A Unified Game-Theoretic Interpretation of Adversarial Robustness

Jie Ren[a*], **Die Zhang**[a*], **Yisen Wang**[b*], **Lu Chen**[a], **Zhanpeng Zhou**[a],
Yiting Chen[a], **Xu Cheng**[a], **Xin Wang**[a], **Meng Zhou**[a,c†], **Jie Shi**[d], **Quanshi Zhang**[a‡]

[a] Shanghai Jiao Tong University
[b]Key Lab. of Machine Perception (MoE), School of EECS, Peking University
[c] Carnegie Mellon University
[d] Huawei technologies Inc.

Abstract

This paper provides a unified view to explain different adversarial attacks and defense methods, *i.e.* the view of multi-order interactions between input variables of DNNs. Based on the multi-order interaction, we discover that adversarial attacks mainly affect high-order interactions to fool the DNN. Furthermore, we find that the robustness of adversarially trained DNNs comes from category-specific low-order interactions. Our findings provide a potential method to unify adversarial perturbations and robustness, which can explain the existing defense methods in a principle way. Besides, our findings also make a revision of previous inaccurate understanding of the shape bias of adversarially learned features. Our code is available online at `https://github.com/Jie-Ren/A-Unified-Game-Theoretic-Interpretation-of-Adversarial-Robustness`.

1 Introduction

Adversarial robustness of deep neural networks (DNNs) has received increasing attention in recent years. Related studies include adversarial defense and attacks [60, 25]. In terms of defense, adversarial training is an effective and the most widely-used method [41, 85, 67, 68, 73]. In spite of their fast development, the essential mechanism of the adversarial robustness is still unclear. Thus, the understanding of adversarial attacks and defense is an emerging direction in recent years. Ilyas et al. [33] demonstrated adversarial examples could be attributed to the presence of non-robust yet discriminative features. Some methods [24, 71] explored the mathematical bound for the model robustness. Zhang and Zhu [86], Tsipras et al. [63] found adversarial training helped DNNs learn a more interpretable (more shape-biased) representation. Besides the feature interpretability, Tsipras et al. [63] further showed an inherent tension between the adversarial robustness and the generalization power.

Unlike above perspectives for explanations, we aim to propose a unified view to explain the essential reason why and how adversarial examples emerge, as well as essential mechanisms of various adversarial defense methods. We rethink the adversarial robustness from the novel perspective of interactions between input variables of a DNN. It is because in an adversarial example, adversarial

[*] Equal contribution

[†] This work was done when Meng Zhou was an undergraduate at Shanghai Jiao Tong University.

[‡] Quanshi Zhang is the corresponding author. `zqs1022@sjtu.edu.cn`. This work is supervised by Dr. Quanshi Zhang. He is with the John Hopcroft Center and the MoE Key Lab of Artificial Intelligence, AI Institute, at the Shanghai Jiao Tong University, China.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

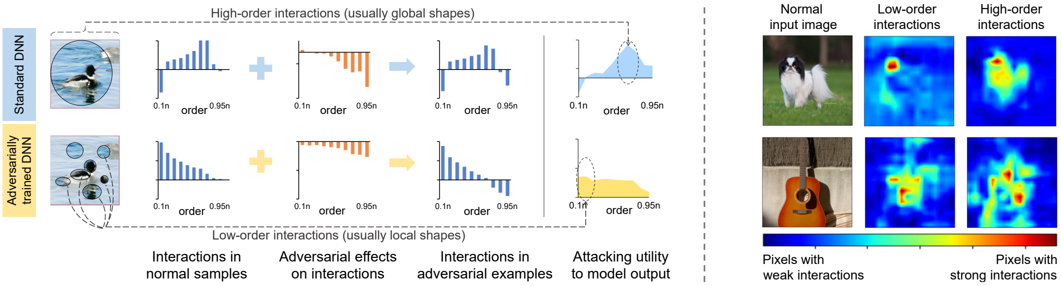

High-order interactions (usually global shapes)
Standard DNN
Adversarially trained DNN
0.1n
order
0.95n
Low-order interactions (usually local shapes)
Interactions in normal samples
Adversarial effects on interactions
Interactions in adversarial examples
Attacking utility to model output
Normal input image
Low-order interactions
High-order interactions
Pixels with weak interactions
Pixels with strong interactions

Figure 1: (Left) Adversarial attacks mainly affect high-order interactions in input samples. High-order interactions in adversarially trained DNNs are more robust than those in standard DNNs. (Right) Regions with strong low-order and high-order interactions, which are visualized by the method extended from [83].

perturbations on different pixels do not attack the DNN independently. Instead, perturbation pixels usually interact with each other to form a specific pattern for attacking. Surprisingly, we find that such interactions can explain various aspects of adversarial robustness. Specifically, this study aims to answer the following three questions.

• **How to disentangle feature representations that are sensitive to adversarial perturbations.** Based on the game-theoretic interactions, we aim to summarize the distinct property of feature representations, which are sensitive to adversarial perturbations, among overall feature representations.

• **How to explain the effectiveness of the adversarial training.** The above summarized property of sensitive feature representations also provides a new perspective to explain the utility of the adversarial training, *i.e.* why and how the adversarial training penalizes such sensitive feature representations.

• **How to unify various adversarial defense methods in a single theoretic system.** Our research provides a unified understanding for the success of adversarial defense methods [78, 18, 35].

As a prerequisite of analyzing the adversarial robustness, let us first revisit the interaction between input variables in DNNs. Let a set of input variables collaborate with each other to form an inference pattern, the Shapley interaction index [26] is a standard metric to measure the numerical benefits to the inference from their collaborations. This metric can be further extended to the multi-order interaction [83]. For the interaction between two input variables (i, j), the interaction order measures the number of contextual variables that influence the significance of the interaction between i and j. In other words, low-order interactions represent simple collaborations between input variables with small contexts, while high-order interactions indicate complex collaborations over large contexts (see Figure 1 (right)).

We further prove that the network output can be decomposed into the sum of multi-order interactions between different pairs of input variables. Thus, the overall effects of adversarial perturbations on the network output can be decomposed into elementary effects on different interaction components. Therefore, we can explain adversarial robustness using such elementary interaction components.

(1) **We discover and partially prove that adversarial perturbations mainly affect high-order interactions, rather than low-order interactions** (see Figure 1 (left)). In comparison, low-order interactions are naturally robust to attacks. In other words, adversarial attacks mainly affect the complex and large-scale collaborations among most pixels in the image. Based on this, we can successfully disentangle sensitive feature representations, *i.e.* high-order interactions.

Interaction vs. frequency & rank. Some studies explained adversarial examples as high-frequency features [79, 64, 27] and high-rank features [35]. We have conducted experiments to show that high-order interactions can better explain the essential property of attacking-sensitive representations, *i.e.* the complex and large-scale visual concepts.

(2) A clear difference between standard DNNs and adversarially trained DNNs is as follows. **Adversarial training significantly increases the robustness of high-order interactions.** In other words, attacks mainly affect complex collaborations in standard DNNs, while for adversarially trained DNNs, complex collaborations are not so vulnerable *w.r.t.* simple collaborations.

Then, we further explain the reason for the high robustness of adversarially trained DNNs. For interactions of each order, we define the disentanglement metric to identify whether interactions of this order are discriminative for the classification of a specific category, or represent common knowledge shared by different categories. For example, in Figure 1 (left), interactions representing the blue water may be shared by different categories, while interactions corresponding to the head of the red-breasted merganser are discriminative for this category. Based on the disentanglement metric, we discover that **compared with standard DNNs, adversarially trained DNNs usually encode more discriminative low-order interactions. Discriminative low-order interactions make high-order interactions of adversarially trained DNNs robust to attacks,** because contexts of high-order interactions are composed of many small contexts of low-order interactions. For example, if the simple (low-order) interactions for the red-breasted merganser are learned to be discriminative, instead of being shared by the bicycle category, then it is difficult to attack this image towards the bicycle category. It is because it is difficult to use the low-order interactions of the red-breasted merganser's head to construct high-order interactions of bicycles.

(3) **Our research provides a unified understanding for the success of several existing adversarial defense methods**, including the attribution-based detection of adversarial examples [78], the recoverability of adversarial examples to normal samples, the cutout method [18], and the rank-based method [35] (which is proved by [35] to be related to frequency-based methods [79, 64, 74]).

Above findings also slightly revise the previous explanation of adversarially learned features [25, 63, 19, 86]. They claimed that adversarial training learned more information about foreground shapes. We discover that these adversarially learned features are actually low-order interactions (usually local shapes), instead of modeling the global shape of the foreground.

Explainable AI system based on game-theoretic interactions. In fact, our research group led by Dr. Quanshi Zhang have proposed game-theoretic interactions, including interactions of different orders [82] and multivariate interactions [84]. The interaction can be used as an typical metric to explain signal processing in DNNs from different perspectives. For example, the game-theoretic metric can be used to guide the learning of baseline values of Shapley values [49]. Furthermore, we have built up a tree structure to explain the hierarchical interactions between words encoded by NLP models [81]. We have also used interactions to explain the generalization power of DNNs [83]. The interaction can also explain how adversarial perturbations contribute to the attacking task [66], and explain the transferability of adversarial perturbations [65]. Furthermore, we have also used the interaction to formulate the visual aesthetics [12] and signal-processing properties of different types of visual concepts [11] in DNNs. As an extension of the system of game-theoretic interactions, in this study, we explain the adversarial robustness based on interactions.

2 Related work

Attacking methods can be roughly summarized into white-box attacks [60, 25, 37, 46, 8] and black-box attacks [38, 47, 10, 6, 32, 72, 4]. For defense, adversarial training is one of the most effective and widely-used defense methods [41, 85, 67, 68, 73]. Other defense methods include masking gradients [47, 44], modifying networks [13, 23], and applying pre-processing on input images for testing [17, 42, 75, 3].

Explanations for adversarial examples. Some previous studies focused on the reason for the existence of adversarial examples. Goodfellow et al. [25] explained adversarial examples as a result of the high linearity of feature representations. Gilmer et al. [24], Ma et al. [40] proved that the existence of adversarial examples was due to the geometry of the high-dimensional manifold. Xie et al. [75], Xu et al. [76, 77], Bai et al. [5] discovered that adversarial perturbations usually activated substantial "noise" and semantically irrelevant features. Engstrom et al. [20] investigated the vulnerability of DNNs to rotations and translations. Tsipras et al. [63], Ilyas et al. [33] demonstrated that adversarial examples were attributed to non-robust yet discriminative features.

Understandings of adversarial training. Athalye et al. [2] suggested that adversarial training did not cause the obfuscated gradients phenomenon, which boosted the robustness. Goodfellow et al. [25], Dong et al. [19], Tsipras et al. [63], Zhang and Zhu [86] found that adversarially trained DNNs learned more shape-biased features than standard DNNs. Chalasani et al. [9] proved that adversarially trained DNNs *w.r.t.* the ℓ_∞ attack exhibited more sparse attributions. Tian et al. [61] showed that there is robustness imbalance among classes in adversarial training. Song et al. [55] considered the adversarial training as the enumeration of all potential adversarial perturbations. Wang et al. [67] explained adversarial training from the perspective of min-max optimization. Yin et al. [79], Wang et al. [69], Harder et al. [27] discovered adversarial training pushed DNNs to utilize low-frequency components in inputs.

Understanding of the robustness. Szegedy et al. [60], Hein and Andriushchenko [29] computed Lipschitz constant to explain the robustness. Ignatiev et al. [31], Boopathy et al. [7] investigated the connection between network interpretability and adversarial robustness. Tsipras et al. [63] proved the inherent tension between adversarial robustness and standard generalization power. Fawzi et al. [21], Gilmer et al. [24], Weng et al. [71] proved lower/upper bounds on the robustness. Pal and Vidal [45] proposed a game-theoretic framework to understand the adversarial robustness, by formulating the game between attackers and defenders *w.r.t.* a trained DNN. Under the game-theoretic framework, Pal and Vidal [45] theoretically proved that the FGM attack and the random smoothing defense formed a Nash Equilibrium under some assumptions.

Unlike previous explanations of the existence of adversarial examples and adversarial robustness, we explain adversarial examples and adversarial training from a new perspective, *i.e.* the complexity of visual concepts that are learned by a DNN.

3 Decomposing attacking utility into interactions of multiple orders

Preliminaries: using Shapley values [52] and the Shapley interaction index [26] to explain a DNN. The Shapley value [52] in game theory is widely considered as an unbiased estimation for the importance or contribution of each player in a game. Given a trained DNN and the input with n variables (*e.g.* an image with n pixels, a sentence with n words) $N = \{1, \cdots, n\}$, we can take input variables as players, and consider the network output as the reward. Shapley values can fairly divide and assign numerical effects on the network output to each input variable. More specifically, let $2^N \overset{\text{def}}{=} \{S|S \subseteq N\}$ denote all potential subsets of N, and each subset S represents a specific context (*e.g.* a set of pixels in an image). $v(S) \in \mathbb{R}$ represents the scalar network output when we keep variables in S unchanged and mask variables in $N \setminus S$ by following settings in [1] (*i.e.* setting a masked variable to the average value over different input samples).[4] In particular, $v(N)$ refers to the network output *w.r.t.* the entire input N (*e.g.* the whole image), and $v(\emptyset)$ denotes the output when we mask all variables. In this way, the Shapley value $\phi(i)$ unbiasedly measures the importance of the variable i to the network output.

$$\phi(i) = \sum\nolimits_{S \subseteq N \setminus \{i\}} p(S) \left[v(S \cup \{i\}) - v(S) \right], \quad p(S) \overset{\text{def}}{=} \frac{(n - |S| - 1)! |S|!}{n!} \tag{1}$$

In this way, the network output can be considered as the sum of elementary importances of input variables, *i.e.* $v(N) = \sum_{i \in N} \phi(i) + v(\emptyset)$. The Shapley value has been proved to satisfy four desirable properties, *i.e.* ***linearity, nullity, symmetry*** and ***efficiency*** properties, thereby being regarded as a fair method to allocate the network output to each input variable [70].

