

## A PROOF OF THEOREM 1

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

*Proof:* By representing deleting the most important  $k$  feature, we have equivalently that  $\mathbf{x}_{\setminus\sigma'[:k]} = \mathbf{x}_{\sigma[:d-k]}$ . Thus  $\sum_{k=0}^d f(\mathbf{x}_{\setminus\sigma'[:k]}) = \sum_{k=0}^d f(\mathbf{x}_{\sigma[:d-k]}) = \sum_{k=0}^d f(\mathbf{x}_{\sigma[:k]})$ . That is, Del-Mo  $\Leftrightarrow$  Ins-Le. Similarly, deleting the least important  $k$  features can be equivalently represented by  $\mathbf{x}_{\setminus\sigma[:k]}$  and  $\mathbf{x}_{\sigma'[:d-k]}$ . Hence  $\sum_{k=0}^d f(\mathbf{x}_{\setminus\sigma[:k]}) = \sum_{k=0}^d f(\mathbf{x}_{\sigma'[:d-k]}) = \sum_{k=0}^d f(\mathbf{x}_{\sigma'[:k]})$ , which means Del-Le  $\Leftrightarrow$  Ins-Mo.  $\square$

## B THE NUMBER OF SUPERPIXEL PATCHES

**The Connection with Adversarial Perturbation.** Due to the high nonlinearity, DNNs are widely known as fragile to adversarial attacks, where they can be easily fooled by slightly but maliciously perturbed input samples. This suggests that the optimum of TRACE-Del-Mo may not be the semantically important features. An illustration of an image from ILSVRC2012 is shown in fig. 8. Optimized trajectories are visualized as attribution maps (fig. 8). From the TRACE-Del-Mo results, we can find that in order to solve for TRACE-Del-Mo (i.e.,  $\{\min_{\sigma} \sum_{k=0}^d f(\mathbf{x}_{\setminus\sigma'[:k]})\}$ ), patches are deleted in an adversarial manners instead of the semantically meaningful way. Especially when  $t = 196$ , the deletion trajectory almost completely becomes an adversarial attack to the input image. In contrast, by trying to keep the highest prediction probability/logit with the gradual deletion process, TRACE-Del-Le and TRACE-Del-(Le-Mo) both highlight the long arms of the crane (i.e. keeps this area to the last moment), and are more robust to the number of feature groups.

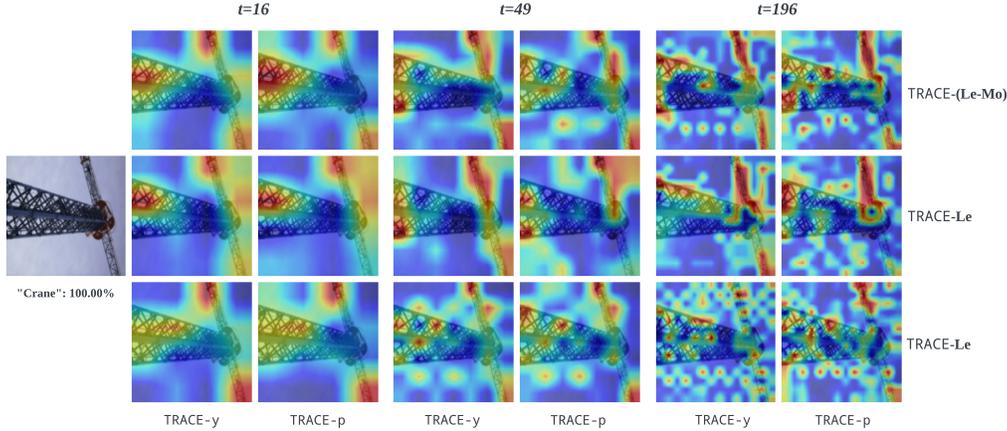


Figure 8: Demonstration of explanations of the image of a “crane” from ILSVRC2012 validation set. A ResNet-18 predicts it correctly with the confidence  $99.999\% \approx 100.00\%$ . The smooth factor  $\alpha = 2$ . We present results of resolutions  $t = 4 \times 4$  (left),  $t = 7 \times 7$  (middle) and  $t = 14 \times 14$  (right). Here  $t$  is the number of square patches of pixels for one image. The top, middle and bottom rows are TRACE-(Le-Mo), TRACE-Le and TRACE-Mo, respectively. Given  $t$ , the left column is TRACE-y and the right column is TRACE-p.

