FORMULATING AND PROVING THE TREND OF DNNS LEARNING SIMPLE CONCEPTS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper theoretically explains the intuition that simple concepts are more likely to be learned by deep neural networks (DNNs) than complex concepts. Beyond empirical studies, our research specifies an exact definition of the complexity of the concept that boosts the learning difficulty. Specifically, it is proven that the inference logic of a DNN can be represented as a causal graph. In this way, causal patterns in the causal graph can be used to formulate interactive concepts learned by the DNN. Based on such formulation, we explain the reason why simple interactive concepts in the data are more likely to be learned than complex interactive concepts. More crucially, we discover that our research provides a new perspective to explain previous understandings of the conceptual complexity. *The code will be released when the paper is accepted.*

1 INTRODUCTION

Deep neural networks (DNNs) have exhibited superior performance in various tasks, but the reason for their superior performance remains an open problem. To this end, many attempts have been made to explain the representation capacity of DNNs from different perspectives. For example, Montufar et al. (2014) used the number of linear response regions in a DNN to evaluate the representation capacity of the DNN. Dinh et al. (2017) and Petzka et al. (2021) used the flatness of loss functions at minima to explain the generalization power. Hardt et al. (2016), Lei & Ying (2020), and Bassily et al. (2020) evaluated the stability of optimization.

Unlike previous studies, we analyze the representation capacity of DNNs from the perspective of conceptual representation. In this paper, **we aim to theoretically explain the reason why it is easier for DNNs to learn simple concepts than complex concepts.**

In fact, the above intuitive phenomenon has been widely observed in previous studies (Arpit et al., 2017; Liu et al., 2021; Mangalam & Prabhu, 2019). However, these studies mainly investigated the phenomenon in an empirical manner, without establishing a clearly theoretical connection between the optimization difficulty and the conceptual complexity. A theoretical connection may shed new light on the understanding of DNNs. Besides, some different metrics (Xu et al., 2020; Wang et al., 2019; Santiago et al., 2021; Chatterji et al., 2019) explained the learning difficulty from other perspectives, instead of directly investigating the complexity of concepts encoded by a DNN.

Therefore, beyond empirical studies on conceptual complexity, in this paper, we aim to specify an exact mathematical form of conceptual complexity that boosts the learning difficulty. Specifically, we face the following two issues, *i.e.,* (1) to faithfully represent concepts encoded by the DNN, and (2) to prove why the DNN is more likely to learn simple concepts than complex concepts.

Representing interactive concepts by causal patterns. Faithfully representing concepts encoded by DNNs has been a significant challenge for decades. Fortunately, Ren et al. (2021a) have proven that the inference logic of a DNN on a specific input sample can be represented as a causal graph. Thus, we consider causal patterns in the casual graph as **interactive concepts**, which are memorized by the DNN. Specifically, given an input sample $\boldsymbol{x}$ with n input variables (*e.g.,* words in a sentence and pixels in an image), Fig. 1 shows a three-layer causal graph. Each source node $A_i \in \{0, 1\}$ in the bottom layer reflects whether the input variable x_i is present ($A_i = 1$) or masked ($A_i = 0$). Each intermediate node C_S corresponds to a causal pattern S, which represents an AND relationship between different input variables in S. For example, the *mouth pattern* in Fig. 1 consists of image

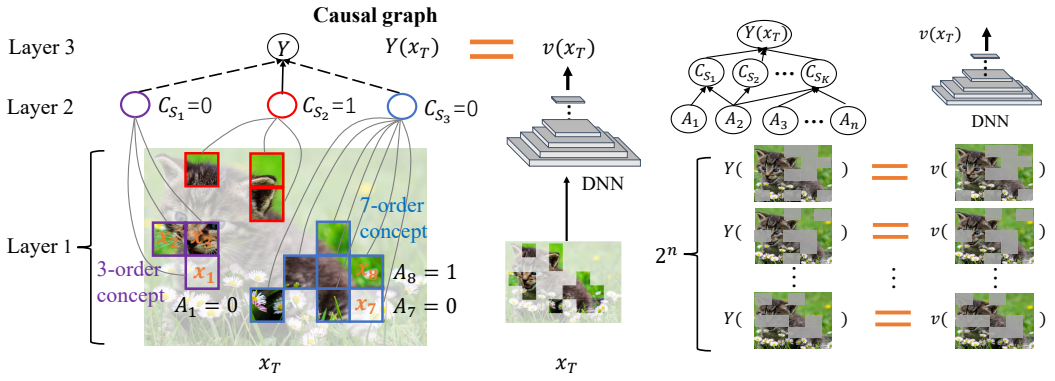

Figure 1: The inference logic of a DNN on a specific input sample can be represented as a causal graph. Given an input sample $\boldsymbol{x}$ with n input variables, there are 2^n differently masked samples $\boldsymbol{x}_T$. The diverse DNN outputs $v(\boldsymbol{x}_T)$ on the 2^n samples can be all accurately mimicked by the output of the corresponding causal graph $Y(\boldsymbol{x}_T)$.

patches x_1, x_2, and x_3, *i.e.,* $S_1 = \{x_1, x_2, x_3\}$. Only when all these three patches are all present, this causal pattern is triggered ($C_{S_1} = 1$). In Fig. 1, we mask the patch x_1, which deactivates this causal pattern ($C_{S_1} = 0$). The sink node Y in the top layer is referred to as the output of this causal graph.

Trustworthiness of causal graph. Given an input sample $\boldsymbol{x} \in \mathbb{R}^n$, there exist as many as 2^n different ways to randomly mask input variables. **It has been proven that diverse outputs of the DNN on all 2^n masked samples can be all accurately mimicked by even a concise causal graph with not so many causal patterns.** Both the conciseness and the universal matching of all arbitrarily masked samples ensure the trustworthiness of using the causal graph to explain the DNN.

Complexity of interactive concepts. In this study, we use the number of variables in S to represent the complexity of an interactive concept, namely *the order of the interactive concept*. Thus, low-order interactive concepts usually reflect simple AND interactions among a few input variables. In contrast, high-order interactive concepts often represent relatively complex AND interactions among a large number of input variables.

Theoretical explanation for the intuition that DNNs mainly learn simple concepts. To this end, we derive an approximate solution to the variance of causal effects of interactive concepts, which shows that high-order interactive concepts encoded by the DNN are more unstable than low-order interactive concepts.

In other words, the same low-order concept is more likely to be shared by different input samples in the same category with consistently positive (or negative) effects on the inference. In comparison, the high-order concept is more likely to have offsetting effects on the inference of different samples. This explains the reason why DNNs mainly learn simple interactive concepts.

Explaining existing findings of adversarial robustness, generalization power, and conceptual complexity of a DNN. We discover that our research may provide a new perspective to explain previous findings/understandings of the conceptual complexity (Arpit et al., 2017; Mangalam & Prabhu, 2019; Xu et al., 2019; Liu et al., 2021). Besides, the conceptual complexity can directly explain the adversarial robustness and generalization power of a DNN. Thus, our study is of considerable value for various research directions in explaining the representation capacity of a DNN.

2 DNNs MAINLY LEARN SIMPLE CONCEPTS

2.1 REPRESENTING A DNN USING INTERACTIVE CONCEPTS

• **Explaining a DNN as a causal graph.** In this subsection, we analyze the representation capacity of a DNN from the perspective of conceptual representation. Ren et al. (2021a) have proven that the inference logic of a trained DNN can be faithfully explained as a causal graph. Specifically, given a pre-trained DNN and an input sample with n input variables $\boldsymbol{x} = [x_1, \ldots, x_n]$, the corresponding

causal graph has three layers. As Fig. 1 shows, *the first layer* of the causal graph contains n leaf nodes. Each leaf node is an indicator variable $A_i \in \{0, 1\}$, which reflects whether the input variable x_i is present ($A_i = 1$) or masked ($A_i = 0$). *The second layer* contains causal patterns. Each causal pattern S represents an AND relationship between different input variables in S. For example, the co-appearance of image patches x_1, x_2, and x_3 form a mouth pattern $S_1 = \{x_1, x_2, x_3\}$. In other words, only when these three patches x_1, x_2, and x_3 are all present, the causal pattern S will be triggered, denoted by $C_{S_1} = 1$; otherwise, not triggered $C_{S_1} = 0$. The *third layer* is the output layer of the causal graph, which only contains a single node Y. Thus, the transition probability of this causal graph can be formulated as follows.

$$P(C_S = 1|A_1, A_2, \ldots, A_n) = \prod\nolimits_{k\in S} A_k, \quad P(Y|\{C_S|S \in \Omega\}) = \mathbb{1}(Y = \sum\nolimits_{S\in\Omega} C_S \cdot U_S), \tag{1}$$

where $P(C_S = 0|A_1, A_2, \ldots, A_n) = 1 - P(C_S = 1|A_1, A_2, \ldots, A_n)$, and $\mathbb{1}(\cdot)$ refers to the indicator function. Moreover, let $N = \{1, 2, \ldots, n\}$ denote the indices of all input variables, and let $\Omega \subseteq 2^N = \{S|S \subseteq N\}$ represent a set of causal patterns. U_S can be understood as the causal effect of the causal pattern S to the output Y of the causal graph.

Trustworthiness of the causal graph. Theorem 1 proves that the inference logic of a DNN can be faithfully explained as the above specific causal graph. To this end, let $Y(\boldsymbol{x}_T)$ represent the output of the causal graph computed in Eq. (3) by setting the input $A_i = \mathbb{1}(i \in T)$, and let $v(\boldsymbol{x}_T)$ denote the output of the DNN on the masked sample $\boldsymbol{x}_T$. Here, $\boldsymbol{x}_T$ is referred to as the input sample, when we mask input variables in $N \setminus T$ using baseline values b_i, and keep variables in T unchanged.

$$(\boldsymbol{x}_T)_i = \begin{cases} x_i, & \text{if } i \in T; \\ b_i, & \text{if } i \in N \setminus T, \end{cases} \quad \text{subject to} \quad b_i = \begin{cases} x_i - \tau & x_i > \mu_i; \\ x_i + \tau, & x_i < \mu_i, \end{cases} \tag{2}$$

where $\tau \in \mathbb{R}$ is a constant, and $\mu_i = \mathbb{E}_{\boldsymbol{x}}[x_i]$ is referred to as the average value of the input variable x_i over all input samples. Because the mean value μ_i is usually considered to represent a non-signal state (Ancona et al., 2019), we consider that pushing the input variable x_i to move a large distance τ towards μ_i has been significant enough to remove its information. Unlike directly setting $b_i = \mu_i$, the above setting ensures comparable perturbation magnitudes over different dimensions.

