Benchmarking Deletion Metrics with the Principled Explanations

Yipei Wang [1] **Xiaoqian Wang** [1]

Abstract

Insertion/deletion metrics and their variants have been extensively applied to evaluate attribution-based explanation methods. Such metrics measure the significance of features by observing changes in model predictions as features are incrementally inserted or deleted. Given the direct connection between the attribution values and model predictions that insertion/deletion metrics enable, they are commonly used as the decisive metrics for novel attribution methods. Such influential metrics for explanation methods should be handled with great scrutiny. However, contemporary research on insertion/deletion metrics falls short of a comprehensive analysis. To address this, we propose the TRAjectory importanCE (`TRACE`) framework, which achieves the best score in the insertion/deletion metric. Our contribution includes two aspects: 1) `TRACE` stands as the principled explanation for explaining the influence of feature deletion on model predictions. We demonstrate that `TRACE` is guaranteed to achieve almost optimal results both theoretically and empirically. 2) Using `TRACE`, we benchmark insertion/deletion metrics across all possible settings and study critical problems such as the out-of-distribution (OOD) issue and provide practical guidance on applying these metrics in practice. The implementation of `TRACE` is available as open source at GitHub.

1. Introduction & Background

With the rapid increase in computational power, deep neural networks have achieved remarkable success in many domains. Despite their impressive performance, DNNs are often criticized for their black-box nature, especially in critical applications where understanding the decision-making process is crucial. To address this opacity, the field of explainable artificial intelligence (XAI) has emerged and developed rapidly, with various explanation methods introduced (Arrieta et al., 2020). Among these, attribution methods stand out and are widely used due to their straightforwardness and intuitive visualizations (Adebayo et al., 2018; Leavitt & Morcos, 2020). Given an input of d features, such as pixels, tokens, or patches, attribution methods assign an attribution value to each feature, illustrating its "importance" to the output. Such an approach offers a clear insight into feature relevance and allows humans to directly comprehend it as it aligns well with the principles of linear models.

While attribution methods often take similar forms, they can originate from various methodologies and objectives. Given the same input data and the same black-box prediction model, different attribution methods can produce vastly different explanations. This variability presents a challenge for both end-users and researchers in selecting the most appropriate explanation method (Kaur et al., 2020; Krishna et al., 2022). To address this issue, evaluation *metrics* for attribution methods have been introduced to evaluate different explanations and identify the most suitable explanation approach. These metrics generally fall into two main categories: alignment and performance. Alignment metrics, such as the pointing game (Zhang et al., 2018), inspect how explanations align with the prior knowledge of the *data*. It has been critiqued that such metrics are actually evaluating the plausibility to humans rather than reflecting actual model behaviors (Jacovi & Goldberg, 2020; Wang & Wang, 2022b). In contrast, performance metrics such as insertion/deletion emphasize the model performance, where input features are perturbed (deleted/inserted, etc.) progressively according to their attribution values. Then the AUCs of the resulting curve, which contrasts model predictions against the proportion of perturbed features, serve as the evaluation criterion for the attribution method. For instance, when the most relevant features are deleted first (denoted as **MoRF**), a low AUC is anticipated. Conversely, when the least relevant features are deleted First (**LeRF**), a high AUC is then expected. Deletion metrics characterize important features as those who *affect the model prediction the most when progressively deleted*. The metric's widespread use suggests that such property is valued in the XAI community.

[1]Elmore Family School of Electrical and Computer Engineering, Purdue University, IN, USA. Correspondence to: Xiaoqian Wang <joywang@purdue.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

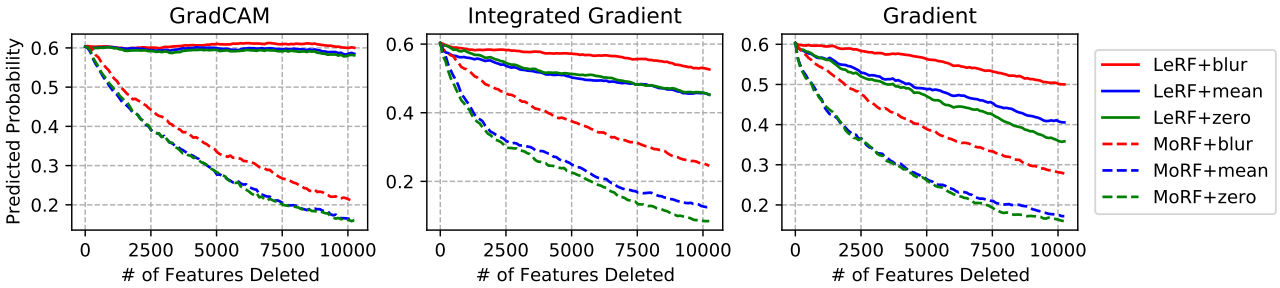


Figure 1. The deletion tests of GradCAM, Integrated Gradient, and Gradient. Solid and dashed curves distinguish between LeRF and MoRF criteria. Different colors represent different reference types. We include the zero, mean, and blurring references.

Related Work of the Studies of Deletion Metrics. Despite the deletion metrics' prominence as a preferred choice for evaluating attribution methods (Samek et al., 2016; Binder et al., 2016; Petsiuk et al., 2018; Chen et al., 2018; Qi et al., 2019; Schulz et al., 2019; Wang & Wang, 2021; Khorram et al., 2021; Covert et al., 2021; Chen et al., 2021), it is crucial to recognize that *metrics should undergo rigorous studies before widespread adoption.* Deletion metrics, with different settings such as the choices of the reference values for the deleted features, LeRF/MoRF criteria, feature sizes, etc., may yield distinct results. Hence it is important to make a judicious choice among these variants in practice.

Besides, as the most paramount problem in the context of the deletion test, the OOD issue refers to the phenomenon that when only a small amount of input features are deleted, the input becomes out-of-distribution. As a result, the model performance decays significantly even if the informative features remain relatively intact. Although there are existing studies pointing out the OOD problem of the deletion metrics and proposing related workarounds (Hooker et al., 2019; Sturmfels et al., 2020; Schulz et al., 2019; Rong et al., 2022), they fall short in certain aspects. Hooker et al. (2019) propose to remove and retrain (ROAR) to alleviate the OOD issue. However, it requires training black-box models from scratch every time the number of deleted features changes, which is computationally expensive and hardly applied. Also, since the models change every time, it leans on explaining the dataset and the model family instead of the specific black-box model of interest (Sturmfels et al., 2020; Zhou et al., 2021; Ras et al., 2022). Schulz et al. (2019) argue that using MoRF or LeRF individually is insufficient and propose to use the difference between them as the measurement. But the statement lacks justifications. Rong et al. (2022) introduce remove and debias (ROAD), a weighted summation of the 8 surrounding pixels of the deleted one as the reference values, which is an intermediate stage between mean imputation and blurring. However, like other existing work, the proposed method is verified simply by observing whether the four selected explanation methods are ranked consistently under LeRF and MoRF. This raises risks because *studies of metrics should not be restricted by specific explanation methods.*

Tethered to popular explanation methods such as Gradient (Simonyan et al., 2014), Integrated Gradient (Sundararajan et al., 2017), GradCAM (Selvaraju et al., 2017) etc., existing studies of the deletion metric fall into circular reasoning – These explanation methods, originally subjects of the deletion metric, are paradoxically used to validate the metric itself. Hence the assessment of the metrics will be highly biased by the selected explanation methods. For instance, to analyze the reference values in deletion metric, studies focusing on discrete attributions such as Gradient or IG are likely to conclude that the difference between reference values is significant, while studies focusing on smooth attributions such as GradCAM may conclude otherwise. Figure 1 shows the deletion tests of three methods, with zero, mean, and blurring references. For Gradient and IG, different reference types lead to completely different scores. However, for GradCAM, the difference between zero reference and blurring reference is much less concerning. These opposite results suggest that studies of the metrics should not rely on existing explanation methods, but instead be approached through the essence of the metric itself.

In response to these problems, we introduce the TRAjectory importanCE (`TRACE`) framework, which achieves the highest score of deletion metric both empirically and *theoretically*. By maximizing the score of the metric, `TRACE` is capable of (1) representing *what the metric really measures*, embodying the principled explanations associated with the deletion metric that reflect the exact influence of feature deletion on model predictions; and (2) benchmarking all the settings of the deletion metric, and providing guidance on how different choices can suffer from or be the remedy to the infamous OOD issue. The main contributions of this paper are summarized as follows.

- We formally study the mathematical essence of deletion metrics to reveal the intrinsic properties of the

metrics without relying on existing XAI methods.

- We propose TRACE, a combinatorial optimization framework to generate the *principled explanation* of the deletion metric and validate its near-optimality both empirically and theoretically. Thus it represents the exact feature importance under feature deletion.
- Using the principled explanations, we present a rigorous study of the various settings, and provide guidelines to effectively mitigate the OOD problem.

2. Methodology

In this section, the details of the TRACE framework are introduced. The discussion covers its solution using combinatorial optimizations. We introduce various settings of the deletion metrics and TRACE in Section 3. This section begins with a formalization of the deletion metric.

Formalization of Deletion Metric. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a black-box model. An attribution method is defined as a mapping $\varphi_f : \mathbb{R}^d \to \mathbb{R}^d, \boldsymbol{x} \mapsto \psi$. For $\forall \delta \subseteq \{1, \cdots, d\}$, let $\boldsymbol{x}_{\setminus\delta}$ denote the input where features indexed by δ are deleted, and $\boldsymbol{x}_\delta$ denote it where features index by δ are kept. Then given the tuple $(f, \boldsymbol{x}, \psi)$, the deletion metric AUC score under the MoRF criterion can be written as

$$\text{MoRF}(\psi) = \sum_{k=0}^{d} f(\boldsymbol{x}_{\setminus\sigma(\psi)[k:]}) = \sum_{k=0}^{d} f(\boldsymbol{x}_{\sigma(\psi)[:k]}) \quad (1)$$

Similarly, $\text{LeRF}(\psi) = \sum f(\boldsymbol{x}_{\setminus\sigma(\psi)[:k]}) = \sum f(\boldsymbol{x}_{\sigma(\psi)[k:]})$. Here σ maps the attribution map ψ to a permutation of feature indices in the bottom-top order. That is, $\psi_{\sigma(\psi)[j]} \leq \psi_{\sigma(\psi)[j+1]}$. And $\boldsymbol{x}_{\setminus\sigma(\psi)[k:]}, \boldsymbol{x}_{\sigma(\psi)[:k]}$ represent the input data where (a) the last k features indexed by $\sigma(\psi)$ are deleted and (b) the first k features indexed by $\sigma(\psi)$ are kept, respectively. For example, if the attributions are $\psi = [0.1, 0.5, 0.3, 0.2]$, then $\sigma(\psi) = [1, 4, 3, 2]$ and $\boldsymbol{x}_{\sigma(\psi)[:1]} = \boldsymbol{x}_{\setminus\sigma(\psi)[3:]} = \boldsymbol{x}_{[1]} = \boldsymbol{x}_{\setminus[4,3,2]} = [x_1, \texttt{ref}_2, \texttt{ref}_3, \texttt{ref}_4]$ denote the input where the features x_4, x_3, x_2 are deleted. With the notations defined above, the best attribution-based explanation of the model prediction $f(\boldsymbol{x})$ under the deletion metric with MoRF criterion is naturally

$$\begin{aligned} \psi^*_{MoRF} &= \arg\min_{\psi \in \mathbb{R}^d} \text{MoRF}(\psi) \\ &= \arg\min_{\psi \in \mathbb{R}^d} \sum\nolimits_{k=0}^{d} f(\boldsymbol{x}_{\sigma(\psi)[:k]}) \end{aligned} \quad (2)$$

Regrettably, the optimization of this objective to find the "best explanation" is infeasible. The study is confined to comparing the scores in Equation (1) between two attributions ψ^1, ψ^2. This limitation underscores why existing studies on the deletion metric rely heavily on specific explanation methods.

Trajectory Importance (TRACE). To address these challenges, we introduce TRACE. A crucial observation is that although attribution explanations are presented as dense vectors in the Euclidean space $\mathbb{R}^d$, their evaluations under the deletion metric are not based on detailed attribution values. In fact, by defining that $\sigma(\psi)[i] < \sigma(\psi)[i+1]$ when $\psi_{\sigma[i]} = \psi_{\sigma[i+1]}$, an attribution ψ can be mapped to a **unique** permutation of the indices $\{1, \cdots, d\}$. We thus define an equivalence relation R, where two attributions ψ^1, ψ^2 are equivalent when they map to the same permutation, i.e., $\psi^1 R \psi^2 \Leftrightarrow \sigma(\psi^1) = \sigma(\psi^2)$. And ψ^1, ψ^2 receive identical scores under the deletion metric. In consideration of this, we quotient out the equivalence class with the projection map $\mathbb{R}^d \to \mathbb{R}^d/R, \psi \mapsto [\psi]$. And since the equivalence class $[\psi]$ can be mapped to the permutation $\sigma(\psi)$ in a 1-to-1 manner, we have $\mathbb{R}^d/R \cong \mathcal{S}_d$. Here $\mathcal{S}_d$ denotes the symmetric group of order d, which consists of all permutations of $\{1, \cdots, d\}$. Proofs are shown in Appendix B.1. As a result, the original problem in Equation (2) transforms into an optimization over a finite, well-structured set $\mathcal{S}_d$ as

$$\begin{cases} \texttt{TRACE-Mo}: \min\limits_{\tau \in \mathcal{S}_d} \sum\nolimits_{k=0}^{d} f(\boldsymbol{x}_{\tau[:k]}); \\ \texttt{TRACE-Le}: \min\limits_{\tau \in \mathcal{S}_d} \sum\nolimits_{k=0}^{d} f(\boldsymbol{x}_{\tau[k:]}) \end{cases} \quad (3)$$

To differentiate between our framework and the test of the deletion metric under different criteria (e.g. MoRF, LeRF), we use the prefix TRACE (e.g. TRACE-Mo). In other words, MoRF is an evaluation criterion defined in Equation (1), where lower values in MoRF indicate better explanations under the deletion metric; while TRACE-Mo is the optimization problem defined in Equation (3). This new formulation sets the stage for combinatorial optimization with adequate tools. Specific algorithms are discussed in Section 4.

