Layerwise Change of Knowledge in Neural Networks

Xu Cheng [*1 2] **Lei Cheng** [*2] **Zhaoran Peng** [2] **Yang Xu** [3] **Tian Han** [4] **Quanshi Zhang** [§ 2]

Abstract

This paper aims to explain how a deep neural network (DNN) gradually extracts new knowledge and forgets noisy features through layers in forward propagation. Up to now, although the definition of knowledge encoded by the DNN has not reached a consensus, Li & Zhang (2023); Ren et al. (2023a; 2024) have derived a series of mathematical evidence to take interactions as symbolic primitive inference patterns encoded by a DNN. We extend the definition of interactions and, for the first time, extract interactions encoded by intermediate layers. We quantify and track the newly emerged interactions and the forgotten interactions in each layer during the forward propagation, which shed new light on the learning behavior of DNNs. The layer-wise change of interactions also reveals the change of the generalization capacity and instability of feature representations of a DNN.

1. Introduction

Recently, understanding the black-box representation of deep neural networks (DNNs) has received increasing attention. This paper investigates how a DNN gradually extracts knowledge from the input for inference during the layerwise forward propagation, although the definition of *knowledge* encoded by an AI model is still an open problem. To this end, the information bottleneck theory (Shwartz-Ziv & Tishby, 2017; Saxe et al., 2018) uses mutual information between the input and the intermediate-layer feature to measure knowledge encoded in each layer. It finds that the DNN fits (learns) task-relevant information, and compresses task-irrelevant information. Liang et al. (2020) extract common feature components shared by different features as the shared knowledge.

In this paper, we aim to define and quantify the knowledge encoded in each layer. In this way, we can accurately decompose and track explicit changes of knowledge (i.e., the learning of new knowledge and the forgetting of old knowledge) through different layers.

However, there is no a widely-accepted definition of knowledge, because we cannot mathematically define/formulate knowledge in human cognition. Instead of focusing on cognitive issues, Ren et al. (2023a); Li & Zhang (2023) have discovered and Ren et al. (2024) have theoretically proven[1] **the sparsity property** and **universal-matching property** of interactions, *i.e., given an input sample $\boldsymbol{x}$, a well-trained DNN usually only implicitly encodes a small number of interactions between the input variables, and the inference score can be explained as numerical effects of these interactions.* **Thus, these two properties mathematically make such interactions (also called *interaction primitives* or *interaction concepts*) be considered as the knowledge encoded by a DNN.** As Fig. 1 shows, given a dog image $\boldsymbol{x}$, each interaction implicitly encoded by the DNN represents a co-appearance relationship between input variables (image patches) in $S = \{\text{eye}, \text{nose}, \text{mouth}\}$. This is actually an AND relationship between image patches in image $\boldsymbol{x}$. Only when all patches in S are present in the image, the interaction S is activated and makes a numerical effect $I(S|\boldsymbol{x})$ on the classification score. Masking[4] any patch will deactivate the interaction S and remove the effect.

Although the above studies make it plausible to define and quantify interactions encoded by a DNN, our target of quantifying and tracking the interactions encoded by different layers presents the following three new challenges.
(1) **Alignment of interaction primitives.** The fair comparison between any arbitrary pair of layers requires interaction primitives extracted from different layers to be aligned, although the physical feature dimensions in different layers do not have a clear correspondence/alignment.
(2) **Decomposability and countability of knowledge.** In-

[*] Equal contribution [1] Nanjing University of Science and Technology. [2] Shanghai Jiao Tong University. [3] Zhejiang University. [4] Stevens Institute of Technology.

Correspondence to: Quanshi Zhang. Quanshi Zhang is the corresponding author. He is with the Department of Computer Science and Engineering, the John Hopcroft Center, at the Shanghai Jiao Tong University, China. <zqs1022@sjtu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1] Ren et al. (2024) have proven that the sparsity of interactions can be guaranteed by three common conditions for the DNN's smooth inferences on randomly masked samples.

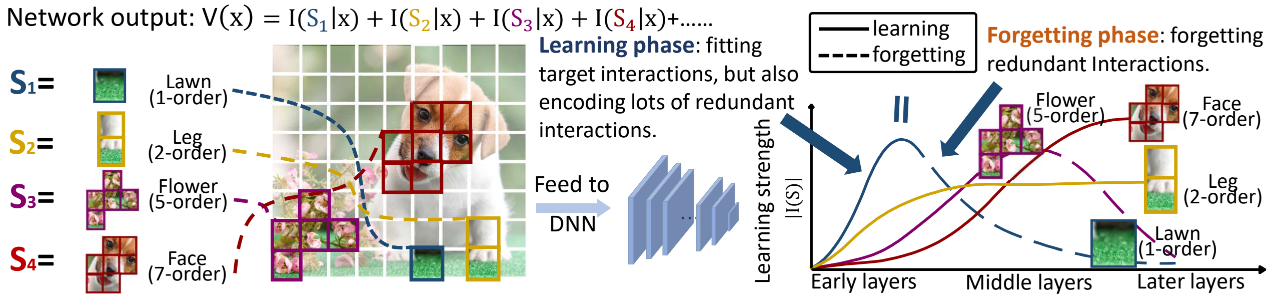


Figure 1. Tracking interactions through layers in the DNN. In most DNNs, early and middle layers usually fit target interactions modeled by the entire network at the cost of encoding lots of redundant interactions, and later layers remove such redundant interactions.