Theorem 1 (proven in Appendix D.1)**.** *For each DNN v, there exists a specific causal graph, such that for any arbitrarily masked sample $\boldsymbol{x}_T$, the output of a DNN $v(\boldsymbol{x}_T)$ can be accurately represented as the output of the causal graph, i.e., $\exists\Omega \subseteq 2^N, \exists\{U_S|S \in \Omega\}, \forall T \subseteq N, v(\boldsymbol{x}_T) = Y(\boldsymbol{x}_T)$.*

Theorem 1 proves the trustworthiness of using the causal graph to explain the DNN. Theoretically, given an input sample $\boldsymbol{x} \in \mathbb{R}^n$ with n variables, there are 2^n ways to mask it, leading to 2^n differently masked samples $\boldsymbol{x}_T$ *w.r.t.* all subsets $T \subseteq N$. Diverse outputs of the DNN $v(\boldsymbol{x}_T)$ on the 2^n samples can be all accurately mimicked by the corresponding causal graph $Y(\boldsymbol{x}_T)$. In this way, the inference logic of a DNN on $\boldsymbol{x}$ is well represented by the causal graph.

• **Considering top-ranked salient causal patterns as interactive concepts encoded by the DNN.** The transition probability in Eq. (1) makes the causal relationship between causal patterns and the output $Y(\boldsymbol{x}_T)$ be represented as a structural causal model (SCM) (Pearl, 2009),

$$Y(\boldsymbol{x}_T) = \sum\nolimits_{S\in\Omega} I(S), \quad I(S) = U_S \cdot C_S(\boldsymbol{x}_T) = U_S \cdot \prod\nolimits_{i\in S} A_i = U_S \cdot \mathbb{1}(S \subseteq T) \tag{3}$$

In fact, each causal pattern S represents an AND relationship, which can be considered as an interactive concept memorized by the DNN. In the above example, the *mouth pattern*$= \{x_1, x_2, x_3\}$ can be considered as an interactive concept. This mouth concept will be triggered, only when these three patches x_1, x_2, and x_3 co-appear, and makes a causal effect $I(S) = U_S$. Otherwise, the absence of any variable (x_1, x_2, and x_3) will remove the causal effect, $I(S) = 0$.

Remark 1. *Ren et al. (2021a) proposed a typical implementation for the computation of the causal graph, i.e., modeling the causal effect U_S as the Harsanyi dividend (Harsanyi, 1963), $\forall S \subseteq N$, $U_S = \sum_{T\subseteq S}(-1)^{|S|-|T|} \cdot v(\boldsymbol{x}_T)$, which satisfies Theorem 1. Hence, the causal effect can be computed as $I(S) = C_S \cdot \sum_{T\subseteq S}(-1)^{|S|-|T|} \cdot v(\boldsymbol{x}_T)$.*

Moreover, people can apply different confident scores to implement $v(\boldsymbol{x}_T)$, *e.g.,* setting $v(\boldsymbol{x}_T)$ as the confidence of classifying the input sample $\boldsymbol{x}_T$ to the ground-truth category y_{truth}, $v(\boldsymbol{x}_T) = \log\frac{p(y=y_{\text{truth}}|\boldsymbol{x}_T)}{1-p(y=y_{\text{truth}}|\boldsymbol{x}_T)}$.

Remark 2. *Based on Remark 1, given a DNN v, causal effects of most patterns are actually ignorable. Thus, we can find a sparse subgraph, which only contains a few causal patterns with top-ranked causal effects in the set $\Omega' \subseteq \Omega$, $|\Omega'| \ll 2^n$, such that the output of the DNN $v(\boldsymbol{x}_T)$ can be approximately represented by the sparse causal graph, i.e., $\forall T \subseteq N$, $v(\boldsymbol{x}_T) \approx Y(\boldsymbol{x}_T|\Omega')$. This is experimentally verified in Appendix C.*

Remark 2 is based on numerous experimental results, which indicates that interactive concepts are very concise (Ren et al., 2021a). In other words, Remark 2 means that causal effects of most interactive concepts S are close to zero ($|U_S| \approx 0$), thereby having ignorable effects on the network output. Not so many interactive concepts make considerable effects $|U_S|$ on the network output. Therefore, these top-ranked salient interactive concepts can be considered as meaningful concepts encoded by the DNN.

• **Complexity of interactive concepts.** We use the number of variables in S to measure the complexity of an interactive concept, namely *the order of the interactive concept*, $\text{order}(S) = |S|$. Thus, low-order interactive concepts usually represent simple AND interactions among not so many input variables. In comparison, high-order interactive concepts are often referred to as relatively complex AND interactions among a large number of input variables.

2.2 Low-order interactive concepts in data are more stable

In this subsection, we derive an approximate analytic solution to the variance of causal effects of interactive concepts, and experimentally verify the correctness of this solution. Specifically, we can re-write the causal effect of each interactive concept as follows.

Theorem 2 (proven in Appendix D.2)**.** *Given a pre-trained DNN v and an arbitrary (masked or not) sample $\boldsymbol{x}' \in \mathbb{R}^n$, we use the Taylor expansion to decompose the output of this DNN by following Deng et al. (2021). The causal effect $I(S) \in \{U_S, 0\}$ in Eq. (3) is a binary variable, since the causal graph only consider the binary masking state of each input variable. Then, we extend the binary causal effect into a continuous function $I(S|\boldsymbol{x}')$ based on the Taylor expansion, which can well fit $I(S)$ on all 2^n samples with 2^n different masking states $\boldsymbol{x}' \in \{\boldsymbol{x}_T|\forall T \subseteq N\}$.*

$$v(\boldsymbol{x}') = \sum\nolimits_{S\subseteq N} \sum\nolimits_{\boldsymbol{\pi}\in Q_s} U_{S,\boldsymbol{\pi}} \cdot J(S, \boldsymbol{\pi}|\boldsymbol{x}') \quad \Rightarrow \quad I(S|\boldsymbol{x}') = \sum\nolimits_{\boldsymbol{\pi}\in Q_s} U_{S,\boldsymbol{\pi}} \cdot J(S, \boldsymbol{\pi}|\boldsymbol{x}'). \tag{4}$$

$J(S, \boldsymbol{\pi}|\boldsymbol{x}') = \prod_{i\in S}(\text{sign}(x'_i - b_i)\frac{x'_i - b_i}{\tau})^{\pi_i}$ denotes a Taylor expansion term of the degree $\boldsymbol{\pi} \in Q_s = \{[\pi_1, \cdots, \pi_n]|\forall i \in S, \pi_i \in \mathbb{N}^+; \forall i \notin S, \pi_i = 0\}$. $U_{S,\boldsymbol{\pi}} = \frac{\tau^m}{\prod_{i=1}^n \pi_i!} \cdot \frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1} \cdots \partial x_n^{\pi_n}} \cdot \prod_{i\in S}(\text{sign}(x'_i - b_i))^{\pi_i}$, $m = \sum_{i=1}^n \pi_i$. $v(\boldsymbol{x}_\emptyset)$ indicates the network output, when we mask all input variables. Moreover, $C_S(\boldsymbol{x}') = I(S|\boldsymbol{x}')/U_S$.

Empirically, people often use the low-order Taylor expansion for approximation. Thus, let us first consider the simplest case, *i.e.,* using the expansion term of the lowest degree $\hat{\boldsymbol{\pi}}$ for approximation, $\forall i \in S, \hat{\pi}_i = 1; \forall i \notin S, \hat{\pi}_i = 0$. In this scenario, the causal effect $I(S|\boldsymbol{x}')$ in Theorem 2 can be computed as $I(S|\boldsymbol{x}') \approx U_{S,\hat{\boldsymbol{\pi}}} \cdot J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}')$. Next, we analyze the sensitivity of $I(S|\boldsymbol{x}')$ when we add a Gaussian perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$. To simplify the analysis, we roughly consider that $|\epsilon_i| \le \tau$, due to the small variance of perturbations $\boldsymbol{\epsilon}$.

Theorem 3 (proven in Appendix D.3)**.** *Let us add a Gaussian perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$ to the input sample $\boldsymbol{x}$. If we only consider the approximation based on the lowest degree $\hat{\boldsymbol{\pi}}$, then according to Theorem 2, the mean and variance of $I(S|\boldsymbol{x} + \boldsymbol{\epsilon})$ over different perturbations are given as*

$$\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x} + \boldsymbol{\epsilon})] = U_{S,\hat{\boldsymbol{\pi}}}, \quad \text{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x} + \boldsymbol{\epsilon})] = U_{S,\hat{\boldsymbol{\pi}}}^2\big[\left(1 + (\sigma/\tau)^2\right)^s - 1\big]. \tag{5}$$

Theorem 3 shows the impact of variations in input samples on the stability of interactive concepts of different complexities. That is, the variance of the causal effect $I(S|\boldsymbol{x} + \boldsymbol{\epsilon})$ increases along with the order s of the interactive concept S exponentially. Here, the Gaussian perturbation $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is used as a rough representation of inevitable variations in data. For example, image classification suffers from different variations, such as small shape deformation and small object rotation variations. We use a Gaussian perturbation because such variations are quite difficult to formulate. In this way, the exponential increase of the variance *w.r.t.* the order s in Theorem 3 reflects that **compared to low-order concepts, high-order concepts are much more sensitive to slight input perturbations**.

Furthermore, we can extend the analytic solution in Theorem 3 to a more general case, *i.e.,* using the higher-order Taylor expansion to represent $I(S|\boldsymbol{x}')$, as follows.

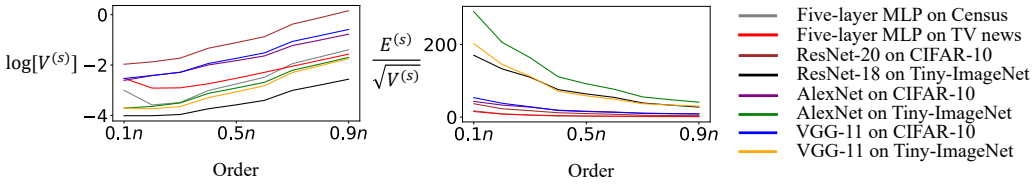

Figure 2: The logarithm of the variance of causal effects $\log[V^{(s)}]$ and the stability of causal effects (measured by $E^{(s)}/\sqrt{V^{(s)}}$). The variance $V^{(s)}$ of causal effects increases along with the order of interactive concepts exponentially. The stability of causal effects decreases along with the order.