Input variables of a DNN do not contribute to the network output independently. Instead, different variables collaborate with each other to affect the network output, and we use the interaction to quantify the numerical utility of such collaborations. To this end, the Shapley interaction index [26] measures the utility of the collaboration between two input variables (i, j) by examining **whether the absence/presence of an input variable j will change the importance of the other variable i.** Their interaction is defined as $I(i, j) = \tilde{\phi}(i)_{j \text{ always present}} - \tilde{\phi}(i)_{j \text{ always absent}}$, where $\tilde{\phi}(i)_{j \text{ always present}}$ and $\tilde{\phi}(i)_{j \text{ always absent}}$ denote the importance of the variable i when j is always present and when j is always absent, respectively (please see [26] for more details about $I(i, j)$).

Decomposition of the existing Shapley interaction index into multiple orders. Zhang et al. [83] further decomposes the Shapley interaction index $I(i, j)$ [26] into interactions of different orders.

$$I(i, j) = \frac{1}{n-1} \sum\nolimits_{m=0}^{n-2} I_{ij}^{(m)}, \quad I_{ij}^{(m)} = \mathbb{E}_{S \subseteq N \setminus \{i,j\}, |S| = m} \left[\Delta v(i, j, S) \right] \tag{2}$$

where $\Delta v(i, j, S) = v(S \cup \{i, j\}) - v(S \cup \{i\}) - v(S \cup \{j\}) + v(S)$.

[4] For the DNN trained for multi-category classification, we set $v(S) = \log p(y = c | \text{given an input with variables in } N \setminus S \text{ masked})$, where c is selected as either the ground-truth category of the input, or the incorrectly classified category after attacks.

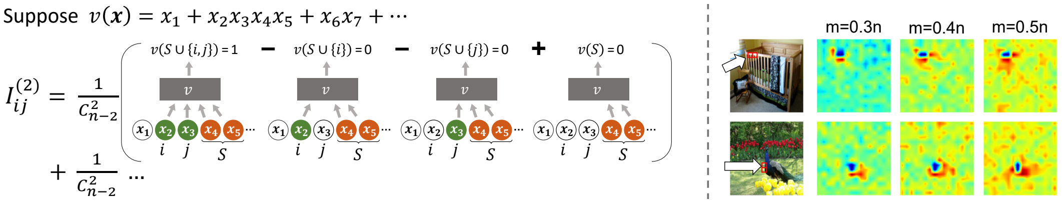

Suppose $v(\boldsymbol{x}) = x_1 + x_2x_3x_4x_5 + x_6x_7 + \cdots$
$I_{ij}^{(2)} = \frac{1}{C_{n-2}^2}$
$v(S\cup\{i,j\})=1$
$v(S\cup\{i\})=0$
$v(S\cup\{j\})=0$
$v(S)=0$
v
x_1 x_2 x_3 x_4 x_5 ⋯
i j S
$+\frac{1}{C_{n-2}^2}\cdots$
m=0.3n
m=0.4n
m=0.5n

Figure 2: (Left) Computation of the m-order interaction between two input variables ($m = 2$). (Right) Contexts of the m-order interaction $I_{ij}^{(m)}$. Variables (i, j) are indicated by red boxes. We extend the visualization method in [84], which visualized contexts of the Shapley interaction index, to show contexts of multi-order interactions (details in the supplementary material). Red (blue) colors indicate pixels that frequently (infrequently) appear in contexts which cause significant interactions.

Proof of decomposing network output into multi-order interactions to explain adversarial attacks. In this study, we successfully prove that the network output can be decomposed as the weighted sum of multi-order interactions. This enables us to use such interactions to explain adversarial attacks. Before the decomposition, let us first use interactions to understand the attack. As Figure 2 (left) shows, the interaction component of each order $I_{ij}^{(m)}$ represents collaborations between two input variables (i, j) with a specific contextual complexity. Let us consider input variables i, j and an arbitrary set of m contextual variables. If $I_{ij}^{(m)} > 0$, it indicates that the presence of the variable j will increase the importance of the variable i. Thus, we consider variables (i, j) have a positive interaction. If $I_{ij}^{(m)} < 0$, it indicates a negative interaction.

The m-order interaction between two input variables (i, j) measures the average interaction between (i, j) when we consider their collaborations with m contextual variables (m pixels). From another perspective, the order m measures the number of contextual variables that influence the interaction between i and j. For a low order m, $I_{ij}^{(m)}$ reflects the interaction between i and j *w.r.t.* simple contexts of a few variables. For a high order m, $I_{ij}^{(m)}$ corresponds to the interaction *w.r.t.* complex contexts with massive variables. Figure 2 (right) visualizes the contexts S corresponding to strong low-order interactions and strong high-order interactions, respectively. Cheng et al. [11] has also proven that low-order interactions (local collaborations) mainly reflect simple and common concepts (features), and high-order interactions (global collaborations) usually represent complex and global features.

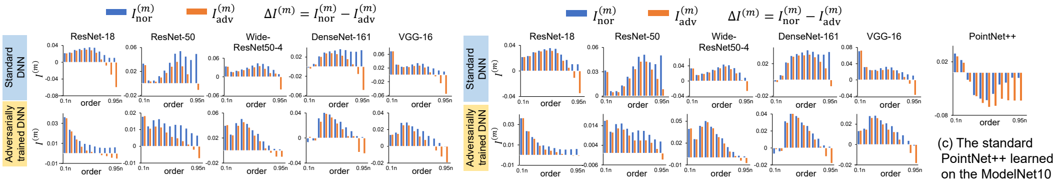
• *Properties of multi-order interactions.* We have proven that $I_{ij}^{(m)}$ satisfies ***linearity, nullity, commutativity, symmetry***, and ***efficiency*** properties (please see the supplementary material for details). In particular, the *efficiency property* is given as follows.

Efficiency property: The output of the DNN given the entire input $v(N)$ can be decomposed into interactions of different orders, *i.e.* $v(N) = v(\emptyset) + \sum_{i\in N}\phi^{(0)}(i) + \sum_{i,j\in N, i\neq j}\sum_{m=0}^{n-2} J_{ij}^{(m)}$, where $J_{ij}^{(m)} \overset{\text{def}}{=} \frac{n-1-m}{n(n-1)} I_{ij}^{(m)}$, and $\phi^{(0)}(i) \overset{\text{def}}{=} v(\{i\}) - v(\emptyset)$.

• *Decomposing the overall attacking utility into interactions of different orders.* According to the *efficiency property* of interactions, the output of the DNN *w.r.t.* the image x can be decomposed into the sum of multi-order interactions between different pairs of input variables. Thus, we can decompose the overall utility of adversarial perturbations on the network output into elementary effects on interactions.

$$\Delta v(N|x) \overset{\text{def}}{=} v(N|x) - v(N|x^{\text{adv}}) = \underbrace{\Delta v(\emptyset|x) + \sum_{i\in N}\Delta\phi^{(0)}(i|x)}_{\text{usually can be ignored}} + \sum_{i\neq j\in N}\sum_{m=0}^{n-2}\Delta J_{ij}^{(m)}(x) \tag{3}$$

where $x \in \mathbb{R}^n$ denotes the normal sample, and $x^{\text{adv}} = x + \Delta x \in \mathbb{R}^n$ denotes the adversarial example. $v(N|x^{\text{adv}})$ denotes the network output given all variables in N of the adversarial example x^{adv}. $v(N|x)$ corresponds to the normal sample. The first term $\Delta v(\emptyset) = v(\emptyset|x) - v(\emptyset|x^{\text{adv}}) = 0$. In the second term, $\Delta\phi^{(0)}(i|x) \overset{\text{def}}{=} (v(\{i\}|x) - v(\emptyset|x)) - (v(\{i\}|x^{\text{adv}}) - v(\emptyset|x^{\text{adv}})) = v(\{i\}|x) - v(\{i\}|x^{\text{adv}})$. Because in most applications, the importance of a single variable (*e.g.* a pixel) is usually small, we can ignore this term. In the third term, $\Delta J_{ij}^{(m)}(x)$ denotes the attacking utility of the m-order interaction between variables (i, j) in x. $\Delta J_{ij}^{(m)}(x) \overset{\text{def}}{=} \frac{n-1-m}{n(n-1)}\Delta I_{ij}^{(m)}(x)$, where $\Delta I_{ij}^{(m)}(x) \overset{\text{def}}{=} I_{ij}^{(m)}(x) - I_{ij}^{(m)}(x^{\text{adv}})$ measures the elementary effects of adversarial perturbations on the m-order interaction.



(a) Untargeted attack

(b) Targeted attack

(c) The standard PointNet++ learned on the ModelNet10 dataset.

Figure 3: The multi-order interaction $I^{(m)}_{\text{nor}}$ and $I^{(m)}_{\text{adv}}$ of standard DNNs and adversarially trained DNNs. Adversarial perturbations mainly affected high-order interactions.

4 Explaining adversarial attacks and defense using interactions

To simplify the story, we only study the simplest and widely-used untargeted and targeted ℓ_∞ PGD attacks [41].

4.1 Attacks mainly affect high-order interactions

Attacking utility of m-order interactions. According to Eq. (3), effects of adversarial attacks mainly depend on changes of interactions $\Delta I^{(m)}_{ij}(x)$ (or $\Delta J^{(m)}_{ij}(x)$). Thus, in this section, we further conduct experiments to show that adversarial attacks mainly affect $\Delta I^{(m)}_{ij}(x)$ of high orders. In order to compare the multi-order interactions in normal samples and those in adversarial examples, we measure the average interaction of a specific order m among different pairs of variables in different input images, *i.e.* $I^{(m)} = \mathbb{E}_{x\in\Omega}\mathbb{E}_{i,j}[I^{(m)}_{ij}(x)]$, where $\Omega \subseteq \mathbb{R}^n$ denotes the set of all samples. We use the normal samples Ω_{nor} to compute $I^{(m)}_{\text{nor}}$ and use adversarial examples Ω_{adv} to compute $I^{(m)}_{\text{adv}}$. In this way, $\Delta I^{(m)} \overset{\text{def}}{=} I^{(m)}_{\text{nor}} - I^{(m)}_{\text{adv}}$ represents effects of attacks on the m-order interactions. Similarly, $\Delta J^{(m)} \overset{\text{def}}{=} \mathbb{E}_{x\in\Omega}\mathbb{E}_{i,j}[\Delta J^{(m)}_{ij}(x)] = \frac{n-1-m}{n(n-1)}\Delta I^{(m)}$ measures the attacking utility of the m-order interactions according to Eq. (3).

Comparing attacking utilities between interactions of different orders (in terms of $I^{(m)}$). In order to measure $I^{(m)}_{\text{nor}}$ and $I^{(m)}_{\text{adv}}$, we conducted experiments on ResNet-18/34/50 [28], Wide-ResNet50-4 [80], DenseNet-161 [30], and VGG-16 [53]. For each DNN, we obtained both the standardly trained version and the adversarially trained version on the ImageNet dataset [50][5]. We also conducted experiments on PointNet++ [48] learned on the ModelNet10 dataset [74], which is a 3D dataset. We measured the interaction $I^{(m)}$ by setting $v(S|x) = \log p(y = y^{\text{truth}}|\text{given variables in } S \text{ in the input } x \text{ and mask variables in } N \setminus S)$, and $x \in \Omega$ was sampled from the validation set. By following settings in [1], the masked variables in $N \setminus S$ were set to the average value over different input samples. However, the computational cost of $I^{(m)}$ was intolerable. To reduce the computational cost, we applied the sampling-based approximation method in [83] and computed interactions at the grid level, rather than the pixel level. Please see the supplementary material for more discussions.

We considered both untargeted and targeted[6] PGD attacks [41] with the ℓ_∞ constraint $\|\Delta x\|_{\ell_\infty} \le \epsilon$ to generate adversarial examples. We set $\epsilon = 32/255$ by following the setting in [75]. The step size was set to $2/255$. For fair comparisons, we controlled the perturbation generated for each image to have similar attacking utility of 8. The attacking utility in untargeted attacks was defined as $U_{\text{untarget}}(x) = h_{y^{\text{truth}}}(N|x) - h_{y^{\text{truth}}}(N|x^{\text{adv}})$, where $h_{y^{\text{truth}}}(N|x)$ and $h_{y^{\text{truth}}}(N|x^{\text{adv}})$ denote network outputs of the ground-truth category y^{truth} before the softmax layer, when taking the normal sample and the adversarial example as input, respectively. The attacking utility in targeted attacks was defined as $U_{\text{target}}(x) = (h_{y^{\text{target}}}(N|x^{\text{adv}}) - h_{y^{\text{truth}}}(N|x^{\text{adv}})) - (h_{y^{\text{target}}}(N|x) - h_{y^{\text{truth}}}(N|x))$, where $h_{y^{\text{target}}}(\cdot)$ denote the network output of the target category[6] y^{target}.

Results. Figure 3 shows the multi-order interactions within normal samples and adversarial examples. We found that adversarial perturbations significantly decreased high-order interactions, in both adversarially trained DNNs and standard DNNs. For example, in Figure 3 (a), low-order interactions

[5] We used pretrained models released by Salman et al. [51].

[6] In the untargeted attack, we considered the misclassified category as the target category. In the targeted attack, the target label was set as the misclassified category in the untarget attack. When we consider interaction *w.r.t.* the target category, we set $v(S) = \log p(y = y^{\text{target}}|\text{given variables in } S \text{ and mask variables in } N \setminus S)$.