Trajectory to Attributions. The optimizer of Equation (3) is a trajectory τ traversing all features in the bottom-top order. While the mapping from the attributions ψ to the corresponding trajectory $\tau = \sigma(\psi)$ is surjective, τ can map back to attribution in its equivalence class. Define $\Pi_\tau = \{\pi | \pi : \mathcal{S}_d \to \mathbb{R}^d, \tau \mapsto \psi, \text{ s.t. } \sigma(\psi) = \tau\}$ as the set of mappings from τ to attributions that preserve the trajectory. Thus $\forall \pi \in \Pi_\tau$, $\pi(\tau) \in \mathbb{R}^d$ is a valid attribution map of τ. We define $\pi(\tau) = (\tau^{-1}/d)^\alpha \in [0, 1]^d$, where $\tau^{-1} = \texttt{argsort}(\tau)$ is the ranking of features in the trajectory τ for simplicity. Here α controls the size of the highlighted region, which is similar to the colormap choices. Using π, we can map the optimizer τ from Equation (3) to attributions ψ, and visualize ψ as a heatmap, offering insights akin to attribution explanation methods. We visualize the TRACE results as heatmaps in Appendix C.

3. Settings of Deletion Metrics and TRACE

As discussed in Section 1, various settings of the deletion metric give rise to a plethora of variants, resulting in distinct evaluation results even for the same $(f, \boldsymbol{x}, \psi)$ tuple, such as the differences shown in Figure 1. However, the judicious choice among these variants remains unclear. Here we discuss these possible variants comprehensively and study how their choices can influence the metric via the principled explanations from TRACE in Section 5. Note that these settings influence both the metric through the term $f(\boldsymbol{x}_{\sigma(\psi)[:k]})$ in Equation (1) (i.e., when using deletion metrics for evaluation in practice), and the TRACE framework through the term $f(\boldsymbol{x}_{\tau[:k]})$ in Equation (3) (when determining the optimization objective). Also, it should be noted that the TRACE framework is compatible with any input data types. In this work, we focus on the image data, which is most influenced by the OOD issue.

Deletion vs. Insertion. Although the insertion metric serves as a popular alternative to the deletion metric and inserts features instead of deleting them, the differences between them are neutralized when the AUC is used for assessment. In fact, we prove in Appendix B.2 that they are equivalent and will focus on deletion in the following context for clarity.

Theorem 3.1. *Insertion-MoRF is equivalent to Deletion-LeRF; Insertion LeRF is equivalent to Deletion-MoRF.*

Logit vs. Probability. Model outputs, denoted by $f(\boldsymbol{x})$, vary in different contexts. For classifiers, both the predicted logit from the final linear layer and the probability yielded by the softmax activation can be seen as the output in standard practice. Notably, previous studies demonstrate that perturbations concerning logits differ from probabilities (Wang & Wang, 2022a). We include this variation with the suffixes -y (for logit) and -p (for probability).

MoRF vs. LeRF. The two criteria MoRF and LeRF, though seem symmetric, have very distinct interpretations. MoRF defines important features as *those who diminish the performance the most when deleted*. Conversely, LeRF sees features as crucial if they *maintain the performance the most when kept*. Taking both aspects into consideration, the "important features" should be able to diminish the model performance when deleted and preserve the model performance when kept. We denote this variant as LeRF−MoRF, which uses the difference $\sum_{k=0}^{d} \left(f(\boldsymbol{x}_{\tau[:k]}) - f(\boldsymbol{x}_{\tau[:k]})\right)$ as the objective in Equation (3). In experiments, we consider all three variants: -Le, -Mo, and -Le−Mo.

Reference Values. Black-box models such as DNNs take inputs of a fixed size. Thus the deleted features have to be replaced with predefined reference values to represent the "null feature". The choices of reference values can significantly affect the results of the metric for some explanation methods. The current conventional way is to use heuristic methods such as zeros, means, and blurrings to avoid introducing exogenous information and overcomplicating problems (Lundberg & Lee, 2017; Sundararajan et al., 2017; Hooker et al., 2019; Shrikumar et al., 2017; Sturmfels et al., 2020; Covert et al., 2021; Rong et al., 2022). In fact, the choices of reference value types are tightly connected with the OOD issue via the trade-off between **deleting the feature** and **preserving the distribution**. The zero reference deletes features completely but breaks the input distribution severely. In contrast, in the context of blurring reference, the original distribution is always preserved. However, the deleted features are also partially recovered, which can lead to problematic deletion tests in practice. To study such influence, we include three types of reference values in our experiments: zeros, means, and blurrings.

Input Feature Size. Within an input image of size 224×224, the semantic meaning of pixel-wise attributions is very limited (Rieger et al., 2020). Grouping pixels and dealing with the superpixel patches, on the other hand, have been demonstrated to achieve great success (Dosovitskiy et al., 2020; Tolstikhin et al., 2021; Yu et al., 2022). It is also observed that the deletion metrics have been implemented with different resolutions. As a result, we operate on t superpixel square patches, where the patch sizes are $\frac{224}{\sqrt{t}} \times \frac{224}{\sqrt{t}}$ (specially, when $t = 224 \times 224$, each patch is a pixel). By comparing the results of different patch sizes, we observe that the OOD issue is greatly mitigated by decreasing the resolution t of the deletion process. Larger patches result in less noisy trajectories, but coarser explanations, while smaller patches lead to finer results but are much more vulnerable to the OOD problem. We study the influence of different patch sizes comprehensively in Section 5.2. And we will abuse the notations a little to denote by $\boldsymbol{x}_{\setminus\tau[k:]}$ or $\boldsymbol{x}_{\tau[:t-k]}$ the input image with the top k patches deleted (i.e. bottom $t-k$ patches kept).

4. Algorithms for TRACE

Complexity Analysis. The TRACE framework in Equation (3) aims at finding a trajectory τ of features that optimize the "cost" defined by $(f, \boldsymbol{x})$. Therefore, it is a non-trivial problem and can be solved by combinatorial optimization with meta-heuristic algorithms. In Appendix B.3, we prove that TRACE is NP-hard by relating to the traveling salesman problem (TSP).

Theorem 4.1. *The optimization problem TRACE-Mo ($\{\min_\tau \sum_{k=0}^{d} f(\boldsymbol{x}_{\tau[:k]})\}$) is NP-hard.*

Heuristic Approaches. In order to quickly identify a trajectory of features that optimizes the objective outline Equation (3), one direct method is the greedy strategy. Instead of seeking an entire trajectory τ dynamically, this approach sequentially deletes one feature in each step. Starting from

the highest ranked feature (the lowest one for TRACE-Le), it finds the feature that minimizes the prediction when deleted in each step. Such an approach, while fast, is usually sub-optimal. Yet it reaches the global optimal of TRACE-Mo/Le if the features' contributions are additive, such as with linear models. It's essential to note, however, that this approach yields distinct trajectories for MoRF and LeRF, and thereby does not apply to the LeRF−MoRF test, resulting in ineluctable trade-offs between the principled explanation's optimality and efficiency. We demonstrate in Section 5 that TRACE-Greedy still outperforms all existing explanation methods significantly, and thus also serves the role of near-principled explanations w.r.t. feature deletion.

Meta-Heuristic Approaches. When benchmarking the deletion metric, the above compromise can cause insufficiency. Addressing the limitation described requires that the entire trajectory τ be optimized comprehensively. In such contexts, meta-heuristic algorithms are the judicious choice given their established efficacy in combinatorial optimization challenges (Baghel et al., 2012). Among them, simulated annealing (SA) (Kirkpatrick et al., 1983) has been actively employed in problems such as TSP to deliver sufficiently good sub-optimal results (Geng et al., 2011). Given its efficacy and theoretical grounding, we too adopt SA in our methodology. The associated pseudo-code is provided in Appendix D. We also explore alternative meta-heuristic algorithms in Appendix E. In the following context, TRACE refers to TRACE-SA unless otherwise claimed.

Neighbor Sets of SA. The performance of SA depends on the apt choice of neighbors, especially on a discrete feasible set where the distance is not well-defined. Meanwhile, TRACE is essentially a harder problem than TSP, where the pairs of directly connected cities determine the total cost. TRACE considers not only the consecutively deleted patches but also the overall ordering of deleting patches matter. For instance, if a segment in the trajectory is reversed, TSP's costs only change for the segment's two endpoints. However, in TRACE, all values post the segment's initial change. As such, common neighbor strategies for TSP like vertex insertion, block insertion, and block reverse (Geng et al., 2011) do not transition to TRACE directly. Our comprehensive study on suitable neighbors for TRACE can be found in Appendix B.4, where we conclude that the optimal neighbor set should comprise all trajectories derived from the initial trajectory, τ_0, by swapping two distinct features: $N(\tau_0) = \{\tau | \exists i, j, i \neq j, \tau_0[i] = \tau[j], \tau_0[j] = \tau[i]\}$.

Optimality of the Algorithms. While TRACE demonstrates exceptional performance in the deletion metrics, it is acknowledged that when employing meta-heuristic/heuristic algorithms, the resultant τ is not necessarily the global optimum. To validate the approximation to the optimum, we undertake an empirical study to bound the deviation between TRACE-SA/TRACE-Greedy and their global optimum. However, as a black-box optimization problem, the theoretical global optimum of Equation (3) (denoted by TRACE-GO) is inaccessible. Exhaustively searching for the global optimum is also impractical since $|S_t| = t!$. Hence instead of comparing with TRACE-GO, we propose complete search (CS), which is proved to be the lower bound of TRACE-GO. Formally, for $k = 1, \cdots, t$, CS-Mo solves for an index set $\boldsymbol{s}_k$ consisting of k deletion features that minimizes the prediction $f(\boldsymbol{x}_{\setminus \boldsymbol{s}_k})$. Therefore, it is the lower bound of the corresponding term of TRACE-Mo in Equation (3): $\forall \tau \in \mathcal{S}_t, \forall k \in \{1, \cdots, t\}, f(\boldsymbol{x}_{\tau[:t-k]}) \geq \min_{\boldsymbol{s}_k \subset \{1,\cdots,t\}, |\boldsymbol{s}_k|=k} f(\boldsymbol{x}_{\setminus \boldsymbol{s}_k})$. The equality hold only if $\forall k \in \{1, \cdots, t-1\}$, the optimizers $\boldsymbol{s}_k^*$ satisfy $\boldsymbol{s}_k^* \subset \boldsymbol{s}_{k+1}^*$,. As a consequence, by summing up over k, we have

$$\text{TRACE-(Greedy/SA)-Mo} \geq \text{TRACE-GO-Mo} \geq \text{CS-Mo}$$

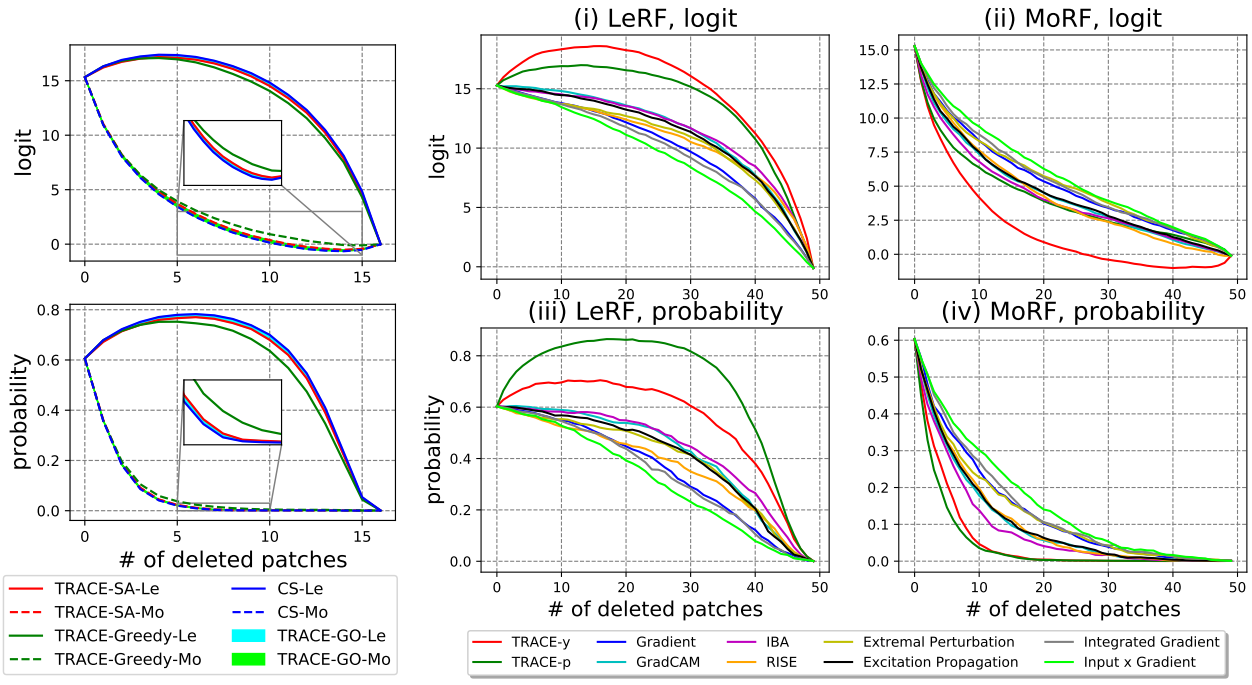
Similar inequality holds for the -Le variant: TRACE-(Greedy/SA)-Le $\leq$ TRACE-GO-Le $\leq$ CS-Le. Therefore, by squeezing TRACE-(Greedy/SA) and CS, we can then verify the near-optimality of the algorithms (i.e., TRACE-(Greedy/SA) is close to the theoretical global optimum TRACE-GO)

5. Experiments

In this section, we conduct experiments to 1) Validate TRACE's optimality and the capability of serving as the principled explanation; and 2) Use TRACE to assess the impact of different settings (as discussed in Section 3) to address the OOD concern in deletion metric. We use a ResNet-18 model (He et al., 2016) as the black box f for the demonstration. Other popular models such as AlexNet (Krizhevsky et al., 2017), VGG-16 (Simonyan & Zisserman, 2015), GoogLeNet (Szegedy et al., 2015), DenseNet-161 (Huang et al., 2017), and MobileNetV3 (Howard et al., 2019) are evaluated, too. We adopt the pre-trained weights from `torchvision`. Experiments utilize the ImageNet-1k (ILSVRC2012) dataset (Deng et al., 2009) with images resized to 224×224. For SA, we use $K = 5000$ iterations, initial temperatures of $T_0 = 2$ for -y and $T_0 = 0.1$ for -p, and a cooling rate of $\eta = 0.999$. Experiments are carried out on Intel(R) Core(TM) i9-9960X CPU @ 3.10GHz with Quadro RTX 6000 GPUs.

