ATTRIBUTION SCORES ARE REDUNDANT: EXPLAINING FEATURE CONTRIBUTION BY TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Opening black boxes and revealing the inner mechanism of deep models is vital in applying them to real-world tasks. As one of the most intuitive and straightforward explanations for deep models, attributive explanation methods have been extensively studied. Existing attribution methods typically assign attribution scores to each individual feature as an explanation. However, when we use or evaluate the explanations in practice, what really matters is not the attribution scores, but the rank order of features (e.g., identifying the top-contributing features, or checking for changes in the model output by masking features in order). In other words, achieving attribution scores is a redundant step in achieving explanations. To address this, we propose a novel framework TRAjectory importanCE (TRACE) which directly provides feature ranking explanations. Our method introduces several improvements. First, TRACE greatly reduces the set of feasible explanations, allowing us to actually solve for the *best* explanation. Second, TRACE is able to achieve the theoretically-grounded *best possible* explanation in commonly used deletion evaluations. Third, we provide extensive experiments to validate that TRACE outperforms attribution methods with a significant margin.

1 INTRODUCTION

With the rapid increase in computational power, deep neural networks (DNNs) have achieved great performance in various tasks, especially in those with high complexity, such as computer vision (CV), natural language processing (NLP), etc. However, DNNs are also notorious for their black-box essence, and DNNs' capability is achieved at the expense of algorithmic transparency. Language models have reached trillions of parameters (BAAI, 2020). Commonly used CNN models, although much smaller, also have hundreds of millions of parameters (Simonyan & Zisserman, 2014). This makes it impossible to track the inner mechanism of the models. However, without adequate explanations, such capable models are hindered to be deployed in reality, especially in high-stake areas.

To reveal the inner mechanism of DNNs, various forms of explanation methods have been proposed (Arrieta et al., 2020). Among these explanation forms, attribution methods are the most extensively studied form since they are very straightforward. Given an input with d features (pixels, tokens, pixel patches, etc.) and the output, attribution methods assign an attribution score for each input feature, representing the contribution (or sensitivity, etc.) of the feature w.r.t. the output. This form also enables appealing visualization since they can be visualized as heatmaps, which is a preferable form of presentation to humans (Leavitt & Morcos, 2020). Because different forms of explanations differ too much to be universally studied together, in this paper, we mainly focus on attribution methods.

Attribution methods provide explanations by assigning attribution scores for features. However, we argue that this is redundant in explanations. First, in real applications, attribution explanations are mostly used to inspect if the *most important features* are properly highlighted. Second, in evaluation metrics of explanations, the focus is not on the attribution scores, but on feature *rankings*. Alignment metrics like pointing game (Zhang et al., 2018) only check if the top *ranked* features correspond with prior knowledge (segmentations, bounding boxes, etc.). And performance metrics like deletion/insertion metrics [1] perturb (deletes, inserts, etc.) features w.r.t. the rankings of attribution scores. Third, recent human-involved experiments also give the same results. (Kaur et al., 2020)

[1] The deletion metric is especially extensively used in evaluating explanations and has many variants, including feature ablation/occlusion, top-k masking, ROAR, image degradation, etc. Here for the sake of consistency we use the term "deletion".

show that even for data scientists, the compatibility between the feature rankings and the intuition contributes greatly to the trust towards black-box models. (Krishna et al., 2022) further demonstrate that when it comes to the "disagreement" of attribution methods, all participants (data scientists) can only think of problems related to the ranking of features.

According to the above observation in attribution explanations, we can find that what is actually used/evaluated is ranking all the features, where assigning the concrete attribution scores is barely a means to fulfill this goal. And this means introducing too much redundant information that entangles the analysis of explanations. As a result, according to Occam's razor, we propose to simplify attributions into *trajectories* of features, where the features are traversed based on the rankings. There are many advantages of using feature trajectories instead of attributions as the explanations. The most significant one is that by aggregating "equivalent" attributions (where attribution scores are highly distinct in values, but keep the same feature ordering), the set of feasible explanations for an input degenerates significantly from the *infinite* Euclidean space $\mathbb{R}^d$ to the symmetric group S_d, which is *finite*. This improvement enables us to actually solve for the *best* explanation. Based on this, we introduce TRAjectory importanCE (`TRACE`), a novel framework to generate high-quality trajectory (ranking) explanations that outperform existing attribution methods in terms of ablation tests by a large margin. The contribution of this paper can be briefly summarized as follows.

- We extract the essence of current usage/evaluation of attribution methods and introduce a novel simplified form of explanation.
- We propose a new framework `TRACE` targeting at the simplified explanation form. `TRACE` is able to achieve the *best possible* explanation in commonly used deletion evaluation for explanation methods.
- To the best of our knowledge, we are the first to formulate perturbation-based explanations as combinatorial optimization problems, where a rich family of tools are available.
- We provide extensive experimental results to validate that `TRACE` outperform attribution explanation methods with significant margin.

2 Related Work

In order to explain DNNs, numerous attribution methods have been developed. Based on the ways explanations are generated, they can be roughly separated into propagation methods and perturbation methods. Propagation methods back-propagate gradients or modified/pseudo gradients in the top-down fashion, while perturbation methods usually generate explanations by modifying the input data and observe the change in the output. Generally, perturbation methods are model-agnostic, meaning that they do not require any information of the explained model. On the contrary, propagation methods need access to the models (layers, parameters, etc.) to perform the propagation. There are also self-interpretable models with attribution scores (Chen et al., 2019; Agarwal et al., 2021a; Wang & Wang, 2021; Li et al., 2021a), where instead of explaining an existing black-box model, they propose entire new models that generate explanations and predictions at the same time. In addition, feature ranking explanations should be distinguished from local feature selection, which aims at selecting a portion of features of given a specific sample. The goal of feature selection is to reduce the dimensionality of the data by selecting the most representative features, while `TRACE` is an explanation method that aims at explaining the feature importance to a specific black-box model instead of the data itself (Chen et al., 2020).