teractions help us overcome the challenge of representing uncountable knowledge as countable primitive patterns. In this way, we can exactly quantify how many interaction primitives are newly emerged and forgotten in each layer.
(3) **Connection to the generalization capacity.** We hope to provide deep insights into how newly merged interaction primitives and forgotten old interaction primitives are related to the generalization capacity of a DNN.

Therefore, considering above challenges, we extend the definition of interactions to intermediate layers of a DNN. Specifically, given features of a certain layer, we train a linear classifier[2] to use these features for classification, and extract a set of interactions from the classifier. We analyze the faithfulness of the newly proposed interaction towards the intermediate layers of a DNN, and we discover that the new interactions provide us with a more straightforward way to analyze how knowledge changes in the layerwise forward propagation. Instead of directly aligning features in different layers, we find that adjacent layers in a DNN usually encode similar sets of interactions. Thus, as illustrated in Fig. 1, we can clarify the emergence of new interactions and the forgetting of old interactions in each layer.

Faithfulness of interactions. More crucially, the newly defined interaction primitives still belong to the typical paradigm of interactions, so that there are a series of theorems (Ren et al., 2023a; Li & Zhang, 2023; Ren et al., 2024) as convincing evidence to take countable/symbolic interactions as primitive inference patterns to represent uncountable knowledge in a DNN. Please See Section 2 and Section 3.1 for details.

In this way, we can use interactions to explain the change of the representation capacity of features in different layers from the following two perspectives, which can help both theoreticians and practitioners gain new insights into the learning behavior of a DNN.

- **The tracking of countable interactions in different layers reveals the change of representation complexity over different layers.** The complexity of an interaction S is defined as the number of input variables in S, which is also termed the *order* of this interaction, *i.e.*, $order(S) = |S|$. In experiments, we discover that in most DNNs, early and middle layers are usually trained to fit target interactions encoded by the entire network at the cost of encoding lots of redundant interactions, and later layers remove such redundant interactions.

- **Redefining the generalization capacity of DNNs and tracking generalizable interactions.** The use of interaction primitives enables us to redefine the generalization power of a DNN from a new perspective. That is, given multiple DNNs trained for the same task, if these DNNs encode similar interactions, then we consider interactions shared by different DNNs generalizable. We discover that low-order interactions usually have stronger generalization capacity than high-order interactions. Besides, we also discover that low-order interactions encoded by the DNN usually exhibit more consistent effects $I(S|\boldsymbol{x'} = \boldsymbol{x}+\epsilon)$ when we add different small noises ϵ to the input sample $\boldsymbol{x}$. In comparison, high-order interactions often exhibit diverse effects $I(S|\boldsymbol{x'})$ on inference scores *w.r.t.* different noises ϵ. This indicates that low-order interactions often have higher stability.

Contributions of this study are summarized as follows.
(1) We redefine the interaction on intermediate layers, and find that the new definition ensures adjacent layers to encode similar interactions.
(2) Our study provides several theoretically verifiable metrics to quantify the newly emerged knowledge and forgotten knowledge in the forward propagation.
(3) The change of interactions is also found to be related to the generalization power of a DNN.

[2] Belinkov (2022) discussed techniques and limitations of classifier probes. Please See Appendix D for the solutions to these problems.

2. Literature in Explaining Knowledge in DNNs

Explaining and quantifying the exact knowledge encoded by a DNN presents a significant challenge to explainable AI. So far, there has not existed a widely accepted definition of knowledge that enables us to accurately disentangle and quantify knowledge encoded by intermediate layers of a DNN, because it covers multiple disciplinary issues, such as cognitive science, neuroscience, *etc*. To explain and quantify the exact knowledge encoded by a DNN, previous studies have either associated units of DNN feature maps with manually annotated semantics/concepts (Bau et al., 2017; Kim et al., 2018) or automatically learned meaningful patterns from data (Chen et al., 2019; Shen et al., 2021; Zhang et al., 2020), but they failed to provide a mathematically guaranteed boundary for the scope of each concept/knowledge. Thus, previous studies could not accurately quantify the exact amount of newly emerged/forgotten/unexplainable knowledge in each layer. Appendix A provides further discussions of more methods (Kolchinsky et al., 2019; Liang et al., 2020; Saxe et al., 2018; Shwartz-Ziv & Tishby, 2017; Wang et al., 2022).