5.1. Verification of the Optimality

SA/Greedy vs. GO vs. CS. To validate the optimization through Greedy and SA achieves the principled explanation of deletion metric, we demonstrate their closeness to CS, and thus squeeze the possible range of TRACE-GO. We test both -Mo with MoRF and -Le with LeRF. Because of the complexity of CS, we let $t = 4 \times 4 = 16$ here. As shown in Figure 2 (a), TRACE-SA (red) constantly out-



(a) SA/Greedy vs. GO vs. CS

(b) Deletion Test between TRACE and Attribution Methods

Figure 2. (a) The comparison between TRACE-SA/Greedy, TRACE-GO and Complete Search (CS). CS-Mo/CS-Le are the lower/upper bounds of TRACE-Mo/TRACE-Le, respectively. The blue, red, and green curves are the results of CS, TRACE-SA and TRACE-Greedy. Solid and dashed curves are -Le and -Mo. The optimum TRACE-GO-Le and TRACE-GO-Mo lie in the cyan and lime color areas, respectively, which are notably marginal. (b) Deletion results of the first 200 images from the validation set of ILSVRC2012. In (i)(iii), patches are deleted following LeRF, and in (ii)(iv), patches are deleted following MoRF. The y-axis of (i)(ii) is the output logits of the network, and the y-axis of (iii)(iv) is the predicted probability. x-axis is the number of deleted patches.

performs TRACE-Greedy (green), suggesting better performance from meta-heuristic algorithms. The difference between TRACE-SA (red) and CS (blue) is almost negligible, resulting in extremely squeezed areas for TRACE-GO between them, as shown in the cyan and lime areas (which are almost invisible). This suggests that TRACE-SA almost achieves the global optimum, and is capable of serving as the principled explanation of the deletion metric. TRACE-Greedy can also be used as the near-principled explanation when it's acceptable to trade performance for efficiency.

The Optimality over Explanation Methods. Conventionally, we compare TRACE with existing explanation methods, to demonstrate the deviation of existing explanation methods from the principled one. The results are demonstrated in Figure 2(b). We present this comparison using TRACE-SA-Le−Mo. And the resolution of both TRACE and the deletion metric is set to $t = 7 \times 7 = 49$. We elaborate on these choices in the next section, where we benchmark all settings of the deletion metric with TRACE. Our observation highlights that existing attribution methods significantly underperform compared to the principled explanation provided by TRACE. Before TRACE's introduction, one might speculate that IBA (purple) is approaching the best AUCs given its superiority to other explanation methods. However, as shown in Figure 2(b) (i)(iii), TRACE reveals that the model performances can even increase substantially when unimportant features are deleted. In Figure 2(b), we use zero references for the demonstration. Further, we show AUCs across different reference values and black-box models in Table 1, where probability is used as the measurement. The same experiments for the logit can be found in Appendix G.

5.2. Benchmarking Deletion Metrics with TRACE

Probability vs. Logit. Comparisons between TRACE-p and TRACE-y in Figure 2(b) and Table 1 suggest that the principled explanations w.r.t. probability and logit align compatibly. For instance, in Figure 2(b):(iii)-(iv) and Table 1 where the evaluation is based on *probabilities*, though both TRACE-p and TRACE-y surpass all attribution methods, their discrepancy is non-negligible. In other words, TRACE-p (as the principled explanation for deletion metric with *probabilities*) performs better than TRACE-y in the evaluation using *probabilities*. Similar results are observed in Figure 2(b):(i)-(ii) when evaluating with *logits*. Thus in

Table 1. The comparison among commonly studied DNNs on ILSVRC2012 with three different reference values. The tested models are (i) ResNet-18, (ii) VGG-16, (iii) AlexNet, (iv) GoogLeNet, (v) MobileNetV3, and (vi) DenseNet-161. The tested methods are Gradient (Grad), Grad-CAM (GC), Information Bottleneck Attribution (IBA), RISE, Extremal Perturbation (EP), Excitation Back-Propagation (EBP), Integrated Gradient (IG), and Input×Gradient. T-y/p stand for TRACE-y/p, respectively. Here we present the difference between AUCs of the *probabilities* for LeRF and MoRF, so larger values are desired.

Ref.	M.	T-y	T-p	Grad	GC	IBA	RISE	EP	EBP	IG	IxG
Zero	(i)	24.98	**31.69**	11.52	16.24	15.92	14.52	13.41	15.39	10.28	8.21
	(ii)	25.80	**31.25**	14.03	16.28	18.77	17.07	15.63	16.36	13.71	10.86
	(iii)	15.40	**21.83**	7.05	7.55	8.13	7.43	7.63	7.55	6.64	5.86
	(iv)	23.70	**28.13**	11.31	14.31	14.06	12.53	11.98	13.67	10.29	8.72
	(v)	27.55	**33.86**	8.15	16.78	13.06	10.0	11.08	10.74	8.49	6.20
	(vi)	28.00	**35.25**	11.35	19.82	18.87	18.18	17.04	19.17	12.20	9.33
Mean	(i)	25.64	**32.51**	11.45	16.22	15.66	14.57	12.96	15.40	10.51	8.48
	(ii)	26.66	**32.64**	14.09	17.00	19.06	17.63	15.73	16.58	13.83	10.77
	(iii)	16.33	**23.32**	8.91	9.82	9.20	10.65	9.83	9.82	8.48	6.73
	(iv)	24.02	**29.17**	11.63	14.49	14.14	12.70	11.85	13.89	10.63	9.06
	(v)	27.25	**34.26**	8.25	17.07	13.64	10.82	11.69	11.45	8.36	5.99
	(vi)	29.02	**36.19**	11.26	20.06	18.80	18.06	17.70	19.48	12.15	8.96
Blurring	(i)	27.34	**33.84**	10.93	17.38	16.41	16.01	14.98	16.24	10.04	7.57
	(ii)	27.50	**34.09**	14.51	17.89	19.27	18.15	15.81	17.06	14.73	11.46
	(iii)	19.55	**26.78**	8.90	9.81	9.20	10.65	10.00	9.81	8.47	6.73
	(iv)	24.86	**29.83**	11.60	14.50	14.31	13.44	12.55	13.89	10.95	9.24
	(v)	24.74	**31.57**	9.43	15.63	14.09	10.22	12.71	12.97	9.81	7.56
	(vi)	29.69	**36.87**	10.93	19.20	18.11	17.95	17.28	18.63	11.55	8.07

practice, one should be aware of the desired goal (probability or logit) of the evaluation and select correspondingly.

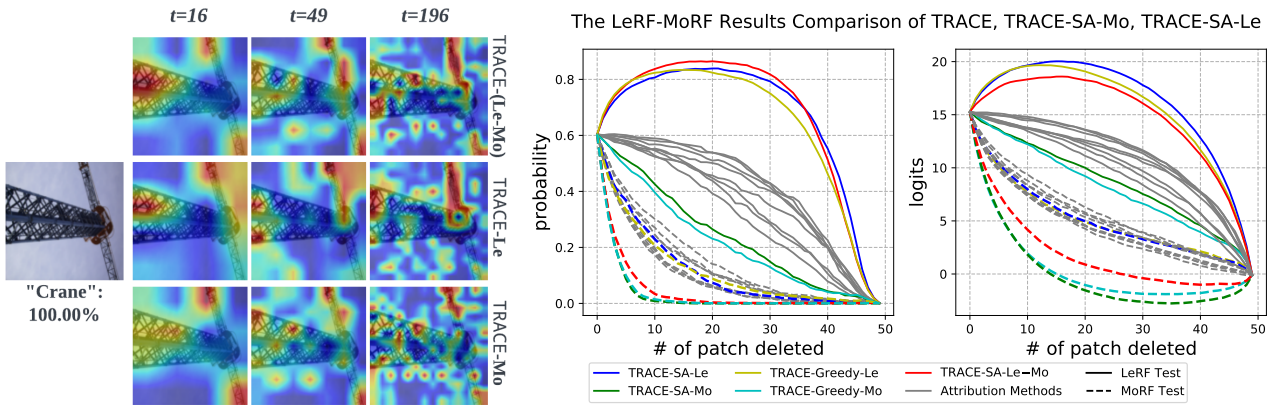
Reference Values. Recall that in Figure 1, where features are defined as pixels, different reference types can affect the deletion test scores of explanation methods that focus on discrete attributions significantly. However, as shown in Table 1, it can be found that the principled explanation TRACE (i.e., the highest-performing explanation) has consistent scores across different reference types.

Patch Sizes. To explain why the reference values no longer have that great influence, we explore how pixel sizes affect the OOD issue. It is observed from the heatmap of a crane image in Figure 3(a) that the principled explanation becomes more noisy as the patch becomes smaller (as t increases from left to right), suggesting more severe OOD problem.

For an impartial and rigorous verification, we execute a randomized deletion test in Figure 4, where different curves represent different patch sizes. Zero reference (left figure in Figure 4) deletes features completely, at the cost of pronounced OOD issue. Since patches are deleted completely at random, when the same amount of features are deleted, the difference in prediction decays among patch sizes is caused almost completely by the different OOD levels. And note Figure 4 reveals that smaller patches lead to a faster decline in prediction quality, which suggests that *using larger patches effectively diminishes the OOD issue.*

In contrast, blurring reference (Figure 4 middle) preserves the distribution, at the cost of not deleting the feature sufficiently. Thus although the decay of model prediction is slower, it might be caused by the information of the lingering features that should have been deleted instead of a mild OOD issue. Interestingly, as patch sizes increase, the difference between zero and blurring references decreases (right figure in Figure 4). Recall that zero reference firmly deletes the features completely but compromises on the OOD issue, while blurring reference firmly solves the OOD issue but compromises on the feature deletion. Therefore, both desiderata can be attained when they behave the same – features are deleted, and the OOD issue is mitigated. This also explains the phenomenon in Table 1 where variances across different reference types are almost negligible.

MoRF vs. LeRF. MoRF defines important features as *those who affect the model prediction the most when deleted*. This, although seems symmetric to LeRF, is problematic. This is because the goal of MoRF is consistent with the OOD problem, where the deletion of a small number of features can bring down the model prediction significantly. We demonstrate TRACE-Le−Mo, TRACE-Le, and TRACE-Mo with different patch sizes in Figure 3(a) using heatmaps. Recall that smaller patches are likely to



(a) Heatmaps (b) Comparisons of -Le, -Mo, and Le−Mo

Figure 3. (a) The illustration of the converted heatmaps of the image of a "crane" from the validation set of ILSVRC2012. A ResNet-18 predicts it correctly with the confidence $\approx 100.00\%$. The smoothing factor $\alpha = 2$. Results of TRACE-SA-Le−Mo (top), TRACE-SA-Le (middle), and TRACE-SA-Mo (bottom) are presented. They are implemented w.r.t. the probability. We set $t = 4 \times 4$ (left), $t = 7 \times 7$ (middle) and $t = 14 \times 14$ (right). Here t is the number of square patches of pixels for one image. (b) The comparison of the TRACE-SA-Le−Mo (red), TRACE-SA-Le (blue) and TRACE-SA-Mo (green), TRACE-Greedy-Le (yellow) and TRACE-Greedy-Mo (cyan) under the LeRF (solid) and MoRF (dashed) tests. All explanation methods are also included but plotted indistinguishably just for reference.

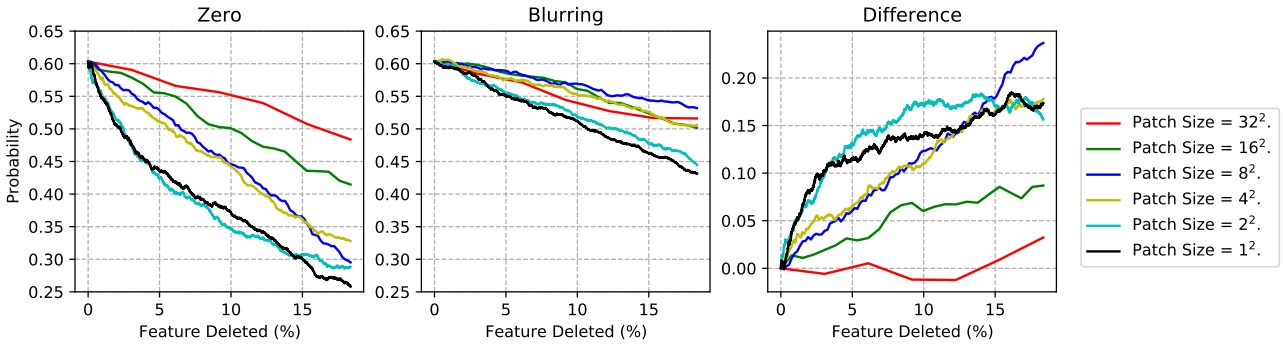


Figure 4. Random deletion tests with (a) zero (b) blurring reference; and (c) difference between (a)(b). 6 different patch sizes are tested.

exacerbate the OOD issue, it can be found that the extent to which the methods are affected by OOD is ranked as TRACE-Le−Mo<TRACE-Le≪TRACE-Mo. We further verify this by a cross-validation between the principled explanations and the associated tests in Figure 3(b), where TRACE-(SA/Greedy)-Le is tested with MoRF and TRACE-(SA/Greedy)-Mo is tested with LeRF. As deduced, TRACE-Mo performs extremely poorly in the LeRF test, indicating that features recognized as "unimportant" by TRACE-Mo (i.e. deleted in the end) are not really unimportant. Because when they are deleted first, the prediction drops fast (green, solid), too. On the other hand, those features that are deleted in the end by TRACE-Le (i.e. important) do cause the prediction to drop fast when they are deleted first (blur, dashed). This result impartially benchmarks that LeRF should be the preferred criterion, while MoRF should be considered with great care. As the combined version, TRACE-Le−Mo compromises slightly under each criterion but demonstrates perfect consistency. Therefore, *in practice, when choosing the criterion in deletion metrics, we suggest that LeRF−MoRF>LeRF≫MoRF.* As a complement, It is also interesting to notice that as t decreases (the patch size increases) to $t = 16$, TRACE-Mo results in semantically meaningful deletions. This is because simply deleting some meaningless features to break the distribution will not reduce the model's prediction significantly. That is, the OOD issue is mitigated when $t = 16$. This consistently supports the previous discussion of the patch size and the OOD issue.