Theorem 4 (proven in Appendix D.4)**.** *For an arbitrary degree* $\boldsymbol{\pi} \in Q_s = \{[\pi_1, \cdots, \pi_n] | \forall i \in S, \pi_i \in \mathbb{N}^+; \forall i \notin S, \pi_i = 0\}$*, the mean and variance of* $J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})$ *can be computed as*

$$\mathbb{E}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] = \mathbb{E}_{\boldsymbol{\epsilon}}[\prod\nolimits_{i\in S}(1+\epsilon_i/\tau)^{\pi_i}], \quad \mathrm{Var}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] = \mathrm{Var}_{\boldsymbol{\epsilon}}[\prod\nolimits_{i\in S}(1+\epsilon_i/\tau)^{\pi_i}]. \quad (6)$$

Remark 3. *According to Theorem 4, we can roughly consider that the variance of* $J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})$ *increases along with the order* s *in an exponential manner. According to Eq. (4), the causal effect* $I(S|\boldsymbol{x}+\boldsymbol{\epsilon})$ *can be represented as the weighted sum of* $J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})$*, and coefficients* $U_{S,\boldsymbol{\pi}}$ *w.r.t. different orders* s *and degrees* $\boldsymbol{\pi}$ *are usually chaotic. Hence, we can roughly consider that the variance of the causal effect* $I(S|\boldsymbol{x}+\boldsymbol{\epsilon})$ *increases along with the order* s *exponentially, as well.*

Experimental verification. Here, we conduct experiments to verify Theorem 3 and Remark 3, *i.e.* checking whether the variance of causal effects increased along with the order of the interactive concepts in an approximately exponential manner. Specifically, to mimic variations in the data, we add Gaussian perturbations $\boldsymbol{\epsilon}$ with zero mean and the variance $\sigma^2 = 0.02^2$ to each training sample. Then, we comput the average mean $E^{(s)}=\mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}}[\mathbb{E}_{S\subseteq N,|S|=s}[|\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]|]]$ and the average variance $V^{(s)}=\mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}}[\mathbb{E}_{S\subseteq N,|S|=s}[\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]]]$ of interactive concepts over different input samples. For verification, we calculate such metrics on DNNs for image classification and DNNs for tabular data. We train AlexNet (Krizhevsky et al., 2012), VGG-11 (Simonyan & Zisserman, 2014), ResNet-18/20 (He et al., 2016) on the CIFAR-10 dataset (Krizhevsky et al., 2009) and the Tiny-ImageNet dataset (Le & Yang, 2015), respectively. We also train a five-layer MLP (Ren et al., 2021a) on the UCI census dataset (namely *census dataset*) and the UCI TV news dataset (namely *TV news dataset*) (Asuncion & Newman, 2007), respectively. Additionally, considering the computational cost of $I(S|\boldsymbol{x}+\boldsymbol{\epsilon})$ in Remark 1 is intolerable in real implementation, we apply the sampling-based approximation method in (Zhang et al., 2020) to calculate $I(S|\boldsymbol{x}+\boldsymbol{\epsilon})$. Please see Appendix F for experimental details.

Thus, we use $E^{(s)}/\sqrt{V^{(s)}}$ to measure the relative stability of causal effects of the s-th order interactive concepts. Fig. 2 shows that the stability decreases significantly, and the variance of causal effects $V^{(s)}$ increases in an exponential manner along with the order of the interactive concept. Such phenomena successfully verified findings in Theorem 3 and Remark 3.

2.3 DNNs MAINLY LEARN SIMPLE LOW-ORDER INTERACTIVE CONCEPTS

Simplifying the conceptual learning as a linear problem based on the SCM. In this subsection, we analyze the trend of a DNN in encoding interactive concepts of different orders. To simplify the analysis of the DNN, the SCM in Eq. (3) and Theorem 1 explain the DNN as a linear function of different interactive concepts, *i.e.,* $v(\boldsymbol{x}_T) = Y(\boldsymbol{x}_T) = \sum_{S\in\Omega} U_S \cdot C_S(\boldsymbol{x}_T)$. Here, C_S can be roughly considered as an input dimension of the linear function, which reflects whether the input sample contains the interactive concept S. The coefficient U_S can be taken as the strength of the DNN encoding the interactive concept S. Most interactive concepts have ignorable coefficients $U_S \approx 0$, and not so many salient concepts S have large absolute value of $|U_S|$. Thus, we can consider that the DNN only learns a small number of salient interactive concepts.

The basic idea of proving the claim that DNNs mainly learn low-order interactive concepts is as follows. The above SCM in Eq. (3) enables us to understand a DNN for the classification task as a pseudo-linear function. If feature dimension (*i.e.,* an interactive concept S) has a stable value (*i.e.,* C_S stably being present/absent) across all samples in the same category, then we consider this feature

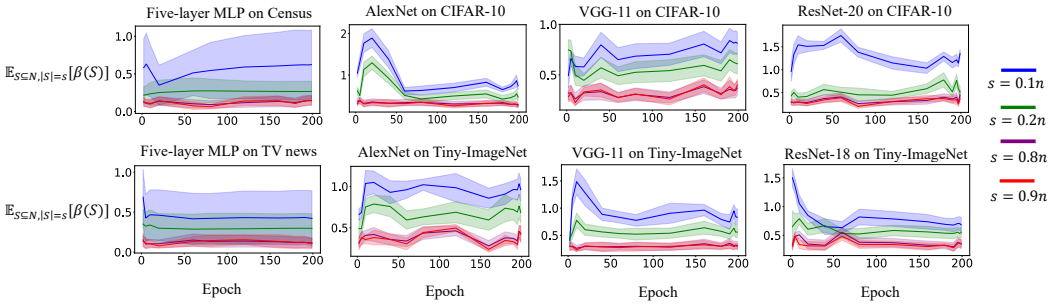

Figure 3: Consistency $\beta(S)$ of s-order interactive concepts. The curve shows the mean consistency $\mathbb{E}_{S\subseteq N,|S|=s}[\beta(S)]$ over different interactive concepts of the s-th order, and the shade indicates the standard deviation $\text{Std}_{S\subseteq N,|S|=s}[\beta(S)]$. Causal effects of low-order concepts are more consistent than those of high-order concepts.

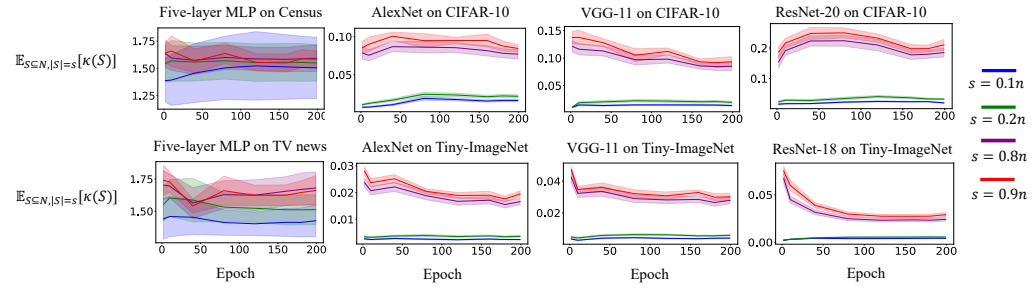

Figure 4: Instability $\kappa(S)$ of s-order interactive concepts to data variations. The curve shows the average instability $\mathbb{E}_{S\subseteq N,|S|=s}[\kappa(S)]$ over different interactive concepts of the s-th order, and the shade represents the standard deviation $\text{Std}_{S\subseteq N,|S|=s}[\kappa(S)]$. Causal effects of high-order concepts are more sensitive to data variations than those of low-order concepts.

dimension (*i.e.,* the concept) is discriminative and easy to learn. In comparison, if the variance of a feature dimension is large, *i.e.,* the concept cannot be consistently present or consistently absent over samples in the same category, then this feature dimension (*i.e.,* the concept) is hard to learn.

Experiment 1: verifying the claim that high-order interactive concepts are more sensitive to data variations than low-order interaction. To this end, we use small perturbations $\boldsymbol{\epsilon}$ to represent various inevitable variations in the data (such as small shape deformation and small object rotation variations). Theorem 3, Remark 3, as well as Fig. 2, all show that compared to low-order interactive concepts, high-order interactive concepts are much more sensitive to inevitable variations in the data. This makes high-order concepts are more likely to be influenced by data variations and less likely to be consistently present or absent in a target sample, which boosts the learning difficulty.

Therefore, we use the following two metrics to evaluate the discrimination power of each interactive concept S (*i.e.,* each single dimension of the above linear function) of a certain order. Specifically, we use the first metric $\beta(S) = \mathbb{E}_c\big[|\mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}_c}[I(S|\boldsymbol{x})]|/\text{Std}_{\boldsymbol{x}\in\boldsymbol{X}_c}[I(S|\boldsymbol{x})]\big]$ to measure the relative consistency of the interactive concept appearing over different input samples in a certain category c. Here, $\boldsymbol{X}_c$ denotes a set of training samples belong to the category c, and $\text{Std}(\cdot)$ indicates the standard deviation over different input samples. A large $\beta(S)$ value means that the interactive concept S in all samples of the category c has similar/consistent causal effects $I(S|\boldsymbol{x})$. It is easier for the DNN to learn such a consistent concept S.

Besides, we use another metric $\kappa(S)$ to verify whether the causal effect of the high-order interactive concept is usually less stable than those of the low-order interactive concept. The metric $\kappa(S) = \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}}[\mathbb{E}_{\boldsymbol{\epsilon}}[|I(S|\boldsymbol{x}+\boldsymbol{\epsilon}) - I(S|\boldsymbol{x})|]]/\mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}}\big[|I(S|\boldsymbol{x})|\big] = \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}}[\mathbb{E}_{\boldsymbol{\epsilon}}[|C_S(\boldsymbol{x}+\boldsymbol{\epsilon}) - C_S(\boldsymbol{x})|]]/\mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}}\big[|C_S(\boldsymbol{x})|\big]$ measures the relative instability of the interactive concept S to the inevitable slight variations $\boldsymbol{\epsilon}$ in the data. Here, $C_S(\boldsymbol{x}+\boldsymbol{\epsilon}) = I(S|\boldsymbol{x}+\boldsymbol{\epsilon})/U_S$ computed in Theorem 2 denotes the trigger state of the interactive concept S on the sample $\boldsymbol{x}+\boldsymbol{\epsilon}$.