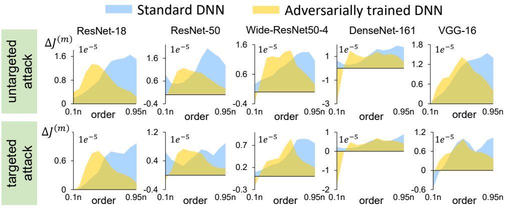


Figure 4: Distribution of compositional attacking utilities caused by interactions of different orders $\{\Delta J^{(m)}\}$.

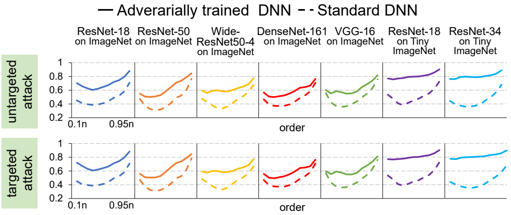


Figure 5: The interaction disentanglement $D^{(m)}$. Interactions of adversarially trained DNNs exhibited higher disentanglement than those of standard DNNs.

($m < 0.5n$) in standard ResNet-18 decreased a little, while the interaction of order $m = 0.95n$ dropped from 0.01 to -0.059. This phenomenon was consistent with the heuristic findings of Dong et al. [19] that neurons corresponding to high-level semantics were ambiguous.

Additional experiments on the target category and other categories. Beyond above analysis about interactions *w.r.t.* the ground-truth category, we also conducted experiments and found that high-order interactions *w.r.t.* the target category[6] increased significantly. Please see the supplementary material for details. In sum, high-order interactions were much more sensitive than low-order interactions, which verified our conclusions.

Theoretic explanation of the sensitivity of high-order interactions. Besides the above empirical observation, we can also theoretically prove that high-order interactions are more sensitive than low-order interactions.

Proposition 1 *(equivalence between the multi-order interaction and the mutual information): Given an input sample $x \in X \subseteq \mathbb{R}^n$ and the network output Y, we define $X_S = \{x_S | x \in X\}$ where $S \subseteq N$; each x_S represents the sample, where variables not in S are masked. If $v(S)$ is set as the entropy of classification $v(S) = H(Y|X_S) = \sum_{x_S} p(x_S) H(Y|X_S = x_S)$,* then $I_{ij}^{(m)} = \mathbb{E}_{S \subseteq N \setminus \{i,j\}, |S|=m} MI(X_i; X_j; Y|X_S)$. The conditional mutual information $MI(X_i; X_j; Y|X_S)$ measures the remaining mutual information[7] between X_i, X_j and Y, when X_S is given. When X_S (with m pixels) is given, $MI(X_i; X_j; Y|X_S)$ can be considered as the benefit to the inference from the interaction between X_i and X_j. Please see the supplementary material for proofs.

Proposition 1 indicates that compared to low-order interactions, high-order interactions are conditioned on larger contexts S, *i.e.* conditioned on more contextual perturbations, thereby suffering more from adversarial perturbations.

Note that in Proposition 1, we set $\hat{v}(S) = H(Y|X_S)$, which is slightly different from setting $v(S) = \log p(y = y^{\text{truth}}|S, x)$ in Section 4.1. Nevertheless, the trend of $v(S)$ can roughly reflect the negative trend of $\hat{v}(S)$, which is discussed in the supplementary material.

Comparison with frequency-based methods. Previous studies [79, 64, 27] explained adversarial perturbations as high-frequency features. Figure 6 (left) shows the difference in the frequency between normal samples and adversarial samples. We find that high-order interactions can better distinguish adversarial examples and normal samples, than the frequency of features, which indicates that the interaction metric can better explain the essential property of adversarial perturbations. Please see the supplementary material for more details.

4.2 Explaining specific interactions encoded by adversarially trained DNNs

Discovery. In order to understand the difference in signal-processing behaviors between adversarially trained DNNs and standard DNNs, we have proven that the overall attacking utility $\mathbb{E}_{x \in \Omega}[\Delta v(N|x)]$ can be approximately decomposed as the sum of attacking utilities on multi-order interactions $\Delta J^{(m)}$, according to Eq. (3). Figure 4 shows the attacking utility of multi-order interactions *i.e.* $\Delta J^{(m)} = \frac{n-1-m}{n(n-1)} \Delta I^{(m)}$.

[7] Note that unlike the bivariate mutual information, $MI(X_i; X_j; Y|X_S)$ can be negative.

We noticed that for standard DNNs, adversarial attacks mainly affected high-order interactions. For adversarially trained DNNs, although high-order interactions were usually sensitive to attacks, low/middle-order interactions took up more attacking utilities. This phenomenon actually can also be observed in Figure 3, which is discussed in the supplementary material. Considering Eq. (3), this indicated that adversarial perturbations towards adversarially trained DNNs penalized both complex features of global collaborations and simple features of local collaborations.

Explanation. We further explain the above phenomena, *i.e.* why high-order interactions in adversarially trained DNNs are more robust to attacks than those in standard DNNs. To this end, we propose the following disentanglement metric for the interaction of a specific order.

$$D^{(m)}=\mathbb{E}_{x\in\Omega}\mathbb{E}_{\substack{i,j\in N\\ i\neq j}}\frac{|I_{ij}^{(m)}(x)|}{\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}|\Delta v(i,j,S|x)|}=\mathbb{E}_{x\in\Omega}\mathbb{E}_{\substack{i,j\in N\\ i\neq j}}\frac{|\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}\Delta v(i,j,S|x)|}{\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}|\Delta v(i,j,S|x)|} \quad (4)$$

The above disentanglement metric examines whether or not the m-order interactions represent discriminative information for a specific category. The high value of $D^{(m)}$ indicates that the m-order interactions relatively purely describe specific categories. In other words, the interactions between (i,j) under different contexts S consistently have the same effects (either positive or negative) on the inference of a specific category. Let us consider the following toy example. When the pair of (i,j) consistently have positive interactions towards a specific category under different contexts, *i.e.* $\forall S\subseteq N\setminus\{i,j\}, |S|=m, \Delta v(i,j,S|x)>0$, then we have $D^{(m)}=1$. This means that the m-th interactions between i and j stably promote the output probability of this category. In contrast, a low value of $D^{(m)}$ indicates that interactions between (i,j) represent diverse categories. *I.e.* given different contexts S, interactions between (i,j) sometimes have positive effects on a specific category, and sometimes have negative effects. Please see the supplementary material for more discussions.

Experiments. Figure 5 compares the interaction disentanglement $D^{(m)}$ of standard DNNs and the disentanglement of adversarially trained DNNs. Interactions of adversarially trained DNNs were more disentangled than those of standard DNNs, especially for low-order interactions. This indicated that low-order interactions in adversarially trained DNNs encoded more category-specific information for inference than low-order interactions in standard DNNs.

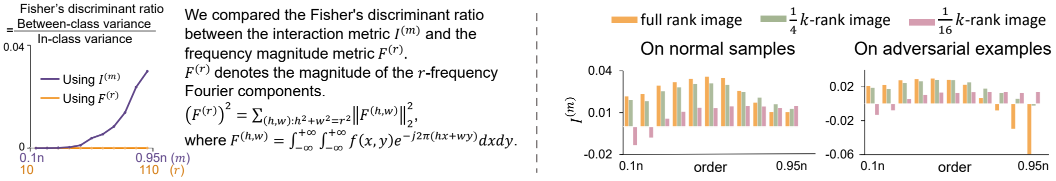
Based on this observation, we could explain the robustness of high-order interactions to attacks in adversarially trained DNNs. Adversarial training learned more category-specific low-order interactions, which boosted the difficulty of attacking high-order interactions. This was because high-order interactions (usually global features) were usually constructed by low-order interactions (usually local features). Let us take the peacock in Figure 2 (right) for example. In this image, low-order interactions (simple features) represented the body of the peacock and the green garden. In this case, it was difficult to attack this image to other categories (*e.g.* the bicycle category) by constructing high-order interactions of the bicycle category using these peacock low-order features. Due to the difficulty of attacking high-order interactions, the adversarial examples towards the adversarially trained DNN had to attack low-order interactions, instead.

4.3 Unifying four existing adversarial defense methods

- **Explaining the attribution-based method of detecting adversarial examples.** Yang et al. [78] proposed a method to use the attribution score of input variables to detect adversarial examples. In order to prove the effectiveness of this method, we define the multi-order Shapley value just like the multi-order interaction, as follows. $\phi^{(m)}(i|x)\overset{\text{def}}{=}\mathbb{E}_{S\subseteq N\setminus\{i\},|S|=m}[v(S\cup\{i\}|x)-v(S|x)]$, where m denotes the order of the Shapley value. Then, we can prove that the attribution score which was used in [78] to detect adversarial exmaples, can be writtern as $\phi^{(n-1)}(i|x)=v(N|x)-v(N\setminus\{i\}|x)$. We further prove that $\phi^{(m)}(i|x)$ can be decomposed into interactions of different orders.

$$v(N|x)-v(\emptyset|x)=\frac{1}{n}\sum_{i\in N}\sum_{m=0}^{n-1}\phi^{(m)}(i|x),\;\; \phi^{(m)}(i|x)=\mathbb{E}_{j\in N\setminus\{i\}}\left[\sum_{k=0}^{m-1}I_{ij}^{(k)}(x)\right]+\phi^{(0)}(i|x) \quad (5)$$

According to the above equation, the overall adversarial effects can also be decomposed into elementary effects on $\phi^{(m)}(i|x)$. $\phi^{(0)}(i|x)=v(i|x)-v(\emptyset|x)$ is usually small and can be ignored, as discussed below Eq. (3). Among Shapley values of all orders, only the $(n-1)$-order component $\phi^{(n-1)}(i|x)$ contains the interactions of the highest order, which are the most sensitive interactions, according to Section 4.1. Therefore, the Shapley value component $\phi^{(n-1)}(i|x)$ is supposed to be



We compared the Fisher's discriminant ratio between the interaction metric $I^{(m)}$ and the frequency magnitude metric $F^{(r)}$.
$F^{(r)}$ denotes the magnitude of the r-frequency Fourier components.
$\left(F^{(r)}\right)^2 = \sum_{(h,w):h^2+w^2=r^2} \left\|F^{(h,w)}\right\|_2^2$,
where $F^{(h,w)} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)e^{-j2\pi(hx+wy)}dxdy$.

Figure 6: (Left) The game-theoretic interaction is a more significant metric to distinguish normal samples and adversarial examples. (Right) Multi-order interactions in images of different ranks.

Table 1: Ratio of adversarial examples whose classification results were corrected.

Standard DNNs	cutout	ours		
		$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$
ResNet-18	7.12%	19.62%	35.55%	**44.14%**
ResNet-50	10.63%	28.53%	45.53%	**52.33%**
DenseNet-161	11.32%	27.16%	46.15%	**56.35%**

Adversarially trained DNNs	cutout	ours		
		$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$
ResNet-18	23.92%	22.46%	33.78%	**39.18%**
ResNet-50	24.27%	23.23%	35.23%	**41.54%**
DenseNet-161	35.65%	24.65%	39.25%	**45.74%**

the most sensitive to adversarial perturbations, which proves the effectiveness of the detection of adversarial examples. Please see the supplementary material for related proofs and discussions.

• **Explaining high recoverability of adversarial examples towards adversarially trained DNNs.** The adversarial recoverability of a DNN is referred to as whether the DNN's adversarial examples can be inverted back to the normal sample by minimizing the classification loss. Specifically, given an adversarial example x^{adv} generated by the untargeted attack [41] via $\max_{\|x^{\text{adv}}-x\|_p \le \epsilon} \ell(x^{\text{adv}}, y^{\text{truth}})$, we conduct a targeted attack [41] on the adversarial sample to invert the classification result back to its ground-truth label, *i.e.* $\min_{\delta'} \ell(x^{\text{adv}} + \delta', y^{\text{truth}})$. Let $\hat{x} = x^{\text{adv}} + \delta'$ denote the recovered sample. If $\|\hat{x} - x\|_2 \le \|x^{\text{adv}} - x\|_2$, it indicates a high recoverability; otherwise, a low recoverability.

We find that ***adversarial examples generated by adversarially trained DNNs usually exhibit higher recoverability than those generated by standard DNNs.*** This phenomenon can be explained by Proposition 1 in Section 4.1, *i.e.* low-order interactions correspond to the mutual information conditioned on smaller adversarial perturbations, thereby suffering less from attacks. Therefore, we can partially explain the high recoverability of adversarial examples towards adversarially trained DNNs. It is because adversarially trained DNNs mainly focus on low-order interactions (see Section 4.2), which leads to high recoverability.

Besides, we conducted experiments on ResNet-18/50 and DenseNet-161 trained on the ImageNet dataset, which also validated the high recoverability of adversarial examples generated on adversarially trained DNNs. Please see the supplementary material for details.

• **Explaining the rank-based method.** Jere et al. [35] discovered that standard DNNs paid certain attention to high-rank (*w.r.t.* the Singular Value Decomposition) features, but adversarial training put more attention on low-rank features. This implied that adversarial examples mainly contained high-rank features.

