Interpreting Attributions and Interactions of Adversarial Attacks: Supplementary Materials

A. Comparisons of Shapley-based attributions and other explanation methods

We define regional attributions and interactions between perturbation pixels based on Shapley values [6]. We compare Shapley-based attributions with other explanation methods from the following perspectives.

• Theoretical rigor. A good attribution method must satisfy certain desirable properties. The Shapley value has been proved to be the unique attribution that satisfies four desirable properties, *i.e.* the linearity property, the dummy property, the symmetry property, and the efficiency property [3]. In comparison, some explanation methods like Grad-CAM [5] and GBP [7] do not have theoretic supports for the correctness of these methods.

• Objectivity. The attribution of one input element depends on contexts of neighboring pixels. The Shapley value considers all possible contexts to compute the attribution of an input unit, which ensures the objectiveness of the attribution. In contrast, some attention methods, such as the adversarial saliency map [4], only consider the marginal gradient, which is biased to a specific context from this perspective.

• Trustworthiness. The theoretic foundation in game theory makes the Shapley values trustworthy. In contrast, some seemingly transparent explanation methods simply do not have clear theoretical support, which hurts the trustworthiness of the explanation. Actually, [1] has shown some explanation methods like GBP [7] can not reflect the true attribution.

• Broad applicability. The Shapley values can be extended to measure interactions between two input elements. [8] has proved the theoretical foundation and advantages for defining interactions using the Shapley value in game theory [3]. However, some gradient-based explanation methods assume the model is locally linear, which fails to measure interactions between two input elements.

B. Details of efficient approximation of interactions

We approximate Shapley values to enable efficient computation of interactions.

B.1. Approximation of attributions of perturbation pixels

In Section Algorithm, we introduce the approximation of the Shapley value of a perturbation pixel. In the supplementary material, we give more discussions about the approximation.

The adversarial perturbation is denoted as $\delta \in \mathbb{R}^n$. Each perturbation pixel i is divided into K sub-pixels with equal values, *i.e.* $\delta_i = \delta_{(i,1)} + \delta_{(i,2)} + \cdots + \delta_{(i,K)}$ and $\delta_{(i,1)} = \delta_{(i,2)} = \cdots = \delta_{(i,K)}$. Instead of directly computing the attribution of each perturbation pixel, we compute the attribution of each sub-pixel. The attribution of the sub-pixel can be efficiently approximated based on the Taylor expansion, which will be discussed later.

Among sub-pixels $(i,1),(i,2)\dots(i,K)$, each sub-pixel plays the same role in attacking, thereby $\phi_{(i,1)} = \phi_{(i,2)} = \cdots = \phi_{(i,K)}$, which is proved as follows. The attribution of each sub-pixel (i,k) is formulated as the Shapley value. The Shapley value satisfies the four axioms (linearity axiom, dummy axiom, symmetry axiom, and efficiency axiom). According to the symmetry axiom, given two sub-pixels (i,k) and (j,k') , if $z(S \cup \{(i,k)\}) = z(S \cup \{(j,k')\})$ holds for any set $S \subseteq \Omega^{\text{pixel}} \backslash \{(i,k),(j,k')\}$, then $\phi_{(i,k)} = \phi_{(j,k')}$, where $\Omega^{\text{pixel}} = \{(1,1),(1,2),\dots,(n,K-1),(n,K)\}$ denotes the set of all sub-pixels. Because the sub-pixel of the same perturbation pixel i has the equal value, given two sub-pixels (i,k) and (i,k') of the same perturbation pixel i, $z(S \cup \{(i,k)\}) = z(S \cup \{(i,k')\})$ holds for any set $S \subseteq \Omega^{\text{pixel}} \backslash \{(i,k),(i,k')\}$, where $1 \le k,k' \le K$, and $k \ne k'$. In this way, $\phi_{(i,1)} = \phi_{(i,2)} = \cdots = \phi_{(i,K)}$. Thus, we approximate the attribution of perturbation pixel i as $\phi_i = \sum_{i=1}^{K} \phi_{(i,k)}$, which equals to $\phi_i = K \cdot \phi_{(i,k)}$.