Faithfulness of using interaction primitives to define knowledge in DNNs. Although there is no theory to guarantee that salient interactions can exactly fit the so-called *knowledge* in human cognition, a series of studies have empirically verified and theoretically ensured the faithfulness of interaction primitives from the following perspectives.

(1) Li & Zhang (2023) have observed and Ren et al. (2024) have partially proven[1] that most DNNs encode a few interactions with salient effect $I(S|\boldsymbol{x})$ on the network output.

(2) Li & Zhang (2023) have observed that interactions exhibited considerable **generalization capacity** across samples and across models. Besides, they have also discovered that salient interactions exhibited remarkable **discrimination power** in classification tasks.

(3) Ren et al. (2023a) have proven seven desirable mathematical properties for interactions.

(4) Interaction primitives can also be used to explain the representation capacity of DNNs. Deng et al. (2022) have proven a counter-intuitive bottleneck of a DNN in encoding interaction primitives of the intermediate complexity. Liu et al. (2023) have proven the learning difficulty of interaction primitives. Zhou et al. (2024) have discovered that low-order interactions have higher generalization power than high-order interactions.

Furthermore, we compare the interaction-based explanation with attribution interpretability methods. Please Appendix B for detailed discussions.

3. Tracking Interactions through Layers

3.1. Preliminaries: using interactions to represent knowledge in DNNs

So far, there is not a widely accepted way to define knowledge encoded by a DNN, because the definition of knowledge is an interdisciplinary problem over cognitive science, neuroscience, and mathematics. Li & Zhang (2023) has derived a series of properties as convincing evidence to define interactions as symbolic primitive inference patterns encoded by a DNN (please see Section 2 for details). Thus, in this paper, we extend the definition to quantify the change of interactions in the layer-wise forward propagation. Specifically, there are two types of interactions, including AND interactions and OR interactions.

Definition 3.1 (**AND interactions**). *Given an input sample $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ comprising n input variables, let $N = \{1, 2, \ldots, n\}$ denote the indices of all n input variables, and let $v(\boldsymbol{x}) \in \mathbb{R}$ denote the scalar output of the DNN or a certain dimension of the DNN*[3]. *Then, the AND interaction $I_{and}(S|\boldsymbol{x})$ is used to quantify the effect of the AND (co-appearance) relationship among a subset $S \subseteq N$ of input variables, which is encoded by the DNN v to compute the inference scores of the label y^{truth}.*

$$I_{\text{and}}(S|\boldsymbol{x}) = \sum\nolimits_{T\subseteq S}(-1)^{|S|-|T|}\cdot v(\boldsymbol{x}_T). \qquad (1)$$

Here, $\boldsymbol{x}_T$ denotes the masked[4] sample obtained by masking variables in $N \setminus T$, $v(\boldsymbol{x}_T)$ represents the output score[3] for the target label y^{truth} on the masked sample $\boldsymbol{x}_T$.

Each AND interaction with non-zero effect $I_{\text{and}}(S|\boldsymbol{x}) \neq 0$ means that the DNN encodes the AND relationship between variables in S. The network output can be represented as the sum of interaction effects $v(\boldsymbol{x}) = \sum_{S\subseteq N} I_{\text{and}}(S|\boldsymbol{x})$.

OR interactions. Ren et al. (2023a); Zhou et al. (2023) have further extended the AND interaction to the OR interaction. To this end, the overall network output is decomposed into the component for AND interactions $v_{\text{and}}(\boldsymbol{x}_T)$ and the component for OR interactions $v_{\text{or}}(\boldsymbol{x}_T)$, subject to $v_{\text{and}}(\boldsymbol{x}_T) = 0.5\cdot v(\boldsymbol{x}_T) + \gamma_T$ and $v_{\text{or}}(\boldsymbol{x}_T) = 0.5\cdot v(\boldsymbol{x}_T) - \gamma_T$. $\{\gamma_T\}$ is a set of

[3] Note that people can apply different settings for $v(\boldsymbol{x})$. Here, we follow (Deng et al., 2022) to set $v(\boldsymbol{x}) = \log \frac{p(y=y^{\text{truth}}|\boldsymbol{x})}{1-p(y=y^{\text{truth}}|\boldsymbol{x})} \in \mathbb{R}$ as the confidence of classifying the sample $\boldsymbol{x}$ to the ground-truth category y^{truth}.

[4] We mask the input variable $i \in N \setminus T$ to the baseline value b_i to represent its masked state. Here, we follow the widely-used setting of baseline values in (Dabkowski & Gal, 2017) to set b_i as the mean value of this variable across all samples in image classification, and follow (Shen et al., 2023) to set b_i as a special token (*e.g.,* [MASK] token) in nature language processing. Note that such settings of baseline values can bring in some biases (Jain et al., 2022). To remove biases, Ren et al. (2023b) proposed a method to learn optimal baseline values based on interactions. Please see Appendix E for details.

learnable parameters to determine the decomposition. In this way, people can simultaneously explain AND interactions and OR interactions encoded by DNN. The AND interaction is extracted as $I_{\text{and}}(S|\boldsymbol{x}) = \sum_{T \subseteq S} (-1)^{|S|-|T|} \cdot v_{\text{and}}(\boldsymbol{x}_T)$, just like in Eq. (1). The OR interaction is defined as follows.