We used [35] to reduce the SVD rank of an image (either a normal sample or an adversarial example) to $\frac{1}{4}k$-rank image and $\frac{1}{16}k$-rank image, where k denoted the full rank of the image. Figure 6 (right) shows that (1) compared with normal samples, the rank-reducing operation on adversarial examples significantly increased high-order interactions. This indicated our conclusions in Section 4.1 could explain that adversarial perturbations were mainly high-rank components in images. (2) When the rank of images was further reduced from $\frac{1}{4}k$ to $\frac{1}{16}k$, the rank-reducing operation affected both low-order and high-order interactions. However, when we reduce the rank of either normal samples or adversarial examples, they exhibited similar effects on low-order interactions, but exhibited dramatically different effects on high-order interactions. This indicated that **the most discriminative factor of distinguishing adversarial examples from normal samples was high-order interactions, rather than the rank-reducing operation.** In other words, the game-theoretic interaction presents a more essential property of adversarial attacks.

• **Explaining the cutout method.** DeVries and Taylor [18] proposed the cutout method to boost the robustness of DNNs by masking out a square region of the input during training. This operation can be roughly considered as applying a random dropout operation [57] to input images. The dropout operation on input images has been proven to destroy interactions between input variables [83]. To this end, we have further proven that the dropout operation mainly hurts high-order interactions, rather than low-order interactions (please see the proof in the supplementary material). In this way, the cutout method can be considered as removing high-order interactions and pushing DNNs to learn more insensitive low-order interactions. Therefore, the learned DNN is robust to adversarial attacks.

In experiments, we applied the dropout operation to the misclassified adversarial examples, in order to verify the attacking-alleviating influence on the network output of removing high-order interactions. If the removal of high-order interactions could alleviate the effects of adversarial attacks, it would prove the attacking-alleviating influence of removing high-order interactions. Given ResNet-18/50 and DenseNet-161 trained on the ImageNet dataset, we selected normal samples that are correctly classified in the validation set of the ImageNet dataset, and generated adversarial examples with the ℓ_∞ untargeted PGD attack based on [75]. The attack was stopped once it succeeded. We applied the dropout operation with different dropout rates α at the pixel level, and the dropped pixels were filled with the average value of surrounding pixels. For the cutout method, we set the length of the side of the masked square regions as 112, which was half of the side length of the input sample (dropping $\alpha = 25\%$ pixels of the input), following settings in [18]. Table 1 reports the ratio of adversarial examples whose classification results were corrected. When we applied dropout to adversarial examples, the classification accuracy of adversarial examples significantly increased, and even performed better than the cutout method. This indicated that the removal of high-order interactions presented a more essential defense mechanism than the cutout method.

5 Conclusion

In this paper, we have used the multi-order interaction to provide a unified understanding for the success of adversarial attacks and defense. Based on the multi-order interaction, we have explained adversarial attacking as mainly affecting high-order interactions between input variables. Furthermore, we have also explained the success of adversarial training, *i.e.* learning more category-specific low-order interactions to boost the robustness. Besides, we have further provided a unified explanation for some existing adversarial defense methods.

Acknowledgments and Disclosure of Funding

This work is partially supported by the National Nature Science Foundation of China (No. 61906120, U19B2043), Shanghai Natural Science Foundation (21JC1403800,21ZR1434600), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). This work is also partially supported by Huawei Technologies Inc. Prof. Yisen Wang is partially supported by the National Natural Science Foundation of China under Grant 62006153, and Project 2020BD006 supported by PKU-Baidu Fund. Xin Wang is supported by Wu Wen Jun Honorary Doctoral Scholarship, AI Institute, Shanghai Jiao Tong University.

References

[1] Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *ICML*, 2019.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

[3] Yang Bai, Yan Feng, Yisen Wang, Tao Dai, Shu-Tao Xia, and Yong Jiang. Hilbert-based generative defense for adversarial examples. In *ICCV*, 2019.

[4] Yang Bai, Yuyuan Zeng, Yong Jiang, Yisen Wang, Shu-Tao Xia, and Weiwei Guo. Improving query efficiency of black-box adversarial attack. In *ECCV*, 2020.

[5] Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adversarial robustness via channel-wise activation suppressing. In *ICLR*, 2021.

[6] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *ECCV*, 2018.

[7] Akhilan Boopathy, Sijia Liu, Gaoyuan Zhang, Cynthia Liu, Pin-Yu Chen, Shiyu Chang, and Luca Daniel. Proper network interpretability helps adversarial robustness in classification. In *ICML*, 2020.

[8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *SP*, 2017.

[9] Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In *ICML*, 2020.

[10] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017.

[11] Xu Cheng, Chuntung Chu, Yi Zheng, Jie Ren, and Quanshi Zhang. A game-theoretic taxonomy of visual concepts in dnns. *arXiv preprint arXiv:2106.10938*, 2021.

[12] Xu Cheng, Xin Wang, Haotian Xue, Zhengyang Liang, and Quanshi Zhang. A hypothesis for the aesthetic appreciation in neural networks. *arXiv preprint arXiv:2108.02646*, 2021.

[13] Moustapha Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured prediction models. *arXiv preprint arXiv:1707.05373*, 2017.

[14] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

[15] Ian Covert, Scott Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. *NeurIPS*, 2020.

[16] Tianyu Cui, Pekka Marttinen, and Samuel Kaski. Learning global pairwise interactions with bayesian neural networks. *arXiv preprint arXiv:1901.08361*, 2019.

[17] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017.

[18] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[19] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017.

[20] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *ICML*, 2019.

[21] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers' robustness to adversarial perturbations. *Machine Learning*, 107(3):481–508, 2018.

[22] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2): 179–188, 1936.

[23] Ji Gao, Beilun Wang, Zeming Lin, Weilin Xu, and Yanjun Qi. Deepcloak: Masking deep neural network models for robustness against adversarial samples. *arXiv preprint arXiv:1702.06763*, 2017.

[24] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, Ian Goodfellow, and G Brain. The relationship between high-dimensional geometry and adversarial examples. *arXiv preprint arXiv:1801.02774*, 2018.

[25] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[26] Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory*, 28(4):547–565, 1999.

[27] Paula Harder, Franz-Josef Pfreundt, Margret Keuper, and Janis Keuper. Spectraldefense: Detecting adversarial attacks on cnns in the fourier domain. *arXiv preprint arXiv:2103.03000*, 2021.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[29] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NeurIPS*, 2017.

[30] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[31] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. On relating explanations and adversarial examples. In *NeurIPS*, 2019.

[32] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *ICML*, 2018.

[33] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

[34] Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: Axiomatic feature interactions for deep networks. *arXiv preprint arXiv:2002.04138*, 2020.

[35] Malhar Jere, Maghav Kumar, and Farinaz Koushanfar. A singular value perspective on model robustness. *arXiv preprint arXiv:2012.03516*, 2020.

[36] Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *ICLR*, 2019.

[37] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[38] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017.

[39] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[40] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018.

[41] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[42] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *SIGSAC*, 2017.

[43] W James Murdoch, Peter J Liu, and Bin Yu. Beyond word importance: Contextual decomposition to extract interactions from lstms. In *ICLR*, 2018.

[44] Aran Nayebi and Surya Ganguli. Biologically inspired protection of deep networks from adversarial attacks. *arXiv preprint arXiv:1703.09202*, 2017.

[45] Ambar Pal and Rene Vidal. A game theoretic analysis of additive adversarial attacks and defenses. In *NeurIPS*, 2020.

[46] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *EuroS&P*, 2016.

[47] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ASIA-CCS*, 2017.

[48] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[49] Jie Ren, Zhanpeng Zhou, Qirui Chen, and Quanshi Zhang. Learning baseline values for shapley values. *arXiv preprint arXiv:2105.10719*, 2021.

[50] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *In International Journal of Computer Vision*, 115(3):211–252, 2015.

[51] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *arXiv preprint arXiv:2007.08489*, 2020.

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

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[54] Chandan Singh, W James Murdoch, and Bin Yu. Hierarchical interpretations for neural network predictions. In *ICLR*, 2018.

[55] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. *NeurIPS*, 2018.

[56] Daria Sorokina, Rich Caruana, Mirek Riedewald, and Daniel Fink. Detecting statistical interactions with additive groves of trees. In *ICML*, 2008.

[57] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of Machine Learning Research*, 15 (1):1929–1958, 2014.

[58] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017.

[59] Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *ICML*, 2020.

[60] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[61] Qi Tian, Kun Kuang, Kelu Jiang, Fei Wu, and Yisen Wang. Analysis and applications of class-wise robustness in adversarial training. In *KDD*, 2021.

[62] Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. In *ICLR*, 2018.

[63] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2018.

[64] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *CVPR*, 2020.

[65] Xin Wang, Jie Ren, Shuyun Lin, Xiangming Zhu, Yisen Wang, and Quanshi Zhang. A unified approach to interpreting and boosting adversarial transferability. *arXiv preprint arXiv:2010.04055*, 2020.

[66] Xin Wang, Shuyun Lin, Hao Zhang, Yufei Zhu, and Quanshi Zhang. Interpreting attributions and interactions of adversarial attacks. In *ICCV*, 2021.

[67] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019.

[68] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

[69] Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *ICML*, 2018.

[70] Robert J Weber. Probabilistic values for games. *The Shapley Value. Essays in Honor of Lloyd S. Shapley*, pages 101–119, 1988.

[71] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *ICLR*, 2018.

[72] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *ICLR*, 2020.

[73] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020.

[74] Zhirong Wu, Shuran Song, Aditya Khosla, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets for 2.5 d object recognition and next-best-view prediction. *arXiv preprint arXiv:1406.5670*, 2014.

[75] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019.

[76] Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Quanfu Fan, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. Structured adversarial attack: Towards general implementation and better interpretability. In *ICLR*, 2018.

[77] Kaidi Xu, Sijia Liu, Gaoyuan Zhang, Mengshu Sun, Pu Zhao, Quanfu Fan, Chuang Gan, and Xue Lin. Interpreting adversarial examples by activation promotion and suppression. *arXiv preprint arXiv:1904.02057*, 2019.

[78] Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan. ML-LOO: detecting adversarial examples with feature attribution. *arXiv preprint arXiv:1906.03499*, 2019.

[79] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *arXiv preprint arXiv:1906.08988*, 2019.

[80] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[81] 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.

[82] Hao Zhang, Xu Cheng, Yiting Chen, and Quanshi Zhang. Game-theoretic interactions of different orders. *arXiv preprint arXiv:2010.14978*, 2020.

[83] Hao Zhang, Sen Li, Yinchao Ma, Mingjie Li, Yichen Xie, and Quanshi Zhang. Interpreting and boosting dropout from a game-theoretic view. In *ICLR*, 2021.

[84] Hao Zhang, Yichen Xie, Longjie Zheng, Die Zhang, and Quanshi Zhang. Interpreting multivariate interactions in dnns. In *AAAI*, 2021.

[85] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[86] Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In *ICML*, 2019.

A Preliminaries: Shapley values

In this section, in order to help readers understand the metric in the paper, we first revisit the definition of the Shapley value [52], which is widely considered as an unbiased estimation of the numerical importance *w.r.t.* each input variable. In game theory, the complex system is usually represented as a game, where each input variable is taken as a player, and the output of this system is regarded as the total reward of all players. Given a game with multiple players (input variables) $N = \{1, 2, \cdots, n\}$, some players cooperate to pursue a high reward. Thus, the task is to divide the total reward, and fairly assign the divided elementary reward to each individual player. In this way, the elementary reward can be considered as the numerical importance of the corresponding variable to the complex system. Let $2^N \overset{\text{def}}{=} \{S|S \subseteq N\}$ indicate all potential subsets of N. The game $v : 2^N \to \mathbb{R}$ is a function, which estimates the overall reward $v(S)$ earned by each specific subset of players $S \subseteq N$. In this way, the Shapley value, denoted by $\phi(i)$, represents the numerical importance of the player i to the game v.

$$\phi(i) = \sum_{S \subseteq N \setminus \{i\}} \frac{(n - |S| - 1)!|S|!}{n!} \Big[v(S \cup \{i\}) - v(S) \Big]. \tag{6}$$

Weber [70] has proven that the Shapley value is a unique method to fairly allocate overall reward to each player that satisfies following properties.

(1) Linearity property: If two independent games can be merged into one game $u(S) = v(S) + w(S)$, then the Shapley value of the new game also can be merged, *i.e.* $\forall i \in N$, $\phi_u(i) = \phi_v(i) + \phi_w(i)$; $\forall c \in \mathbb{R}$, $\phi_{c \cdot u}(i) = c \cdot \phi_u(i)$.

(2) Nullity property: The dummy player i is defined as a player satisfying $\forall S \subseteq N \setminus \{i\}$, $v(S \cup \{i\}) = v(S) + v(\{i\})$, which indicates that the player i has no interactions with other players in N, $\phi(i) = v(\{i\}) - v(\emptyset)$.

(3) Symmetry property: If $\forall S \subseteq N \setminus \{i, j\}$, $v(S \cup \{i\}) = v(S \cup \{j\})$, then $\phi(i) = \phi(j)$.

(4) Efficiency property: The overall reward can be assigned to all players, $\sum_{i \in N} \phi(i) = v(N) - v(\emptyset)$.

Using Shapley values to explain DNNs. Given a trained DNN and the input with n variables $N = \{1, \cdots, n\}$, we can take the input variables as players, and consider the network output as the reward. The Shapley value $\phi(i)$ of each input variable $i \in N$ is regarded as the importance of the variable i to the network output. Each subset of variables $S \subseteq N$ represents a specific context. $v(S)$ represents the network output, when we keep variables in S unchanged and mask variables in $N \setminus S$ by following settings in [1]. In particular, $v(N)$ refers to the network output *w.r.t.* the entire input N, and $v(\emptyset)$ denotes the output when we mask all variables. Note that for the DNN trained for multi-category classification, $v(S)$ can be taken as an arbitrary dimension of the network output, so as to measure the variable importance to the corresponding category.