B.2. Properties of the approximated attribution

In Section Algorithm, we approximate the attribution of perturbation pixel i as $\phi_i = \sum_{i=1}^{K} \phi_{(i,k)}$. In the supplementary material, we further discuss properties of the approximated attribution.

The approximated attribution still satisfies the linearity axiom and the efficiency axiom.

Proof of the linearity axiom: Given two score functions $v(S)$ and $w(S)$, we use ϕ_i^v and ϕ_i^w to denote the attribution of perturbation pixel i to score v and score w respectively.

Let there be a new score function $f'(S) = v(S) + w(S)$. We use ϕ_i^{v+w} to denote the approximated attribution of perturbation pixel i to the new score function. The approximated attribution of perturbation pixel i is the sum of attributions sub-pixels, *i.e.* $\phi_i^{v+w} = \sum_{k=1}^{K} \phi_{(i,k)}^{v+w}$. The attribution of each sub-pixel is defined as the Shapley value. The Shapley value satisfies the linearity axiom. Then $\sum_{k-1}^{K} \phi_{(i,k)}^{v+w} = \sum_{k=1}^{K} (\phi_{(i,k)}^{v} + \phi_{(i,k)}^{w}) = \phi_i^v + \phi_i^w$. In this way, the approximated attribution is proved to satisfy the linearity axiom, *i.e.* $\phi_i^{v+w} = \phi_i^v + \phi_i^w$.

Proof of the efficiency axiom: The approximated attribution of each perturbation pixel is the sum of attributions of corresponding sub-pixels. Thus, the sum of approximated attributions of all perturbation pixels is the sum of attributions of all sub-pixels, *i.e.* $\sum_{i=1}^{n} \phi_i = \sum_{i=1}^{n} \sum_{k=1}^{K} \phi_{(i,k)}$.

Attributions of sub-pixels satisfy the efficiency axiom, *i.e.* $\sum_{i=1}^{n} \sum_{k=1}^{K} \phi_{(i,k)} = z(\Omega^{\text{pixel}}) - z(\emptyset)$. $z(\Omega^{\text{pixel}})$ is the score gained with all sub-pixels, *i.e.* the score made by the whole adversarial perturbation δ, and $z(\emptyset)$ is the score produced without the adversarial perturbation, *i.e.* the score made by the original image. $z(\Omega)$ also represents the score made by the whole adversarial perturbation δ, where $\Omega = \{1, 2, \ldots, n\}$ is the set of all perturbation pixels. Thus, $z(\Omega^{\text{pixel}}) = z(\Omega)$, and $\sum_{i=1}^{n} \sum_{k=1}^{K} \phi_{(i,k)} = z(\Omega^{\text{pixel}}) - z(\emptyset) = z(\Omega) - z(\emptyset)$. In this way, the approximated attribution is proved to satisfy the efficiency axiom, *i.e.* $\sum_{i=1}^{n} \phi_i = z(\Omega) - z(\emptyset)$.

B.3. Approximation for attributions of sub-pixels based on the Taylor expansion

In the paper, we approximate attributions of sub-pixels based on the Taylor expansion as Equation (8). In the supplementary material, we aim to derive the approximation in details.

Given a function $f(x_1, x_2, \ldots, x_n) : \mathbb{R}^n \to \mathbb{R}$, the Taylor expansion at $(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)})$ is

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) = & f(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}) \\ & + \sum_{i=1}^{n} (x_i - x_i^{(k)}) \frac{\partial f(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)})}{\partial x_i} \\ & + o((x_1 - x_1^{(k)}, x_2 - x_2^{(k)}, \ldots, x_n - x_n^{(k)})) \end{aligned}$$