Propagation Methods. Saliency (Simonyan et al., 2013) makes use of the gradient of input as the attribution scores. Guided back-propagation (Springenberg et al., 2014) modifies the behaviour of ReLU layers in backpropagations. LRP (Bach et al., 2015) and DeepLift Shrikumar et al. (2017) change the back-propagation rule to propagate attribution scores layer-wise in the top-down fashion. Input $\times$ Gradient (Shrikumar et al., 2016) uses the Hadamard product between input and its gradient as attributions. Sundararajan et al. (2017) propose axioms for attribution methods and introduce Integrated Gradient, which is the line integral of the input gradient. Grad-CAM (Selvaraju et al., 2017) generalizes the class activation mapping to all CNNs through the gradient of the CNN activations.

Perturbation Methods. LIME (Ribeiro et al., 2016) locally approximates the prediction with a simple surrogate model. Occlusion (Zeiler & Fergus, 2014) identifies the object locations by replacing different portions of image with gray squares. SHAP (Lundberg & Lee, 2017) utilizes the approximated Shapley values (Shapley, 1953) as attribution scores. RISE (Petsiuk et al., 2018) defines attribution scores based on many randomly sampled masks. IBA (Schulz et al., 2020) generates

Table 1: The deletion/insertion tests with MoRF and LeRF criteria. The first row is the k-th point values of the corresponding curve showing the change in model output when k features are perturbed ($0 \leq k \leq d$). The second row is the corresponding AUCs estimated by Riemann sum (i.e. the summation of all point values). The third row is the desiderata for the corresponding AUCs.

Criteria	Del-Mo	Del-Le	Ins-Mo	Ins-Le
k-th Point	$f(\boldsymbol{x}_{\setminus\sigma'[:k]})$	$f(\boldsymbol{x}_{\setminus\sigma[:k]})$	$f(\boldsymbol{x}_{\sigma'[:k]})$	$f(\boldsymbol{x}_{\sigma[:k]})$
AUC	$\sum_{k=0}^{d} f(\boldsymbol{x}_{\setminus\sigma'[:k]})$	$\sum_{k=0}^{d} f(\boldsymbol{x}_{\setminus\sigma[:k]})$	$\sum_{k=0}^{d} f(\boldsymbol{x}_{\sigma'[:k]})$	$\sum_{k=0}^{d} f(\boldsymbol{x}_{\sigma[:k]})$
Desiderata	min	max	max	min

explanations via per-sample information bottleneck. I-GOS (Qi et al., 2019; Khorram et al., 2021) optimizes small masks to maximally decrease prediction scores. (Agarwal et al., 2021b) formulate attribution generation as a Markov Decision Process and use reinforcement learning to solve it.

3 METHODOLOGY

Notations. Let $f : \mathbb{R}^d \to \mathbb{R}$ be the black-box model to be explained. Here the output is constrained in scalars. This is valid since even for multi-class tasks, explanations are generated w.r.t. one class of interest. Given such a model f and an input $\boldsymbol{x} \in \mathbb{R}^d$, an attribution method is defined as a mapping $\phi_f : \mathbb{R}^d \to \mathbb{R}^d$. Here the RGB channels are omitted and only the spatial dimension is considered. $[d] = \{1, \cdots, d\}$ denotes the index set of features. Given f, ϕ, a unique trajectory that traverses from the least important feature (w.r.t. f) to the most important feature is defined for input $\boldsymbol{x}$ based on the attribution $\phi_f(\boldsymbol{x})$. We denote it by $\sigma(f, \phi; \boldsymbol{x})$ and simplify it as $\sigma(\boldsymbol{x})$ or σ for brevity since f, ϕ and $\boldsymbol{x}$ are general. Here $\sigma(\boldsymbol{x})[i] \in [d]$ is the index of the i-th non-important feature, and $\forall i \in [d-1]$, $x_{\sigma(\boldsymbol{x})[i]}$ is seen as less important than $x_{\sigma(\boldsymbol{x})[i+1]}$ to model f. And σ is thereby a permutation of $[d]$.

For any index subset $\tau \subset [d]$, let $\boldsymbol{x}_\tau, \boldsymbol{x}_{\setminus\tau}$ denote when only the features indexed by τ are preserved or deleted, respectively. And for an ordered index set such as σ, we denote by $\sigma[:i], \sigma[i:j], \sigma[j:]$ the subset of σ consisting of the first i, the i-th to the j-th, and the j-th to the last elements of σ, respectively. All endpoints are included. In addition, σ' represents the reverse of the trajectory σ.

Insertion/Deletion Measures v.s. Perturbation Methods. As the most popularly and widely used evaluation metrics for attribution explanations, insertion/deletion measure the output change w.r.t. the input perturbations. The insertion metric gradually inserts features to a `null` input (e.g. zeros, means, etc.) while the deletion metric gradually deletes features from the original input. These processes actually correspond to the essence of perturbation methods. Existing perturbation methods tend to optimize masks according to $f(\boldsymbol{x} \odot \boldsymbol{m} + \boldsymbol{r} \odot (1 - \boldsymbol{m}))$, where $\odot$ stands for the Hadamard product, $\boldsymbol{r}$ is some reference values (zeros, means, etc.), and $\boldsymbol{m} \in \{0, 1\}^d$ represents a binary mask. Sometimes $\boldsymbol{m}$ is relaxed in $[0, 1]^d$ for continuity and smoothness. The objective function of perturbation methods can be formulated as[2]:

$$\boldsymbol{m}^* = \arg \min_{\boldsymbol{m} \in \{0,1\}^d} f(\boldsymbol{x} \odot \boldsymbol{m} + \boldsymbol{r} \odot (1 - \boldsymbol{m})), \text{ s.t. } \Omega(\boldsymbol{m}), \tag{1}$$

where Ω is some imposed constraints. we denote by $\boldsymbol{m}_k^*$ if $\|\boldsymbol{m}^*\|_1 = k$. Given $k \in [d]$, $\boldsymbol{m}_k^*$ represents the group of informative features when exactly k features are kept. This choice can be distinct for $\boldsymbol{m}_j^*, j \neq k$. This is where perturbation methods do **not** align with deletion test, which requires features to be deleted in a trajectory. We deduce that this might be the reason why even perturbation methods are more closely related to the deletion test than back-propagation methods, they still show no significant advantages under such metric (Li et al., 2021b).