TRACE-Greedy as the Baseline. The difference between TRACE-SA and TRACE-Greedy can be small according to Figure 3 (b). This illustrates that when associating with the MoRF and LeRF tests individually, the greedy scheme is an acceptable compromise to the meta-heuristic algorithms. As discussed above, both TRACE-(SA/Greedy)-Le can out-

perform all attribution methods in the LeRF−MoRF by a significant margin. Hence TRACE-Greedy can be used as a compromise between performance and efficiency. Furthermore, TRACE-Greedy-Le can also be used as an initialization of TRACE-SA to improve the speed of convergence. We provide an assessment of the trade-off between performance and efficiency in Appendix F.

6. Conclusions

In this paper, we study the deletion/insertion metric, the most popular metric for evaluating attribution methods. We propose a framework TRACE that assesses the deletion metric in an unbiased manner. It solves for the optimal deletion trajectories that approach the theoretical global minimum closely. In doing so, TRACE not only emerges as the principled explanation for the deletion metric, but also provides a standardized lens to inspect and benchmark all kinds of variants of deletion metrics.

Benchmarking the Deletion Metric. Our rigorous study offers several insights into the effective application of deletion metrics: (i) The image features should be deleted as superpixel patches instead of pixels. (ii) While MoRF and LeRF tests seem symmetric, the comparison between TRACE-Mo and TRACE-Le reveals that LeRF is preferred over MoRF. Besides, LeRF−MoRF retains both sides to characterize important features. (iii) It is verified that, unlike pixel-wise deletion, the reference values' influence can be negligible for superpixel deletions. (iv) We also emphasize that using probabilities and logits yields distinct evaluation results, and thus the goal of the test should be explicit.

Intrinsic Explanation for Feature Deletion. Deletion metrics are popular because they are intuitive and do not require prior knowledge – a feature is important when deleting it affects the prediction significantly, and is not important when it does not. However, finding such rankings of features has been infeasible because of various difficulties. Previous works compromise with other perturbation-based methods and only used this as a metric instead, to measure how close their methods are to the optimal deletion trajectories. This provides evidence that such optimal deletion trajectories are desired as an explanation. TRACE resolves this issue obtains the ranking of features consistent with the deletion objective. Therefore, TRACE not only benchmarks the deletion metric but also serves as an explanation method that fulfills the needs in highlighting the feature importance regarding feature deletion/insertion. Such an explanation provides benefits in (a) *Faithfulness.* TRACE is directly connected to the prediction and thus faithful to it. It passes tests such as sanity checks easily. Other applications based on faithfulness such as revealing spurious features can also be achieved. (b) *Model Agnosticism.* Powered by combinatorial optimizations, TRACE does not require access to the parameters, weights, or gradients of the black-box models. This ensures a larger use of fields, especially in the era where many deep models are not fully accessed. (c) *Robustness to Attacks.* TRACE provides explanations in a non-differentiable manner. Hence explanation-targeted attacks such as explanation sneaking attack (i.e. attack the prediction while keeping the explanation invariant), explanation manipulations (i.e. attack the explanation while keeping the prediction invariant), etc. will not work on TRACE. (d) *Global View.* TRACE considers all features (i.e. optimized w.r.t. τ) globally. Highly ranked features of TRACE not only reduce the predictions significantly when being deleted but also preserve the predictions significantly when being preserved. (e) *Clear Interpretation.* Attribution maps are accused of being non-applicable since the specific meaning of attribution values remains unclear. A higher value only indicates that the feature is "somewhat important" to the model. The attribution values require further interpretations to be useful. This ambiguity hinders the use cases of feature attributions. Differently, TRACE is clearly defined and connected to the deletion task. And the explanations can be easily interpreted. For example, in medical imaging analysis, if TRACE highlights an area of the figure, it directly indicates that the prediction will change as long as this area is masked or will not change as long as this area is preserved. This is valuable in model debugging and clinical practices.

Implications on Evaluating XAI Methods. By the optimality nature, it is suggested that when the deletion metric is utilized, TRACE is **the one**. This phenomenon is a warning that we should rethink how we develop and evaluate explanation methods. Since every time a metric is employed, there is a potential principled explanation for that metric that achieves the optimum. And the question remains, "is that the desired explanation?"

As we conclude, this work also leaves many interesting topics. For example, the deletion trajectories are closely related to path-based attribution methods. For example, SHAP takes the average of the incremental values of features overs $d!$ trajectories as attribution scores (Lundberg & Lee, 2017), while TRACE considers the optimal deletion/insertion that minimizes the global AUCs. These are two different perspectives of feature deletion that might result in distinct concepts, even with the same deletion trajectories. The exploration of the differences between path-based methods and the principles of deletion metrics may reveal deeper understanding of feature importance.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

This work was partially supported by NSF IIS #1955890 and IIS #2146091.

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., and Hinton, G. E. Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems*, 34: 4699–4711, 2021a.

Agarwal, S., IQBAL, O., Buridi, S. A., Manjusha, M., and Das, A. Reinforcement explanation learning. In *eXplainable AI approaches for debugging and diagnosis.*, 2021b.

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

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

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

Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., and Samek, W. Layer-wise relevance propagation for neural networks with local renormalization layers. In *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*, pp. 63–71. Springer, 2016.

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., and Su, J. K. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

Chen, H., Lundberg, S., and Lee, S.-I. Explaining models by propagating shapley values of local components. *Explainable AI in Healthcare and Medicine: Building a Culture of Transparency and Accountability*, pp. 261–270, 2021.

Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. L-shapley and c-shapley: Efficient model interpretation for structured data. In *International Conference on Learning Representations*, 2018.

Covert, I. C., Lundberg, S., and Lee, S.-I. Explaining by removing: A unified framework for model explanation. *The Journal of Machine Learning Research*, 22(1):9477–9566, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and 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.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437, 2017.

Geng, X., Chen, Z., Yang, W., Shi, D., and Zhao, K. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing*, 11(4):3680–3689, 2011.

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

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

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Jacovi, A. and Goldberg, Y. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4198–4205, 2020.

Kaur, H., Nori, H., Jenkins, S., Caruana, R., Wallach, H., and Wortman Vaughan, J. 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.

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

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

Krishna, S., Han, T., Gu, A., Pombra, J., Jabbari, S., Wu, S., and Lakkaraju, H. The disagreement problem in explainable machine learning: A practitioner's perspective. *arXiv preprint arXiv:2202.01602*, 2022.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

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

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

Li, L., Wang, B., Verma, M., Nakashima, Y., Kawasaki, R., and Nagahara, H. Scouter: Slot attention-based classifier for explainable image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1046–1055, 2021.

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

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Magnus, W., Karrass, A., and Solitar, D. *Combinatorial group theory: Presentations of groups in terms of generators and relations*. Courier Corporation, 2004.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the 29th British Machine Vision Conference*, pp. 151, 2018.

Qi, Z., Khorram, S., and Li, F. Visualizing deep networks by optimizing with integrated gradients. In *CVPR Workshops*, volume 2, 2019.

Ras, G., Xie, N., Van Gerven, M., and Doran, D. Explainable deep learning: A field guide for the uninitiated. *Journal of Artificial Intelligence Research*, 73:329–396, 2022.

Ribeiro, M. T., Singh, S., and 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.

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

Rong, Y., Leemann, T., Borisov, V., Kasneci, G., and Kasneci, E. A consistent and efficient evaluation strategy for attribution methods. In *International Conference on Machine Learning*, pp. 18770–18795. PMLR, 2022.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 2016.

Schulz, K., Sixt, L., Tombari, F., and Landgraf, T. Restricting the flow: Information bottlenecks for attribution. In *International Conference on Learning Representations*, 2019.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and 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.

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

Shi, Y., Wang, S., and Han, Y. Curls & whey: Boosting black-box adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6519–6527, 2019.

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

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015.

Simonyan, K., Vedaldi, A., and 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.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Sturmfels, P., Lundberg, S., and Lee, S.-I. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020.

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

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.

Tomsett, R., Harborne, D., Chakraborty, S., Gurram, P., and Preece, A. Sanity checks for saliency metrics. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6021–6029, 2020.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

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

Wang, Y. and Wang, X. "why not other classes?": Towards class-contrastive back-propagation explanations. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9085–9097, 2022a.

Wang, Y. and Wang, X. A unified study of machine learning explanation evaluation metrics. *arXiv preprint arXiv:2203.14265*, 2022b.

Wang, Y. and Wang, X. On the effect of key factors in spurious correlation: A theoretical perspective. In *International Conference on Artificial Intelligence and Statistics*, pp. 3745–3753. PMLR, 2024.

Wolberg, William, M. O. S. N. and Street, W. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository, 1995. DOI: https://doi.org/10.24432/C5DW2B.

Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., and Yan, S. 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.

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

Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., and Sclaroff, S. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126 (10):1084–1102, 2018.

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

Zhou, J., Gandomi, A. H., Chen, F., and Holzinger, A. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5):593, 2021.

Zhou, Y. and Shah, J. The solvability of interpretability evaluation metrics. In *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 2354–2370, 2023.

A. Extensive Related Work

Attribution Methods. 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 a top-down fashion. Saliency (Simonyan et al., 2014) makes use of the gradient of input as the attribution values. Guided back-propagation (Springenberg et al., 2014) modifies the behavior of ReLU layers in backpropagations. LRP (Bach et al., 2015) and DeepLift (Shrikumar et al., 2017) change the back-propagation rule to propagate attribution values layer-wise. Input $\times$ Gradient (Shrikumar et al., 2017) 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, on the other hand, usually generate explanations by modifying the input data and observing the change in the output. 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 images with gray squares. SHAP (Lundberg & Lee, 2017) utilizes the approximated Shapley values (Shapley, 1953) as attribution values. RISE (Petsiuk et al., 2018) defines attribution values based on many randomly sampled masks. IBA (Schulz et al., 2019) generates explanations via per-sample information bottleneck. I-GOS (Qi et al., 2019; Khorram et al., 2021) optimize small masks to maximally decrease prediction scores. Fong & Vedaldi (2017) similarly optimize a relaxed continuous mask with L_1 regularization so that the predictions of the masked inputs are minimized. Agarwal et al. (2021b) formulate attribution generation as a Markov Decision Process and use reinforcement learning to solve it. Generally, perturbation methods are model-agnostic, meaning that they do not require any information about 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 values (Chen et al., 2019; Agarwal et al., 2021a; Wang & Wang, 2021; Li et al., 2021), where instead of explaining an existing black-box model, they propose entire new models that generate explanations and predictions at the same time.

Insertion/Deletion Metrics. As the most popular genre of evaluation metrics, insertion/deletion metrics are also called "faithfulness" in other works. They have a lot of variants, which, although have different names, all share the same essence. Samek et al. (2016) propose pixel flipping, where pixels are gradually replaced with zero values. This vanilla form is equivalent to most applications of such kinds of metrics (with names like ablations, maskings, etc.). Petsiuk et al. (2018) introduce the insertion metric in addition to the deletion one, where features are gradually inserted instead of deleted. Tomsett et al. (2020) carry out a sanity check and explore the AUC scores of multiple attribution explanation methods such as SHAP, Input $\times$ Gradient. Hooker et al. (2019) argue that deleting features from the input tends to break the original distribution. They propose ROAR to alleviate this issue. However, it requires training black-box models from scratch every time the number of deleted features changes, which is computationally expensive and hardly applied. To alleviate the out-of-distribution issue, there are other workarounds such as replacing feature values with reference values from mean/median/blurring instead of zeros (Wang & Wang, 2022a). Rong et al. (2022) propose to use the weighted summation of the 8 surrounding pixels of the deleted one as the reference values, which is actually an intermediate stage between mean and blurring. Recently, Zhou & Shah (2023) propose to use beam search to explore the limit of deletion metrics. However, as a step-by-step search, it is a relaxed greedy scheme that does not take the entire deletion process into consideration. Therefore, it suffers from all the drawbacks of the standard greedy search that simply searches for the next feature to insert/delete to maximize/minimize the prediction in the next step. Also, for image data, it has been shown that deleting tiles of square pixels can also alleviate such issue (Schulz et al., 2019; Agarwal et al., 2021b). Schulz et al. (2019) also argue that removing features either from the top-down manner or the down-top manner individually is insufficient. They propose to use the difference between them as the measurement.

B. Proofs & Analysis

B.1. Details of the Formulation

In this section, we elaborate on the definitions of the equivalence relation R defined over $\mathbb{R}^d$. The relation R is defined so that every attribution map ψ can be identified by its equivalence class $[\psi]$. However, note that when $\exists i, j \in \{1, \cdots, d\}, i \neq j$ such that $\psi_i = \psi_j$, the permutation $\sigma(\psi)$ is not well-defined. Therefore, we define that $\sigma(\psi)[i] < \sigma(\psi)[i+1]$ when $\psi_{\sigma[i]} = \psi_{\sigma[i+1]}$, i.e., when the attributions are equal, features with smaller indices are put ahead. Hence the relation R is defined as follows.