$z(S \cup \{(i,k)\})$ denotes the change of the prediction score of the DNN made by sub-pixels in $S \cup \{(i,k)\}$, where $S \subseteq \Omega^{\text{pixel}} \backslash \{(i,k)\}$. The Taylor expansion for $z(S \cup \{(i,k)\})$ at S is given as

$$z(S \cup \{(i,k)\}) \approx z(S) + \delta_{(i,k)} \cdot \frac{\partial z(S)}{\partial \delta_{(i,k)}}$$

Thus, the approximation for the Shapley value of the sub-pixel (i,k) is given as

$$\begin{aligned} \phi_{(i,k)} &= \frac{1}{nK} \sum_{S \subseteq \Omega^{\text{pixel}} \backslash \{(i,k)\}} \binom{nK-1}{|S|}^{-1} \Big[z(S \cup \{(i,k)\}) - z(S) \Big] \\ &\approx \frac{1}{nK} \sum_{S \subseteq \Omega^{\text{pixel}} \backslash \{(i,k)\}} \binom{nK-1}{|S|}^{-1} \left(\frac{\partial z(S)}{\partial \delta_{(i,k)}} \delta_{(i,k)} \right) \end{aligned}$$

Let there be m components in a certain clustering step. $C_k^{(u)} = \bigcup_{i \in C^{(u)}} (i,k)$ denotes a sub-component. We use $\Omega^{\text{comp}} = \{C_1^{(1)}, C_2^{(1)}, \ldots, C_{K-1}^{(m)}, C_K^{(m)}\}$ to denote the set of all sub-components. The Shapley value of the sub-component $C_k^{(u)}$ is

approximated as

$$\begin{aligned}\phi_{C_k^{(u)}} &= \frac{1}{mK}\sum_{S\subseteq\Omega^{\text{comp}}\setminus\{C_k^{(u)}\}}\binom{mK-1}{|S|}^{-1}\Big[z(S\cup\{C_k^{(u)}\})-z(S)\Big]\\ &\approx \frac{1}{mK}\sum_{S\subseteq\Omega^{\text{comp}}\setminus\{C_k^{(u)}\}}\binom{mK-1}{|S|}^{-1}\sum_{(i,k)\in C_k^{(u)}}\Big[z(S\cup\{i,k\})-z(S)\Big]\\ &\approx \frac{1}{mK}\sum_{S\subseteq\Omega^{\text{comp}}\setminus\{C_k^{(u)}\}}\binom{mK-1}{|S|}^{-1}\sum_{(i,k)\in C_k^{(u)}}(\frac{\partial z(S)}{\partial\delta_{(i,k)}}\delta_{(i,k)})\end{aligned}$$

B.4. Implementation & computational complexity:

Clarification: In both the paper and the supplementary material, the computational complexity is quantified as times of network inference, *i.e.* the number of input (masked) images on which we conduct the forward/backward propagation. We do not count the number of detailed operations during the forward/backward propagation *w.r.t.* each specific input image, in order to simplify the analysis. It is because given a specific DNN, the number of detailed operations during the forward/backward propagation is the same for different input images.

In the paper, we introduce the implementation of the approximation of Shapley values and analyze the computational complexity. In the supplementary material, we aim to further explain the computational complexity of our approximation for attributions and how we approximate the attribution of components in detail.

We use a sampling-based method to reduce the complexity of computing Shapley values. The original formulation of the Shapley value considers all combinations of pixels to compute the Shapley value for each pixel. Thus, the computational complexity of the Shapley value of each pixel is $O(2^n)$. We implement the approximation of Shapley values of sub-pixels with a sampling method. In this way, the complexity of computing the Shapley value of one sub-pixel is reduced to $O(nKT)$. Note that $\phi_i \approx K\cdot\phi_{(i,k)}$. Therefore, the complexity of approximating the Shapley value of each pixel is also $O(nKT)$. The derivatives towards all sub-pixels can be computed simultaneously via back-propagation. Thus, the computational complexity of computing Shapley values of all pixels remains $O(nKT)$.