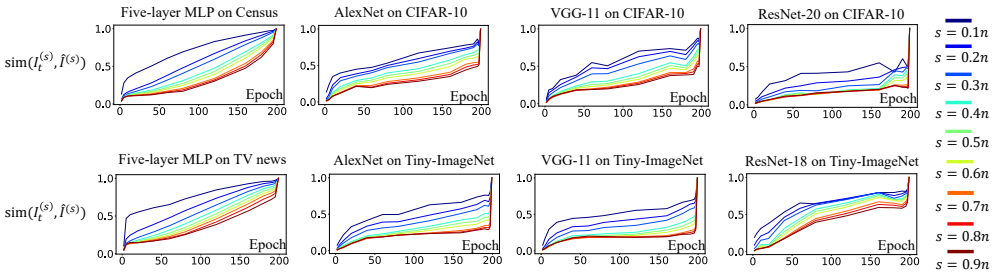

Figure 5: The weighted Jaccard similarity $\text{sim}(\boldsymbol{I}_t^{(s)}, \hat{\boldsymbol{I}}^{(s)})$ between s-order interactive concepts learned after intermediate epochs $\boldsymbol{I}_t^{(s)}$ and those learned after the final epoch $\hat{\boldsymbol{I}}^{(s)}$. Low-order concepts usually have higher Jaccard similarity during the learning process, which indicates that DNNs first mainly learn low-order concepts and then gradually learn more about high-order concepts.

To this end, we use DNNs trained in the ***"experimental verification"*** paragraph of Section 2.2 for evaluation. Please see Appendix F for more details of experimental settings. Fig. 3 and Fig. 4 show the change of the average consistency $\beta(S)$ and the average instability $\kappa(S)$ of s-order interactive concepts, respectively, through the learning process. At each training epoch, low-order concepts usually obtain higher consistency $\beta(S)$ and lower instability $\kappa(S)$ than high-order concepts. It means that low-order concepts usually are more consistent and more stable. Thus, this experiment explains the reason why low-order interactive concepts are easier to learn.

Experiment 2: verifying the phenomenon that low-order interactive concepts are usually learned faster than high-order concepts. We have theoretically proven in Section 2.2 and experimentally verified in Experiment 1 that low-order interactive concepts are more consistently present or consistently absent in different samples of the same category, which makes low-order interactive concepts easier to be learned. Then, Experiment 2 is conducted to check whether low-order concepts are really learned faster than high-order concepts.

Specifically, we examine whether interactive concepts encoded by the finally-learned DNN $\hat{v}(x)$ have already been encoded by the DNN that has not been fully optimized after t training epochs $v_t(x)$. If so, we consider such interactive concepts are learned fast. Specifically, let $\boldsymbol{I}_t^{(s)} = [I_t(S_1|\boldsymbol{x}), \cdots, I_t(S_d|\boldsymbol{x})]^\top \in \mathbb{R}^d$ denote a vector for all $d = \binom{n}{s}$ combinations corresponding to s-order interactive concepts. Then, we compute the Jaccard similarity between s-order interactive concepts encoded by the DNN $\hat{v}(x)$ and those encoded by the DNN $v_t(x)$, *i.e.*, $\text{sim}(\boldsymbol{I}_t^{(s)}, \hat{\boldsymbol{I}}^{(s)}) = r \cdot \text{Jaccard}(\boldsymbol{I}_t^{(s),+}, \hat{\boldsymbol{I}}^{(s),+}) + (1-r) \cdot \text{Jaccard}(\boldsymbol{I}_t^{(s),-}, \hat{\boldsymbol{I}}^{(s),-})$. Because the computation of the Jaccard similarity requires all elements in these two vectors are non-negative, we compute $\text{Jaccard}(\boldsymbol{I}_t^{(s),+}, \hat{\boldsymbol{I}}^{(s),+}) = \|\min(\boldsymbol{I}_t^{(s),+}, \hat{\boldsymbol{I}}^{(s),+})\|_1 / \|\max(\boldsymbol{I}_t^{(s),+}, \hat{\boldsymbol{I}}^{(s),+})\|_1$ as the Jaccard similarity between positive elements in $\boldsymbol{I}_t^{(s),+} = \max(\boldsymbol{I}_t^{(s)}, 0)$ and $\hat{\boldsymbol{I}}^{(s),+} = \max(\hat{\boldsymbol{I}}_t^{(s)}, 0)$, where $\|\cdot\|_1$ denotes L1-norm. Similarly, $\text{Jaccard}(\boldsymbol{I}_t^{(s),-}, \hat{\boldsymbol{I}}^{(s),-})$ measures the Jaccard similarity between negative elements in $\boldsymbol{I}_t^{(s),-} = \max(-\boldsymbol{I}_t^{(s)}, 0)$ and $\hat{\boldsymbol{I}}^{(s),-} = \max(-\hat{\boldsymbol{I}}_t^{(s)}, 0)$. The weight r is calculated as $r = \|\max(\boldsymbol{I}_t^{(s),+}, \hat{\boldsymbol{I}}^{(s),+})\|_1 / [\|\max(\boldsymbol{I}_t^{(s),+}, \hat{\boldsymbol{I}}^{(s),+})\|_1 + \|\max(\boldsymbol{I}_t^{(s),-}, \hat{\boldsymbol{I}}^{(s),-})\|_1]$. In this way, a large similarity $\text{sim}(\boldsymbol{I}_t^{(s)}, \hat{\boldsymbol{I}}^{(s)})$ at an earlier epoch t indicates that interactive concepts are easier to learn.

To this end, we use DNNs trained in the ***"experimental verification"*** paragraph of Section 2.2 for evaluation. Please see Appendix F for more details of experimental settings. Fig. 5 shows that the DNN first learns low-order interactive concepts, and then learns high-order interactive concepts. Such a phenomenon verifies the conclusion that the DNN mainly learns simple interactive concepts.

3 Explaining findings in previous studies

3.1 Explaining generalization power and adversarial robustness

Previous studies (Zhang et al., 2020; Ren et al., 2021b) have discovered that low-order interactions have stronger generalization power and are more robust to adversarial attacks than high-order interactions. Notice that their high/low-order interactions are highly related to our high/low-order

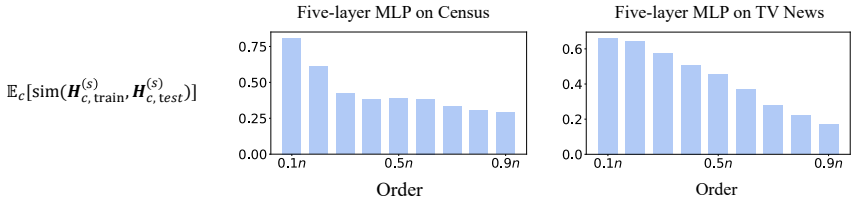

Figure 6: Similarity between the distribution of s-order interactive concepts in training samples and that in testing samples. The distribution of low-order interactive concepts in training samples has a high Jaccard similarity with that in testing samples, while high-order concepts do not have such a property. This proves the strong generalization power of low-order interactive concepts.

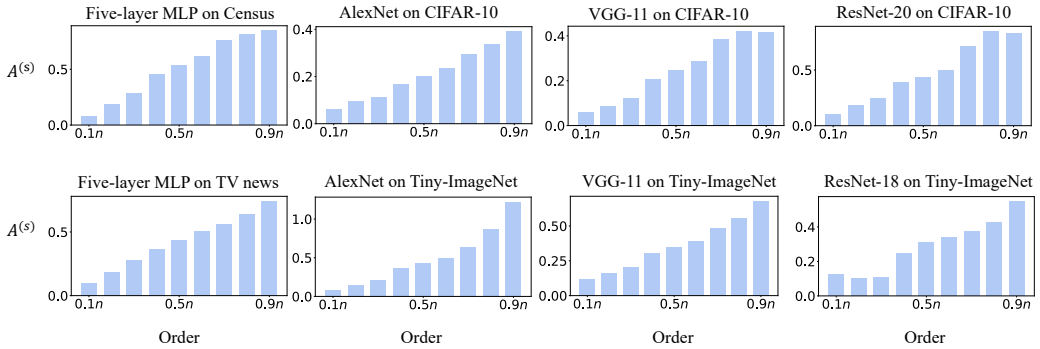

Figure 7: Average sensitivity $A^{(s)}$ of s-order interactive concepts to adversarial perturbations. Low-order interactive concepts are usually much less sensitive to adversarial attacks than high-order interactive concepts.

interactive concepts. Specifically, our interactive concepts can be explained as elementary components for such multi-order interactions (proven in Appendix E). *Therefore, from this perspective, our conclusion that* low-order interactive concepts are easy to learn *can also explain how a DNN encodes concepts of different generalization power and adversarial robustness.*

Can we use interactive concepts in this paper to verify the heuristic findings of generalization power and adversarial robustness in (Zhang et al., 2020; Ren et al., 2021b)? According to the SCM in Eq. (3), we can represent the inference logic of a DNN by a set of interactive concepts, *i.e.,* $v(\boldsymbol{x}_T) = \sum_{S\in\Omega} I(S)$. In this way, we conduct experiments to evaluate the generalization power and the adversarial robustness of each interactive concept S. The symbolic conceptual representation of interactive concepts allows us to define the generalization power in a more direct way. That is, if an interactive concept that frequently appears in training samples also frequently appears in testing samples, we consider this interactive concept generalizes well to the test dataset. Similarly, a frequent concept in testing samples is also expected to frequently appear in training samples. To this end, we use the aforementioned weighted Jaccard similarity $\text{sim}(\boldsymbol{H}^{(s)}_{c,\text{train}}, \boldsymbol{H}^{(s)}_{c,\text{test}})$ to evaluate the generalization power of s-order interactive concepts in the category c. Here, $\boldsymbol{H}^{(s)}_{c,\text{train}} = [H_{c,\text{train}}(S_1), \cdots, H_{c,\text{train}}(S_d)]^\top \in \mathbb{R}^d$ is a vector that enumerates all the s-order interactive concepts, where each dimension $H_{c,\text{train}}(S_i) = \mathbb{E}_{x\in\boldsymbol{X}_c}[I(S_i|\boldsymbol{x})]$ denotes the average causal effects of the interactive concept S_i over all training samples in the same category c. In this way, a high similarity represents that s-order interactive concepts generalize well.

For the adversarial robustness, we use the metric $\alpha(S) = \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}}[|I(S|\boldsymbol{x}+\boldsymbol{\delta}) - I(S|\boldsymbol{x})|] / \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}}[|I(S|\boldsymbol{x})|]$ to evaluate the sensitivity of the interactive concept S to adversarial perturbations. δ denotes the adversarial perturbation generated by the ℓ_∞ attack (Madry et al., 2018). In this way, a small $\alpha(S)$ value indicates that the interactive concept S is robust to the adversarial attack.