## C PROOF OF THEOREM 2

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

*Proof:* Note that  $f(\mathbf{x}_{\setminus\sigma'[:k]}) = f(\mathbf{x})$  is constant w.r.t.  $\sigma$  when  $k = 0$ , it suffices to minimize  $\sum_{k=1}^t f(\mathbf{x}_{\setminus\sigma'[:k]})$ . Here we show this by demonstrating the corresponding decision problem “Given a cost  $f^* \in \mathbb{R}$ , is there a trajectory  $\sigma$  s.t.  $\sum_{k=1}^t f(\mathbf{x}_{\setminus\sigma'[:k]}) \leq f^*$ .”

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

$$f(\mathbf{x}_{\setminus\sigma'[:i]}) := \delta_{\sigma[i]\sigma[i+1]} \quad (6)$$

In this way for any trajectory  $\sigma$ , we have

$$f_{tsp}(\sigma) = \sum_{i=1}^{p-1} \delta_{\sigma[i]\sigma[i+1]} = \sum_{i=1}^t f(\mathbf{x}_{\setminus\sigma'[:k]}) \quad (7)$$

Therefore, this polynomial time algorithm also serves as an algorithm for TSP, a contradiction.  $\square$

## D PSUEDO-CODE FOR TRACE WITH SIMULATED ANNEALING

---

### Algorithm 1 Simulated Annealing for TRACE

---

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

```

 $T \leftarrow T_0$ 
 $\sigma_0 \leftarrow \text{RandomInitialTrajectory}$ 
 $auc_0 \leftarrow \sum_{k=1}^t (f(\mathbf{x}_{\setminus\sigma_0[:k]}) - f(\mathbf{x}_{\setminus\sigma'_0[:k]}))$ 
 $k \leftarrow 0$ 
while  $k < K$  do
   $\sigma_1 \leftarrow \text{RandomChoice}(\text{neighbor}(\sigma_0))$ 
   $auc_1 \leftarrow \sum_{k=1}^t (f(\mathbf{x}_{\setminus\sigma_1[:k]}) - f(\mathbf{x}_{\setminus\sigma'_1[:k]}))$ 
   $\delta = auc_1 - auc_0$ 
  if  $\delta > 0$  then
     $\sigma_0 \leftarrow \sigma_1$ 
     $auc_0 \leftarrow auc_1$ 
  else
     $r \leftarrow \text{RandomUniform}(0, 1)$ 
    if  $r < \exp(\delta/T)$  then
       $\sigma_0 \leftarrow \sigma_1$ 
       $auc_0 \leftarrow auc_1$ 
    end if
  end if
   $k \leftarrow k + 1$ 
   $T \leftarrow \eta T$ 
end while
return  $\sigma_0$ 

```

---

## 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  $\sim 200$  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.

The results are shown in fig. 9, where Simulated Annealing outperform other algorithms in the experiments. It should be noticed that one of the most important factor 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 less evaluation times. For instance, Tabu Search requires to evaluate the 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 fig. 2. 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 optimum in the feasible set  $S_t$ .