B Multi-order interactions and multi-order Shapley values

In this section, we first revisit the definition of the Shapley interaction index [26] and multi-order interactions [83], in order to improve the readability. Then, we provide proofs of extended properties of multi-order interactions, proofs of the relationship between multi-order interactions and multi-order Shapley values, and the proof of Proposition 1 in Section 4.1 of the paper.

B.1 Multi-order interactions

Shapley interaction index. Input variables of a DNN do not contribute to the network output independently. Instead, there are interactions/collaborations between different variables. To this end, the Shapley interaction index [26] measures the influence of a variable on the Shapley value (importance) of another variable. *I.e.* for two variables (i, j), it examines whether the absence/presence of j can change the importance of i. Thus, the Shapley interaction index is defined as the change in the Shapley value (importance) of variable i when the variable j is always present *w.r.t.* the case when j is always absent, as follows.

$$I(i, j) = \tilde{\phi}(i)_{j \text{ always present}} - \tilde{\phi}(i)_{j \text{ always absent}} \tag{7}$$

where $\tilde{\phi}(i)_{j \text{ always present}}$ denotes the Shapley value of the variable i computed under the specific condition that j is always present. $\tilde{\phi}(i)_{j \text{ always absent}}$ is computed under the specific condition that j is always absent. Note that $I(i,j) = I(j,i)$. If $I(i,j) > 0$, it indicates that the presence of the variable j will increase the importance of the variable i. Thus, we consider variables (i,j) have a positive interaction. If $I(i,j) < 0$, it indicates a negative interaction.

Multi-order interactions. Zhang et al. [83] decomposed the Shapley interaction index into interactions of different orders, as follows.

$$I_{ij}^{(m)} = \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[\Delta v(i,j,S)], \tag{8}$$

where $\Delta v(i,j,S) \overset{\text{def}}{=} v(S\cup\{i,j\}) - v(S\cup\{i\}) - v(S\cup\{j\}) + v(S)$. $I_{ij}^{(m)}$ denotes the interaction of the m-th order, which measures the average interaction between variables (i,j) under all contexts consisting of m variables (*e.g.* a visual context consisting of m pixels). For a low order m, $I_{ij}^{(m)}$ reflects the interaction between i and j *w.r.t.* simple contextual collaborations with a few variables. For a high order m, $I_{ij}^{(m)}$ corresponds to the interaction *w.r.t.* complex contextual collaborations with massive variables.

Proofs of new properties of multi-order interactions claimed in the paper. In Section 3 of the paper, we claim that $I_{ij}^{(m)}$ satisfies *linearity, nullity, commutativity, symmetry,* and *efficiency* properties. Proofs are given as follows.

(1) Linearity property: If we merge outputs of two DNNs, $u(S) = w(S) + v(S)$, then, $\forall i,j \in N$, the interaction $I_{ij,u}^{(m)}$ *w.r.t.* the new output u can be decomposed into $I_{ij,u}^{(m)} = I_{ij,w}^{(m)} + I_{ij,v}^{(m)}$.

• *Proof*:

$$\begin{aligned} I_{ij,u}^{(m)} &= \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[\Delta u(S,i,j)] \\ &= \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[\Delta v(i,j,S) + \Delta w(S,i,j)] \\ &= I_{ij,v}^{(m)} + I_{ij,w}^{(m)} \end{aligned}$$

□

(2) Nullity property: The dummy variable $i \in N$ satisfies $\forall S \subseteq N\setminus\{i\}$, $v(S\cup\{i\}) = v(S)+v(\{i\})$. It means that the variable i has no interactions with other variables, *i.e.* $\forall j \in N$, $I_{ij}^{(m)} = 0$.

• *Proof*:

$$\begin{aligned} I_{ij}^{(m)} &= \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[v(S\cup\{i,j\}) - v(S\cup\{i\}) - v(S\cup\{j\}) + v(S)] \\ &= \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}\big[v(S\cup\{j\}\cup\{i\}) - v(S\cup\{j\}) - \big(v(S\cup\{i\}) - v(S)\big)\big] \\ &= \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[v(\{i\}) - v(\{i\})] = 0 \end{aligned}$$

□

(3) Commutativity property: $\forall i,j \in N$, $I_{ij}^{(m)} = I_{ji}^{(m)}$.

• *Proof*:

$$\begin{aligned} I_{ij}^{(m)} &= \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[v(S\cup\{i,j\}) - v(S\cup\{i\}) - v(S\cup\{j\}) + v(S)] \\ &= \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[v(S\cup\{i,j\}) - v(S\cup\{j\}) - v(S\cup\{i\}) + v(S)] \\ &= I_{ji}^{(m)} \end{aligned}$$

□

(4) Symmetry property: If input variables $i,j \in N$ have same cooperations with other variables $\forall S \subseteq N\setminus\{i,j\}$, $v(S\cup\{i\}) = v(S\cup\{j\})$, then they have same interactions, $\forall k \in N\setminus\{i,j\}$, $I_{ik}^{(m)} = I_{jk}^{(m)}$.

• *Proof*:

$$\begin{aligned}
I_{ik}^{(m)} &= \mathbb{E}_{\substack{S\subseteq N\setminus\{i,j\},\\ |S|=m}}[\Delta v(i,k,S)] \\
&= \frac{m!(n-2-m)!}{(n-2)!} \sum_{\substack{S\subseteq N\setminus\{i,k\},\\ |S|=m}} [\Delta v(i,k,S)] \\
&= \frac{m!(n-2-m)!}{(n-2)!} \left(\sum_{\substack{S\subseteq N\setminus\{i,j,k\},\\ |S|=m-1}} [\Delta v(i,k,S\cup\{j\})] + \sum_{\substack{S\subseteq N\setminus\{i,j,k\},\\ |S|=m}} [\Delta v(i,k,S)] \right) \\
&= \frac{m!(n-2-m)!}{(n-2)!} \left(\sum_{\substack{S\subseteq N\setminus\{i,j,k\},\\ |S|=m-1}} [\Delta v(j,k,S\cup\{i\})] + \sum_{\substack{S\subseteq N\setminus\{i,j,k\},\\ |S|=m}} [\Delta v(j,k,S)] \right) \\
&= \mathbb{E}_{\substack{S\subseteq N\setminus\{i,j\},\\ |S|=m}}[\Delta v(j,k,S)] \\
&= I_{jk}^{(m)}
\end{aligned}$$

□

(5) Efficiency property: The output of the DNN can be decomposed into interactions of different orders, $v(N) = v(\emptyset) + \sum_{i\in N}\phi^{(0)}(i) + \sum_{i\in N}\sum_{j\in N\setminus\{i\}}[\sum_{m=0}^{n-2}\frac{n-1-m}{n(n-1)}I_{ij}^{(m)}]$, where $\phi^{(0)}(i) \overset{\text{def}}{=} v(i) - v(\emptyset)$.

• *Proof*:

$$\begin{aligned}
v(N) &= v(\emptyset) + \frac{1}{n}\sum_{i\in N}\sum_{m=0}^{n-1}\phi^{(m)}(i) \\
&= v(\emptyset) + \frac{1}{n}\sum_{i\in N}\phi^{(0)}(i) + \frac{1}{n}\sum_{i\in N}\sum_{m=1}^{n-1}\left[\mathbb{E}_{j\in N\setminus\{i\}}\left[\sum_{k=0}^{k=m-1} I_{ij}^{(k)}\right] + \phi^{(0)}(i)\right] \\
&= v(\emptyset) + \sum_{i\in N}\phi^{(0)}(i) + \frac{1}{n(n-1)}\sum_{i\in N}\sum_{j\in N\setminus\{i\}}\left[\sum_{m=1}^{n-1}\sum_{k=0}^{k=m-1} I_{ij}^{(k)}\right] \\
&= v(\emptyset) + \sum_{i\in N}\phi^{(0)}(i) + \sum_{i\in N}\sum_{j\in N\setminus\{i\}}\left[\sum_{m=0}^{n-2}\frac{n-1-m}{n(n-1)} I_{ij}^{(m)}\right]
\end{aligned}$$

□

B.2 Multi-order Shapley values

In Section 4.3 of the paper, we define multi-order Shapley values. In the supplementary material, this section provides more details about multi-order Shapley values to help readers understand, and also provides proofs of extended properties of multi-order Shapley values.

We decompose the Shapley value $\phi(i)$ into different orders, as follows.

$$\phi(i) = \frac{1}{n}\sum_{m=0}^{n-1}\phi^{(m)}(i), \tag{9}$$

$$\phi^{(m)}(i) = \mathbb{E}_{S\subseteq N\setminus\{i\},|S|=m}[v(S\cup i) - v(S)], \tag{10}$$

where $\phi^{(m)}(i)$ denotes the Shapley value of the m-th order. It measures the importance of the input variable i to the network output with contexts consisting of $m \in \{0,\dots,n-1\}$ variables.

For a low order m, $\phi^{(m)}(i)$ denotes the importance of the variable i, when i cooperates with a few contextual variables for inference. For a high order m, $\phi^{(m)}(i)$ describes the importance of the variable i, which cooperates with massive contextual variables. In particular, $\phi^{(0)}(i) = v(i) - v(\emptyset)$ represents the numerical importance of i without taking into account any contexts.

In addition, we have proven that $\phi^{(m)}(i)$ satisfies following properties.

(1) Linearity property: If we merge the outputs of two DNNs $u(S) = w(S) + v(S)$, then the Shapley values of input variables also can be added, *i.e.* $\forall i \in N$, $\phi_u^{(m)}(i) = \phi_w^{(m)}(i) + \phi_v^{(m)}(i)$.

- *Proof*:

$$\begin{aligned}\phi_u^{(m)}(i) &= \mathbb{E}_{S\subseteq N\setminus\{i\},|S|=m}[u(S\cup\{i\}) - u(S)]\\ &= \mathbb{E}_{S\subseteq N\setminus\{i\},|S|=m}[w(S\cup\{i\}) + v(S\cup\{i\}) - w(S) - v(S)]\\ &= \mathbb{E}_{S\subseteq N\setminus\{i\},|S|=m}[w(S\cup\{i\}) - w(S)] + \mathbb{E}_{S\subseteq N\setminus\{i\},|S|=m}[v(S\cup\{i\}) - v(S)]\\ &= \phi_w^{(m)}(i) + \phi_v^{(m)}(i)\end{aligned}$$

□

(2) Nullity property: An input variable $i \in N$ is considered as a dummy player if $\forall S \subseteq N \setminus \{i\}$, $v(S \cup \{i\}) = v(S) + v(\{i\})$. Thus, the variable i has no interactions with other variables, *i.e.* $\phi^{(m)}(i) = v(\{i\})$.

- *Proof*:

$$\begin{aligned}\phi^{(m)}(i) &= \mathbb{E}_{S\subseteq N\setminus\{i\},|S|=m}[v(S\cup\{i\}) - v(S)]\\ &= \mathbb{E}_{S\subseteq N\setminus\{i\},|S|=m}[v(\{i\})] = v(\{i\})\end{aligned}$$

□

(3) Symmetry property: Given two input variables $i, j \in N$, if these two variables have same cooperations with all other variables $\forall S \subseteq N\setminus\{i,j\}$, $v(S \cup \{i\}) = v(S \cup \{j\})$, then $\phi^{(m)}(i) = \phi^{(m)}(j)$.

- *Proof*:

$$\begin{aligned}\phi^{(m)}(i) &= \mathbb{E}_{S\subseteq N\setminus\{i\},|S|=m}[v(S\cup\{i\}) - v(S)]\\ &= \frac{m!(n-1-m)!}{(n-1)!}\sum_{\substack{S\subseteq N\setminus\{i\},\\ |S|=m}}[v(S\cup\{i\}) - v(S)]\\ &= \frac{m!(n-1-m)!}{(n-1)!}\Bigg(\sum_{\substack{S\subseteq N\setminus\{i,j\},\\ |S|=m-1}}[v(S\cup\{i,j\}) - v(S,j)] + \sum_{\substack{S\subseteq N\setminus\{i,j\},\\ |S|=m}}[v(S\cup\{i\}) - v(S)]\Bigg)\\ &= \frac{m!(n-1-m)!}{(n-1)!}\Bigg(\sum_{\substack{S\subseteq N\setminus\{i,j\},\\ |S|=m-1}}[v(S\cup\{i,j\}) - v(S,i)] + \sum_{\substack{S\subseteq N\setminus\{i,j\},\\ |S|=m}}[v(S\cup\{j\}) - v(S)]\Bigg)\\ &= \mathbb{E}_{S\subseteq N\setminus\{j\},|S|=m}[v(S\cup\{j\}) - v(S)]\\ &= \phi^{(m)}(j)\end{aligned}$$

□

(4) Efficiency property: The overall reward can be assigned to all players, $\frac{1}{n}\sum_{i\in N}\sum_{m=0}^{n-1}\phi^{(m)}(i) = v(N) - v(\emptyset)$.

- *Proof*:

$$v(N) - v(\emptyset) = \sum\nolimits_{i\in N}\phi(i) = \frac{1}{n}\sum\nolimits_{i\in N}\sum\nolimits_{m=0}^{n-1}\phi^{(m)}(i)$$

□

B.3 Relationship between multi-order Shapley values and multi-order interactions

This section provides proofs of the relationship between multi-order Shapley values and multi-order interactions. We have proven that multi-order Shapley values and multi-order interactions satisfy the following *marginal attribution* and *accumulation* properties.

(1) Marginal attribution property: The marginal attribution of the $(m+1)$-th order Shapley values beyond the m-th order is equal to the average interaction of the m-th order between i and all other variables. $\forall i, j \in N, i \neq j$, $\phi^{(m+1)}(i) - \phi^{(m)}(i) = \mathbb{E}_{j\in N\setminus\{i\}}[I_{ij}^{(m)}]$.