Definition 3.2 (**OR interactions**)**.** *The OR interaction is used to quantify the effect of the OR relationship between a set $S \subseteq N$ of input variables encoded by the DNN..*

$$I_{\text{or}}(S|\boldsymbol{x}) = -\sum\nolimits_{T \subseteq S} (-1)^{|S|-|T|} \cdot v_{\text{or}}(\boldsymbol{x}_{N \setminus T}). \qquad (2)$$

Eq. (2) indicates that the presence of any input variable in S will activate the OR interaction and make an effect $I_{\text{or}}(S|\boldsymbol{x})$ to the output score $v(\boldsymbol{x})$. Ren et al. (2023a); Zhou et al. (2023) proposed to learn parameters $\{\gamma_T\}$ to generate the sparsest AND-OR interactions. The AND-OR interactions is determined when $\{\gamma_T\}$ are learned. Please see Zhou et al. (2023) for detailed technique of learning $\{\gamma_T\}$ for the optimal decomposition of AND-OR interactions.

Faithfulness. The sparsity property and universal-matching property mathematically guarantee the faithfulness of interaction-based explanation. Let us randomly mask[4] an input sample x and generate a total of 2^n masked samples $\boldsymbol{x}_T$. Then, Theorem 3.3 shows that output scores $v(\boldsymbol{x}_T)$ on all 2^n masked samples $\boldsymbol{x}_T$ can always be well matched by AND-OR interactions.

Theorem 3.3. ***(Proven in Appendix F)*** *Given an input sample $\boldsymbol{x} \in \mathbb{R}^n$, the network output score $v(\boldsymbol{x}_T)$ on each masked input samples $\{\boldsymbol{x}_T | T \subseteq N\}$ can be decomposed into effects of AND interactions and OR interactions, subject to $I_{and}(\emptyset|\boldsymbol{x}) = v_{and}(\boldsymbol{x}_\emptyset) = v(\boldsymbol{x}_\emptyset)$ and $I_{or}(\emptyset|\boldsymbol{x}) = v_{or}(\boldsymbol{x}_\emptyset) = 0$.*

$$\begin{aligned} v(\boldsymbol{x}_T) &= v_{and}(\boldsymbol{x}_T) + v_{or}(\boldsymbol{x}_T) \\ &= \sum\nolimits_{S \subseteq T} I_{and}(S|\boldsymbol{x}_T) + \sum\nolimits_{S \cap T \neq \emptyset} I_{or}(S|\boldsymbol{x}_T) \end{aligned} \qquad (3)$$

Ren et al. (2024) have proven[1] that most AND-OR interactions have negligible effects $I(S|\boldsymbol{x}) \approx 0$ on inference, which can be regarded as noisy patterns. Only a small number of interactions have considerable effects. Given an input sample $\boldsymbol{x} \in \mathbb{R}^n$, **we can use a small set of salient AND interactions $\Omega^{\text{and}}_{\text{salient}}$ and OR interactions $\Omega^{\text{or}}_{\text{salient}}$ to universally match network outputs $v(\boldsymbol{x}_T)$ on all 2^n masked samples**. This indicates that salient interactions can serve as **primitive inference patterns** encoded by the DNN.

Lemma 3.4. ***(Proving interactions as primitive inference patterns**, c.f. Appendix G)* *Given an input sample $\boldsymbol{x} \in \mathbb{R}^n$, the network output on all 2^n masked input samples $\{\boldsymbol{x}_T | T \subseteq N\}$ can be universally matched by a small set of salient interactions.*

$$v(\boldsymbol{x}_T) \approx v(\boldsymbol{x}_\emptyset) + \sum_{\substack{S \in \Omega^{and}_{salient} \\ \emptyset \neq S \subseteq T}} I_{and}(S|\boldsymbol{x}_T) + \sum_{\substack{S \in \Omega^{or}_{salient} \\ S \cap T \neq \emptyset}} I_{or}(S|\boldsymbol{x}_T) \qquad (4)$$

3.2. Tracking interactions through layers

Despite the universal-matching property, the transferability, and the discrimination power of interactions in Section 2, the definition, quantification, and tracking of interactions through layers present distinctive challenges in real applications. Specifically, we aim to define the interaction for the high-dimensional features, while previous interactions are all defined on a scalar output score. Besides, the newly defined interaction successfully ensures that neighboring layers encode similar interactions. This enables us to propose various metrics to track the newly emerged interaction primitives and the forgotten interaction primitives in each layer, which provide new insights into the learning of DNNs.