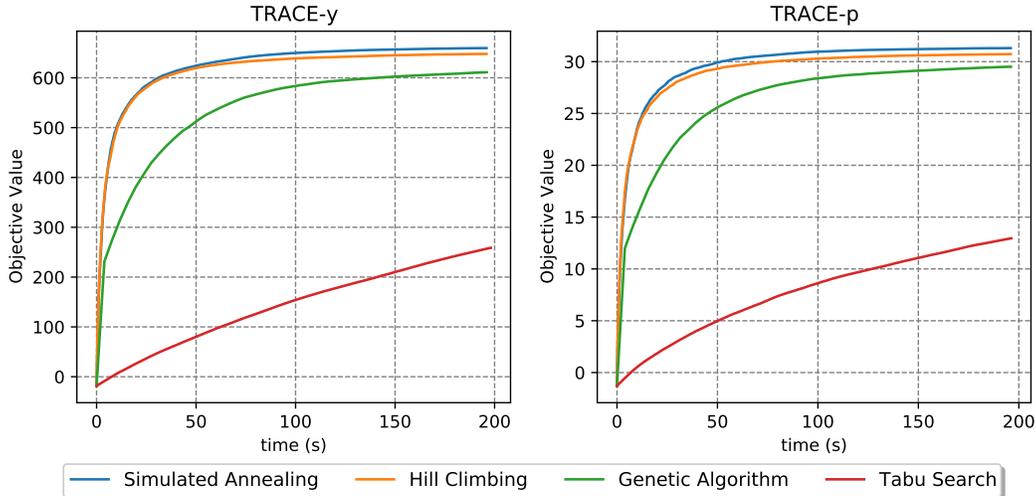


Figure 9: The comparison of different algorithms on solving TRACE.

## F PROOF OF THEOREM 3

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

*Proof:* Given any two permutations of length  $t$ :  $\forall \sigma_1, \sigma_2 \in S_t, \sigma_1 \neq \sigma_2$ , we have

$$\sigma_2 = (t - 1 \sigma_1^{-1}[\sigma_2[t - 1]]) \cdots (2 \sigma_1^{-1}[\sigma_2[2]]) \circ (1 \sigma_1^{-1}[\sigma_2[1]]) \circ \sigma_1 \quad (8)$$

where  $t - 1$  transpositions are applied to  $\sigma_1$ . 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  $\text{Cay}(S_t, S_{\text{complete}})$ .  $\square$

## G DELETION TRAJECTORIES

Here we visualize the trajectories in the form of gradual deletion of patches for images from ILSVRC 2012 validation sets. The results are shown in fig. 10. We include the three figures demonstrated in the previous experiments and two other images. It can be found that by solving for TRACE, we can find a deletion trajectory that preserve the model’s prediction with semantically meaningful features. Compared with heatmaps, such visualization is much more meaningful – it clearly illustrates how the explanations of TRACE are obtained how the black-box model is explained.

## H SANITY CHECK

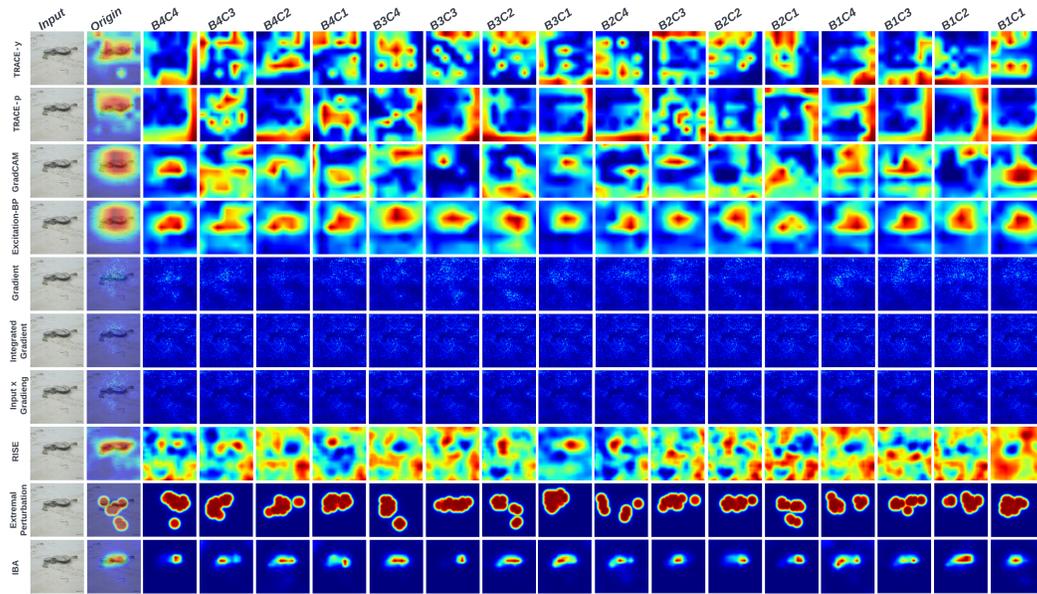
Here we present the sanity check visualizations for the first image of ILSVRC 2012 validation set same as the previous experiment. But we also include attribution explanation methods as benchmarks. The results are shown in fig. 11.