We use hierarchical clustering to iteratively merge several components into a larger component based on interactions. We use the following approximation method, to compute and reduce the complexity of computing the attribution of the pair of components. Let there be m components in a certain clustering step. Given a component $C^{(u)}$, we can use the sampling method to get their attributions $\phi_{C^{(u)}}$. Here, from the perspective of game theory, each component is a player, and there are m players in the game. As mentioned above, the complexity of computing $\phi_{C^{(u)}}$ is $O(mKT)$. We use $S^c = C^{(i_1)}\cup C^{(i_2)}\cup\cdots\cup C^{(i_q)}$ to denote a component candidate. To determine the interaction inside S^c, we need to compute ϕ'_{S^c}. The computation of ϕ'_{S^c} regards $C^{(i_1)}, C^{(i_2)}, \ldots, C^{(i_q)}$ as a single component. Then, the set of components changes to $\{S^c, C^{(i_{k+1})}\ldots, C^{(i_m)}\}$ with $m-q+1$ players. In this way, the computational complexity of ϕ'_{S^c} is $O((m-q)KT)$. Whereas, considering all potential pairs of components, the computational complexity grows. We only consider the interaction between neighboring components. There are m potential pairs of components, and the complexity of computing all potential pairs of components is $O(m(m-q)KT)$.

Considering the local property [2], we can further approximate $\phi'_{C^{(u)}\bigcup C^{(v)}}$ by simplifying contextual relationships of far-away pixels. Here, instead of computing ϕ'_{S^c} in the set $\{S^c, C^{(i_{k+1})}\ldots, C^{(i_m)}\}$, we randomly merge $\tilde{m}$ components to get $\tilde{m}/q$ component candidates, including S^c. In this way, the new set includes $\tilde{m}/q$ component candidates and $m-\tilde{m}$ components, *i.e.* $\{S^c, \bigcup_{a=q+1}^{2q} C^{(i_a)}, \ldots, \bigcup_{a=\tilde{m}-q+1}^{\tilde{m}} C^{(i_a)}, C^{(i_{\tilde{m}+1})}, \ldots, C^{(i_m)}\}$. We can simultaneously compute attributions of $\tilde{m}/q$ candidates in the new set, and the computational complexity is $O((m-(q-1)\tilde{m}/q)KT)$. To compute attributions of all potential component candidates, we need to sample $qm/\tilde{m}$ different sets. In this way, the overall complexity for the computation of attributions of candidates is reduced from $O(m(m-q)KT)$ to $O(m(qm/\tilde{m}-q)KT)$.

C. Pseudo code of extracting perturbation components

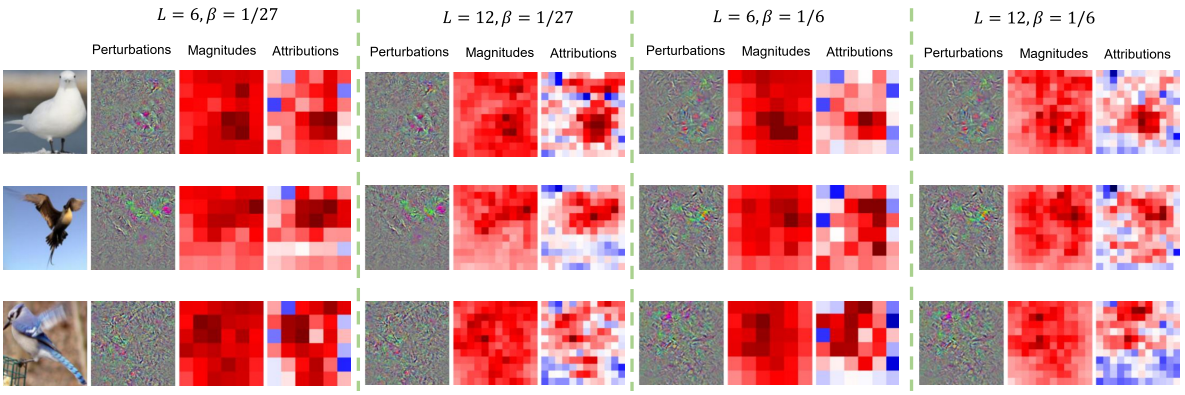
Algorithm 1 Extraction of perturbation components via hierarchical clustering