Our Explanation Method: Trajectory Importance (`TRACE`). In order to dominate the deletion test, the perturbation should be based trajectories, which means $\boldsymbol{m}_k^* \leq \boldsymbol{m}_{k+1}^*$. This is exactly compatible with the observation that only trajectories (rankings) of attribution methods are made use of. In fact, when the explanations degenerate to trajectories $\sigma(\boldsymbol{x})$, we can optimize for an explanation that aces the insertion/deletion test. Note that by differentiating between acting on the most or the least relevant feature first (MoRF/LeRF), there are totally four different ways in evaluating explanations by combining insertion/deletion and most/least together. We denote them by Ins/Del-Mo/Le[3]. In other words, the MoRF criterion includes Ins/Del-Mo and the LeRF criterion

[2]This simple form is of course not the exact form of perturbation methods. But it is indeed the basic factor of most perturbation methods.

[3]For example, in Del-Mo, important features are deleted first and the desired curve should drop fast.

includes Ins/Del-Le. Formally, when the top k features are deleted, the input becomes $\boldsymbol{x}_{\setminus\sigma'(\boldsymbol{x})[:k]}$. Then by deleting/inserting d features, four curves regarding the change in model output can be drawn respectively. Details of the four curves are shown in table 1. According to the desiderata, when Del-Mo is used to evaluate explanation methods ϕ_f, it is desired that the curve to drop fast (min AUC). Therefore, when we have access to the trajectory σ directly, it it natural to optimize σ directly:

$$\texttt{TRACE}\text{-Del-Mo} : \min_{\sigma} \sum_{k=0}^{d} f(\boldsymbol{x}_{\setminus\sigma'[:k]}), \tag{2}$$

and the three other cases can be optimized similarly. In the remaining context, notations like Del-Mo or MoRF refer to the metrics. And we add the prefix `TRACE` ahead (e.g. `TRACE`-Del-Mo, `TRACE`-Mo, `TRACE`-p, etc.) to represent the proposed method `TRACE` and variants.

4 Remarks on Trajectory Explanations and TRACE

Deletion v.s. Insertion. It is worth noticing that deletion and insertion tests works from different directions of the trajectory. However, when summarizing all point values together to calculate the Riemann sum, the influence of such directional difference will be omitted. In fact, we have:

Theorem 1 *Ins-Le and Ins-Mo are equivalent to Del-Mo and Del-Le up to AUCs, respectively.*

Please refer to appendix A for the proof. As a result, we only consider the deletion test, and denote the two criteria only by Mo and Le for brevity. And eq. (2) is thereby denoted as `TRACE`-Mo.

Mo v.s. Le. As two criteria of the deletion test, Mo and Le have very distinct interpretations. Del-Mo defines important features as *those affecting the performance the most when we delete them*, while Del-Le defines important features as *those maintaining the performance the most when we only keep them*. A desired trajectory σ should correspond to both directions, i.e., achieving Mo and Le on the same feature trajectory. Hence a better metric that evaluate the two desiderata at the same time is $\sum_{k=1}^{d} \left(f(\boldsymbol{x}_{\setminus\sigma[:k]}) - f(\boldsymbol{x}_{\setminus\sigma'[:k]})\right)$, which is also known as the normalized AUC (Schulz et al., 2020). Here we denote the metric as (Le$-$Mo) and the method as `TRACE`-(Le$-$Mo). **In the following context, if not specified, `TRACE` refers to `TRACE`-(Le$-$Mo).**

The Out-of-Distribution Problem. The feature deletion test is also affected by the out-of-distribution issue (Hooker et al., 2019; Wang & Wang, 2022a). It is suspected that finer deletions break the input distribution, which contributes more to the performance decay. Besides, pixel-wise attributions tend to lose semantic information present by the image (Rieger et al., 2020). Grouping pixels and dealing with patches have also been demonstrated to achieve great success (Dosovitskiy et al., 2020; Tolstikhin et al., 2021; Yu et al., 2022). As a result, instead of performing deletion pixel-wise over $\boldsymbol{x} \in \mathbb{R}^d$, we operate on t superpixel square patches. By comparing different t values, we observe that the out-of-distribution issue is alleviated by decreasing the resolution t of the deletion process. Smaller t results in less noisy trajectories, but coarser explanations, while larger t leads to finer but more noisy explanations. Also, a large t brings more difficulties to the optimization as the feasible set is of size $t!$. In order to balance these issues above, we set $t = 7^2 = 49$ in our main experiments. And we will abuse the notations a little to denote by $\boldsymbol{x}_{\setminus\sigma'[:k]}$ the input image with the top k patches (instead of pixels) deleted. We put more results with other t values in appendix B

Logit v.s. Probability. The output of the model, which is represented by $f(\boldsymbol{x})$, can have different meanings. As a classifier, the standard output of the model is the predicted logit of the last linear layer, while the predicted probability is the softmax activation. Perturbations w.r.t. the logits and probabilities have different behaviours since a perturbation increases the logit of class i may also increase that of class $j \neq i$, resulting drop of the probability of class i (Wang & Wang, 2022b). We distinguish the two variants of our `TRACE` method by suffix -`y` (explanation w.r.t. logits) and -`p` (explanation w.r.t. probabilities), respectively.

