denotes the LeRF protocol.

and Figure S6); (3) Epoch-consistency results on other settings (in Table Tab. S5 and Table Tab. S6). All of the aforementioned experimental results strongly corroborate the conclusions stated in the main body of the text.

E Detailed Limitations of ROAD

Corresponding to RQ2, we also evaluate the attributions obtained by "NWPU-ResNet18" case and "Food-Inceptionv4" case using the ROAD method [13]. The curves of accuracy decay and time cost are recorded in Figure S7. We have identified the following observations:

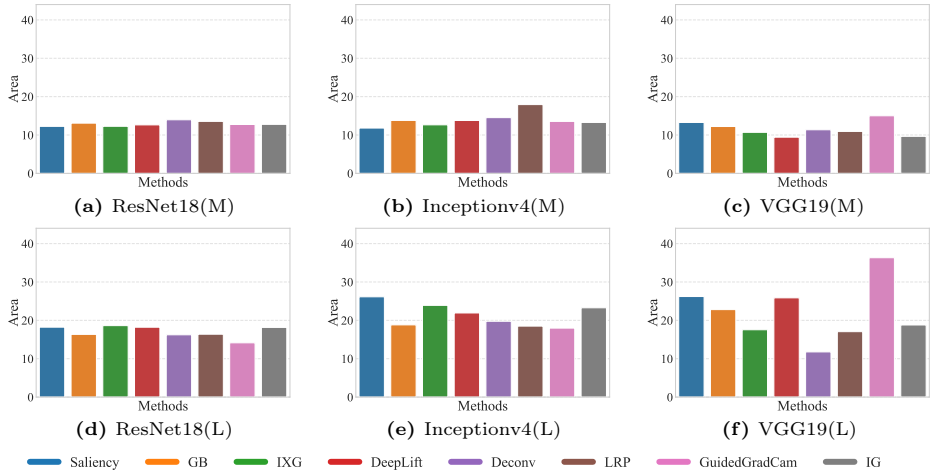


Fig. S4: AUC results on the Food-101 dataset. The settings applied in this figure are concordant with those described in Fig. S3.

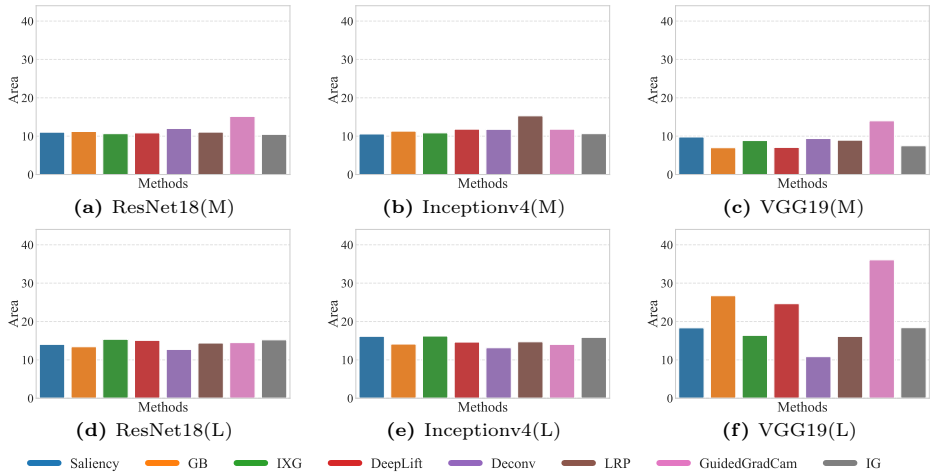


Fig. S5: AUC results on the ImageNet-1k dataset. The settings adopted in this figure are consistent with those detailed in Fig. S3.

- **ROAD cannot produce completely consistent results.** By comparing the experimental results of "Accuracy" between (a) and (b), as well as (c) and (d) in Figure S7, the evaluation results generated by ROAD exhibit a general consistency. However, they are not entirely identical in detail. For instance, LRP achieves the highest performance in "Food-Inceptionv4(M)" case (c), but exhibits the lowest performance in "Food-Inceptionv4(L)" case (d). Thus,

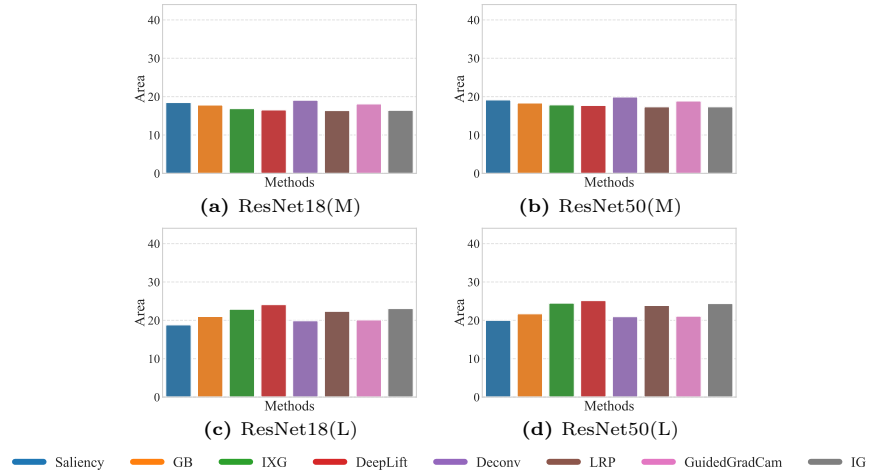


Fig. S6: AUC results on the Places-365 dataset. The settings in this figure also align with the descriptions provided in Fig. S3.