3.2.1. Verifying the sparsity of interactions

Before we define interactions encoded by intermediate-layer features, we need to first examine whether the final layer of the DNN encodes a small number of interactions. Although the sparsity of interactions has been partially proven under three common conditions[2], it is still a challenge to strictly examine whether the DNN fully satisfies these conditions in real applications. Besides, the sparsity of interactions has not been proven when we simultaneously use AND interactions and OR interactions to explain a DNN.

The interactions used by the final layer are directly extracted based on the network output score $v(\boldsymbol{x})$[3], according to Eq. (1) and Eq. (2). Thus, we can consider interactions extracted from the final layer as the target interactions used for the inference. If these interactions are sparse, then the utility of all layers can be simplified as pushing features towards a specific small set of sparse interactions. This will significantly simplify feature analysis.

Experiments. We conducted experiments to illustrate the sparsity of interactions. Given a well-trained DNN and an input sample $\boldsymbol{x} \in \mathbb{R}^n$, we calculated AND interactions $I_{\text{and}}(S|\boldsymbol{x})$ and OR interactions $I_{\text{or}}(S|\boldsymbol{x})$ of all 2^n possible subsets[5] $S \subseteq N$. To this end, we trained VGG-11 (Simonyan & Zisserman, 2014), ResNet-20 (He et al., 2016) on the MNIST dataset (LeCun et al., 1998) and CIFAR-10 datasets (Krizhevsky et al., 2009), respectively. We also learned a seven-layer MLP (namely *MLP-7*) on the MNIST dataset and CIFAR-10 dataset, respectively, where each layer contained 1024 neurons. Please see Appendix M.4 for experimental details.

Fig. 2 shows the strength of all AND-OR interactions extracted from different samples $\boldsymbol{x}$, $|I(S|\boldsymbol{x})|$ *w.r.t.* different S and $\boldsymbol{x}$, in descending order. We discovered only about 21.8 AND/OR salient interactions in each MNIST image and

[5] Appendix M.1 introduces the details of selecting a relatively small number of input variables (image patches or words) to compute interactions in order to reduce computational cost.

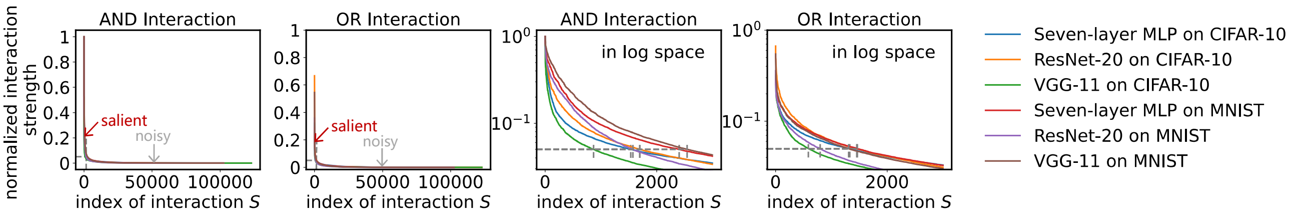


Figure 2. Sparsity of interactions. We visualized strength of all AND-OR interactions extracted from different samples $\boldsymbol{x}$, $|I(S|\boldsymbol{x})|$ *w.r.t.* different S and $\boldsymbol{x}$, in a descending order. Only about 21.8 AND/OR interactions in each sample of the MNIST dataset and about 45.6 AND/OR interactions in each sample of the CIFAR-10 dataset made salient effects on the network output.

about 45.6 AND/OR salient interactions in each CIFAR-10 image. All other interactions exhibited very small effects. Such a phenomenon verified the sparsity of interactions.

3.2.2. Extracting interactions from intermediate layers

In comparison with extracting interactions from the network output score $v(\boldsymbol{x})$[3], defining and extracting interactions from intermediate layers present a new challenge. It is because the intermediate-layer features are usually high-dimensional vectors/tensors/matrices, rather than a scalar output. Thus, we need to define a new scalar metric $v^{(l)}(\boldsymbol{x})$, which faithfully identify signals in the high-dimensional feature directly related to the classification task, to compute interactions encoded by the l-th layer of the DNN.

To this end, given an input sample $\boldsymbol{x}$, we propose to train a linear classifier $p^{(l)}(y|\boldsymbol{x}) = \textit{softmax/sigmoid}((w^{(l)})^T f^{(l)}(\boldsymbol{x}) + b^{(l)})$ based on the cross-entropy loss, which uses the feature $f^{(l)}(\boldsymbol{x})$ of the l-th layer to conduct the same classification task as the DNN[6]. We can define the following $v^{(l)}(\boldsymbol{x})$ to represent signals encoded by the l-th layer of the DNN.