Trajectories to Attributions. The mapping from the attributions $\phi_f(\boldsymbol{x})$ to the corresponding trajectory $\sigma(\boldsymbol{x})$ is a surjective, but $\sigma(\boldsymbol{x})$ can still be mapped back to attributions as $\phi(\boldsymbol{x}) = \texttt{bilinear}((\sigma^{-1}(\boldsymbol{x})/p)^{\alpha}) \in [0,1]^d$, where $\sigma^{-1} = \texttt{argsort}(\sigma)$ is the ranking of features in the trajectory σ. And $\sigma^{-1}[i]$ is the rank of the feature x_i in the less-important-first manner. The power $\alpha > 0$ is a smoothing factor to constrain the portion of the highlighted area, and the denominator t is to normalize the rankings to $[0, 1]$. This is only to visualize trajectories as heatmaps so that we can have conventional qualitative comparisons with attribution methods. The converted visualizations are shown in fig. 1(a), where $\alpha \in \{1, 2, 5\}$. Here we use the power function family for simplicity. Any monotonic continuous mapping $[d] \to [0, 1]$ can achieve this task. This further

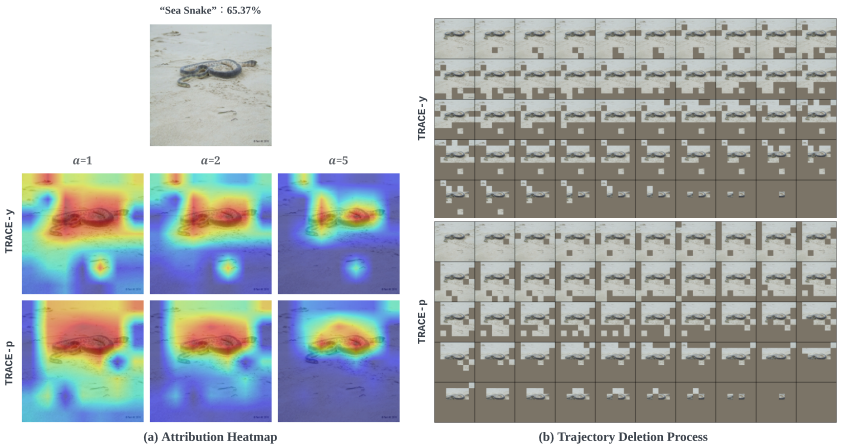


Figure 1: Explanations from our method TRACE-y and TRACE-p on the "sea snake" image of ILSVRC2012 validation set. (a) The converted heatmap explanations with different smooth factor α. (b) The deletion process based on the trajectory σ from our method.

shows the redundancy of attribution scores in attribution methods. Also, compared with deletion trajectories (b), which lucidly shows how these patches interact, the converted attribution heatmaps shown in (a) can not properly reflect all the details contained in the trajectory σ, even visually.

5 Algorithms for TRACE

Despite the promising properties of TRACE, the optimization of trajectory in TRACE is not trivial (and is NP-hard). Here we propose the algorithm based on combinatorial optimization.

The Relation to TSP. The most well-known problem related to TRACE optimization is classic traveling salesman problem (TSP), where a salesman is supposed to traverse all t cities, and the minimal cost is sought. It is defined by a cost matrix $\Delta = [\delta_{ij}]_{p\times p}$ where δ_{ij} is the cost going from city i to j. Given a trajectory σ, the cost function is defined as $f_{tsp}(\sigma) = \sum_{i=1}^{t} \delta_{\sigma[i]\sigma[i+1]}$, where we extend $\sigma[p+1] := \sigma[1]$. Based on this, we have the following theorem (proved in appendix C):

Theorem 2 *The optimization problem TRACE-Mo ($\{\min_\sigma \sum_{k=0}^{d} f(\boldsymbol{x}_{\setminus\sigma'[:k]})\}$) is NP-hard.*

Simulated Annealing. Optimization over all permutations is a typical combinatorial optimization problem. For TRACE, since the objective is dependent on the DNN model f, whose analytical formulae is not available, meta-heuristic algorithms are the judicious choice. They have been demonstrated effective over combinatorial optimization problems (Baghel et al., 2012). Among them, simulated annealing (SA) (Kirkpatrick et al., 1983) has been successfully applied to problems such as TSP to to generate sufficiently good sub-optimal results (Geng et al., 2011). Therefore, we too employ SA as the tool. The pseudo-code can be found in appendix D.

It should be noticed that developing better tools for combinatorial optimization is beyond the scope of this paper. TRACE is introduced as a pipeline to formulate perturbation explanations as a combinatorial optimization problem so that they can be solved directly with a rich family of tools. We employ SA because it is efficient, effective and theoretically sound. Provided with better tools, TRACE can generate explanations of higher quality. We also test other algorithms in appendix E.

Neighbor Sets. As a searching algorithm, one of the most important factors of SA is the choice of neighbors, which is not trivial on a discrete feasible set, given that the objective function is a black box. Meanwhile, closely related to TSP as it is, TRACE is essentially a harder problem. In TSP, the directly connected cities can determine the total cost, while in TRACE, not only the consecutively deleted patches, but also the overall ordering of deleting patches matter. For example, if a block in the trajectory is reversed, then in (symmetric) TSP, the only values that change are the connections to the two endpoints of the block. However, in TRACE, all the values after the first point of the reverse block will change. Therefore, common neighbors for TSP such as the vertex insertion, block insertion, and block reverse (Geng et al., 2011) do not apply to TRACE trivially. We thus explore valid and efficient neighbors for TRACE.

Note that σ can be any permutation of length t, which corresponds to S_t, the symmetric group of order t. Specifically, since $i = \sigma[\sigma^{-1}[i]] = \sigma^{-1}[\sigma[i]]$, we have $\forall\sigma$, $\exists \boldsymbol{s} \in S_t$ s.t.