Table S5: Checkpoint-level consistency on NWPU-RESISC45 × Inception-v4.

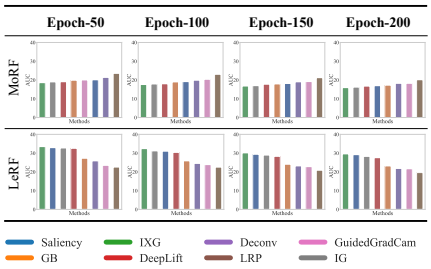
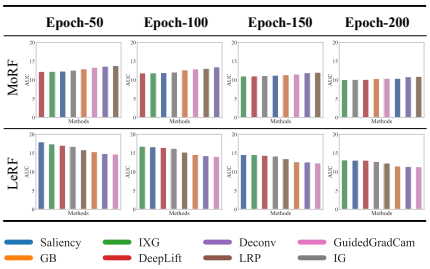


Table S6: Checkpoint-level consistency on Food-101 × ResNet-18.



we still cannot derive a unified ranking result for attribution methods using this approach.

- **ROAD generates an evaluation result that contradicts the facts.** According to the concepts of MoRF and LeRF, the decrease in accuracy of an attribution method should be faster in MoRF than in LeRF. Contrary to expectations, the obtained results demonstrate that numerous methods, such as IG, Input⊙Gradient, and DeepLift, display a more rapid decrease in LeRF compared to MoRF.
- **ROAD comes with sizable time expenses during execution.** The "Time" row in Figure S7 clearly demonstrates that as the number of masked pixels increases, the time cost of ROAD increases exponentially. Especially for large-scale datasets, such a time cost is even more expensive. This find-

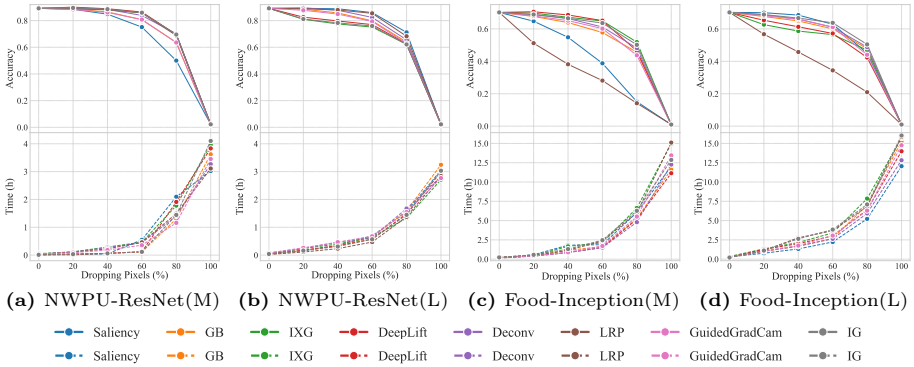


Fig. S7: The evaluation results of ROAD on NWPU-RESISC45 (a)(b) and Food-101 (c)(d). The eight methods are evaluated for accuracy variation (the solid line) and time cost (the dashed line) under two masking modes, MoRF (a)(c) and LeRF (b)(d). Observation points range from 0% to 100% with intervals of 20%.

ing contradicts our expectations and renders ROAD unsuitable for practical applications.

Considering the above shortcomings, ROAD is not utilized as a feature ablation method for our attribution evaluation on the test cases. In the future, we aim to investigate solutions that substantially mitigate the missingness bias in feature ablation.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. Advances in neural information processing systems **31** (2018)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
3. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13. pp. 446–461. Springer (2014)
4. 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)
5. Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: Benchmark and state of the art. Proceedings of the IEEE **105**(10), 1865–1883 (2017)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

7. 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)
8. Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: A benchmark for interpretability methods in deep neural networks. Advances in neural information processing systems **32** (2019)
9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
10. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Advances in neural information processing systems **30** (2017)
11. Petsiuk, V., Das, A., Saenko, K.: Rise: Randomized input sampling for explanation of black-box models. arXiv preprint arXiv:1806.07421 (2018)
12. Ribeiro, M.T., Singh, S., Guestrin, C.: " 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. pp. 1135–1144 (2016)
13. Rong, Y., Leemann, T., Borisov, V., Kasneci, G., Kasneci, E.: A consistent and efficient evaluation strategy for attribution methods. In: International Conference on Machine Learning. pp. 18770–18795. PMLR (2022)
14. 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)
15. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: International conference on machine learning. pp. 3145–3153. PMLR (2017)
16. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: Learning important features through propagating activation differences. arXiv preprint arXiv:1605.01713 (2016)
17. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. In: Proceedings of the International Conference on Learning Representations (ICLR). ICLR (2014)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
19. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
20. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)
21. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International conference on machine learning. pp. 3319–3328. PMLR (2017)
22. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)
23. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)
24. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13. pp. 818–833. Springer (2014)

25. 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)
26. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1452–1464 (2017)