• *Proof*:

$$
\begin{aligned}
\phi^{(m+1)}(i)-\phi^{(m)}(i) =& \mathbb{E}_{\substack{S'\subseteq N\setminus\{i\}\\|S'|=m+1}}\left[v(S'\cup\{i\})-v(S')\right]-\mathbb{E}_{\substack{S\subseteq N\setminus\{i\}\\|S|=m}}\left[v(S\cup\{i\})-v(S)\right]\\
=&\mathbb{E}_{\substack{S\subseteq N\setminus\{i\}\\|S|=m}}\Big[\mathbb{E}_{j\in N\setminus(S\cup\{i\})}\big[v(S\cup\{j\}\cup\{i\})\\
&-v(S\cup\{j\})\big]\Big]-\mathbb{E}_{\substack{S\subseteq N\setminus\{i\}\\|S|=m}}\left[v(S\cup\{i\})-v(S)\right]\\
=&\mathbb{E}_{\substack{S\subseteq N\setminus\{i\}\\|S|=m}}\Big[\mathbb{E}_{j\in N\setminus(S\cup\{i\})}\big[v(S\cup\{j\}\cup\{i\})-v(S\cup\{j\})-v(S\cup\{i\})+v(S)\big]\Big]\\
=&\mathbb{E}_{\substack{S\subseteq N\setminus\{i\}\\|S|=m}}\mathbb{E}_{j\in N\setminus(S\cup\{i\})}\left[\Delta v(i,j,S)\right]\\
=&\mathbb{E}_{j\in N\setminus\{i\}}\mathbb{E}_{\substack{S\subseteq N\setminus\{i,j\}\\|S|=m}}\left[\Delta v(i,j,S)\right]\\
=&\mathbb{E}_{j\in N\setminus\{i\}}[I_{ij}^{(m)}]
\end{aligned}
$$

□

(2) Accumulation property: The m-th ($m>0$) order Shapley value of the variable $i\in N$ can be decomposed into interactions of lower orders, $\phi^{(m)}(i)=\mathbb{E}_{j\in N\setminus\{i\}}[\sum_{k=0}^{m-1}I_{ij}^{(k)}]+\phi^{(0)}(i)$.

• *Proof*:

$$
\begin{aligned}
\phi^{(m)}(i)&=\phi^{(m)}(i)-\phi^{(m-1)}(i)+\phi^{(m-1)}(i)-\phi^{(m-2)}(i)+\cdots-\phi^{(0)}(i)+\phi^{(0)}(i)\\
&=\mathbb{E}_{j\in N\setminus\{i\}}[I_{ij}^{(m-1)}]+\mathbb{E}_{j\in N\setminus\{i\}}[I_{ij}^{(m-2)}]+\cdots+\mathbb{E}_{j\in N\setminus\{i\}}[I_{ij}^{(0)}]+\phi^{(0)}(i)\\
&=\mathbb{E}_{j\in N\setminus\{i\}}\Big[\sum\nolimits_{k=0}^{k=m-1}I_{ij}^{(k)}\Big]+\phi^{(0)}(i)
\end{aligned}
$$

□

B.4 Equivalence between the multi-order interaction and the mutual information

Proposition 1 in Section 4.1 of the paper shows the equivalence between the multi-order interaction and the mutual information. In the supplementary material, this section provides the proof of this proposition.

When a DNN outputs a probability distribution, we prove that the interaction between input variables can be represented in the form of mutual information. Without loss of generality, let us take the image classification task for example. Let $x\in X\subseteq\mathbb{R}^n$ denote an input image of the DNN. x_i denotes the i-th pixel, and $X_i=\{x_i\}$. $\forall S\subseteq N$, we define $X_S=\{x_S|x\in X\}$; each x_S represents the image, where pixels in S remain unchanged, and other pixels $j\in N\setminus S$ are masked following settings of [1]. Let $y\in Y=\{y^1,\cdots,y^C\}$ denote the network prediction. In this way, given x_S as the input, $p(y|x_S)$ denotes the output probability of the DNN. Let us set $v(S)=H(Y|X_S)=\sum_{x_S}p(x_S)H(Y|X_S=x_S)$, which measures the entropy of y given the input x_S. Then we prove that

$$I_{ij}^{(m)}=\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}MI(X_i;X_j;Y|X_S) \tag{11}$$

• *Proof*:

$$
\begin{aligned}
I_{ij}^{(m)}&=\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}\Big[v(S\cup\{i,j\})-v(S\cup\{i\})-v(S\cup\{j\})+v(S)\Big]\\
&=\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}\Big[-H(Y|X_S,X_i,X_j)+H(Y|X_S,X_i)+H(Y|X_S,X_j)-H(H|X_S)\Big]\\
&=\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}\Big[H(Y|X_S,X_j)-H(Y|X_S,X_j,X_i)+H(Y|X_S,X_i)-H(Y|X_S)\Big]\\
&=\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}\Big[MI(X_i;Y|X_S,X_j)-MI(X_i;Y|X_S)\Big]\\
&=\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}\Big[MI(X_i;X_j;Y|X_S)\Big]
\end{aligned}
$$

□

The conditional mutual information $MI(X_i;X_j;Y|X_S)$ measures the remaining mutual information between X_i,X_j and Y when X_S is given. Note that unlike the bivariate mutual information, $MI(X_i;X_j;Y|X_S)$ can be negative. When X_S (each $x_S\in X_S$ containing m pixels) is given, we can roughly understand the conditional mutual information $MI(X_{\{i,j\}};Y|X_S)$ as the additional benefits from X_i and X_j to classification. We prove that $MI(X_{\{i,j\}};Y|X_S)$ can be decomposed into the exclusive benefits of X_i (*i.e.* $MI(X_i;Y|X_j,X_S)$), the exclusive benefits of X_j (*i.e.* $MI(X_j;Y|X_i,X_S)$), and the benefit shared by X_i and X_j (*i.e.* $MI(X_i;X_j;Y|X_S)$). Thus, $MI(X_i;X_j;Y|X_S)$ can be considered as the benefits from the interaction between X_i and X_j.

$$MI(X_{\{i,j\}};Y|X_S) = MI(X_i;Y|X_j,X_S) + MI(X_j;Y|X_i,X_S) + MI(X_i;X_j;Y|X_S) \quad (12)$$

• *Proof*:

$$\begin{aligned}
\text{right} &= MI(X_i;Y|X_j,X_S) + MI(X_j;Y|X_i,X_S) + MI(X_i;X_j;Y|X_S)\\
&= MI(X_i;Y|X_j,X_S) + MI(X_j;Y|X_i,X_S) + MI(X_i;Y|X_S) - MI(X_i;Y|X_j,X_S)\\
&= MI(X_i;Y|X_j,X_S) + MI(X_i;Y|X_S)\\
&= \sum_{x_i,x_j,x_S,y} p(x_i,x_j,x_S,y)\log\frac{p(x_j,y|x_i,x_S)}{p(x_j|x_i,x_S)p(y|x_i,x_S)} + \sum_{x_i,x_S,y} p(x_i,x_S,y)\log\frac{p(x_i,y|x_S)}{p(x_i|x_S)p(y|x_S)}\\
&= \sum_{x_i,x_j,x_S,y} p(x_i,x_j,x_S,y)\log\frac{p(x_j,y|x_i,x_S)}{p(x_j|x_i,x_S)p(y|x_i,x_S)} + \sum_{x_i,x_j,x_S,y} p(x_i,x_j,x_S,y)\log\frac{p(x_i,y|x_S)}{p(x_i|x_S)p(y|x_S)}\\
&= \sum_{x_i,x_j,x_S,y} p(x_i,x_j,x_S,y)\log\frac{p(x_j,y|x_i,x_S)p(x_i,y|x_S)}{p(x_j|x_i,x_S)p(y|x_i,x_S)p(x_i|x_S)p(y|x_S)}\\
&= \sum_{x_i,x_j,x_S,y} p(x_i,x_j,x_S,y)\log\frac{p(x_j,y|x_i,x_S)p(x_i,y|x_S)p(x_i,x_S)p(x_S)}{p(x_j|x_i,x_S)p(y|x_i,x_S)p(x_i|x_S)p(y|x_S)p(x_i,x_S)p(x_S)}\\
&= \sum_{x_i,x_j,x_S,y} p(x_i,x_j,x_S,y)\log\frac{p(x_i,x_j,x_S,y)p(x_i,x_S,y)}{p(x_j|x_i,x_S)p(x_i,x_S,y)p(x_i|x_S)p(y|x_S)p(x_S)}\\
&= \sum_{x_i,x_j,x_S,y} p(x_i,x_j,x_S,y)\log\frac{p(x_i,x_j,x_S,y)}{p(x_j|x_i,x_S)p(x_i|x_S)p(y|x_S)p(x_S)}\\
&= \sum_{x_i,x_j,x_S,y} p(x_i,x_j,x_S,y)\log\frac{p(x_i,x_j,y|x_S)}{p(x_i,x_j|x_S)p(y|x_S)}\\
&= MI(X_{\{i,j\}};Y|X_S) = \text{left}
\end{aligned}$$

□

C Related works about interactions

In Section 2 of the paper, we have discussed related works about understandings of adversarial attacks, defense, and robustness. Due to the page limit, we discuss related works about interactions in this section of the supplementary material. Unlike previous studies about interactions, we firstly use the interaction to explain adversarial perturbations and robustness, and provide a unified view to understand existing defense methods.

Interactions between input variables of a DNN have been widely investigated in recent years. In game theory, Grabisch and Roubens [26] and Lundberg et al. [39] proposed and used the Shapley interaction index based on Shapley values [52]. Covert et al. [15] investigated the relationship between the Shapley value and the mutual information. Sorokina et al. [56] measured the interaction of multiple input variables in an additive model. Tsang et al. [62] calculated interactions of weights in a DNN. Wang et al. [65] applied the interaction of adversarial perturbations to understand adversarial transferabilitiy. Murdoch et al. [43], Singh et al. [54], and Jin et al. [36] used the contextual decomposition (CD) technique to extract variable interactions. Cui et al. [16] proposed a non-parametric probabilistic method to measure interactions using a Bayesian neural network. Janizek et al. [34] extended the Integrated Gradients method [58] to explain pairwise feature interactions in DNNs. Sundararajan et al. [59] defined the Shapley-Taylor index to measure interactions over binary features. In comparison, we novelly use the multi-order interaction to understand the detailed interaction behaviors *w.r.t.* adversarial attacks, which enables us to explain adversarial examples and adversarial training.

D Information reflected by $\Delta I^{(m)}$

In section 4.1 of the paper, we propose the metric $I^{(m)} = \mathbb{E}_{x\in\Omega}\mathbb{E}_{i,j}[I_{ij}^{(m)}(x)]$, and $\Delta I^{(m)} \overset{\text{def}}{=} I_{\text{nor}}^{(m)} - I_{\text{adv}}^{(m)}$, which measures the difference in interactions between normal samples and adversarial examples. In this section of the supplementary material, we prove the following property of the metric $\Delta I^{(m)}$.

$$\Delta I^{(m)} = \mathbb{E}_{x\in\Omega}\mathbb{E}_{i,j}[\Delta I_{ij}^{(m)}(x)] \tag{13}$$

where $\Delta I_{ij}^{(m)}(x) = I_{ij}^{(m)}(x) - I_{ij}^{(m)}(x^{\text{adv}})$.

• *Proof*:

$$\begin{aligned}\Delta I^{(m)} &= I_{\text{nor}}^{(m)} - I_{\text{adv}}^{(m)}\\ &= \mathbb{E}_{x\in\Omega_{\text{nor}}}\mathbb{E}_{i,j}[I_{ij}^{(m)}(x)] - \mathbb{E}_{x\in\Omega_{\text{adv}}}\mathbb{E}_{i,j}[I_{ij}^{(m)}(x)]\\ &= \mathbb{E}_{x\in\Omega_{\text{nor}}}\mathbb{E}_{i,j}[I_{ij}^{(m)}(x)] - \mathbb{E}_{x\in\Omega_{\text{nor}}}\mathbb{E}_{i,j}[I_{ij}^{(m)}(x+\Delta x)]\\ &= \mathbb{E}_{x\in\Omega_{\text{nor}}}\mathbb{E}_{i,j}[I_{ij}^{(m)}(x) - I_{ij}^{(m)}(x+\Delta x)]\\ &= \mathbb{E}_{x\in\Omega_{\text{nor}}}\mathbb{E}_{i,j}[I_{ij}^{(m)}(x) - I_{ij}^{(m)}(x^{\text{adv}})]\\ &= \mathbb{E}_{x\in\Omega_{\text{nor}}}\mathbb{E}_{i,j}[\Delta I_{ij}^{(m)}(x)]\end{aligned}$$

□

E Relationship between $\hat{v}(S)$ and $v(S)$

In Section 4.1 of the main paper, we claim that the trend of $v(S)$ can roughly reflect the negative trend of $\hat{v}(S)$. In this section, we discuss the negative correlation between $\hat{v}(S)$ and $v(S)$.

According to Proposition 1, $\hat{v}(S) = H(Y|X_S)$ denotes the entropy of the classification probability given variables in S of the image x. Thus, $\hat{v}(S)$ measures the uncertainty of the prediction. If the model prediction is correct and confident, *i.e.* the value of $v(S) = \log p(y = y^{\text{truth}}|x, S)$ is large, then the uncertainty $\hat{v}(S)$ is very low. In comparison, if the model prediction is correct but with a small value of $v(S)$, then the uncertainty is large, yielding a large value of $\hat{v}(S)$. Therefore, the trend of $v(S)$ can roughly reflect the negative trend of $\hat{v}(S)$ when the model prediction is correct.