To this end, we use DNNs trained in the "***experimental verification***" paragraph of Section 2.2 for evaluation. Fig. 6 and Fig. 7 show that compared to high-order interactive concepts, low-order concepts usually obtain larger $\mathbb{E}_c[\text{sim}(\boldsymbol{H}^{(s)}_{c,\text{train}}, \boldsymbol{H}^{(s)}_{c,\text{test}})]$ values and smaller $A^{(s)} = \mathbb{E}_{S\subseteq N,|S|=s}[\alpha(S)]$

values, respectively. Such phenomena demonstrate that low-order interactive concepts have stronger generalization power, and are more robust to adversarial attacks.

3.2 EXPLAINING EXISTING FINDINGS ABOUT WHAT ARE LEARNED FIRST.

In this subsection, we discuss some related studies on which kind of knowledge is usually first learned by a DNN. Most previous studies conducted experiments to explore the knowledge that was easier to be learned by a DNN, without providing much theoretical support. However, we find that our theorems can partially explain mechanisms behind some previous findings.

• Arpit et al. (2017) trained DNNs to classify both normal samples and white-noise samples to different object categories. In this way, they considered that the DNN encoded simple concepts to classify normal samples, but the DNN had to learn complex concepts to classify white noises to randomly-assigned labels. They observed that the DNN usually learned normal samples first, because the classification accuracy of normal samples increased before that of white noises. To this end, our research provides more insights into such an observation. Specifically, Cheng et al. (2021) have proven that the classification of noisy data usually depends on high-order concepts. Let us combine this conclusion with our finding that high-order interactive concepts are hard to learn. Then, we can easily owe the slow learning of white-noise samples observed by Arpit et al. (2017) to the difficulty of learning high-order interactive concepts.

• Mangalam & Prabhu (2019) considered that easy samples as training samples that could be correctly classified by shallow machine learning models, such as support vector machine (SVM) and random forests (RF). They discovered that DNNs first mainly learned easy samples, and then gradually learned more about hard samples. To this end, our research verifies such observation. Specifically, in this paper, we claim that such hard samples may mainly contain high-order interactive concepts, which correspond to complex AND interactions between numerous variables. This claim just fits the finding in (Mangalam & Prabhu, 2019), because it is difficult to use shallow models (*e.g.*, the SVM and the RF) to classify complex interactions between lots of input variables, which correspond to high-order concepts. In this way, the fast learning of low-order concepts is another understanding of the finding in (Mangalam & Prabhu, 2019).

• Xu et al. (2019) discovered that during the training process, DNNs usually first learned samples of low frequencies (*e.g.,* robust to noises), and then encoded samples of high frequencies (*e.g.,* sensitive to noises). However, the original design of DNNs is not towards learning specific spectrums, and techniques of deep learning are not developed by assuming a periodic loss landscape of training samples. Therefore, we believe there should be a more direct explanation for the spectrum-learning phenomenon discovered by (Xu et al., 2019). To this end, our research explains this phenomenon as the difficulty of learning high-order concepts. According to Section 3.1, low-order concepts usually are less sensitive to input perturbations, thereby corresponding to low-frequency components defined in the loss landscape in (Xu et al., 2019). Accordingly, high-order concepts correspond to high-frequency components.

• Liu et al. (2021) discovered that during adversarial training, the training loss of the DNN trained on easy samples decreased faster than that of the DNN trained on hard samples. To this end, we consider easy samples in adversarial training mentioned by Liu et al. (2021) may mainly contain low-order interactive concepts. It is because, as discussed in Section 3.1, Ren et al. (2021b) discovered that low-order interactions were robust to adversarial perturbations. Thus, the fast learning of easy samples in adversarial training can be roughly owing to the learning of low-order interactive concepts.

4 CONCLUSION

In this paper, we theoretically explain the trend of DNNs learning simple concepts. Our research specifies an explicit definition of the conceptual complexity that makes the DNN difficult to learn. Specifically, we use causal patterns in a causal graph to represent interactive concepts encoded by the DNN. We prove that low-order interactive concepts in the data are much more stable than high-order concepts, which makes low-order interactive concepts more likely to be encoded. Besides, our research can also provide new insights into several empirical findings *w.r.t.* the conceptual representation of DNNs.

ETHIC STATEMENT

This paper theoretically explains the intuition that simple concepts are more likely to be learned by deep neural networks (DNNs) than complex concepts. Beyond empirical studies, our research specifies an exact definition of the complexity of the concept that boosts the learning difficulty. There are no ethic issues with this paper.

REPRODUCIBILITY STATEMENT

We have provided the proof for all theoretical results in Appendix D. We have also provided experimental details in Section 2.2 and Appendix F. Besides, we will release the code when the paper is accepted.

REFERENCES

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40 (12):2897–2905, 2018.

Rana Ali Amjad and Bernhard C Geiger. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2225–2239, 2019.

Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pp. 272–281. PMLR, 2019.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pp. 233–242. PMLR, 2017.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Raef Bassily, Vitaly Feldman, Cristóbal Guzmán, and Kunal Talwar. Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 33: 4381–4391, 2020.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Niladri S Chatterji, Behnam Neyshabur, and Hanie Sedghi. The intriguing role of module criticality in the generalization of deep networks. *arXiv preprint arXiv:1912.00528*, 2019.

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.

Huiqi Deng, Na Zou, Mengnan Du, Weifu Chen, Guocan Feng, and Xia Hu. A general taylor framework for unifying and revisiting attribution methods. *arXiv preprint arXiv:2105.13841*, 2021.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

Stanislav Fort, Paweł Krzysztof Nowak, Stanislaw Jastrzebski, and Srini Narayanan. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*, 2019.

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.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pp. 1225–1234. PMLR, 2016.

John C Harsanyi. A simplified bargaining model for the n-person cooperative game. *International Economic Review*, 4(2):194–220, 1963.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. *Advances in neural information processing systems*, 7, 1994.

Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: Axiomatic feature interactions for deep networks. *J. Mach. Learn. Res.*, 22:104–1, 2021.

S Kornblith, M Norouzi, H Lee, and G Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Yunwen Lei and Yiming Ying. Fine-grained analysis of stability and generalization for stochastic gradient descent. In *International Conference on Machine Learning*, pp. 5809–5819. PMLR, 2020.

Chen Liu, Zhichao Huang, Mathieu Salzmann, Tong Zhang, and Sabine Süsstrunk. On the impact of hard adversarial instances on overfitting in adversarial training. *arXiv preprint arXiv:2112.07324*, 2021.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Karttikeya Mangalam and Vinay Uday Prabhu. Do deep neural networks learn shallow learnable examples first? 2019.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *Advances in neural information processing systems*, 27, 2014.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31, 2018.

Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

Razvan Pascanu, Guido Montufar, and Yoshua Bengio. On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv preprint arXiv:1312.6098*, 2013.

Judea Pearl. *Causality*. Cambridge university press, 2009.

William Peebles, John Peebles, Jun-Yan Zhu, Alexei Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *European Conference on Computer Vision*, pp. 581–597. Springer, 2020.

Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. *Advances in Neural Information Processing Systems*, 34:18420–18432, 2021.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Jie Ren, Mingjie Li, Qihan Ren, Huiqi Deng, and Quanshi Zhang. Towards axiomatic, hierarchical, and symbolic explanation for deep models. *arXiv preprint arXiv:2111.06206*, 2021a.

Jie Ren, Die Zhang, Yisen Wang, Lu Chen, Zhanpeng Zhou, Yiting Chen, Xu Cheng, Xin Wang, Meng Zhou, Jie Shi, et al. A unified game-theoretic interpretation of adversarial robustness. *arXiv preprint arXiv:2103.07364*, 2021b.

Carlos Santiago, Catarina Barata, Michele Sasdelli, Gustavo Carneiro, and Jacinto C Nascimento. Low: Training deep neural networks by learning optimal sample weights. *Pattern Recognition*, 110:107585, 2021.

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

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

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

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *International Conference on Machine Learning*, pp. 9259–9268. PMLR, 2020.

Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. Faith-shap: The faithful shapley interaction index. *arXiv preprint arXiv:2203.00870*, 2022.

Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. *arXiv preprint arXiv:1705.04977*, 2017.

Michael Tsang, Hanpeng Liu, Sanjay Purushotham, Pavankumar Murali, and Yan Liu. Neural interaction transparency (nit): Disentangling learned interactions for improved interpretability. *Advances in Neural Information Processing Systems*, 31, 2018.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. Crossweigh: Training named entity tagger from imperfect annotations. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pp. 5154–5163. Association for Computational Linguistics, 2019.

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. *arXiv preprint arXiv:1801.10578*, 2018.

Da Xu, Yuting Ye, and Chuanwei Ruan. Understanding the role of importance weighting for deep learning. In *International Conference on Learning Representations*, 2020.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

Zhiqin John Xu. Understanding training and generalization in deep learning by fourier analysis. *arXiv preprint arXiv:1808.04295*, 2018.

Hao Zhang, Sen Li, Yinchao Ma, Mingjie Li, Yichen Xie, and Quanshi Zhang. Interpreting and boosting dropout from a game-theoretic view. *arXiv preprint arXiv:2009.11729*, 2020.

A MORE DISCUSSIONS ABOUT RELATED WORK

We put the related work section to the Appendix for more discussion.

Evaluating and explaining the representation capacity of DNNs. Formulating and evaluating the representation ability of DNNs is an emerging perspective to explain DNNs. Pascanu et al. (2013) and Montufar et al. (2014) evaluated the representation capacity of DNNs based on the number of linear response regions. Kornblith et al. (2019), Raghu et al. (2017), and Morcos et al. (2018) analyzed representations similarity between DNNs by using canonical correlation analysis. The information-bottleneck theory (Shwartz-Ziv & Tishby, 2017) quantified information encoded in DNNs, and was extended to improve the representation capacity of DNNs (Achille & Soatto, 2018; Amjad & Geiger, 2019; Hjelm et al., 2018). Xu (2018) and Xu et al. (2019) explained the generalization of DNNs from the perspective of Fourier analysis. Furthermore, several metrics were proposed to evaluate the robustness or generalization capacity of DNNs, including the flatness of loss functions at minima (Hochreiter & Schmidhuber, 1994; Dinh et al., 2017; Petzka et al., 2021), the stability of optimization (Bousquet & Elisseeff, 2002; Hardt et al., 2016; Lei & Ying, 2020; Bassily et al., 2020), the CLEVER score (Weng et al., 2018), the stiffness (Fort et al., 2019), and the sensitivity metrics (Novak et al., 2018).