$$\begin{aligned} v^{(l)}(\boldsymbol{x}) &= \log \frac{p^{(l)}(y = y^{\text{truth}}|\boldsymbol{x})}{1 - p^{(l)}(y = y^{\text{truth}}|\boldsymbol{x})} - \delta_N, \\ v^{(l)}(\boldsymbol{x}_T) &= \log \frac{p^{(l)}(y = y^{\text{truth}}|\boldsymbol{x}_T)}{1 - p^{(l)}(y = y^{\text{truth}}|\boldsymbol{x}_T)} - \delta_T, \end{aligned} \tag{5}$$

where δ_T is a learnable residual proposed to model and remove the tiny noise from the output $v^{(l)}(\boldsymbol{x}_T)$, so as to extract relatively clean interactions. δ_T is constrained to a small range $\kappa = 0.04 \cdot |v^{(l)}(\boldsymbol{x}_N) - v^{(l)}(\boldsymbol{x}_\emptyset)|$. We discover that small noise in output function $v^{(l)}(\boldsymbol{x}_T)$ may significantly change the interaction effect. In this way, parameters $\{\gamma_T, \delta_T\}$ are learned by minimizing $\sum_{T \subseteq N} |I_{\text{and}}(T|\boldsymbol{x}, v^{(l)})| + |I_{\text{or}}(T|\boldsymbol{x}, v^{(l)})|, s.t.\ \forall T \subseteq N, |\delta_T| < \kappa$. An ablation study in Appendix J shows that the extraction of interactions is relatively robust to the κ value.

Comparing interaction complexity over different layers. The new function $v^{(l)}(\boldsymbol{x})$ enables a fair comparison between interactions extracted from different layers. The classification score $v^{(l)}(\boldsymbol{x})$ potentially reflects a set of interactions, which are encoded by $f^{(l)}(\boldsymbol{x})$ and can be directly used for classification. Specifically, we conducted experiments to extract interactions from different layers of different DNNs[5]. We used the MLP-7, VGG-11, and ResNet-20 trained on the MNIST dataset and CIFAR-10 dataset, which were introduced in Section 3.2.1. We also fine-tuned pre-trained DistilBERT (Sanh et al., 2019) and BERT$_{\text{BASE}}$ (Devlin et al., 2019) models on the SST-2 dataset (Socher et al., 2013) for binary sentiment classification.

We used the order of an interaction to measure the complexity of the interaction. The order was defined as the number of input variables involved in this interaction, *i.e.*, *order*$(S) = |S|$. As illustrated in Fig. 3, linear classifiers trained on features of early layers encode less high-order interactions than later layers. We can consider that features in early layers usually represent lots of local and simple nonlinear patterns between a few input variables, but most of such patterns cannot be directly used by the classifier for the classification task. Besides, compared to linear classifiers trained on early-layer features, classifiers trained on features of later layers usually share more similar interactions with the final layer of the DNN.

Emergence of new interactions and discarding of old interactions. In this experiment, we quantified how the DNN gradually learned new interactions and discarded useless interactions in the forward propagation and obtained the target interactions in the last layer. To this end, given all AND-OR interactions encoded by the l-th layer, let $\Omega_{\text{and}}^{(l),m} = \{S \subseteq N : |S| = m, |I_{\text{and}}(S|\boldsymbol{x}, v^{(l)})| > \tau\}$[7] denote the set of salient AND interactions of the m-th order extracted from the l-th layer. Accordingly, $\Omega_{\text{or}}^{(l),m} = \{S \subseteq N : |S| = m, |I_{\text{or}}(S|\boldsymbol{x}, v^{(l)})| > \tau^7\}$ represented the set of salient OR interactions of the m-th order extracted from the l-th layer. To this end, we used $\textit{all}_{\text{and}}^{(l),m}$ and $\textit{all}_{\text{and}}^{(L),m}$ to

[6] Appendix M.2 introduces the details of training the classifier. Note that the network parameters in the DNN are all fixed without being tuned, when we learn classifiers.

[7] We set $\tau = 0.05 \cdot \max_S(|I_{\text{and}}(S|\boldsymbol{x}, v^{(l)})|, |I_{\text{or}}(S|\boldsymbol{x}, v^{(l)})|)$ to select a set of salient interactions from all interactions extracted from the l-th layer of the target DNN.