F More discussions about the disentanglement metric

This section provides more discussions about the disentanglement metric in Eq. (4) of the main paper. The motivation of the disentanglement is to measure the discrimination power of interactions of a specific order, as discussed in Section 4.2 of the paper. According to the efficiency property of the multi-order interactions, the model output can be decomposed into the weighted sum of massive interaction components $\Delta v(i, j, S|x)$:

$$v(N|x)= v(\emptyset|x) + \sum_{i\in N}\phi^{(0)}(i|x) + \sum_{i,j\in N, i\neq j}\sum_{m=0}^{n-2}\frac{n-1-m}{n(n-1)}\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[\Delta v(i,j,S)] \tag{14}$$

In this way, if interaction components of a certain order m are all positive (or negative) and do not conflict with each other, it indicates that these components jointly promote or suppress the model output, showing a strong discrimination power. Otherwise, if some interaction components are positive and others are negative, their effects on the model output will be eliminated. In this case, the discrimination power of interactions is poor.

Therefore, we design the disentanglement metric $D^{(m)}$ in Eq. (4) to model the above phenomenon. The physical meaning of this metric is shown in the following equation.

$$D^{(m)} = \mathbb{E}_{x\in\Omega}\mathbb{E}_{i,j\in N,i\neq j}\frac{\overbrace{|\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}\Delta v(i,j,S|x)|}^{\text{the strength of the average utility of all components}}}{\mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}\underbrace{|\Delta v(i,j,S|x)|}_{\text{the strength of each component}}} \tag{15}$$

The numerator measures the strength of the average utility of all interaction components between (i,j) under different contexts S. The dominator represents the average strength of each interaction component. If $D^{(m)}$ approximates to 1, then it indicates that almost all interaction components have similar effects (either positive or negative) on the model output. If $D^{(m)}$ approximates to 0, then it shows that most interaction components conflict with each other and are eliminated. Therefore, the disentanglement $D^{(m)}$ measures the discrimination power of interactions.

G The attribution-based method of detecting adversarial examples

In Section 4.3 of the paper, we claim that the attribution score used in [78] to detect adversarial exmaples can be writtern as $\phi^{(n-1)}(i|x)$. In the supplementary material, this section provides proofs for this claim.

Yang et al. [78] proposed an attribution-based method to detect adversarial examples, which used the attribution score of input variables. The attribution score of the variable i in [78] is defined as

$$\phi(x)_i := f(x)_c - f(x_{(i)})_c, \;\; \text{where} \;\; c = \arg\max_{j\in C} f(x)_j \tag{16}$$

where x denotes the original input sample, and $x_{(i)}$ denotes the input sample with the i-th variable masked by 0. $f(x)_c$ denotes the network output of the c-th category. Actually, $f(x)_c$ can also be written as $v(N|x)$, and $f(x_{(i)})_c$ can be written as $v(N\setminus\{i\}|x)$. Thus, the attribution score can be represented as $v(N|x) - v(N\setminus\{i\}|x)$. We prove that $v(N|x) - v(N\setminus\{i\}|x) = \phi^{(n-1)}(i|x)$.

• *Proof*:

$$\begin{aligned} v(N|x) - v(N\setminus\{i\}|x) &= v((N\setminus\{i\})\cup\{i\}|x) - v(N\setminus\{i\}|x) \\ &= v(S\cup\{i\}|x) - v(S|x) \qquad \% \; S \overset{\text{def}}{=} N\setminus\{i\}, |S| = n-1 \\ &= \mathbb{E}_{S\subseteq N\setminus\{i\},|S|=n-1}\left[v(S\cup\{i\}|x) - v(S|x)\right] \\ &= \phi^{(n-1)}(i|x) \end{aligned}$$

□

According to the *accumulation property* of multi-order Shapley values and interactions, we have $\phi^{(n-1)}(i|x) = \mathbb{E}_{j\in N\setminus\{i\}}\left[\sum_{m=0}^{n-2} I_{ij}^{(m)}\right] + \phi^{(0)}(i|x)$. This indicates that $\phi^{(n-1)}(i|x)$ contains the interaction components with the highest order ($m = n-2$), which are not included in Shapley values with orders lower than $n-1$. Section 4.1 of the paper has pointed that high-order interactions are the most sensitive to adversarial perturbations, thereby enabling the detection of adversarial examples.

H Effectiveness of the dropout method to alleviate adversarial utilities

In Section 4.3 of the paper, we claim that the dropout operation mainly hurts high-order interactions, rather than low-order interactions. In this section, we theoretically prove such effects of the dropout operation.

Given the input sample $x \in \mathbb{R}^n$ and the dropout rate α, let $\mathcal{K} = \{K|K \subset N, |K| = \lfloor(1-\alpha)n\rfloor\}$ denote all possible sets of remained variables after the dropout operation. Let $v^{\alpha}(N|x) = \mathbb{E}_{K\in\mathcal{K}}[v(K|x)]$ denote the average network output among all inputs after the dropout operation with rate α. According to the efficiency property of the multi-order interaction, we have

$$v(N|x) = v(\emptyset|x) + \sum_{i\in N}\phi^{(0)}(i|x) + \sum_{i\neq j\in N}\sum_{m=0}^{n-2}\frac{n-1-m}{n(n-1)}I_{ij,N}^{(m)}(x) \tag{17}$$

where $I^{(m)}_{ij,N}(x)$ denotes the m-order interaction between variables (i,j) of the input x with all variables N. Similarly,

$$v(K|x) = v(\emptyset|x) + \sum_{i\in K}\phi^{(0)}(i|K,x) + \sum_{i\neq j\in K}\sum_{m=0}^{k-2}\frac{k-1-m}{k(k-1)}I^{(m)}_{ij,K}(x) \tag{18}$$

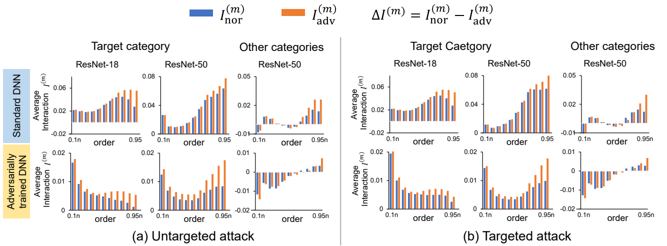
where $k = |K| = \lfloor(1-\alpha)n\rfloor$. Thus,

$$\begin{aligned}
v^{\alpha}(N|x) &= \mathbb{E}_{K\in\mathcal{K}}[v(K|x)] \\
&= \mathbb{E}_{K\in\mathcal{K}}\left[v(\emptyset|x) + \sum_{i\in K}\phi^{(0)}(i|K,x) + \sum_{i\neq j\in K}\sum_{m=0}^{k-2}\frac{k-1-m}{k(k-1)}I^{(m)}_{ij,K}(x)\right] \\
&= v(\emptyset|x) + \mathbb{E}_{K\in\mathcal{K}}\left[\sum_{i\in K}(v(i|x)-v(\emptyset|x))\right] + \mathbb{E}_{K\in\mathcal{K}}\left[\sum_{i\neq j\in K}\sum_{m=0}^{k-2}\frac{k-1-m}{k(k-1)}I^{(m)}_{ij,K}(x)\right] \\
&= v(\emptyset|x) + \mathbb{E}_{\substack{K\subset N\\k=(1-\alpha)n}}\left[\sum_{i\in K}\phi^{(0)}(i|x)\right] + \mathbb{E}_{\substack{K\subset N\\k=(1-\alpha)n}}\left[\sum_{i\neq j\in K}\sum_{m=0}^{k-2}\frac{k-1-m}{k(k-1)}I^{(m)}_{ij,K}(x)\right] \\
&= v(\emptyset|x) + (1-\alpha)\sum_{i\in N}\phi^{(0)}(i|x) + \mathbb{E}_{\substack{K\subset N\\k=(1-\alpha)n}}\left[\sum_{i\neq j\in K}\sum_{m=0}^{k-2}\frac{k-1-m}{k(k-1)}I^{(m)}_{ij,K}(x)\right] \\
&= v(\emptyset|x) + (1-\alpha)\sum_{i\in N}\phi^{(0)}(i|x) + \mathbb{E}_{\substack{K\subset N\\k=(1-\alpha)n}}\left[\sum_{i\neq j\in K}\sum_{m=0}^{k-2}\frac{k-1-m}{k(k-1)}\mathbb{E}_{\substack{S\subseteq K\setminus\{i,j\}\\|S|=m}}[\Delta v(i,j,S|x)]\right] \\
&= v(\emptyset|x) + (1-\alpha)\sum_{i\in N}\phi^{(0)}(i|x) + \frac{k(k-1)}{n(n-1)}\sum_{i\neq j\in N}\sum_{m=0}^{k-2}\frac{k-1-m}{k(k-1)}\mathbb{E}_{\substack{K\subset N\\k=(1-\alpha)n}}\mathbb{E}_{\substack{S\subseteq K\setminus\{i,j\}\\|S|=m}}[\Delta v(i,j,S|x)] \\
&= v(\emptyset|x) + (1-\alpha)\sum_{i\in N}\phi^{(0)}(i|x) + \sum_{i\neq j\in N}\sum_{m=0}^{k-2}\frac{k-1-m}{n(n-1)}\mathbb{E}_{\substack{K\subset N\\k=(1-\alpha)n}}\mathbb{E}_{\substack{S\subseteq K\setminus\{i,j\}\\|S|=m}}[\Delta v(i,j,S|x)] \\
&= v(\emptyset|x) + (1-\alpha)\sum_{i\in N}\phi^{(0)}(i|x) + \sum_{i\neq j\in N}\sum_{m=0}^{k-2}\frac{k-1-m}{n(n-1)}\mathbb{E}_{\substack{S\subset N\setminus\{i,j\}\\|S|=m}}[\Delta v(i,j,S|x)] \\
&= v(\emptyset|x) + (1-\alpha)\sum_{i\in N}\phi^{(0)}(i|x) + \sum_{i\neq j\in N}\sum_{m=0}^{k-2}\frac{k-1-m}{n(n-1)}I^{(m)}_{ij,N}(x)
\end{aligned} \tag{19}$$

Thus, the change in the network output caused by the dropout operation can be represented as follows,

$$\begin{aligned}
v(N|x)-v^{\alpha}(N|x) &= \alpha\sum_{i\in N}\phi^{(0)}(i,x) + \sum_{i\neq j\in N}\left[\sum_{m=0}^{n-2}\frac{n-1-m}{n(n-1)}I^{(m)}_{ij,N}(x) - \sum_{m=0}^{k-2}\frac{k-1-m}{n(n-1)}I^{(m)}_{ij,N}(x)\right] \\
&= \alpha\sum_{i\in N}\phi^{(0)}(i,x) + \sum_{i\neq j\in N}\sum_{m=0}^{k-2}\frac{n-k}{n(n-1)}I^{(m)}_{ij,N}(x) + \sum_{i\neq j\in N}\sum_{m=k-1}^{n-2}\frac{n-1-m}{n(n-1)}I^{(m)}_{ij,N}(x) \\
&= \alpha\sum_{i\in N}\phi^{(0)}(i,x) + \frac{\alpha}{n-1}\sum_{i\neq j\in N}\sum_{m=0}^{k-2}I^{(m)}_{ij,N}(x) + \underbrace{\sum_{i\neq j\in N}\sum_{m=k-1}^{n-2}\frac{n-1-m}{n(n-1)}I^{(m)}_{ij,N}(x)}_{\text{high-order interactions}}
\end{aligned} \tag{20}$$

According to Eq. (20), the dropout operation removes all high-order interactions ($m>(1-\alpha)n-2$), while slightly affects low-order interactions. Thus, the dropout operation can remove sensitive interaction components of the DNN, thereby reducing the attacking utility of perturbations and correcting the network output.



(a) Untargeted attack

(b) Targeted attack

Figure 7: Interactions *w.r.t.* the target category and other categories.

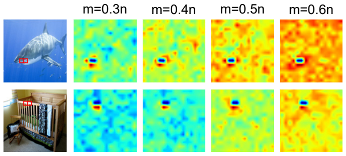


Figure 8: Contexts of the m-order interaction $I_{ij}^{(m)}$ in normal samples of standard ResNet-50.

I More experimental results

I.1 More visualization results based on ResNet-50

The method in [83] could be directly extended to visualize salient concepts of the multi-order interaction without much change. For a specific order m, we visualized all contexts $\{S\}$ of $I^{(m)}(i,j)$ for each pair (i,j) with top 10% interaction strengths. We exclusively visualized contexts $\{S\}$ that boosted the strength of $I^{(m)}(i,j)$, *i.e.* $\Delta v(S,i,j) \cdot I^{(m)}(i,j) > 0$. Let $\text{map}(S) \in \{0,1\}^n$ denote the map corresponding to the context S. If the k-th pixel was contained in S, then $\text{map}_k(S) = 1$; otherwise, $\text{map}_k(S) = 0$. In this way, we visualized the weighted average contexts *w.r.t.* pixels (i,j) as $\sum_{S\subseteq N, |S|=m} |\Delta v(S,i,j)| \cdot \text{map}(S)$.

Figure 2 (right) in the paper has shown visualization results on ResNet-18. Here, we presented more visualization results of interaction contexts based on ResNet-50 [28] in Figure 8. As Figure 8 shows, low-order interactions usually represented simple features of local collaborations, and high-order interactions usually reflected complex features of global collaborations.

I.2 More discussions about the difference between standard DNNs and adversarially trained DNNs in Figure 3 of the main paper

In Figure 3 of the main paper, it seems that standard DNNs and adversarially trained DNNs are similar under the metric $\Delta I^{(m)}$, but Figure 3 does reflect the significant difference of interaction behaviors between standard DNNs and adversarially trained DNNs. Figure 3 shows that although high-order interactions are penalized in both standard and adversarially trained DNNs, high-order interactions in the standard DNN decreased much more than those in the adversarially trained DNN.