In contrast to previous empirical studies, we mathematically formulate concepts encoded by DNNs and theoretically prove that DNNs mainly learn simple concepts.

Interactions. Many studies investigated interactions between input variables of DNNs in recent years. Grabisch & Roubens (1999) proposed the Shapley interaction index to measure the interaction between players in a cooperative game, based on the Shapley value (Shapley, 1953). Lundberg et al. (2018) used the Shapley interaction index to analyze tree ensembles. Sundararajan et al. (2020) proposed the Shapley-Taylor interaction index, and Tsai et al. (2022) defined Faith-Shap, which was another interaction index. Janizek et al. (2021) explained the pairwise feature interaction in DNNs by extending Integrated Gradients (Sundararajan et al., 2017). Tsang et al. (2018) and Peebles et al. (2020) disentangled features via restricting interactions. Tsang et al. (2017) proposed to interpret DNNs via detecting statistical interactions between DNNs' weights.

In this paper, we use interactions between input variables of a DNN to represent concepts encoded by the DNN. In this way, We theoretically explain and empirically verify that DNN is easier to learn simple interactive concepts.

B AXIOMS AND THEOREMS FOR THE HARSANYI DIVIDEND INTERACTION

In this section, we introduce that the Harsanyi dividend interaction $I(S)$ satisfies several desirable axioms and theorems.

The Harsanyi dividend interactions $I(S)$ satisfies the *efficiency, linearity, dummy, symmetry, anonymity, recursive* and *interaction distribution* axioms, as follows.

(1) *Efficiency axiom* (proved by (Harsanyi, 1963)). The reward of a neural network can be decomposed into interaction effects of different contexts, *i.e.* $v(N) = \sum_{S\subseteq N} I(S)$.

(2) *Linearity axiom*. If we merge rewards of two neural networks w and v as the reward of model u, *i.e.* $\forall S \subseteq N,\ u(S) = w(S) + v(S)$, then their interaction effects $I_v(S)$ and $I_w(S)$ can be represented as $\forall S \subseteq N, I_u(S) = I_v(S) + I_w(S)$.

(3) *Dummy axiom*. If a variable $i \in N$ has no interaction with other variables, $\forall S \subseteq N\backslash\{i\}, I(S \cup \{i\}) = 0$, then this variable is denoted as a dummy variable, *i.e.* $\forall S \subseteq N\backslash\{i\}, v(S \cup \{i\}) = v(S) + v(\{i\})$.

(4) *Symmetry axiom*. Given two input variables $i, j \in N$, if $\forall S \subseteq N\backslash\{i, j\}, v(S \cup \{i\}) = v(S \cup \{j\})$, then $\forall S \subseteq N\backslash\{i, j\}, I(S \cup \{i\}) = I(S \cup \{j\})$.

(5) *Anonymity axiom*. For any permutations π on N, we have $\forall S \subseteq N, I_v(S) = I_{\pi v}(\pi S)$, where $\pi S\{\pi(i)|i \in S\}$, and the new model πv is defined by $(\pi v)(\pi S)=v(S)$. This indicates that interaction effects are not changed by permutation.

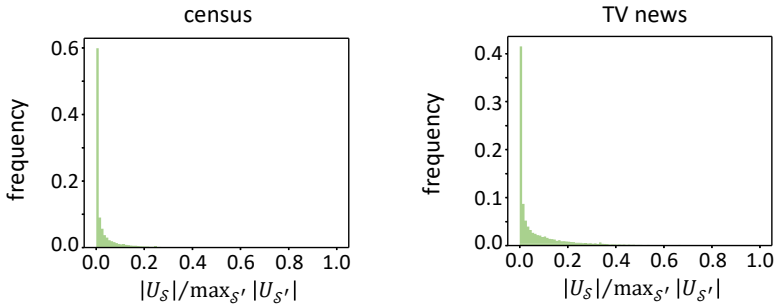

Figure 8: The average histograms of absolute causal effects of causal patterns encoded by five-layer MLPs.

(6) *Recursive axiom*. The interaction effects can be calculated recursively. Given $i \in N$ and $S \subseteq N \backslash \{i\}$, the interaction effect of the pattern $S \cup \{i\}$ can be represented as the interaction effect of S with i minus the interaction effect of S without i, *i.e.* $\forall S \subseteq N \backslash \{i\}, I(S \cup \{i\}) = I(S|i \text{ is always present}) - I(S)$. $I(S|i \text{ is always present})$ denotes the interaction effect when the variable i is always present as a constant context, *i.e.* $I(S|i \text{ is always present}) = \sum_{L \subseteq S} (-1)^{|S|-|L|} \cdot v(L \cup \{i\})$.

C CONCISENESS OF INTERACTIVE CONCEPTS

In this section, we conducted experiments to verify the conciseness of causal effects, which is introduced in Section 2.1. To this end, we computed causal effects U_S of all 2^n causal patterns encoded by a DNN by following (Ren et al., 2021a). Specifically, we trained a five-layer MLP on the *census* dataset and the *TV news* dataset) (Asuncion & Newman, 2007), respectively. For better visualization, we re-scaled causal effects U_S by $|U_S| / \max_{S' \subseteq N} |U_{S'}|$. Moreover, the strength of causal effects are average over different samples in each dataset. Fig. 8 shows histograms of absolute causal effects of causal patterns.

D PROOF OF THEOREMS

D.1 PROOF OF THEOREM 1 IN THE MAIN PAPER

Theorem 1. *For each DNN v, there exists a specific causal graph, such that for any arbitrarily masked sample $\boldsymbol{x}_T$, the output of a DNN $v(\boldsymbol{x}_T)$ can be accurately represented as the output of the causal graph. I.e., $\exists \Omega \subseteq 2^N, \exists \{U_S | S \in \Omega\}$, such that*

$$\forall\, T \subseteq N, \quad v(\boldsymbol{x}_T) = Y(\boldsymbol{x}_T) = \sum\nolimits_{S \subseteq T} U_S \tag{7}$$

In fact, Ren et al. (2021a) have provided proofs of Theorem 1. Specifically, they proved that when the causal effect U_S of the causal graph is measured by the Harsanyi dividend (Harsanyi, 1963), *i.e.*, $U_S = \sum_{T \subseteq S} (-1)^{|S|-|T|} v(\boldsymbol{x}_T)$, the output of the specific causal graph $Y(\boldsymbol{x}_T)$ can well mimic the output of a DNN $v(\boldsymbol{x}_T)$ on all potential masked samples $\boldsymbol{x}_T$, *i.e.*, $\forall T \subseteq N, v(\boldsymbol{x}_T) = Y(\boldsymbol{x}_T)$.

Proof. According to the SCM in Eq. (3), we have $Y(\boldsymbol{x}_T) = \sum_{S\in\Omega} U_S \cdot C_S(\boldsymbol{x}_T) = \sum_{S\subseteq T} U_S$. Hence, we only need to prove that $\forall\, T \subseteq N, v(\boldsymbol{x}_T) = \sum_{S\subseteq T} U_S$. Specifically,

$$\begin{aligned}\sum_{S\subseteq T} U_S &= \sum_{S\subseteq T}\sum_{L\subseteq S}(-1)^{|S|-|L|}v(\boldsymbol{x}_L)\\ &= \sum_{L\subseteq T}\sum_{S\subseteq T:S\supseteq L}(-1)^{|S|-|L|}v(\boldsymbol{x}_L)\\ &= \sum_{L\subseteq T}\sum_{s=1}^{|S|}\sum_{S\subseteq T:S\supseteq L,|S|=s}(-1)^{s-|L|}v(\boldsymbol{x}_L)\\ &= \sum_{L\subseteq T} v(\boldsymbol{x}_L)\sum_{m=0}^{|T|-|L|}\binom{|T|-|L|}{m}(-1)^m = v(\boldsymbol{x}_T)\end{aligned} \tag{8}$$

□

D.2 PROOF OF THEOREM 2 IN THE MAIN PAPER

Theorem 2. *Given a pre-trained DNN v and an arbitrary (masked or not) sample $\boldsymbol{x}' \in \mathbb{R}^n$, we use the Taylor expansion to decompose the output of this DNN by following (Deng et al., 2021). The causal effect $I(S) \in \{U_S, 0\}$ in Eq. (3) is a binary variable, since the causal graph only consider the binary masking state of each input variable. Then, we extend the binary causal effect into a continuous function $I(S|\boldsymbol{x}')$ based on the Taylor expansion, which can well fit $I(S)$ on all 2^n samples with 2^n different masking states $\boldsymbol{x}' \in \{\boldsymbol{x}_T|\forall T \subseteq N\}$*

$$v(\boldsymbol{x}') = \sum\nolimits_{S\subseteq N}\sum\nolimits_{\boldsymbol{\pi}\in Q_s} U_{S,\boldsymbol{\pi}}\cdot J(S,\boldsymbol{\pi}|\boldsymbol{x}') \quad\Rightarrow\quad I(S|\boldsymbol{x}') \quad= \sum\nolimits_{\boldsymbol{\pi}\in Q_s} U_{S,\boldsymbol{\pi}}\cdot J(S,\boldsymbol{\pi}|\boldsymbol{x}'). \tag{9}$$

$J(S,\boldsymbol{\pi}|\boldsymbol{x}') = \prod_{i\in S}(\text{sign}(x'_i - b_i)\frac{x'_i-b_i}{\tau})^{\pi_i}$ denotes a Taylor expansion term of the degree $\boldsymbol{\pi} \in Q_s = \{[\pi_1,\cdots,\pi_n]|\forall i \in S, \pi_i \in \mathbb{N}^+; \forall i \notin S, \pi_i = 0\}$. $U_{S,\boldsymbol{\pi}} = \frac{\tau^m}{\prod_{i=1}^n \pi_i!}\cdot\frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1}\cdots\partial x_n^{\pi_n}}\cdot\prod_{i\in S}(\text{sign}(x'_i-b_i))^{\pi_i}$, $m = \sum_{i=1}^n \pi_i$. $v(\boldsymbol{x}_\emptyset)$ indicates the network output, when we mask all input variables. Moreover, $C_S(\boldsymbol{x}') = I(S|\boldsymbol{x}')/U_S$.

Proof. Let us denote the continuous function on the right of Eq.(9) by $\tilde{I}(S|\boldsymbol{x}')$, *i.e.*,

$$\tilde{I}(S|\boldsymbol{x}') = \sum\nolimits_{\pi\in Q_S} U_{S,\pi} J(S,\pi|\boldsymbol{x}')$$

We need to prove that on all the 2^n masked samples, $\tilde{I}(S|\boldsymbol{x}') = I(S|\boldsymbol{x}') \in \{U_S, 0\}$.