$$\boldsymbol{s} = \begin{pmatrix} 1 & 2 & \cdots & d \\ \sigma^{-1}[1] & \sigma^{-1}[2] & \cdots & \sigma^{-1}[d] \end{pmatrix} = \boldsymbol{s}(\sigma), \tag{3}$$

which is a bijective. Since the feasible set S_t is a discrete space, SA is modeled as a search method over a graph, where the vertices are feasible states, and the edges are possible movements between corresponding states, i.e. neighboring relations. Besides, it is also desired that each state has exactly the same number of neighbors. For the symmetric group S_t, such graph is perfectly modeled by Cayley's graph (Cayley, 1878). Given a generating set $S \subset S_t$, the Cayley graph is defined as a directed graph $\mathrm{Cay}(S_t, S) = G(V, E)$ where the set of vertices V are the same as S_t, and the arcs are defined by $E = \{[s_1, s_2] | \exists g \in S,\ gs_1 = s_2\}$, which results in an $|S|$-regular graph. Therefore, from any state $\forall \boldsymbol{s} \in S_t$, we can move to $|S|$ other states. And there are also $|S|$ states that can move directly to s. For neighbors, we expect: 1) sufficiently small change between neighbored states and 2) the neighboring should be symmetric (i.e. $[\boldsymbol{s}_1, \boldsymbol{s}_2] \in E \Leftrightarrow [\boldsymbol{s}_2, \boldsymbol{s}_1] \in E$). Hence we only include transpositions (permutations that only exchange two elements) in S (known as transposition set). For a transposition set S, we have $\forall \boldsymbol{s} \in S$, $\boldsymbol{s} = \boldsymbol{s}^{-1}$, which means that $\mathrm{Cay}(S_t, S)$ is a symmetric directed graph and hence can be seen as undirected. In this case $G([t], S)$ is known as the transposition graph, where the vertices are $[t]$, and the edges are the transpositions in S. Then

Proposition 1. (Hahn & Sabidussi, 2013) S generates S_t if and only if $G(S)$ is connected.

This indicates that $t - 1 \leq |S| \leq \frac{t(t-1)}{2}$, where the two equalities hold at spanning trees of the complete graph and the complete graph, respectively. (Lakshmivarahan et al., 1993) propose several well-structured transposition generating set for S_t:

- Complete Transpositions: $S_{complete} = \{(i\ j) | 1 \leq i < j \leq d\}$
- Bubble-Sort Transpositions: $S_{bubble} = \{(i\ i+1) | 1 \leq i < t\}$
- Star Transpositions: $S_{star,i} = \{(i\ j) | 1 \leq j \leq d, j \neq i\}, 1 \leq i \leq t$

When applying SA over S_t, the number of states $t!$ is easy to explode compared with the neighbor size. This requires: 1) sufficiently many movements from each state; 2) sufficiently few steps between any two states. In fact, let $\mathrm{diam}(G)$ denote the diameter of the graph G, then we have

Theorem 3 $\mathrm{diam}(\mathrm{Cay}(S_t, S_{complete})) \leq t - 1$.

Plese refer to appendix F for the proof. On the other hand, for the bubble-sort transposition and star transposition, the diameters are (Akers & Krishnamurthy, 1989):

Proposition 2. $\mathrm{diam}(\mathrm{Cay}(S_t, S_{bubble})) = \frac{t(t-1)}{2}$; $\mathrm{diam}(\mathrm{Cay}(S_t, S_{star})) = (\lfloor 3(t-1) \rfloor)/2$

As a result, even there are $t! = 49! \approx 6.08 \times 10^{62}$, the distance between the any pair of vertices is only $t - 1 = 48$ in the complete graph. And this is the smallest value among all transposition sets. Because $S_{complete} = \cup_{S \text{ is a transposition set}} S$.

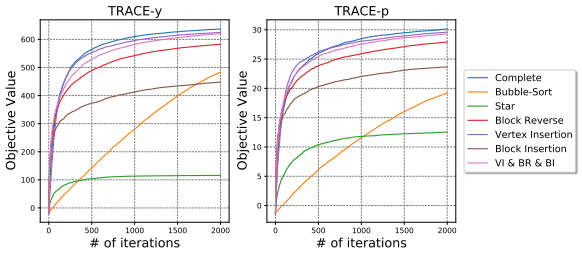


Figure 2: The comparison of different neighbor sets.

We present empirical results of different neighbor settings, including complete graph, bubble-sort graph, star-graph, vertex insertion (VI), block reverse (BR), block insertion (BI), and mix (89%BR + 10%VI + 1%BI) (Geng et al., 2011). The SA optimization process for the first 100 images of the validation set of ILSVRC2012 on pretrained ResNet-18 provided by `torchvision` are plotted. The results are shown in fig. 2. It can be found that the complete graph outperform other neighbor sets.

6 Experiments

In this section, we carry out multiple experiments to demonstrate the advantages of using `TRACE` as the explanations to DNNs. Without specifically clarified, we use a ResNet-18 model as the black-box f to be explained. Experiments are carried out on the ImageNet-1k (ILSVRC2012) dataset (Deng et al., 2009). The input image are resized to 224×224. The explanations trajectory are

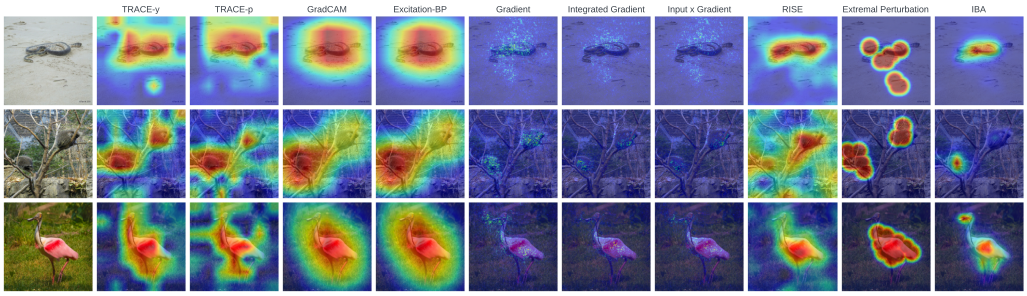


Figure 3: Visualizations of TRACE and popular attribution methods on images from ILSVRC2012.