Inputs: pixel set Ω; reward function $z(\cdot)$; component size q; iteration times T
Outputs: Component set Ω';
Initialization: $\Omega' = \Omega$
for $iter = 1$ to T **do**
  $\forall C \in \Omega'$, compute ϕ_C with reward function $z(\cdot)$
  while *not* all possible component candidates are considered **do**
    Get component candidate set $\Omega^{\text{candidate}}$ by randomly merging each group of neighboring q components in Ω'
    $\forall C_{\text{candidate}} \in \Omega^{\text{candidate}}$, compute $\phi_{C_{\text{candidate}}}$ with reward function $z(\cdot)$
    Compute interaction in each component candidate: $I = \phi_{C_{\text{candidate}}} - \sum_{C \in C_{\text{candidate}}} \phi_C$
  end while
  $\Omega' = \emptyset$
  Update the component set Ω' by greedily adding the component candidate with highest interaction strength $|I|$ to Ω'
end for

D. Additional experimental results of regional attributions

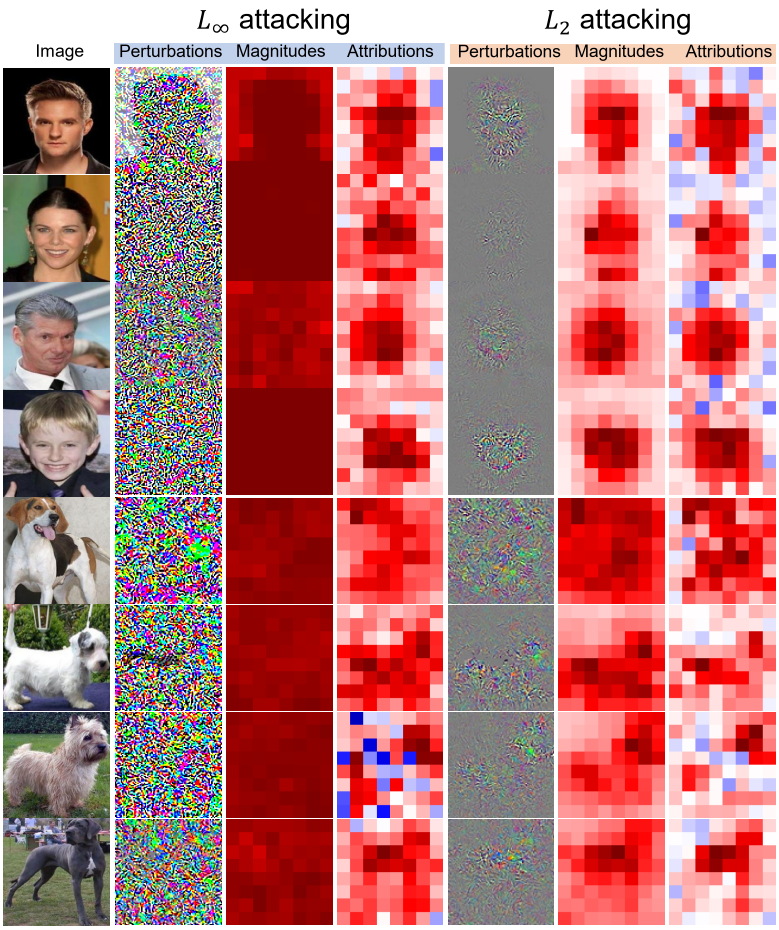
D.1. Regional attributions computed with different hyper-parameters