We prove this theorem in two steps. (i) In the first step, we prove that $\tilde{U}_S \overset{\text{def}}{=} \tilde{I}(S|\boldsymbol{x})$ on the given sample $\boldsymbol{x}$ also satisfies the faithfulness requirement in Eq. (7). Furthermore, Grabisch & Roubens (1999) and Ren et al. (2021a) has proved that the Harsanyi dividend $U_S = I(S|\boldsymbol{x})$ is the unique metric to satisfy Eq. (7). Therefore, we can obtain that $\tilde{U}_S = U_S$. (ii) In the second step, we prove that on all the 2^n masked samples $\boldsymbol{x}' \in \{\boldsymbol{x}_T|\forall T \subseteq N\}$, $\tilde{I}(S|\boldsymbol{x}') = I(S|\boldsymbol{x}') \in \{U_S, 0\}$.

Proof of Step 1. We aim to prove that $\tilde{U}_S = \tilde{I}(S|\boldsymbol{x}) = \sum_{\pi\in Q_S} U_{S,\pi} J(S,\pi|\boldsymbol{x})$ also satisfies Eq. (7). Specifically, for an arbitrary masked sample $\boldsymbol{x_T}$, let us consider the Taylor expansion of $v(\boldsymbol{x}_T)$ which is expanded at $\boldsymbol{x}_\emptyset$. Then, we have

$$\forall T\subseteq N,\quad v(\boldsymbol{x}_T) = \sum_{\pi_1=0}^{\infty}\sum_{\pi_2=0}^{\infty}\cdots\sum_{\pi_n=0}^{\infty}\frac{1}{\prod_{i=1}^n\pi_i!}\frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1}\cdots\partial x_n^{\pi_n}}\cdot\prod_{i=1}^n[(\boldsymbol{x}_T)_i - b_i]^{\pi_i} \tag{10}$$

where $\boldsymbol{\pi} \in \{[\pi_1,\ldots,\pi_n]|\forall i \in N, \pi_i \in \mathbb{N}\}$ denotes the degree vector of Taylor expansion terms. In addition, $m = \sum_{i=1}^n \pi_i$.

According to the definition of the masked sample $\boldsymbol{x}_T$, we have that $\forall i \notin T$, $(\boldsymbol{x}_T)_i = b_i$ and hence $\forall i \notin T, [(\boldsymbol{x}_T)_i - b_i]^{\pi_i} = 0$. Then, among all Taylor expansion terms, only terms corresponding to degrees $\boldsymbol{\pi}$ in the set $P = \{[\pi_1,\ldots,\pi_n]|\forall i \in T, \pi_i \in \mathbb{N}; \forall i \notin T, \pi_i = 0\}$ may not be zero. Therefore, Eq. (10) can be re-written as follows.

$$\forall T\subseteq N,\quad v(\boldsymbol{x}_T) = \sum_{\pi\in P}\frac{1}{\prod_{i=1}^n\pi_i!}\frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1}\cdots\partial x_n^{\pi_n}}\cdot\prod_{i\in T}(x_i - b_i)^{\pi_i}$$

We find that the set P can be divided into multiple disjoint sets as follows, $P = \cup_{S\subseteq T} Q_S$, where $Q_S = \{[\pi_1, \ldots, \pi_n] | \forall i \in S, \pi_i \in \mathbb{N}^+; \forall i \notin S, \pi_i = 0\}$. Then, we can derive that

$$\begin{aligned}\forall T \subseteq N, \quad v(\boldsymbol{x}_T) &= \sum_{S\subseteq T} \sum_{\pi\in Q_S} \frac{1}{\prod_{i=1}^n \pi_i!} \frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1} \cdots \partial x_n^{\pi_n}} \cdot \prod_{i\in S} (x_i - b_i)^{\pi_i} \\ &= \sum_{S\subseteq T} \sum_{\pi\in Q_S} \underbrace{\frac{\tau^m}{\prod_{i=1}^n \pi_i!} \frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1} \cdots \partial x_n^{\pi_n}} \prod_{i\in S} (\delta_i)^{\pi}}_{\text{termed } U_{S,\pi}} \cdot \underbrace{\prod_{i\in S'} (\delta_i \frac{x_i - b_i}{\tau})^{\pi_i}}_{\text{termed } J(S,\pi|\boldsymbol{x})},\end{aligned} \tag{11}$$

where $\tau \in \mathbb{R}$ is a pre-defined constant and $\delta_i = \text{sign}(x_i - b_i)$. Then, Eq. (11) can be re-written as,

$$\forall T \subseteq N,\ v(\boldsymbol{x}_T) = \sum\nolimits_{S\subseteq T} \tilde{U}_S$$

i.e., $\{\tilde{U}_S | S \subseteq N\}$ also satisfies the faithfulness requirement in Eq. (7). Furthermore, Grabisch & Roubens (1999) and Ren et al. (2021a) has proved that the Harsanyi dividend $U_S = I(S|\boldsymbol{x})$ is the unique metric to satisfy Eq. (7). Therefore, we can obtain that $\tilde{U}_S = U_S$.

Proof of Step 2. We aim to prove that for a specific interactive concept S, $I(S|\boldsymbol{x}') = \tilde{I}(S|\boldsymbol{x}')$ holds for all the 2^n masked samples $\boldsymbol{x}' \in \{\boldsymbol{x}_T | \forall T \subseteq N\}$.

Specifically, for the interactive concept S, let us divided all masked samples $\boldsymbol{x}_T$ into two groups, (i) $\{\boldsymbol{x}_T | S \subseteq T\}$ and (ii) $\{\boldsymbol{x}_T | S \not\subseteq T\}$. According to the SCM in Eq. (3), we can obtain that

$$I(S|\boldsymbol{x}_T) = U_S \cdot \mathbb{1}(S \subseteq T) = \begin{cases} U_S, & \text{if } S \subseteq T; \\ 0, & \text{if } S \not\subseteq T. \end{cases} \tag{12}$$

According to the definition of $\tilde{I}(S|\boldsymbol{x}')$, it is easy to obtain that when $S \subseteq T$, $\tilde{I}(S|\boldsymbol{x}_T) = \tilde{U}_S = U_S$; otherwise, $\tilde{I}(S|\boldsymbol{x}_T) = 0$. Then, Theorem 2 holds. □

D.3 PROOF OF THEOREM 3 IN THE MAIN PAPER

Theorem 3. *Let us add a Gaussian perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ to the input sample $\boldsymbol{x}$. If we only consider the approximation based on the lowest degree $\hat{\boldsymbol{\pi}}$, then according to Theorem 2, the mean and variance of $I(S|\boldsymbol{x} + \boldsymbol{\epsilon})$ over different perturbations are given as*

$$\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x} + \boldsymbol{\epsilon})] = U_{S,\hat{\boldsymbol{\pi}}}, \quad \text{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x} + \boldsymbol{\epsilon})] = U_{S,\hat{\boldsymbol{\pi}}}^2 \left[\left(1 + (\sigma/\tau)^2\right)^s - 1 \right].$$

Proof. If we only consider Taylor expansion term of the lowest degree, then $I(S|\boldsymbol{x}') \approx U_{S,\hat{\boldsymbol{\pi}}} \cdot J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}')$, where $J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}') = \prod_{i\in S} \text{sign}(x_i' - b_i) \cdot \frac{x_i' - b_i}{\tau}$.

Let us add a Gaussian perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ to the input sample $\boldsymbol{x}$. Then, we have

$$\begin{aligned} I(S|\boldsymbol{x} + \boldsymbol{\epsilon}) &\approx U_{S,\hat{\boldsymbol{\pi}}} \cdot J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x} + \boldsymbol{\epsilon}) \\ J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x} + \boldsymbol{\epsilon}) &= \prod_{i\in S} \text{sign}(x_i + \epsilon_i - b_i) \cdot \frac{x_i + \epsilon_i - b_i}{\tau} \\ &= \prod_{i\in S} \left(\text{sign}(x_i + \epsilon_i - b_i) \cdot \frac{x_i - b_i}{\tau} + \text{sign}(x_i + \epsilon_i - b_i) \cdot \frac{\epsilon_i}{\tau} \right) \end{aligned} \tag{13}$$

According to the setting of the baseline value, we have $\forall i \in S, x_i - b_i \in \{-\tau, \tau\}$. In Section 2.2, we have assumed that the perturbation is small, *i.e.*, $\forall i \in S, |\epsilon_i| \leq \tau$. In this way, we have $\text{sign}(x_i + \epsilon_i - b_i) = \text{sign}(x_i - b_i)$, and we can obtain

$$\begin{aligned} J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x} + \boldsymbol{\epsilon}) &= \prod_{i\in S} \left(\text{sign}(x_i - b_i) \cdot \frac{x_i - b_i}{\tau} + \text{sign}(x_i - b_i) \cdot \frac{\epsilon_i}{\tau} \right) \\ &= \prod_{i\in S} \left(1 + \text{sign}(x_i - b_i) \cdot \frac{\epsilon_i}{\tau} \right) \end{aligned} \tag{14}$$

$$\begin{aligned}\Rightarrow \mathbb{E}_{\boldsymbol{\epsilon}}[J(S,\hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S}\left(1+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)\right]\\ \operatorname{Var}_{\boldsymbol{\epsilon}}[J(S,\hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \operatorname{Var}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S}\left(1+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)\right]\end{aligned} \tag{15}$$

Since $\operatorname{sign}(x_i-b_i)\in\{-1,1\}$, we have $1+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\sim\mathcal{N}(1,(\sigma/\tau)^2),\forall i\in S$.