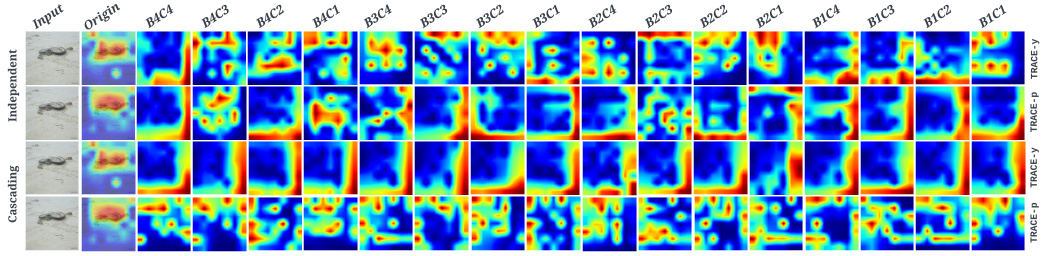


Figure 4: Sanity check using cascading randomization for TRACE. Convolutional layers of pre-trained ResNet-18 are randomized in the independent (upper) and cascading (lower) manners. In the independent randomization, other layers are kept at the pre-trained values. And in the cascading randomization, layers are progressively randomized from left to right (top-down). Here "BaCb" means the b-th convolutional layer in the a-th block.

generated using SA algorithm 1 to solve TRACE. As for parameters for SA, we use the complete graph as the neighbor sets. The max iteration is $K = 10000$. The initial temperature is $T_0 = 2$ for -y and $T_0 = f(\boldsymbol{x})/10$ for -p. And the cooling rate is $\eta = 0.999$. All experiments are carried out over Intel(R) Core(TM) i9-9960X CPU @ 3.10GHz with Quadro RTX 6000 GPUs.

Visualizations. As a convention, we first present the heatmaps of TRACE and popular attribution methods, including GradCAM, Excitation Back-Propagation, Gradient, Integrated Gradient, Input $\times$ Gradient, RISE, Extremal Perturbation, and IBA. The results are shown in fig. 3. Please refer to appendix G for the deletion trajectories. Although heatmaps are very preferred in the interpretable machine learning community due to the intuitive and straightforward forms, it should be emphasized that it can be dangerous to attach undue importance to them (Leavitt & Morcos, 2020). The model can make correct prediction using biased/wrong features that we humans cannot understand. And those explanations who do not highlight the objects in the image are not necessarily bad, and vice versa. Thus heatmaps should be used only as a supplement for the evaluation of explanations.

Sanity Check. In order to make better use of visualizations given this situation, Adebayo et al. (2018) propose the sanity check for visualizations, where a "ground truth" is created artificially as "When the layers of the black-box model are randomized, the heatmaps should **not** stay invariant." We follow the criteria and present the sanity check results of TRACE with both the cascading and the independent randomization. As shown in fig. 4, both TRACE-y and TRACE-p change immediately once any layer is randomized, which indicates that TRACE successfully passes the sanity check. For completeness, we present the sanity check results of other comparing methods in appendix H.

Deletion Test. Here we demonstrate that TRACE outperforms attribution methods in the deletion test, the most commonly used quantitative experiment. The input image is first split into $t = 49$ square patches. Then given an attribution map, patches are deleted following Most relevant Remove First (MoRF) or Least relevant Remove First (LeRF). Under MoRF, the prediction is expected to drop fast, and vice versa. From the plots in fig. 5 and AUCs in table 2, we find that since TRACE-y and TRACE-p are optimized respectively, they outperform each other in the corresponding experiments (logit v.s. probability). In (a)(b), TRACE-y achieves better results, while in (c)(d), TRACE-p does. But both of them outperforms attribution methods by a significant margin. Besides, it is ar-

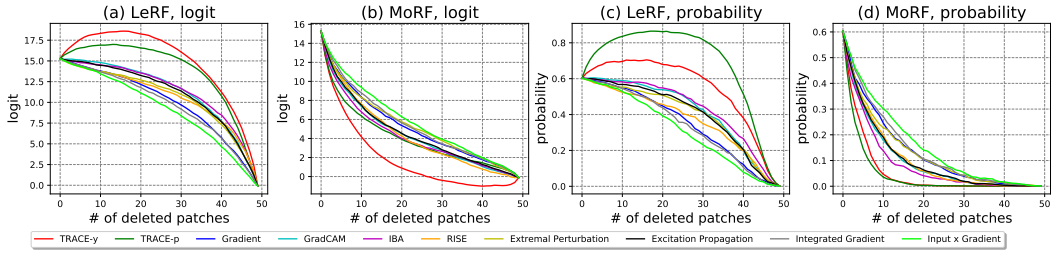


Figure 5: Deletion results of the first 200 images from the validation set of ILSVRC2012. Each image is split to $t = 49$ square patches. In (a)(c), patches are deleted following LeRF, and in (b)(d), patches are deleted following MoRF. The y-axis of (a)(b) is the output logits of the network, and the y-axis of (c)(d) is the predicted probability. x-axis is the number of masked patches.

Table 2: AUCs of curves shown in fig. 5. AUC values are computed by the Riemann sum of the corresponding curves. T-y, T-p, Grad, GC, EP, EBP, IG, I×G are abbreviations for TRACE-y, TRACE-p, Gradient, GradCAM, Extremal Perturbation, Excitation-BP, Integrated Gradient, Input × Gradient, respectively. The (a)(b)(c)(d) rows correspond to subfigures from fig. 5 respectively. (a-b) and (c-d) are differences between (a)(b) and (c)(d) respectively. For (a)(a-b)(c)(c-d), larger values are desired, and for (b)(d) smaller values are desired.