Definition B.1. We say the relation R holds for $\psi^1, \psi^2 \in \mathbb{R}^d$ if they have they have the same permutation of features, i.e.,

$\psi^1 R \psi^2 \Leftrightarrow \sigma(\psi^1) = \sigma(\psi^2)$.

Next, we prove the following theorem that the relation R is an equivalence relation.

Theorem B.2. *R is an equivalence relation.*

Proof: (i) $\forall \psi \in \mathbb{R}^d$, since an attribution map defines a unique permutation of features, we have $\sigma(\psi) = \sigma(\psi)$, and hence the reflexivity is proved by $\psi R \psi$. (ii) $\forall \psi^1, \psi^2 \in \mathbb{R}^d$, $\psi^1 R \psi^2 \Leftrightarrow \sigma(\psi^1) = \sigma(\psi^2) \Leftrightarrow \sigma(\psi^2) = \sigma(\psi^1) \Leftrightarrow \psi_2 R \psi_1$. Thus R satisfies symmetry. (iii) For transitivity, $\forall \psi^1, \psi^2, \psi^3 \in \mathbb{R}^d$, if $\psi^1 R \psi^2$ and $\psi^2 R \psi^3$, then $\sigma(\psi^1) = \sigma(\psi^2) = \sigma(\psi^3)$. Therefore, $\psi^1 R \psi^3$ and the transitivity is proved.□

Now that R is an equivalence relation, we can effectively focus on the quotient set $\mathbb{R}^d/R = \{[\psi] : \psi \in \mathbb{R}^d\}$ that consists of all the equivalence classes instead of the original Euclidean space $\mathbb{R}^d$. Since the set $\mathbb{R}^d/R$ of an equivalence class is not intuitive to deal with, we map the equivalence class $[\psi]$ to the permutation $\sigma(\psi)$ in a 1-to-1 manner, we have $\mathbb{R}^d/R \cong \mathcal{S}_d$. Here $\mathcal{S}_d$ denotes the set of all permutations of $\{1, \cdots, d\}$.

Theorem B.3. *$\mathbb{R}^d/R \cong \mathcal{S}_d$ with the bijection $[\psi] \mapsto \sigma(\psi)$.*

Proof: On the one hand, $\forall \tau \in \mathcal{S}_d$, let $\psi \in \mathbb{R}^d$ s.t. $\psi_{\tau[i]} = i$, then $\sigma(\psi) = \tau$. And thus $\exists [\psi] \in \mathbb{R}^d/R$ s.t. $[\psi] \mapsto \tau$. Hence it is surjective. On the other hand, $\forall \psi^1, \psi^2 \in \mathbb{R}$ s.t. $[\psi^1] \neq [\psi^2]$, by the definition of R, $\sigma(\psi^1) \neq \sigma(\psi^2)$. Hence $[\psi] \mapsto \sigma(\psi)$ is an injective. Therefore, $[\psi] \mapsto \mathcal{S}_d$ is bijective.□

B.2. Proof of Theorem 3.1

Theorem 3.1. The insertion metric is equivalent to the deletion metric up to AUCs with MoRF/LeRF.

Proof: We show that deletion-MoRF is equivalent to insertion-LeRF. Deleting the most important k features results in $\boldsymbol{x}_{\setminus \tau[k:]}$. On the other hand, inserting the least important k features results in $\boldsymbol{x}_{\tau[:k]}$. Taking the summation of both of them, we have the equivalent AUCs:

$$\sum_{k=0}^{d} f(\boldsymbol{x}_{\setminus \tau[k:]}) = \sum_{k=0}^{d} f(\boldsymbol{x}_{\tau[:d-k]}) = \sum_{k=0}^{d} f(\boldsymbol{x}_{\tau[:k]}) \tag{4}$$

Thus it proves that deletion-MoRF is equivalent to insertion-LeRF. Similarly, it can be easily shown that deletion-LeRF is equivalent to insertion-MoRF. □

B.3. Proof of Theorem 4.1

Theorem 4.1. The optimization problem TRACE-Mo ($\{\min_\tau \sum_{k=0}^{d} f(\boldsymbol{x}_{\tau[:k]})\}$) is NP-hard.

Proof: In TSP, a salesman traverses all t cities, and the minimal cost is sought. It is defined by a cost matrix $\Delta = [\delta_{ij}]_{t \times t}$ where δ_{ij} is the cost going from city i to j. Given a trajectory τ, the cost function is defined as $f_{tsp}(\tau) = \sum_{i=1}^{t} \delta_{\tau[i]\tau[i+1]}$, where we extend $\tau[t+1] := \tau[1]$.

Note that $f(\boldsymbol{x}_{\tau[:k]}) = f(\texttt{ref})$ is constant w.r.t. τ when $k = 0$, it sufficies to minimize $\sum_{k=1}^{t} f(\boldsymbol{x}_{\tau[:k]})$. Here we show this by demonstrating the corresponding decision problem "Given a cost $f^* \in \mathbb{R}$, is there a trajectory τ s.t. $\sum_{k=1}^{t} f(\boldsymbol{x}_{\tau[:k]}) \leq f^*$."

Now assume that there's a polynomial time algorithm for TRACE. Note that $f(\boldsymbol{x})$ is a black-box neural network and thereby can be any continuous function, and also $\forall i \neq j$ we have $\boldsymbol{x}_{\tau[:i]} \neq \boldsymbol{x}_{\tau[:j]}$, therefore, we define for any trajectory τ of length t and $\forall i \in \mathbb{N}, 1 \leq i \leq t$,

$$f(\boldsymbol{x}_{\tau[:i]}) := \delta_{\tau[i]\tau[i+1]} \tag{5}$$

In this way for any trajectory τ, we have

$$f_{tsp}(\tau) = \sum_{i=1}^{t-1} \delta_{\tau[i]\tau[i+1]} = \sum_{i=1}^{t} f(\boldsymbol{x}_{\tau[:i]}) \tag{6}$$

Therefore, this polynomial time algorithm also serves as an algorithm for TSP, a contradiction. □

B.4. Neighbor Sets Analysis

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

$$\boldsymbol{s} = \begin{pmatrix} 1 & 2 & \cdots & d \\ \tau^{-1}[1] & \tau^{-1}[2] & \cdots & \tau^{-1}[d] \end{pmatrix} = \boldsymbol{s}(\tau), \tag{7}$$

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 the same number of neighbors. For the symmetric group S_t, such a graph is perfectly modeled by Cayley's graph (Magnus et al., 2004). 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 $E = \{[s_1, s_2] | \exists g \in S,\ g s_1 = s_2\}$, 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 $G(\{1, \cdots, t\}, S)$ is known as the transposition graph, where the vertices are $\{1, \cdots, t\}$, and the edges are the transpositions in S. Then

Proposition B.4. *(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 B.5. $\mathrm{diam}(\mathrm{Cay}(S_t, S_{complete})) \leq t - 1$.

Proof: Given any two permutations of length t: $\forall \sigma_1, \sigma_2 \in S_t$, $\sigma_1 \neq \sigma_2$, we have where $t - 1$ transpositions are applied to σ_1. Note that $\forall i \in \mathbb{N}, i < t$, if $i = \sigma_1^{-1}[\sigma_2[i]]$, then the operation can be skipped. Therefore, there is always a path of length at most $t - 1$ connecting any two vertices in $\mathrm{Cay}(S_t, S_{complete})$.□

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

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 pre-trained ResNet-18 provided by `torchvision` is plotted. The results are shown in Figure 5. It can be found that the complete graph outperforms other neighbor sets.

C. Visualizing Trajectories as Heatmaps

In this section we visualize trajectory τ as heatmaps using $\psi = \pi(\tau) = ((\tau^{-1}/d)^{\alpha}) \in [0, 1]^d$. Since in practice, we implement `TRACE` in $t = 7 \times 7$ superpixel patches as discussed, we use the common practice in XAI, the bilinear

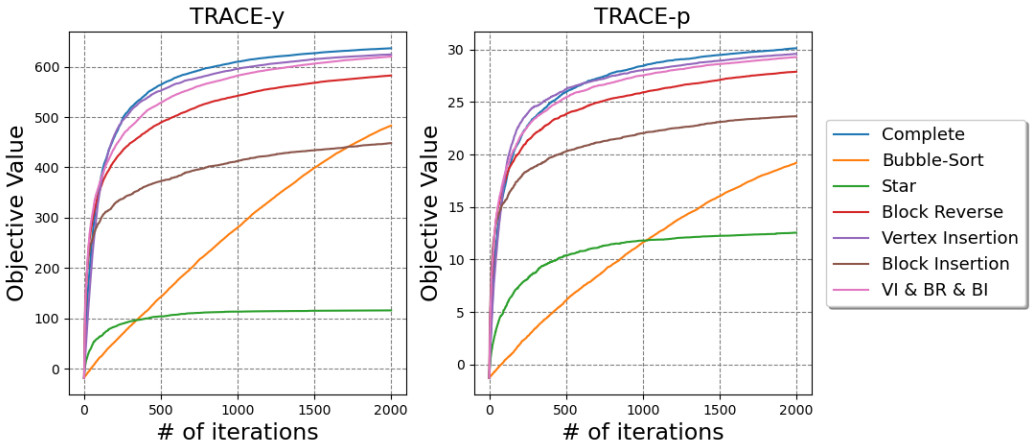


Figure 5. The comparison of different neighbor sets.

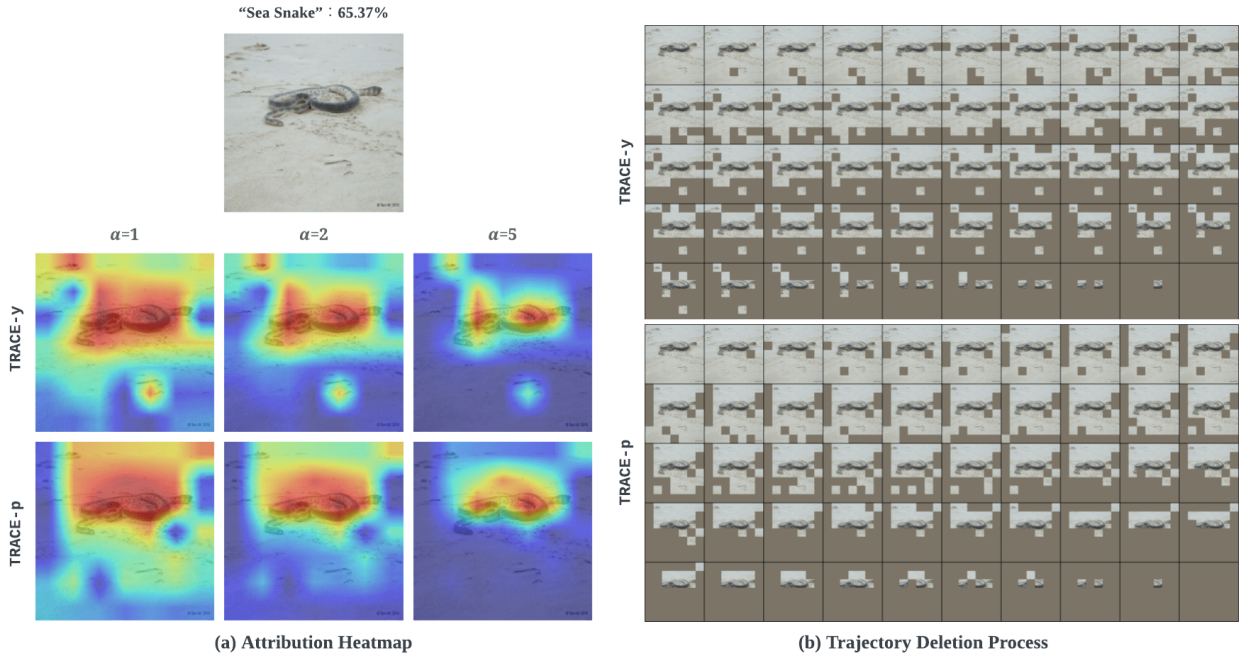


Figure 6. Visualizations of the optimized results of TRACE-y and TRACE-p on the "sea snake" image of ILSVRC2012 validation set. (a) The converted heatmaps ψ with different smoothing factor α. (b) The deletion process is based on the trajectory τ.

upsampling, to interpolate the ranking of features back to the input space. Different choices of α are compared in Figure 6 (a). It can be found that α is independent from the deletion process shown in Figure 6 (b). Instead, it controls the visually highlighted area, serving similar purposes as colormaps in visualizations. In Figure 7, we demonstrate the deviation between existing attribution methods and the principled explanations of the deletion metric provided by TRACE.

As a convention test for attribution explanations, we also perform the sanity check for the converted heatmaps of TRACE. The results are shown in Figure 8, where the DNNs are randomized in either the independent or the cascading fashion. It is observed that TRACE passes the sanity check for explanation methods.