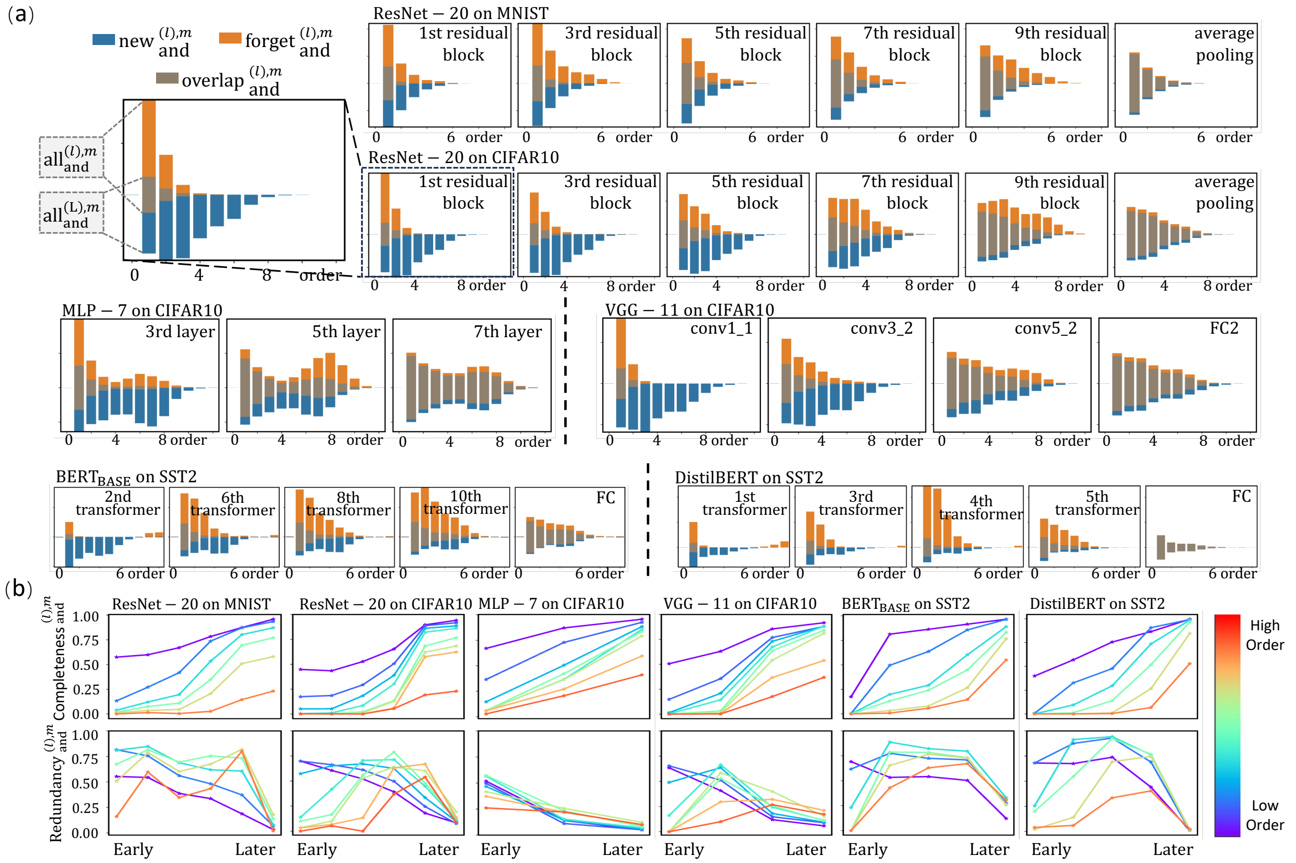


Figure 3. (a) Tracking the change of the average strength of the overlapped (*overlap*$_{\text{and}}^{(l),m}$), forgotten (*forget*$_{\text{and}}^{(l),m}$), and newly emerged interactions (*new*$_{\text{and}}^{(l),m}$) through different layers. For each subfigure, the total length of the orange bar and the grey bar equals to *all*$_{\text{and}}^{(l),m}$, and the total length of the blue bar and the grey bar equals to *all*$_{\text{and}}^{(L),m}$ (b) Tracking the change of *completeness*$_{\text{and}}^{(l),m}$ and *redundancy*$_{\text{and}}^{(l),m}$ through different layers. We do not show interactions of the highest four orders, because almost no interactions of extremely high orders were learned. Please see Appendix I for results of OR interactions and results on tabular datasets.

quantify the overall strength of all m-order salient AND interactions encoded by the l-th layer and those encoded by the final layer (the L-th layer), respectively.

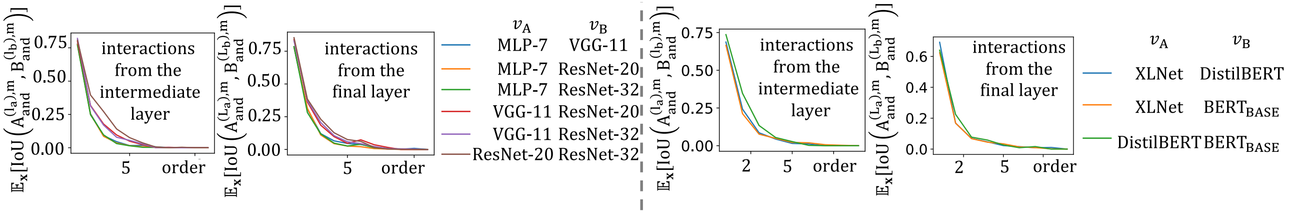
$$\begin{aligned} \textit{all}_{\text{and}}^{(l),m} &= \sum\nolimits_{S\in\Omega_{\text{and}}^{(l),m}} |I_{\text{and}}(S|\boldsymbol{x}, v^{(l)})|, \\ \textit{all}_{\text{and}}^{(L),m} &= \sum\nolimits_{S\in\Omega_{\text{and}}^{(L),m}} |I_{\text{and}}(S|\boldsymbol{x}, v^{(L)})|. \end{aligned} \tag{6}$$