	T-y	T-p	Grad	GC	IBA	RISE	EP	EBP	IG	I×G
(a)	**740.30**	687.76	499.83	566.15	571.36	534.46	533.41	554.74	488.42	460.45
(b)	**98.31**	207.07	263.34	225.94	215.64	221.77	270.17	232.22	272.87	292.31
(a-b)	**641.99**	480.69	236.49	340.21	355.72	312.70	263.24	322.52	211.55	168.14
(c)	27.20	**33.99**	17.44	20.85	21.61	18.59	19.82	20.16	16.98	15.65
(d)	2.93	**2.49**	6.79	5.22	4.57	5.41	6.55	5.40	7.17	8.00
(c-d)	24.27	**31.50**	10.65	15.63	17.04	13.18	13.27	14.76	9.81	7.65

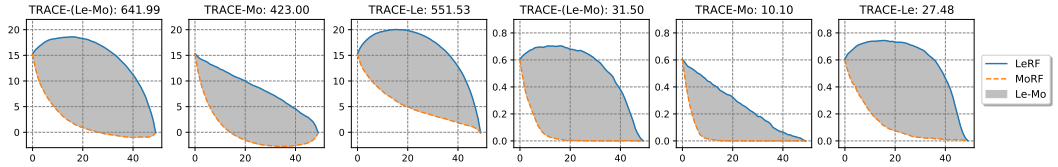
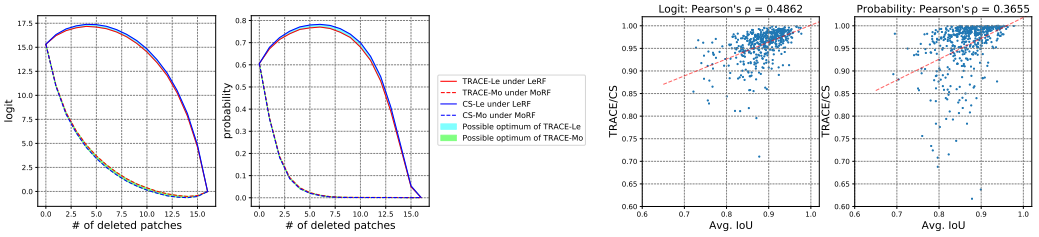


Figure 6: The deletion tests of variants of TRACE. The blue curves are the results of LeRF test, while the dashed orange curves are the results of MoRF test. The value after each method in the title is the difference between LeRF and MoRF (i.e., size of the gray area). The left three figures are w.r.t. the logit and the right three figures are w.r.t. the probability.

gued that using MoRF or LeRF alone can be insufficient. (Schulz et al., 2020) propose to use the difference between them as the metric. However, TRACE already outperforms other methods in both directions, it automatically aces this test by a large margin as shown in (a-b) and (c-d) in table 2. We visualize the results of attribution methods as the form in (Schulz et al., 2020) in appendix I

The Necessity of (Le−Mo). To demonstrate the necessity of solving TRACE-(Le−Mo) instead of just TRACE-Le or TRACE-Mo, we compare them together in the MoRF and LeRF deletion tests shown in fig. 6. Since TRACE-Mo and TRACE-Le are optimized w.r.t. the MoRF and LeRF, they perform better in the corresponding tests. However, the resulted trajectories of TRACE-Le/Mo perform poorly in the other tests. This means: 1) when the patches recognized as important by TRACE-Le are deleted, the prediction does not drop drastly; 2) when the patches recognized as important by TRACE-Mo are preserved, the prediction can not be kept in a relatively high level. As a consequence, the trajectories that only focus on MoRF or LeRF solely would not be sufficiently meaningful in the semantic ways. And it can be found that TRACE-(Le−Mo) successfully combine these two aspects.

Error Analysis of TRACE. Solved using meta-heuristic algorithms, the resulting σ is not guaranteed to be the global optimum. Although TRACE-Le and TRACE-Mo achieve the best results in LeRF



(a) TRACE v.s. CS (b) Results of $\frac{\text{TRACE}}{\text{CS}}$-Avg.IoU

Figure 7: Comparision between TRACE and Complete Search (CS). In (a), the red and blue curves are the results of TRACE (solved by SA) and CS, respectively. Solid and dashed curves are -Le and -Mo, respectively. The optimum of TRACE-Le and TRACE-Mo lie in the cyan and lime color areas, respectively. In (b), $\frac{\text{TRACE}}{\text{CS}}$-Avg.IoU results are plotted for logit y (left) and probability t (right).

and MoRF tests respectively according to fig. 5 and fig. 6, the error between them and the global optimum can still be large. In this section we empirically demonstrate that this error is very marginal.

Since $|S_t| = t!$, searching for the global optimum is impossible even for a small $t = 4^2$ case as $16! \approx 2.1 \times 10^{13}$. However, on the other hand, finding global optimum of eq. (1) for all fixed $k \in [t]$ requires only 2^t states to search, which is feasible for $t = 16$. This can be written as $\sum_{k=0}^{t} \min_{\boldsymbol{s}_k \subset [t], |\boldsymbol{s}_k| = k} f(\boldsymbol{x}_{\setminus \boldsymbol{s}_k})$, where the solved $\boldsymbol{s}_k^*$ are independent from each other. This means generally $\boldsymbol{s}_k^* \not\subset \boldsymbol{s}_{k+1}^*$. Hence the solved sequence $\{\boldsymbol{s}_k^*\}_{k=1}^t$ does not form a valid trajectory. However, since for each k the optimization is unconstrained, it is a lower bound for TRACE-Mo:

$$\sum_{k=0}^{t} f(\boldsymbol{x}_{\setminus \sigma'[:k]}) \geq \sum_{k=0}^{t} \min_{\boldsymbol{s}_k \subset [t], |\boldsymbol{s}_k| = k} f(\boldsymbol{x}_{\setminus \boldsymbol{s}_k}), \quad \forall \sigma. \tag{4}$$