## I THE AREA BETWEEN MORF AND LERF

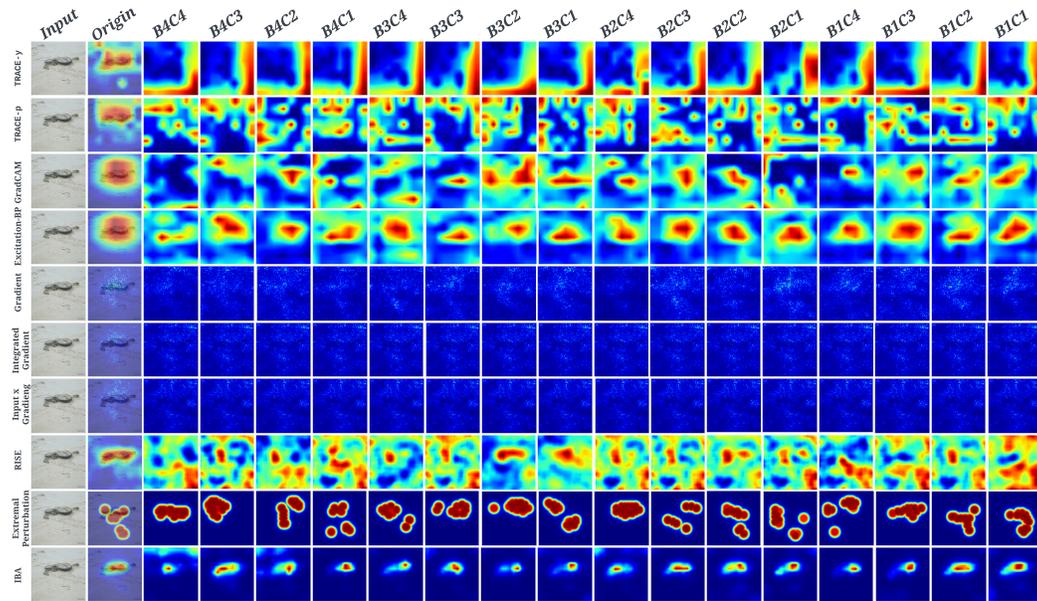
We visualize the results from fig. 5 and table 2 in the form introduced by (Schulz et al., 2020). As shown in fig. 12, MoRF and LeRF curves are plotted in blue solid and orange dashed fashions respectively. The gray areas between them (i.e. Mo–Le) are used to measure the explanations. This clearly visualize the significant improvement of TRACE in the deletion test.



Figure 10: The deletion trajectories of 5. The left column is the results of TRACE-y and the right column is TRACE-p. From the deletion trajectory, we can gain more insights of how specific features are ranked than simple attribution heatmaps.