Proposition 1. *If random variables $X_1,X_2,\cdots,X_k$ are independent of each other, then $\mathbb{E}[X_1X_2\cdots X_k]=\prod_{i=1}^k\mathbb{E}[X_i]$, and $\operatorname{Var}[X_1X_2\cdots X_k]=\prod_{i=1}^k(\mathbb{E}[X_i]^2+\operatorname{Var}[X_i]^2)-\prod_{i=1}^k\mathbb{E}[X_i]^2$.*

According to the above proposition, we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon}}[J(S,\hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \prod_{i\in S}1=1\\ \operatorname{Var}_{\boldsymbol{\epsilon}}[J(S,\hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \prod_{i\in S}\left(1^2+(\sigma/\tau)^2\right)-\prod_{i\in S}1^2\\ &= \left(1+(\sigma/\tau)^2\right)^{|S|}-1\end{aligned} \tag{16}$$

Therefore,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \mathbb{E}_{\boldsymbol{\epsilon}}[U_{S,\hat{\boldsymbol{\pi}}}\cdot J(S,\hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})]=U_{S,\hat{\boldsymbol{\pi}}}\\ \operatorname{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \operatorname{Var}_{\boldsymbol{\epsilon}}[U_{S,\hat{\boldsymbol{\pi}}}\cdot J(S,\hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})]=U_{S,\hat{\boldsymbol{\pi}}}^2\left(\left(1+(\sigma/\tau)^2\right)^{|S|}-1\right)\end{aligned} \tag{17}$$

□

D.4 PROOF OF THEOREM 4 IN THE MAIN PAPER

Theorem 4. *For an arbitrary degree $\boldsymbol{\pi}\in Q_s=\{[\pi_1,\cdots,\pi_n]|\forall i\in S,\pi_i\in\mathbb{N}^+;\forall i\notin S,\pi_i=0\}$, the mean and variance of $J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})$ can be computed as*

$$\mathbb{E}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]=\mathbb{E}_{\boldsymbol{\epsilon}}[\textstyle\prod_{i\in S}(1+\epsilon_i/\tau)^{\pi_i}],\quad \operatorname{Var}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]=\operatorname{Var}_{\boldsymbol{\epsilon}}[\textstyle\prod_{i\in S}(1+\epsilon_i/\tau)^{\pi_i}].$$

Proof. According to Theorem 2, given an arbitrary input sample $\boldsymbol{x}'$, we have

$$J(S,\boldsymbol{\pi}|\boldsymbol{x}')=\prod\nolimits_{i\in S}\left(\operatorname{sign}(x_i'-b_i)\cdot\frac{x_i'-b_i}{\tau}\right)^{\pi_i} \tag{18}$$

Let us add a Gaussian perturbation $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})$ to the input sample $\boldsymbol{x}$. In this way, we have

$$\begin{aligned}J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon}) &= \prod_{i\in S}\left(\operatorname{sign}(x_i+\epsilon_i-b_i)\cdot\frac{x_i+\epsilon_i-b_i}{\tau}\right)^{\pi_i}\\ &= \prod_{i\in S}\left(\operatorname{sign}(x_i+\epsilon_i-b_i)\cdot\frac{x_i-b_i}{\tau}+\operatorname{sign}(x_i+\epsilon_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)^{\pi_i}\end{aligned} \tag{19}$$

According to the setting of the baseline value, $\forall i\in S, x_i-b_i\in\{-\tau,\tau\}$. We assume that the perturbation is small, *i.e.*, $\forall i\in S,|\epsilon_i|\ll\tau$. In this way, $\operatorname{sign}(x_i+\epsilon_i-b_i)=\operatorname{sign}(x_i-b_i)$, and we can obtain

$$\begin{aligned}J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon}) &= \prod_{i\in S}\left(\operatorname{sign}(x_i-b_i)\cdot\frac{x_i-b_i}{\tau}+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)^{\pi_i}\\ &= \prod_{i\in S}\left(1+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)^{\pi_i}\end{aligned} \tag{20}$$

$$
\begin{aligned}
\Rightarrow \mathbb{E}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S}\left(1+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)^{\pi_i}\right] \\
\operatorname{Var}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \operatorname{Var}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S}\left(1+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)^{\pi_i}\right]
\end{aligned} \tag{21}
$$

Since $\forall i \in S$, ϵ_i is independent of each other, according to Proposition 1 and Eq. (21), we have

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \prod_{i\in S}\mathbb{E}_{\epsilon_i}\left[\left(1+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)^{\pi_i}\right] \\
\operatorname{Var}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \prod_{i\in S}\mathbb{E}_{\epsilon_i}\left[\left(1+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)^{2\pi_i}\right] - \prod_{i\in S}\left(\mathbb{E}_{\epsilon_i}\left[\left(1+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)^{\pi_i}\right]\right)^2
\end{aligned} \tag{22}
$$

Since $\operatorname{sign}(x_i - b_i) \in \{-1, 1\}$, we have $\mathbb{E}_{\epsilon_i}\left[\left(1+\operatorname{sign}(x_i-b_i)\cdot\frac{\epsilon_i}{\tau}\right)^k\right] = \mathbb{E}_{\epsilon_i}\left[\left(1+\frac{\epsilon_i}{\tau}\right)^k\right], \forall k \in \mathbb{N}^+$. Therefore, we obtain

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \prod_{i\in S}\mathbb{E}_{\epsilon_i}\left[\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i}\right] \\
&= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S}\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i}\right] \\
\operatorname{Var}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \prod_{i\in S}\mathbb{E}_{\epsilon_i}\left[\left(1+\frac{\epsilon_i}{\tau}\right)^{2\pi_i}\right] - \prod_{i\in S}\left(\mathbb{E}_{\epsilon_i}\left[\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i}\right]\right)^2 \\
&= \operatorname{Var}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S}\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i}\right].
\end{aligned}
$$

□

E RELATION BETWEEN INTERACTIVE CONCEPTS AND MULTI-ORDER INTERACTIONS

In this section, we derive that high-order interactive concepts (computed via the Harsanyi dividend (Harsanyi, 1963)) can be considered as elementary components for high-order interactions used in (Ren et al., 2021b).

Given a pre-trained DNN v and a masked sample $\boldsymbol{x}_S$, the multi-order interaction $I^{(m)}(i,j)$ used in (Ren et al., 2021b) is given as follows:

$$I^{(m)}(i,j) = \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[\Delta v(i,j,S)], \tag{23}$$

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

Let $\Delta v_T(S) = \sum_{L\subseteq T}(-1)^{|T|-|L|}v(\boldsymbol{x}_{L\cup S})$ denote the marginal benefit of variables in $T \subseteq N\setminus S$, given the environment S. In this way, $\Delta v_T(S)$ can be represented as the sum of interaction effects inside T and sub-environments of S, *i.e.* $\Delta v_T(S) = \sum_{S'\subseteq S} I(T\cup S')$ Ren et al. (2021a).

Thus, the $I^{(m)}(i,j)$ can be represented as follows,

$$\begin{aligned}
I^{(m)}(i,j) =& \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[\Delta v(i,j,S)] \\
=& \mathbb{E}_{S\subseteq N\setminus\{i,j\},|S|=m}[\sum_{L\subseteq S} I(L\cup\{i,j\})] \\
=& \frac{1}{\binom{n-2}{m}} \sum_{\substack{S\subseteq N\setminus\{i,j\}\\|S|=m}} [\sum_{L\subseteq S} I(L\cup\{i,j\})] \\
=& \sum_{\substack{L\subseteq N\setminus\{i,j\}\\|L|\le m}} I(L\cup\{i,j\}) \sum_{\substack{L\subseteq S\subseteq N\setminus\{i,j\}\\|S|=m}} \frac{1}{\binom{n-2}{m}} \\
=& \sum_{\substack{L\subseteq N\setminus\{i,j\}\\|L|\le m}} I(L\cup\{i,j\}) \frac{\binom{n-2}{m-l}}{\binom{n-2}{m}} \\
=& \sum_{l=0}^{m} \sum_{\substack{L\subseteq N\setminus\{i,j\}\\|L|=m}} \frac{\binom{n-2}{m-l}}{\binom{n-2}{m}} I(L\cup\{i,j\}).
\end{aligned}$$

Therefore, we prove that $I^{(m)}(i,j) = \sum_{l=0}^{m}\sum_{\substack{L\subseteq N\setminus\{i,j\}\\|L|=m}} \frac{\binom{n-2}{m-l}}{\binom{n-2}{m}} I(L\cup\{i,j\})$.

F EXPERIMENTAL SETTINGS

Training details. We trained AlexNet (Krizhevsky et al., 2012), VGG-11 (Simonyan & Zisserman, 2014), ResNet-18/20 (He et al., 2016) on the CIFAR-10 dataset (Krizhevsky et al., 2009) and the Tiny-ImageNet dataset (Le & Yang, 2015), respectively. We also trained a five-layer MLP (Ren et al., 2021a) on the UCI census dataset (namely *census dataset*) and the UCI TV news dataset (namely *TV news dataset*) (Asuncion & Newman, 2007), respectively. Each layer of the MLP contained 100 neurons. We trained each neural networks for 200 epochs with the SGD optimizer.

Sampling details. Since, the computational cost of $I(S|\boldsymbol{x})$ in Remark 1 was intolerable in real implementation, we applied the sampling-based approximation method in (Zhang et al., 2020) to calculate $I(S|\boldsymbol{x})$. Due to the high dimension of image data (*e.g.* 224×224 for ImageNet), we uniformly split the input image into 8×8 patches. Furthermore, we random sampled 12 patches and considered these patches as input variables for each image. The remaining 52 patches are set to the baseline value.

Implementations details. Here, we introduce how to measure $\beta(S)$ and $\kappa(S)$ in Section 2.3. On the Tiny-ImageNet dataset, we randomly sampled 100 training images. These training images were randomly sampled from different 10 classes. On the CIFAR-10 dataset, we randomly sampled 10 training images from each class. For tabular datasets, we randomly sampled 50 training samples from each class. For image datasets, we set $\tau = 2$. For tabular datasets, we set $\tau = 1$. In this way, we set the baseline $b_i = max(x_i - \tau, \mu)$, if $x_i > \mu_i$, and we set the baseline $b_i = min(x_i + \tau, \mu)$, if $x_i < \mu_i$. For Gaussian perturbation $\boldsymbol{\epsilon}$, we set $\sigma = 0.02$. Besides, for each training sample, we randomly sampled five Gaussian perturbation with five different seeds, respectively.

Generalization power. Here, we introduce how to measure $\mathbb{E}_c[\text{sim}(\boldsymbol{H}^{(s)}_{c,\text{train}}, \boldsymbol{H}^{(s)}_{c,\text{test}})]$ in Section 3.1. For tabular datasets, we randomly sampled 50 training samples from each class from the training set. We also randomly sampled 50 testing samples from each class from the test set. For the baseline value, we set $\tau = 1.5$.

Adversarial attack. Here, we introduce how to measure $A^{(s)}$ in Section 3.1. For tabular datasets, we randomly sampled 50 training samples from each class from the training set. On the Tiny-ImageNet dataset, we randomly sampled 100 training images. These training images were randomly sampled from different 10 classes. On the CIFAR-10 dataset, we randomly sampled 10 training images from each class. We used the l_∞ untargeted PGD attack by following (Madry et al., 2018), in which the constraint $\epsilon = 16/255$, and the attack was conducted with 5 steps with the step size $\epsilon = 3/255$.