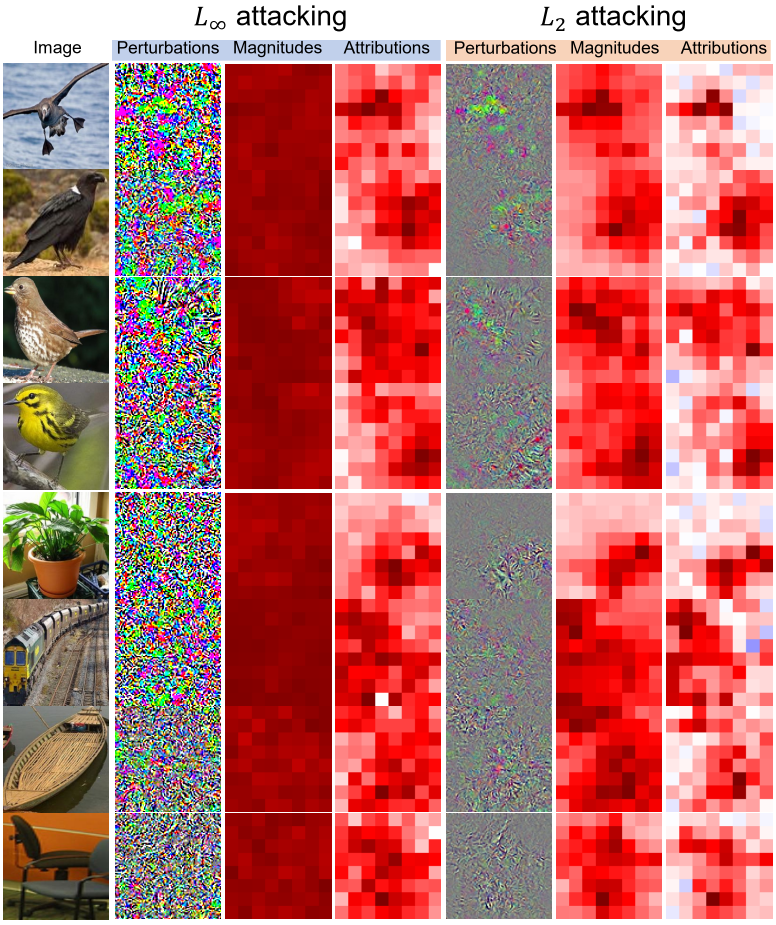
In this section, we have compared regional attributions with different hyper-parameters (β and L). The results are shown as follows. We found that important regions indicated by attributions were similar under the same selection of β, such as the belly region of the pigeon (in the first row) and the wing region of the jaeger (in the second row). Note that when β were different, the generated adversarial perturbations would be slightly different, which lead to slightly different regional attributions. However, compared with the difference between the magnitudes and attributions, the difference between regional attributions computed with different hyper-parameters was smaller.



D.2. More experimental results of regional attributions

Experimental results of regional attributions have been shown in Fig. 3 in the paper. In the supplementary material, we give additional results of regional attributions. The visualization shows that although the distribution of L_2 adversarial perturbations and the distribution of L_∞ adversarial perturbations were dissimilar, their regional attributions were similar to each other.