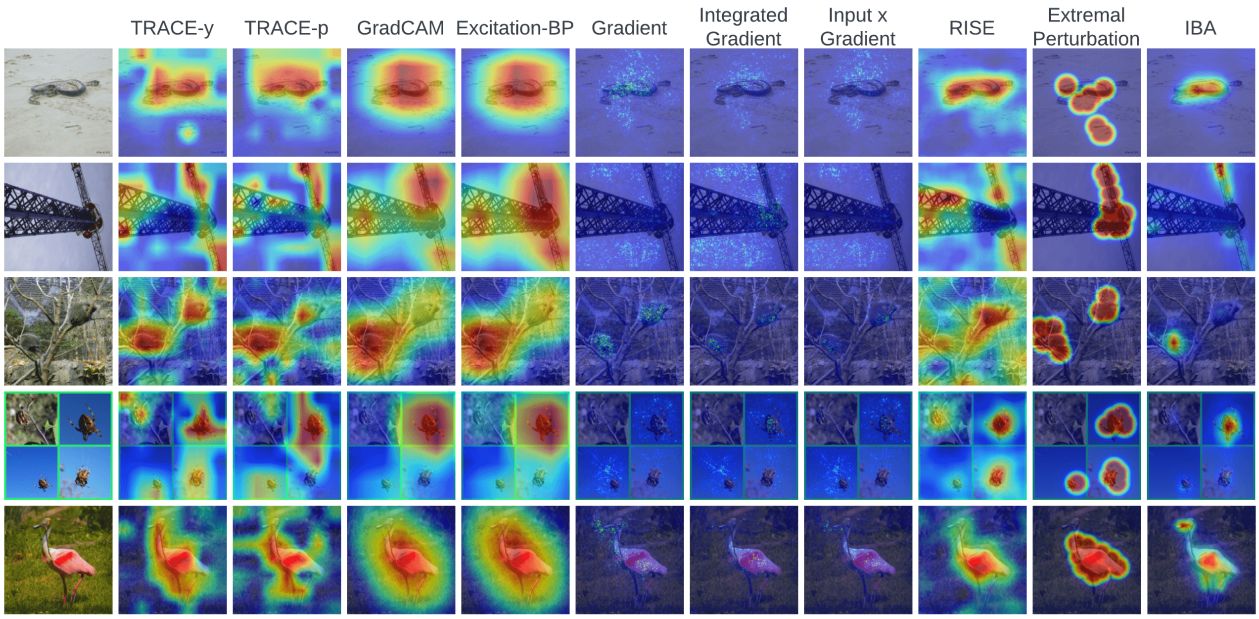


Figure 7. Visualizations of TRACE and popular attribution methods on images from ILSVRC2012.

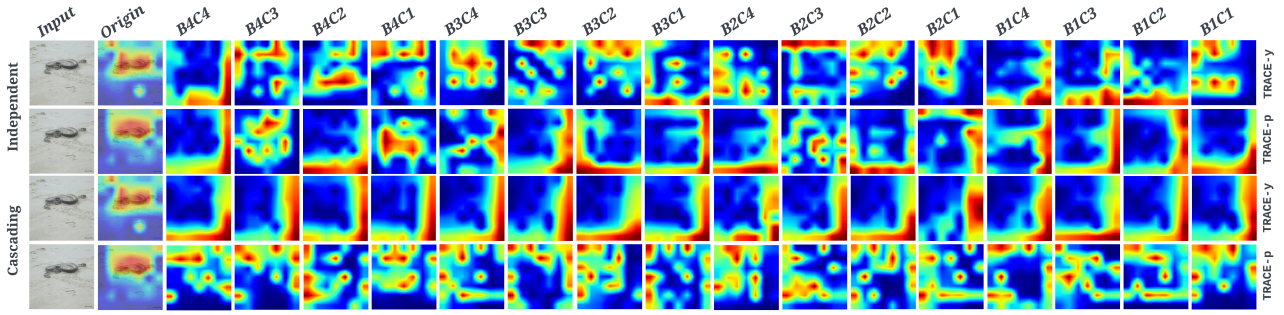


Figure 8. Sanity check using cascading randomization for TRACE. Convolutional layers of pre-trained ResNet-18 are randomized in the independent (upper) and cascading (lower) manners. In the independent randomization, other layers are kept at the pre-trained values. And in the cascading randomization, layers are progressively randomized from left to right (top-down). Here "BaCb" means the b-th convolutional layer in the a-th block.

D. Psuedo-Code for TRACE

Algorithm 1 Simulated Annealing for TRACE

Require: black box f, input $\boldsymbol{x}$, number of patches t, max iteration K, neighbor set function `neighbor()`, initial temperature T_0, cooling rate η

$T \leftarrow T_0$
$\tau_0 \leftarrow$ `RandomInitialTrajectory`
$auc_0 \leftarrow \sum_{k=1}^{t} \left(f(\boldsymbol{x}_{\setminus \tau_0[:k]}) - f(\boldsymbol{x}_{\tau_0[k:]})\right)$
$k \leftarrow 0$
while $k < K$ **do**
  $\tau_1 \leftarrow$ `RandomChoice(neightbor(`τ_0`))`
  $auc_1 \leftarrow \sum_{k=1}^{t} \left(f(\boldsymbol{x}_{\setminus \tau_1[:k]}) - f(\boldsymbol{x}_{\tau_1[k:]})\right)$
  $\delta = auc_1 - auc_0$
  if $\delta > 0$ **then**
    $\tau_0 \leftarrow \tau_1$
    $auc_0 \leftarrow auc_1$
  else
    $r \leftarrow$ `RandomUniform(`$0, 1$`)`
    if $r < \exp(\delta/T)$ **then**
      $\tau_0 \leftarrow \tau_1$
      $auc_0 \leftarrow auc_1$
    end if
  end if
  $k \leftarrow k + 1$
  $T \leftarrow \eta T$
end while
return τ_0

Algorithm 2 Greedy Scheme for TRACE

Require: black box f, input $\boldsymbol{x}$, number of patches t
$k \leftarrow 0$
$\tau \leftarrow$ `EmptyList`
$\delta \leftarrow$ `[1,...,t]`
while $k < t$ **do**
  $F \leftarrow$ `EmptyList`
  $N \leftarrow$ `EmptyList`
  $n \leftarrow 1$
  while $n <$ `len(`δ`)` **do**
    $\epsilon \leftarrow \tau \cup \{\delta[n]\}$
    $F \leftarrow F \cup \{f(\boldsymbol{x}_{\setminus \epsilon}\}$
    $N \leftarrow N \cup \{n\}$
    $n \leftarrow n + 1$
  end while
  $i \leftarrow$ `argmin(`F`)`
  $\tau \leftarrow \tau \cup \{\delta[N[i]]\}$
  $\delta \leftarrow \delta \setminus \{\tau[-1]\}$
  $k \leftarrow k + 1$
end while
return `flip(`τ`)`

E. Comparisons among Different Algorithms on TRACE

As a supplementary, we test TRACE with several other popular algorithms for combinatorial optimizations. We include local search algorithms such as Hill Climbing, Tabu Search (GS) (Glover, 1986), and global search algorithms such as Genetic Algorithm (GA) (Holland, 1992). Note that these algorithms can have different complexity per iteration. Therefore, we compare the average optimization process within the same amount of time. As the benchmark, SA takes ~ 200 seconds for 5000 iterations when $t = 49$. Hence here we compare the results of these algorithms within 200 seconds, no matter how many iterations there are.

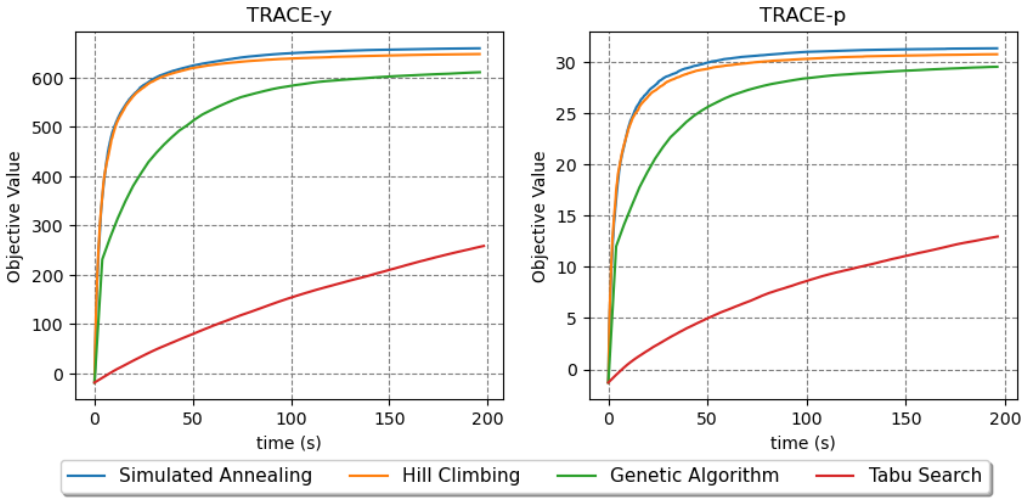


Figure 9. The comparison of different algorithms on solving TRACE.

The results are shown in Figure 9, where Simulated Annealing outperforms other algorithms in the experiments. It should be noticed that one of the most important factors in TRACE is that the objective function is more expensive to evaluate than common combinatorial optimization problems like TSP. Thereby, an algorithm that fits TRACE well should require fewer evaluation times. For instance, Tabu Search requires evaluating all neighbors to update the tabu list, which means the complete graph cannot be applied as the neighbor size is $\frac{t(t-1)}{2} = 49 \times 48/2 = 1176$. The bubble-sort graph is applied instead, which is the reason why it is the slowest. This also corresponds to the results of the neighbor comparison experiments shown in Figure 5. Another interesting result is that Hill Climbing, which is not a meta-heuristic algorithm but a simple heuristic method instead, has the second-best result. This may suggest TRACE do not have many local optima in the feasible set S_t.

F. The Trade-Off between Performance and Efficiency

The trade-off between the optimality of TRACE and the efficiency is inevitable given the nature of combinatorial optimization algorithms. And the running time is affected by the number of iterations.

When efficiency is preferred for explanations, TRACE-Greedy-Le or TRACE-SA-Le−Mo with a smaller number of iterations are preferred for explanations. When the ground truth is required for benchmarking of the metrics, we push the optimization process of TRACEto the limit by applying a larger number of iterations. We include the running time and the LeRF−MoRF deletion scores of all explanation methods and TRACE-Greedy, TRACE-SA. The results are shown in Table 2. Here Extremal Perturbation and RISE are implemented in the default settings suggested in the original papers. It can be found that (a) All TRACE variants outperform explanation methods in the deletion scores by a significant margin. (b) Increasing the number of iterations of SA consistently gives rise to the deletion score. (c) The running time of TRACE is very comparable and even outperforms popular perturbation-based explanation methods. With the trade-off on the efficiency, TRACE pushes the deletion score to the limit gradually. (d) Back-propagation-based explanations do have the best efficiency.

G. Supplementary results.

Visualizations and Qualitative Inspections of TRACE. Instead of attribution-based visualizations in the main manuscript, TRACE is better visualized as the deletion process. Because it is proposed that way. Here we demonstrate this in Figures 13 and 14, where the deletions are w.r.t. the probability and the logit respectively. The deletion process is visualized in the LeRF criterion. Hence the lastly deleted features are those that preserve the model's performance (or even significantly increase it as shown previously) when preserved. We focus on the deletion process following the probability in Figure 14.

In the first figure, we can see that the top left corner has a small bump on the ground, which is preserved till the last few steps. This indicates that keeping this bump can greatly preserve/increase the predicted probability of the snake class, suggesting

Table 2. The comparison between the running time and the LeRF−MoRF

Methods	LeRF-MoRF Score	Time (s)
Gradient	11.52	∼0.005
GradCAM	16.24	∼0.005
Excitation Back-Propagation	15.38	∼0.014
Integrated Gradient	10.28	∼0.116
Input x Gradient	8.21	∼0.006
IBA	15.92	∼0.144
RISE	14.52	∼5.161
Extremal Perturbation	13.41	∼28.975
TRACE-Greedy-Le	**27.09**	∼0.656
TRACE-SA-Le−Mo ($K = 100$)	**28.42**	∼4.079
TRACE-SA-Le−Mo ($K = 500$)	**29.32**	∼20.105
TRACE-SA-Le−Mo ($K = 1000$)	**29.83**	∼40.245
TRACE-SA-Le−Mo ($K = 5000$)	**31.69**	∼200.635

that the model might recognize that feature as a part of snakes by mistake. Similarly, in the figure of row 3 column 2, the wings are the features that are preserved till the end, suggesting the importance of wings in recognizing birds. Also, it can be found in the figure of row 3 column 4 that the arms contribute greatly for the prediction of the crane.

Exhaustive Results w.r.t. Logits. Due to the space limit, we only present the exhaustive results w.r.t. the probability in Table 1. Here we include the results of the same experiment, where we focus on the logits instead. The results are shown in Table 3, and are consistent with the probability experiments.

Extensive Comparisons of Attribution Methods. In order to compare attribution methods comprehensively, we include more explanation methods such as guided GradCAM (Selvaraju et al., 2017), LRP (Bach et al., 2015), guided back-propagation (Springenberg et al., 2014), deconvolution (Zeiler & Fergus, 2014), GradientSHAP (Lundberg & Lee, 2017). The results are visualized in Figure 11. 13 methods apart from TRACE have been included in the test.

Tabular Data. Based on perturbing input features, the optimality of TRACE is independent of the data modality, making it readily applicable to other domains. Here we briefly demonstrate how TRACE works for tabular datasets. We train an MLP over the breast cancer dataset (Wolberg & Street, 1995), and test the explanation methods and TRACE using the deletion metric. Note that methods like GradCAM, Excitation-BP, IBA, etc. are specifically designed for image understanding. Therefore, here we only include explanation methods that are universal for all data types, including Gradient, Input $\times$ Gradient (IxG), Integrated Gradient (IG), Layerwise Relevance Propagation (LRP), and Gradient SHAP. The deletion results are shown in Figure 12. It can be found that TRACE (red) explores the principled deletion trajectory of features that push the limit of the deletion metric to a new level. Without the red curves, one can hardly imagine the seemingly good AUCs are still far from the optimum.