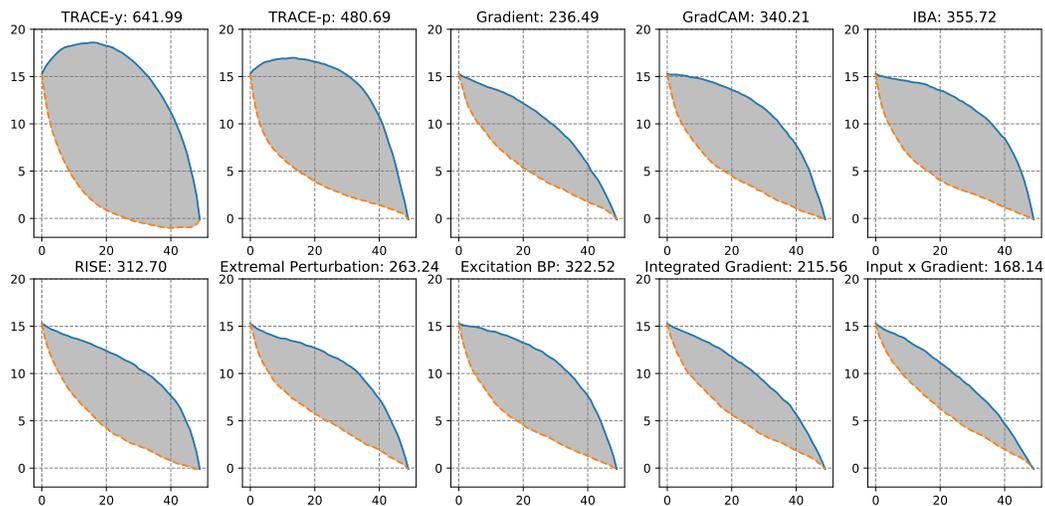


(a) Independent

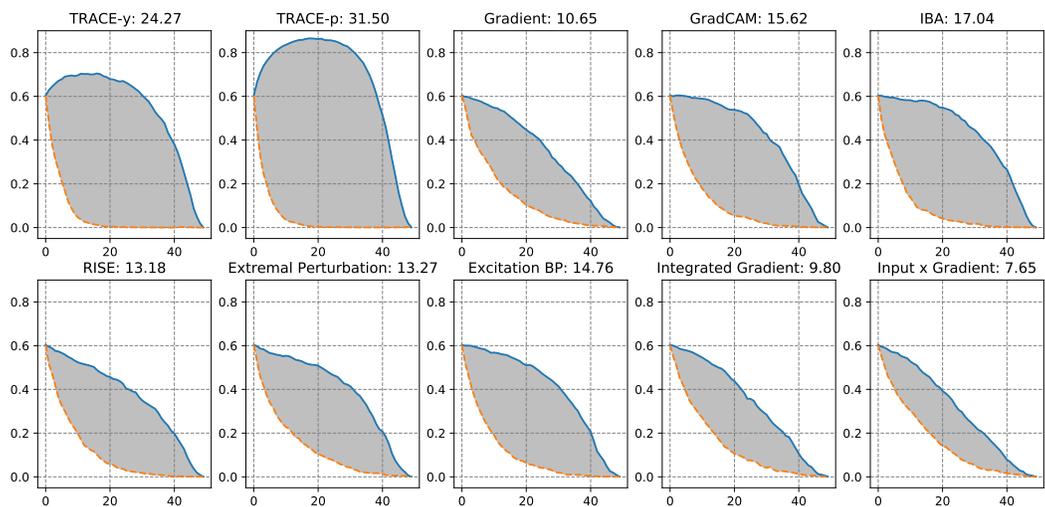


(b) Cascading

Figure 11: Sanity check using cascading randomization for TRACE and other tested explanation methods. (a) Convolutional layers of pre-trained ResNet-18 are randomized in the independent manners. That is, other layers are kept at the pre-trained values. (b) Convolutional layers of pre-trained ResNet-18 are randomized in the cascading manners, where layers are progressively randomized from left to right. Here “BaCb” means the  $b$ -th convolutional layer in the  $a$ -th block.



(a) The results w.r.t. the output logit



(b) The results w.r.t. the probability

Figure 12: The visualizations of the results in fig. 5 and table 2 in the form introduced by (Schulz et al., 2020). MoRF and LeRF curves are plotted in blue solid and orange dashed fashions respectively. The gray areas between them (i.e. Mo–Le) are used to measure the explanations. The value in each subtitle is the corresponding area (same as table 2).