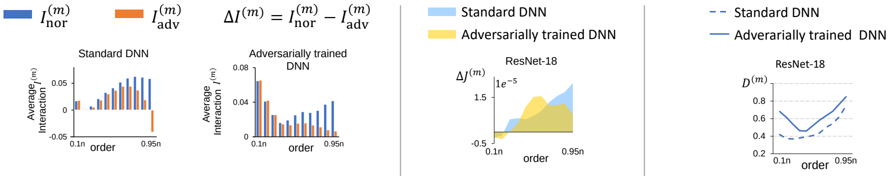
More specifically, we define the metric $\frac{|\Delta I^{(m)}|}{\sum_{m'} |\Delta I^{(m')}|}$ to compare the change of multi-order interactions in different DNNs, where $\Delta I^{(m)} = I_{\text{ori}}^{(m)} - I_{\text{adv}}^{(m)}$ measures the difference of the m-th order interaction between normal samples and adversarial examples. $\sum_{m'} |\Delta I^{(m')}|$ is the total strength of differences of multi-order interactions, which is used for normalization. Table 2 shows that the interaction behaviors in standard DNNs and adversarially trained DNNs were quite different. In the standard ResNet-18, low-order interactions were penalized slightly, and high-order interactions dropped a lot. In comparison, in the adversarially trained ResNet-18, low-order interactions and middle-order interactions were penalized more than those in the standard ResNet-18, while high-order interactions were less penalized than those in the standard ResNet-18. Therefore, the adversarially trained DNN was supposed to exhibit a larger value of $\frac{|\Delta I^{(m)}|}{\sum_{m'} |\Delta I^{(m')}|}$ for low-order interactions (with small m) than the standard DNN. In comparison, the adversarially trained DNN exhibited a smaller value of $\frac{|\Delta I^{(m)}|}{\sum_{m'} |\Delta I^{(m')}|}$ for high-order interactions (with large m) than the standard DNN.

I.3 Discussions about attacking utilities of multi-order interactions.

Figure 4 of the paper shows that low-order interactions in adversarially trained DNNs usually have more attacking utilities. Note that the attacking utility on high/middle-order interactions may make $\Delta J^{(m)}$ on low-order interactions negative, as a trade-off. According to Eq. (3), the sum of attacking utilities on low-order, middle-order, and high-order interactions is a constant ($\Delta v(N|x)$). Therefore,

Table 2: Interaction behaviors in standard DNNs and adversarially trained DNNs are quite different *w.r.t.* the metric $\frac{|\Delta I^{(m)}|}{\sum_{m'} |\Delta I^{(m')}|}$.

m	$0.1n$	$0.2n$	$0.3n$	$0.4n$	$0.5n$	$0.6n$	$0.7n$	$0.8n$	$0.85n$	$0.9n$	$0.95n$
$\frac{\|\Delta I^{(m)}\|}{\sum_{m'} \|\Delta I^{(m')}\|}$ in standard ResNet-18	0.003	0.005	0.008	0.014	0.021	0.040	0.066	0.103	**0.138**	**0.219**	**0.383**
$\frac{\|\Delta I^{(m)}\|}{\sum_{m'} \|\Delta I^{(m')}\|}$ in adversarially trained ResNet-18	**0.016**	**0.023**	**0.055**	**0.082**	**0.095**	**0.100**	**0.100**	**0.108**	0.117	0.144	0.160



(a)The multi-order interaction $I^{(m)}_{\text{nor}}$ and $I^{(m)}_{\text{adv}}$ based on ResNet-18.

(b) Distribution of compositional attacking utilities $\{\Delta J^{(m)}\}$.

(c) The interaction disentanglement $D^{(m)}$.

Figure 9: Experimental results on ResNet-18 when setting the masked variables to zero.

the penalization of high-order interactions may also cause the increase of low-order interactions to some extent. In this way, there are two effects on low-order interactions. First, adversarial attacks penalize low-order interactions. Second, the penalization of high-order interactions also boosts low-order interactions as a side effect. In such a trade-off, low-order interactions decrease in most DNNs. In this way, low-order interactions are increased in a few special cases, which can be explained by the above analysis.

I.4 More details about the approximation method of computing interactions

To reduce the computational cost, we applied the sampling-based approximation method in [83] and did not compute interactions at the pixel-wise level. Instead, we split the image into 16 × 16 grids, and took each grid as a single input variable, thereby $n = 256$. Then, we randomly sampled 200 pairs of grids (i, j), and for each pair of grids and each order m, we sampled contexts S s.t. $|S| = m$ for 100 times to approximate the interaction. Then we computed the m-th order interactions based on Eq. (2) in the main paper by using the sampled pairs of grids and contexts.

I.5 More details about the setting of masking in the computation of interactions

In the main paper, we measured the interaction $I^{(m)}$ by setting $v(S|x) = \log p(y = y^{\text{truth}}|\text{given variables in } S \text{ in the input } x \text{ and mask variables in } N \setminus S)$. Variables not in S were set to the average value over different input samples following settings in [1], to represent their absence. In this section, we conducted experiments by using setting masked variables to zero, to verify that the effects of the choice of masking on the results did not affect our conclusions. Figure 9 shows that the choice of masking did not affect the conclusion that adversarial attacks mainly affect high-order interactions.

I.6 Analysis about interactions *w.r.t.* the target category and other categories

In the main paper, experimental results on the ground-truth category have verified our conclusions that adversarial attacks mainly affect high-order interactions. In this section of the supplementary material, we show more results on the target category and other categories, which also verified our conclusions.

In the targeted attack, the target category was referred to as the target label $y^{\text{target}} \neq y^{\text{truth}}$. In the untargeted attack, we considered the misclassified category $y^{\text{pred}} \neq y^{\text{truth}}$ as the target category. Given each input image, we measured the interaction *w.r.t.* the output of its target category by setting $v(S|x) = \log p(y = y^{\text{target}}|S, x)$. Besides, we also measured the interactions *w.r.t.* outputs of other categories, except the ground-truth category. Considering the softmax operation $p(y = y^{\text{truth}}|S, x) =$

Table 3: The close relationship between the attacking strength and changes of high-order interactions.

Attacking strength (iteration number)	$\Delta I^{(0.7n)}$	$\Delta I^{(0.8n)}$	$\Delta I^{(0.85n)}$	$\Delta I^{(0.9n)}$	$\Delta I^{(0.95n)}$
4	0.0518	0.0990	0.1411	0.1944	**0.2274**
8	0.0789	0.1741	0.2399	0.3118	**0.3632**
16	0.1113	0.2565	0.3342	0.4157	**0.4657**
32	0.1512	0.3338	0.4391	0.5480	**0.6085**
Pearson correlation coefficient	0.9799	0.9603	0.9635	0.9667	0.9634

$\frac{\exp(h_{y^{\text{truth}}}(S|x))}{\sum_{y'}\exp(h_{y'}(S|x))}$, we set $v(S|x) = \log\sum_{y'\neq y^{\text{truth}}}\exp(h_{y'}(S|x))$ to measure the interaction *w.r.t.* effects on other categories. $h_{y'}(S|x)$ denoted the network output of the catagory y' before the softmax layer, when we took variables in S of x as the input. Figure 7 shows interactions *w.r.t.* outputs of the target category and other categories. We found that high-order interactions *w.r.t.* other categories usually increased. Meanwhile, high-order interactions *w.r.t.* the target category also significantly increased. This indicated that adversarial perturbations adversely affected complex features corresponding to the ground-truth category, while encouraging features for other categories, especially for the target category.

I.7 Extended experiments on the relationship between the strength of attacks and the change of high-order interactions

In Section 4.1 of the paper, we find that adversarial attacks mainly affect high-order interactions in DNNs. In this section, we further explore the relationship between the strength of attacks and the change of high-order interactions.

Using the standard ResNet-18 learned on the ImageNet dataset, we conducted untargeted PGD attacks on input samples with different attacking strengths, in order to test the effects of attacking strength on high-order interactions. The strength of attacks was represented by the iteration numbers of the PGD attack. Then, we computed the change in high-order interactions $\Delta I^{(m)}$ caused by adversarial attacks of different strengths. We also computed the Pearson correlation coefficient between the strength of attacks and $\Delta I^{(m)}$. Table 3 shows a close relationship between changes in high-order interactions and the strength of adversarial attacks. Besides, for both weak and strong attacks, the highest-order (here we set it as $0.95n$-order) interactions were much more sensitive than interactions of not-so-high orders (e.g., $0.7n$-order).

I.8 Extended experiments based on models with certified robustness

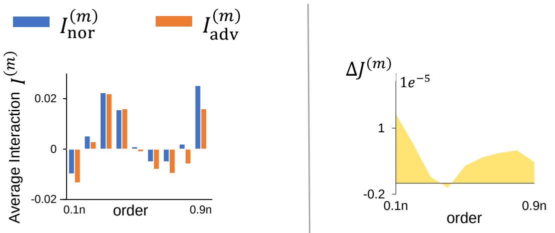
In the main paper, experimental results on DNNs trained via adversarial training [41] have verified our conclusions that adversarial attacks mainly affect high-order interactions, and low-order interactions in adversarial examples towards robust DNNs have more attacking utilities. In this section, we conducted extended experiments on other defending method [14]. Given the ResNet-50, which was trained using randomized smoothing [14], we generated adversarial examples using the PGD attack on samples from the validation set of the ImageNet dataset. We followed experimental settings of the classic PGD attack (see Section 4.1 in the main paper). Then, we computed multi-order interactions in normal samples and adversarial examples on the pre-trained ResNet-50.

Figure 10 shows that in the model trained via randomized smoothing, adversarial attacks also mainly affected high-order interactions, and low-order interactions had more attacking utilities on the model output, which verified our conclusions.

I.9 More discussions about the comparison with frequency-based methods

In Section 4.1 of the main paper, we have compared our interaction-based metric with the frequency-based metric. This section provides more discussions about the comparative experiment.

Previous studies [79, 64, 27] explained adversarial perturbations as high-frequency features. In this paper, we compared the frequency metric with the interaction metric, and showed that our interaction metric could better explain the essential property of adversarial perturbations. In order to measure the significance of features of different frequencies in an input image x, we applied the Fourier transform to the image to obtain the frequency spectrum F. Then, we shifted the low frequency components



(a) The multi-order interaction $I_{\text{nor}}^{(m)}$ and $I_{\text{adv}}^{(m)}$ based on the ResNet-50 trained via randomized smoothing.

(b) Distribution of compositional attacking utilities $\{\Delta J^{(m)}\}$ on the ResNet-50 trained via randomized smoothing.

Figure 10: Experimental results on the ResNet-50 trained via randomized smoothing.

to the center of the spectrum. The magnitude of the r-frequency component was quantified as $F^{(r)} = \sqrt{\sum_{(h,w):h^2+w^2=r^2} \|F^{(h,w)}\|_2^2}$. In order to fairly compare these two metrics, we computed the Fisher's discriminant ratio [22] using two metrics, which measured the ratio of the variance between the classes (between normal samples and adversarial examples) to the variance within the classes. Such experimental settings has also been introduced in Figure 6 (left) of the main paper.

I.10 Explaining high recoverability of adversarial examples on adversarially trained DNNs

In Section 4.3 of the paper, we claim that adversarial examples towards adversarially trained DNNs usually exhibit higher recoverability than adversarial examples towards standard DNNs. In the supplementary material, this section shows experimental results to verify the above claim.

We generated adversarial examples for normal validation samples in the ImageNet dataset by following settings of the untargeted PGD attack [41], in which $\epsilon = 16/255$, and the attack was conducted with 10 steps with the step size $2/255$. Adversarial examples were generated based on ResNet-18/50 and DenseNet-161 trained on the ImageNet dataset. Then, we used same parameters to conduct the targeted PGD attack and recover normal samples.

Table 4: The distance between normal samples x and adversarial examples x^{adv}, and the distance between x and the recovered samples $\hat{x}$.

	Standard DNN		Adversarially trained DNN	
	$\mathbb{E}\|x - x^{\text{adv}}\|_2$	$\mathbb{E}\|x - \hat{x}\|_2$	$\mathbb{E}\|x - x^{\text{adv}}\|_2$	$\mathbb{E}\|x - \hat{x}\|_2$
ResNet-18	**9.72**	13.57	18.69	**11.45**
ResNet-50	**9.56**	13.40	18.34	**12.68**
DenseNet-161	**9.67**	13.51	18.55	**13.26**

Table 4 shows that adversarially trained DNNs usually exhibited higher recoverability than standard DNNs. This can be explained Proposition 1 in the paper. As we have discussed in Section 4.1 of the paper, adversarial perturbations towards adversarially trained DNNs usually pay more attention to low-order interactions than perturbations towards standard DNNs. On the other hand, the low-order interaction $I_{ij}^{(m)}(x)$ is equivalent to the conditional mutual information $MI(X_i; X_j; Y|X_S)$ given small contexts X_S according to Proposition 1, while the high-order interaction corresponds to such a mutual information conditioned on large contexts with massive variables. In general, compared to high-order interactions, low-order interactions are conditioned on less contextual variables, thereby suffering less from adversarial perturbations, *i.e.* obviously $MI(X_i; X_j; Y|X_S)$ usually suffers less from adversarial perturbations, if the condition x_S only contains very few variables. In other words, low-order interactions are more transferable among different contexts x_S, so it is easy to invert adversarial perturbations of low-order interactions. In this way, because adversarial perturbations for adversarially trained DNNs mainly focus on low-order interactions, such perturbations are easy to be recovered.