H. Identifying Spurious Correlations

Spurious correlation refers to the scenario where irrelevant (spurious) features are correlated with the core features in the training distribution (Wang & Wang, 2024). As a result, a model trained on the dataset with spurious correlation may mistakenly rely on spurious features to make the prediction. By highlighting the areas that are most related to the prediction through deletions, TRACE is able to identify whether a model relies on spurious features. We take the waterbird dataset (Sagawa et al., 2019) as an example. In this synthetic binary classification dataset, the core feature is the bird type (waterbird vs. landbird). The bird images from CUB-200-2011 (Wah et al., 2011) are cropped and attached to different backgrounds (water vs. land). The backgrounds are from the Places dataset (Zhou et al., 2017). This leads to four groups: (waterbird, water), (waterbird, land), (landbird, water), (landbird, land). We compare two models: (a) The model is trained without the spurious correlation (four groups have the ratio $1:1:1:1$). (b) The model is trained with highly correlated features, where the ratio is $999:1:1:999$ instead. Therefore, model (b) will be more likely to rely on the spurious feature, i.e. the background, to make the prediction. Then we explore the deletion trajectory of TRACE for these two models. The results are presented in Figure 15. It can be found that TRACE is capable of detecting spurious correlations sensitively. This is because

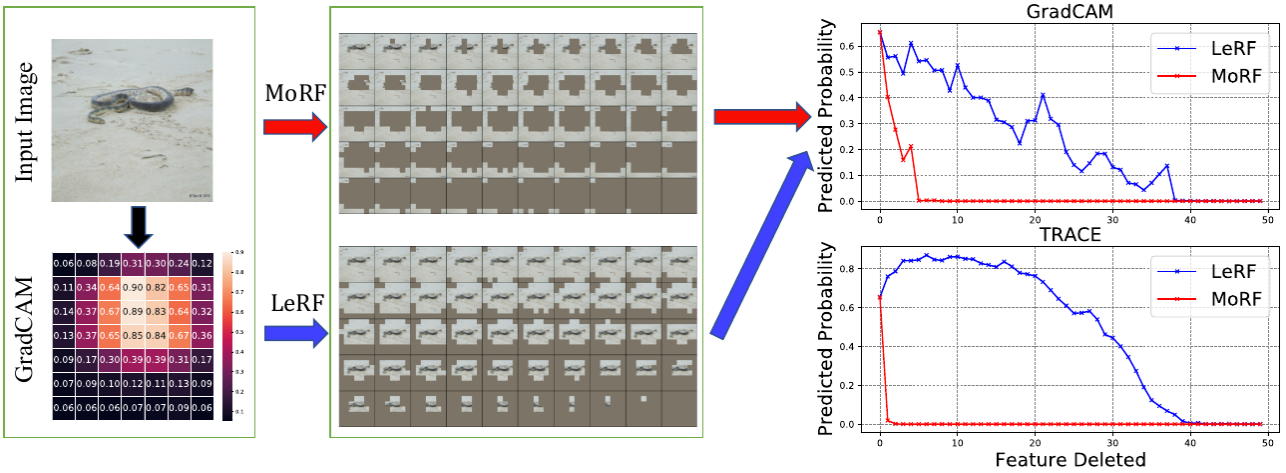


Figure 10. Demonstration of the deletion metrics under different criteria (MoRF or LeRF). The demonstration is based on ResNet-18 and GradCAM applied to the first image in ILSVRC2012. In the middle, MoRF (top) deletes the features with the highest attribution first, while LeRF (bottom) deletes the lowest feature first. On the top right, it shows the deletion metric results of MoRF (red curve) and LeRF (blue curve) for the single input using the probability as the indicators. On the bottom right, it shows the `TRACE` result for the same model and input.

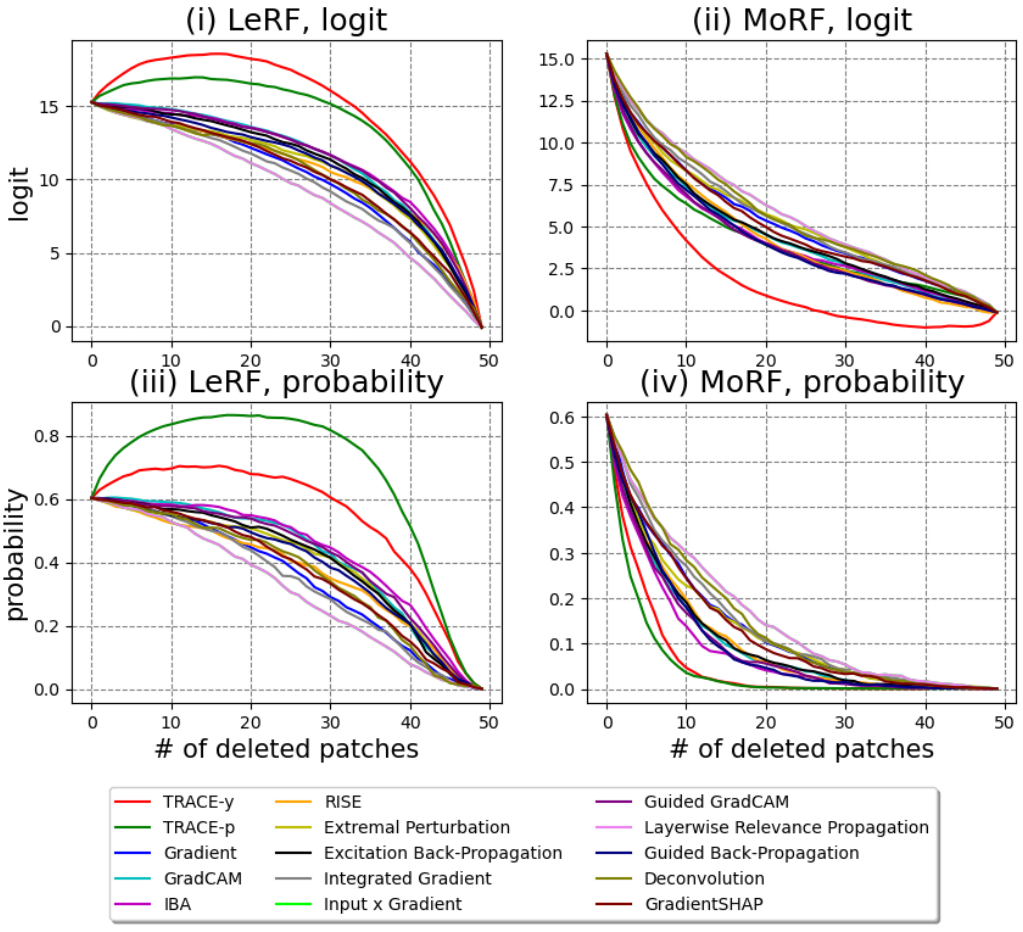


Figure 11. The comparison between the deletion tests of `TRACE` and more attribution methods

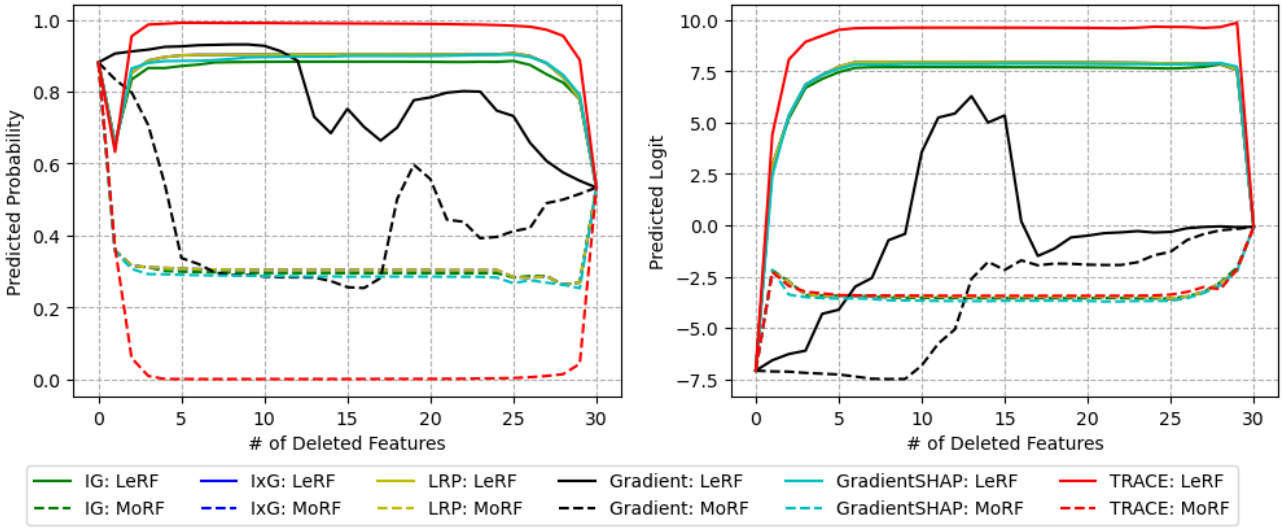


Figure 12. Deletion test results of the breast cancer dataset.

TRACE is directly related to the model predictions – If the model relies on the spurious feature to make the prediction, preserving those features will retain a high prediction confidence.

I. Transferability of the Optimal Trajecotires

Now that TRACE can be seen as a principled explanation of a model reflecting the influence of feature deletion, it can serve other uses. Previously, given $\boldsymbol{x}$, f_1, f_2 and an explanation method φ, the explanation method provides explanations $\phi_{f_1}(\boldsymbol{x}), \phi_{f_2}(\boldsymbol{x})$ for the two models, respectively. It can be observed that although the two explanations reflect the mechanism of the corresponding models, they do not provide an opportunity for the evaluation of the cross-model mechanism – e.g. the explanation $\phi_{f_1}(\boldsymbol{x})$ of model f_1 has very limited meaning for model f_2. Thus explanations are usually compared through $d(\varphi_{f_1}(\boldsymbol{x}), \varphi_{f_2}(\boldsymbol{x}))$ where $d(\cdot, \cdot)$ is usually a distance defined on $\mathbb{R}^d \times \mathbb{R}^d$. It is well-known such distance is limited due to the curse of dimensionality. Inspired by (Madry et al., 2018; Shi et al., 2019), we study the transferability of TRACE among different models. The correlation matrices are reported in Figure 16. All six matrices are the combinations of the two indicators: the logit and the probability, and the three references: zeros, means, and Gaussian blurs. By considering the transferability of the trajectory between the two models, we are able to quantify the difference between models in a more reasonable way.

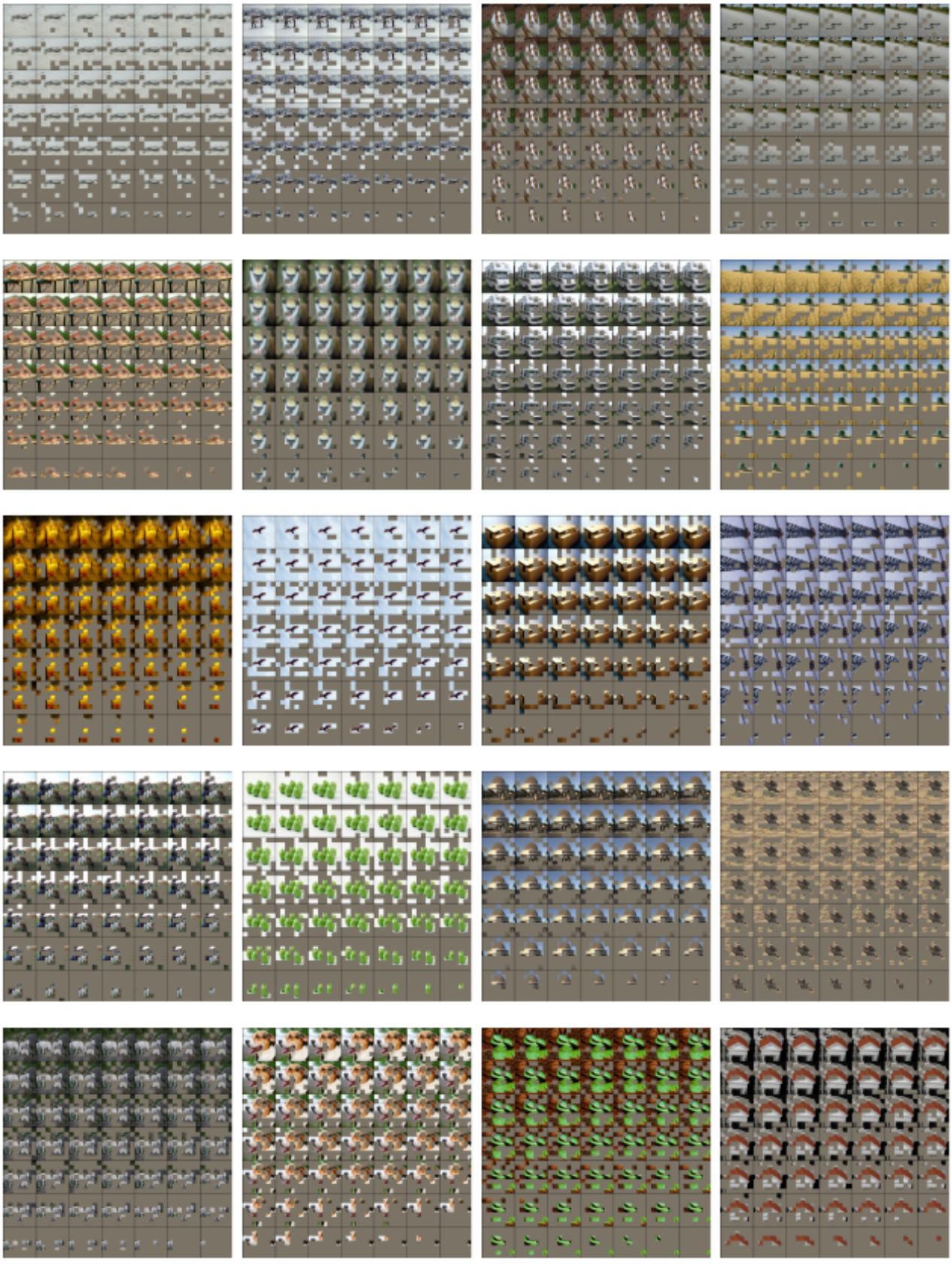


Figure 13. The deletion process of images from ILSVRC2012 on ResNet-18. Here all the trajectories are generated under `TRACE-y`. That is, the predicted logits are the measurement.