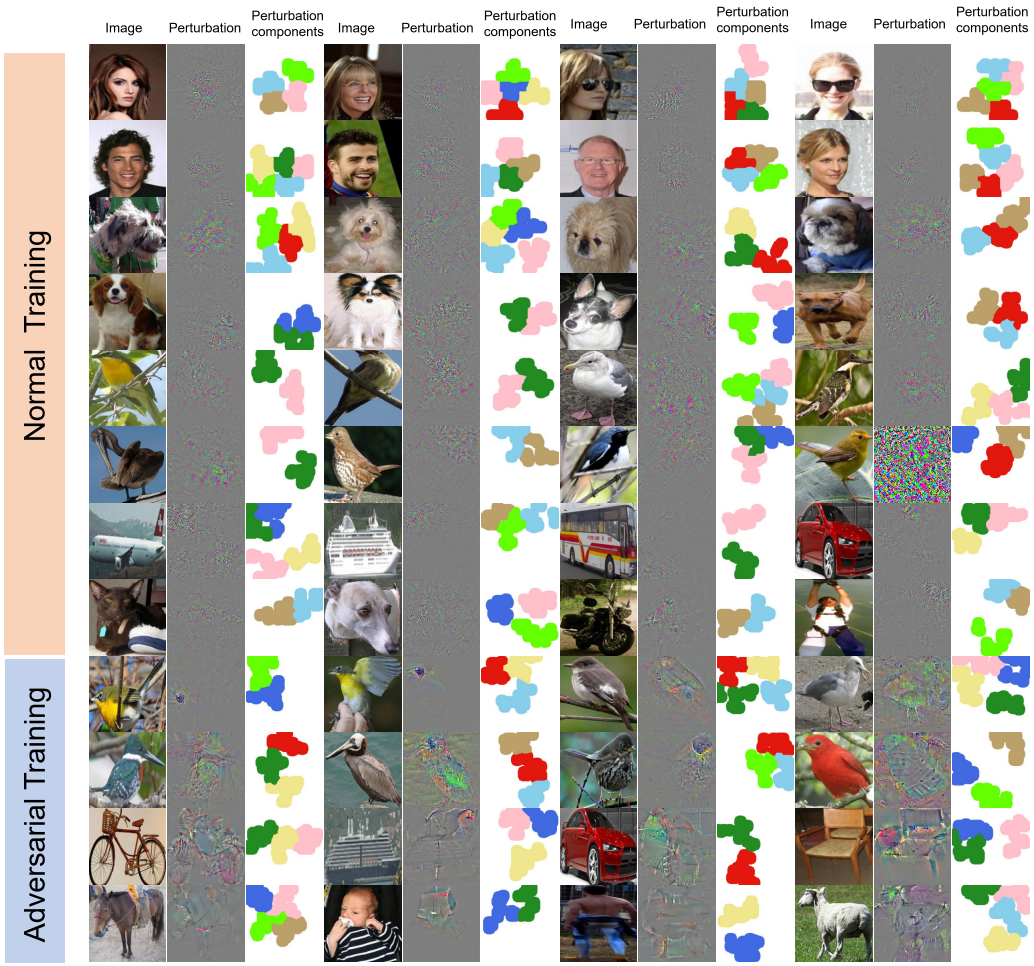

L_∞ attacking
L_2 attacking
Image
Perturbations
Magnitudes
Attributions
Perturbations
Magnitudes
Attributions

E. Comparisons of attributions

There are two types of attributions in the paper, *i.e.* regional attributions to the attacking cost and pixel-level attributions to the change of prediction score. We visualize regional attributions to the cost of L_2 attacking and L_∞ attacking and pixel-wise attribution to the change of the prediction score (under L_2 attacking). In most cases, important regions indicated by these attributions were similar. For example, in the third row, the dog's head and the horse's body were indicated to be important by all three kinds of attributions. In other cases, important regions indicated by different attributions were different. For example, important regions of the potted plant in the last row indicated by these three kinds of attributions were dissimilar.

F. More experimental results of interactions and perturbation components

Experimental results of interactions and perturbation components have been shown in Fig. 4 in the paper. In the supplementary material, we give more examples of visualizations. Perturbation components usually were not aligned with visual concepts.



References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[2] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. *In arXiv:1808.02610*, 2018.

[3] Grabisch Michel and Roubens Marc. An axiomatic approach to the concept of interaction among players in cooperative games. *In International Journal of Game Theory*, 1999.

[4] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. *In IEEE European Symposium on Security & Privacy*, 2016.

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

[6] Lloyd S Shapley. A value for n-person games. *In Contributions to the Theory of Games*, 2(28):307–317, 1953.

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

[8] Die Zhang, Huilin Zhou, Hao Zhang, Xiaoyi Bao, Da Huo, Ruizhao Chen, Xu Cheng, Mengyue Wu, and Quanshi Zhang. Building interpretable interaction trees for deep nlp models. In *AAAI*, 2021.