The equality can hold for some σ^* only if the minimizer $\{\boldsymbol{s}_k^*\}_{k=0}^t$ satisfies $\varnothing = \boldsymbol{s}_0^* \subset \cdots \subset \boldsymbol{s}_t^* = [t]$ (i.e., the selected features in each k form a trajectory). Similarly, $\sum_{k=0}^{t} \max_{\boldsymbol{s}_k \subset [t], |\boldsymbol{s}_k| = k} f(\boldsymbol{x}_{\boldsymbol{s}_k})$ is an upper bound for TRACE-Le. We term this Complete Search (CS-Mo and CS-Le). The comparison between CS and TRACE are shown in fig. 7(a). Bounded by CS, the optimum of TRACE-Le and TRACE-Mo lie in the cyan and lime color area, which is very marginal to the SA results given the already achieved improvement to attribution methods shown in fig. 5. In addition, since the extend to which the sequence $\{\boldsymbol{s}_k^*\}$ do not form a trajectory can affect the error of SA, we introduce the average IoU to measure how distant the sequence $\{\boldsymbol{s}_k^*\}$ is from a trajectory, which is defined as

$$\text{Avg.IoU} = \mathbb{E}\left[\frac{1}{t-1} \sum_{k=1}^{t-1} \frac{|\boldsymbol{s}_k^* \cap \boldsymbol{s}_{k+1}^*|}{|\boldsymbol{s}_k^* \cup \boldsymbol{s}_{k+1}^*|} \cdot \frac{k+1}{k}\right], \tag{5}$$

where $\frac{k+1}{k}$ is to balance each term as $|\boldsymbol{s}_k^*| = k$. The expectation is taken over input samples. Obviously, Avg.IoU $= 1$ when $\boldsymbol{s}_k^* \subset \boldsymbol{s}_{k+1}^*$ and Avg.IoU $= 0$ when $\boldsymbol{s}_k^* \cap \boldsymbol{s}_{k+1}^* = \varnothing, \forall k \in [t-1]$. We plot $\frac{\text{TRACE}}{\text{CS}}$-Avg.IoU in fig. 7(b). Here the y-axis represents the ratio of (TRACE-Le)−(TRACE-Mo) to (CS-Le)−(CS-Mo), which is 1 only if (but not if) TRACE reaches global optimum. Results show that $\frac{\text{TRACE}}{\text{CS}}$ does increase towards 1 when Avg.IoU approaches 1. It can also be found that even when $\{\boldsymbol{s}_k^*\}$ are far from a trajectory, TRACE may still approach the global minimum closely.

7 Conclusion

In this paper, we propose TRACE, a novel model-agnostic explanation framework based on trajectories. TRACE constrains the feasible set of perturbation-based methods from the Euclidean space into a symmetric group, and introduces combinatorial optimization tools to solve for the problem. We point out the problems in the existing studies of attribution methods – the meaning of attribution values are ambiguous and they are not made use of. As the essence of existing attribution methods, after converted to attributions, TRACE not only outperforms attribution methods in the most commonly used deletion/insertion metrics by a significant margin, but also is demonstrated to be potentially able to achieve the optimality in the deletion/insertion test. That being said, either TRACE is recognized as the best explanation, or this is a sanity check for the deletion/insertion metric that it does not pass. In this way, with different objective functions, TRACE can be a guideline for developing better metrics when concrete attribution scores are not involved. We leave this to the future work. Also, it would be interesting to explore more efficient tools for the combinatorial optimization problem.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems*, 34:4699–4711, 2021a.

Siddhant Agarwal, Owais Iqbal, Sree Aditya Buridi, Madda Manjusha, and Abir Das. Reinforcement explanation learning. *arXiv preprint arXiv:2111.13406*, 2021b.

Sheldon B. Akers and Balakrishnan Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE transactions on Computers*, 38(4):555–566, 1989.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

BAAI. Wu dao 2.0. `https://gpt3demo.com/apps/wu-dao-20`, 2020. Accessed: 2022-09-22.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Malti Baghel, Shikha Agrawal, and Sanjay Silakari. Survey of metaheuristic algorithms for combinatorial optimization. *International Journal of Computer Applications*, 58(19), 2012.

Professor Cayley. Desiderata and suggestions: No. 2. the theory of groups: graphical representation. *American journal of mathematics*, 1(2):174–176, 1878.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Xiutang Geng, Zhihua Chen, Wei Yang, Deqian Shi, and Kai Zhao. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing*, 11(4):3680–3689, 2011.

Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549, 1986.

Gena Hahn and Gert Sabidussi. *Graph symmetry: algebraic methods and applications*, volume 497. Springer Science & Business Media, 2013.

John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pp. 1–14, 2020.

Saeed Khorram, Tyler Lawson, and Li Fuxin. igos++ integrated gradient optimized saliency by bilateral perturbations. In *Proceedings of the Conference on Health, Inference, and Learning*, pp. 174–182, 2021.

Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. The disagreement problem in explainable machine learning: A practitioner's perspective. *arXiv preprint arXiv:2202.01602*, 2022.

Sivaramakrishnan Lakshmivarahan, Jung-Sing Jwo, and Sudarshan K. Dhall. Symmetry in interconnection networks based on cayley graphs of permutation groups: A survey. *Parallel computing*, 19(4):361–407, 1993.

Matthew L Leavitt and Ari Morcos. Towards falsifiable interpretability research. *arXiv preprint arXiv:2010.12016*, 2020.

Liangzhi Li, Bowen Wang, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, and Hajime Nagahara. Scouter: Slot attention-based classifier for explainable image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1046–1055, 2021a.

Xiao-Hui Li, Yuhan Shi, Haoyang Li, Wei Bai, Caleb Chen Cao, and Lei Chen. An experimental study of quantitative evaluations on saliency methods. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3200–3208, 2021b.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4768–4777, 2017.

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

Zhongang Qi, Saeed Khorram, and Fuxin Li. Visualizing deep networks by optimizing with integrated gradients. In *CVPR Workshops*, volume 2, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International conference on machine learning*, pp. 8116–8126. PMLR, 2020.

Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. *arXiv preprint arXiv:2001.00396*, 2020.

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

L Shapley. Quota solutions op n-person games1. *Edited by Emil Artin and Marston Morse*, pp. 343, 1953.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

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

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.

Yipei Wang and Xiaoqian Wang. Self-interpretable model with transformation equivariant interpretation. *Advances in Neural Information Processing Systems*, 34:2359–2372, 2021.

Yipei Wang and Xiaoqian Wang. A unified study of machine learning explanation evaluation metrics. *arXiv preprint arXiv:2203.14265*, 2022a.

Yipei Wang and Xiaoqian Wang. "why not other classes?": Towards class-contrastive back-propagation explanations. volume 36, 2022b.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10819–10829, 2022.

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

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