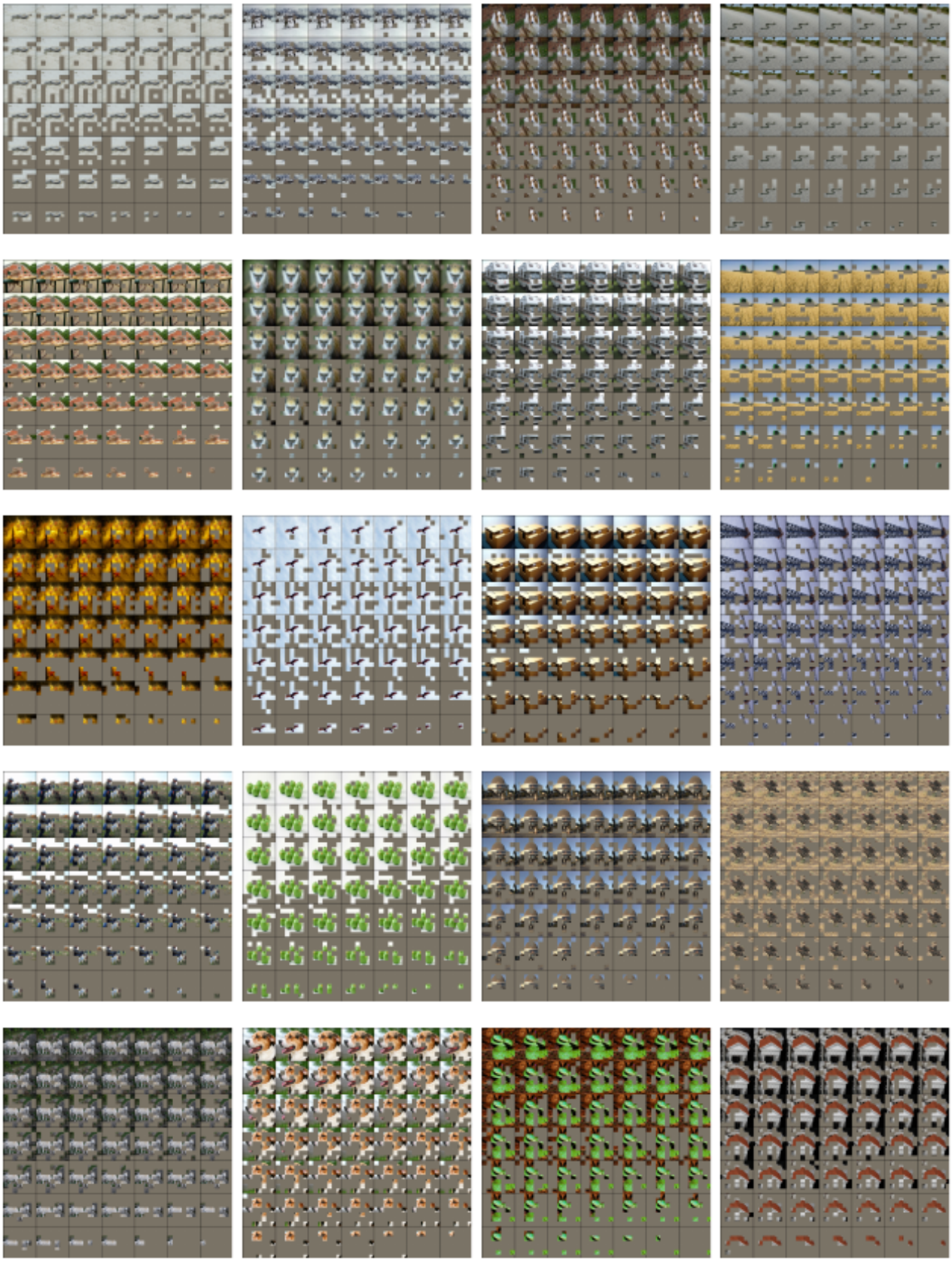
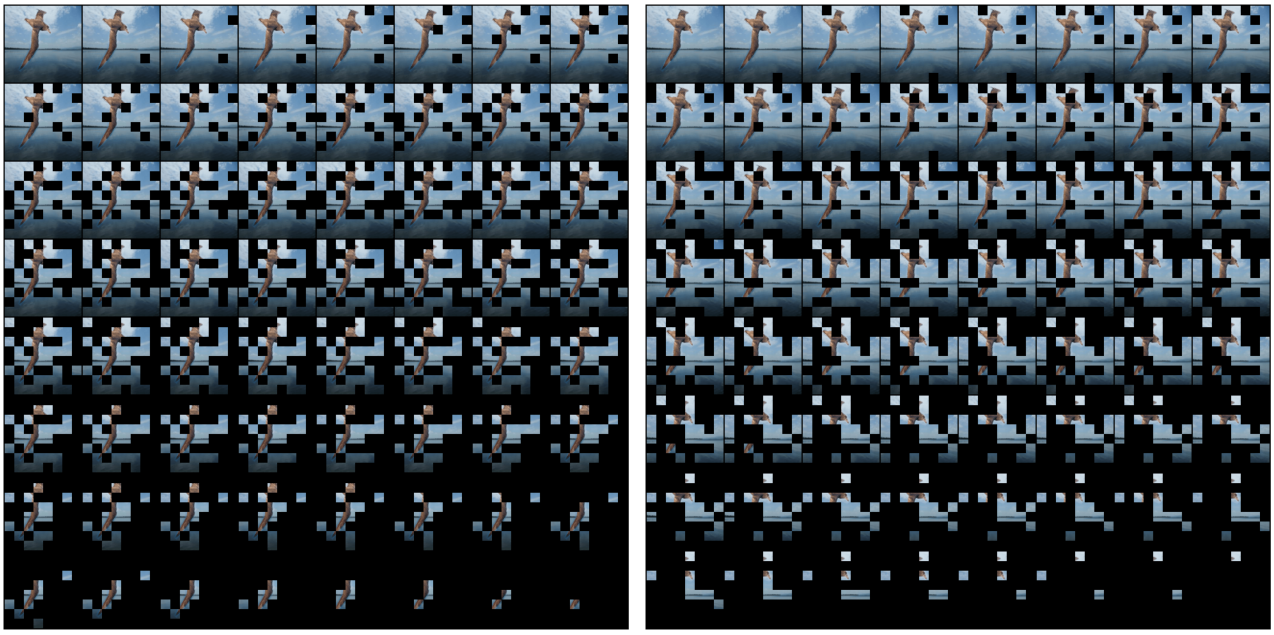
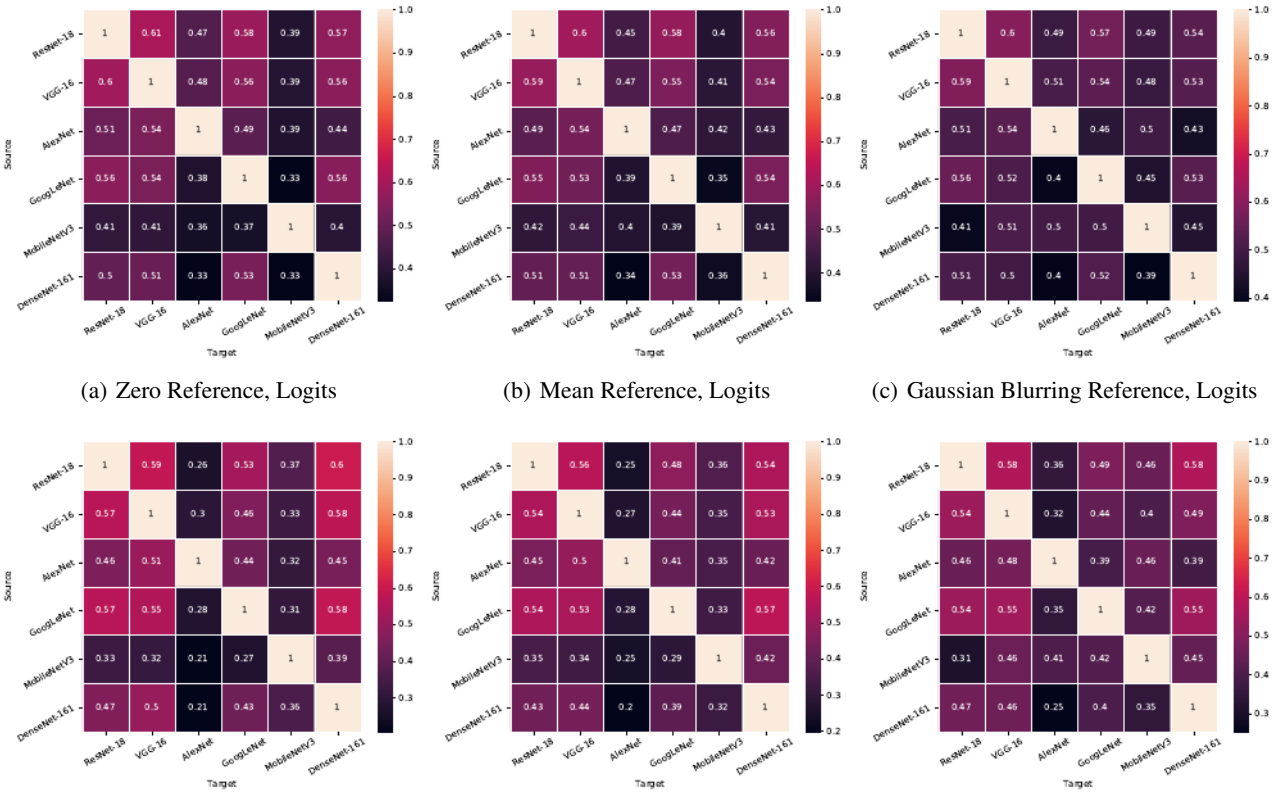


Figure 14. The deletion process of images from ILSVRC2012 on ResNet-18. Here all the trajectories are generated under `TRACE-p`. That is, the predicted probability is the measurement.



(a) Model Trained without Spurious Correlation (b) Model Trained with Spurious Correlation

Figure 15. The demonstration of the LeRF deletion process of TRACE-Le−Mo on a testing image of the waterbird dataset. The label is associated with the object (bird). In (a), the model is trained on the non-spurious training set, where the ratio of the group sizes is (waterbird, water):(waterbird, land):(landbird, water):(landbird,land) = $1:1:1:1$. On the contrary, in (b), the model is trained on the spurious correlated training set, where the ratio among the four groups is $999:1:1:999$. This suggests that model (b) may rely on the background to make the prediction instead of the object. Since the testing sample is waterbird + water, both models achieve correct predictions with confidence $> 1 - 10^{-4}$. However, as illustrated in the bottom row of the two deletion processes, the non-spuriously trained model (a) highlights birds' wings, while the spuriously trained model (b) highlights the coastline of the water background. This demonstrates the power of TRACE in detecting the spurious correlated features due to the direct connection between TRACE and model predictions.



(a) Zero Reference, Logits (b) Mean Reference, Logits (c) Gaussian Blurring Reference, Logits

(d) Zero Reference, Probability (e) Mean Reference, Probability (f) Gaussian Blurring Reference, Probability

Figure 16. The transferability of the trajectory among 6 different DNNs. In each subfigure, the y-axis is the list of source models, which the trajectories are optimized over. The trajectories are then tested over the target models listed on the x-axis. The upper row is the result when the predicted logits are used as the measurement. The lower row is the result of the predicted probability. And each column represents a certain type of reference value. Results show that AlexNet and MobileNetV3's best trajectories are the most distant from others.

Table 3. The comparison among commonly studied DNNs on ILSVRC2012 with three different reference values. Here we present the difference between AUCs of the *logits* for LeRF and MoRF.

Reference	Model	T-y	T-p	Grad	GC	IBA	RISE	EP	EBP	IG	IxG
Zero	ResNet-18	**654.63**	494.39	246.28	345.30	368.63	330.01	271.24	330.95	217.46	170.96
	VGG-16	**733.33**	570.08	354.31	389.71	446.96	422.11	342.14	412.27	340.36	273.04
	AlexNet	**528.92**	387.99	235.47	215.60	234.70	262.67	225.70	215.60	228.04	199.65
	GoogLeNet	**436.03**	373.45	186.15	230.28	217.85	217.04	184.37	219.79	170.78	147.61
	MobileNetV3	**564.07**	412.34	140.08	279.24	228.40	198.11	173.48	200.90	146.53	111.26
	DenseNet-161	**748.21**	553.13	231.28	389.69	361.18	375.25	312.25	377.49	236.95	73.57
Mean	ResNet-18	**677.82**	490.32	255.54	349.48	332.02	332.77	259.02	334.96	225.60	179.44
	VGG-16	**761.62**	563.88	360.30	401.50	456.45	432.31	352.19	423.36	348.54	276.86
	AlexNet	**541.18**	396.74	252.16	242.20	241.08	313.22	257.51	242.20	235.24	197.67
	GoogLeNet	**446.35**	368.47	189.38	233.79	218.98	218.78	182.57	223.79	175.87	152.19
	MobileNetV3	**568.10**	411.44	146.17	284.28	238.15	210.70	183.48	212.35	148.70	109.74
	DenseNet-161	**755.73**	521.30	231.56	389.62	357.16	368.78	315.32	380.62	237.05	173.82
Blurring	ResNet-18	**667.50**	488.25	237.36	347.15	329.25	338.98	270.65	332.68	218.25	169.08
	VGG-16	**785.37**	572.89	355.51	409.08	457.24	450.51	356.16	420.88	350.53	279.70
	AlexNet	**580.66**	424.06	252.31	242.59	240.95	313.36	256.44	242.59	235.26	197.64
	GoogLeNet	**441.68**	369.25	178.26	218.80	206.72	221.96	180.74	207.93	169.69	144.81
	MobileNetV3	**567.15**	419.61	180.00	296.19	264.50	221.05	227.01	251.26	177.09	135.86
	DenseNet-161	**724.62**	507.72	194.39	340.50	317.74	337.16	286.58	331.22	199.23	141.73