As Fig. 3 shows, we designed the following three metrics to further disentangle the overall strength *all*$_{\text{and}}^{(l),m}$ and *all*$_{\text{and}}^{(L),m}$ into three terms: (1) the overall strength of interactions shared by both the l-th layer and the final layer, *overlap*$_{\text{and}}^{(l),m}$, (2) the overall strength of interactions encoded by the l-th layer but later forgotten in the final layer, *forget*$_{\text{and}}^{(l),m}$, (3) the overall strength of interactions that were encoded in the final layer, but were not encoded by the l-th layer, *new*$_{\text{and}}^{(l),m}$.

$$\begin{aligned} \textit{overlap}_{\text{and}}^{(l),m} &= \sum\nolimits_{S\in\Omega_{\text{and}}^{(l),m}\cap\Omega_{\text{and}}^{(L),m}} \left|I_{\text{and, shared}}^{(l,L)}(S|\boldsymbol{x})\right|, \\ \textit{forget}_{\text{and}}^{(l),m} &= \sum\nolimits_{S\in\Omega_{\text{and}}^{(l),m}} \left|I_{\text{and}}(S|\boldsymbol{x}, v^{(l)}) - I_{\text{and, shared}}^{(l,L)}(S|\boldsymbol{x})\right|, \\ \textit{new}_{\text{and}}^{(l),m} &= \sum\nolimits_{S\in\Omega_{\text{and}}^{(L),m}} \left|I_{\text{and}}(S|\boldsymbol{x}, v^{(L)}) - I_{\text{and, shared}}^{(l,L)}(S|\boldsymbol{x})\right|, \end{aligned} \tag{7}$$

where $I_{\text{and, shared}}^{(l,L)}(S|\boldsymbol{x})$ measured the shared AND interactions between $I_{\text{and}}(S|\boldsymbol{x}, v^{(l)})$ extracted from the l-th layer and $I_{\text{and}}(S|\boldsymbol{x}, v^{(L)})$ encoded by the final L-th layer. If $I_{\text{and}}(S|\boldsymbol{x}, v^{(l)})$ and $I_{\text{and}}(S|\boldsymbol{x}, v^{(L)})$ had opposite interaction effects, then $I_{\text{and, shared}}^{(l,L)}(S|\boldsymbol{x}) = 0$; Otherwise, the shared AND interaction was defined as $I_{\text{and, shared}}^{(l,L)}(S|\boldsymbol{x}) = \text{sign}(I_{\text{and}}(S|\boldsymbol{x}, v^{(l)})) \cdot \min(|I_{\text{and}}(S|\boldsymbol{x}, v^{(l)})|, |I_{\text{and}}(S|\boldsymbol{x}, v^{(L)})|)$.

Thus, *overlap*$_{\text{and}}^{(l),m}$, *forget*$_{\text{and}}^{(l),m}$, and *new*$_{\text{and}}^{(l),m}$ formed a decomposition of overall interaction strength, as follows.



(a) DNNs trained on CIFAR10

(b) DNNs trained on SST2

Figure 4. Average IoU values of AND interactions extracted from two DNNs trained for the same task over different input samples. Low-order interactions usually exhibited higher IoU values, thereby being better generalized across DNNs. Please see Appendix I for results of OR interactions and Appendix M.3 for the selected intermediate layer.

$$\begin{aligned} all_{\text{and}}^{(l),m} &= overlap_{\text{and}}^{(l),m} + forget_{\text{and}}^{(l),m}, \\ all_{\text{and}}^{(L),m} &= overlap_{\text{and}}^{(l),m} + new_{\text{and}}^{(l),m}. \end{aligned} \tag{8}$$

Metrics for OR interactions $overlap_{\text{or}}^{(l),m}$, $forget_{\text{or}}^{(l),m}$, and $new_{\text{or}}^{(l),m}$ were defined in the similar way.

To evaluate the progress of learning target interactions and redundant interactions, we also define the completeness of interactions encoded by each l-th layer *w.r.t.* all interactions $all_{\text{and}}^{(L),m}$ encoded by the final layer, as $completeness_{\text{and}}^{(l),m} = overlap_{\text{and}}^{(l),m} / \, all_{\text{and}}^{(L),m}$. We define the redundancy of interactions in each l-th layer as the ratio of the interactions $forget_{\text{and}}^{(l),m}$ that are finally forgotten, as $redundancy_{\text{and}}^{(l),m} = forget_{\text{and}}^{(l),m} / \, all_{\text{and}}^{(l),m}$. What's more, we can define the metrics for OR interaction $completeness_{\text{or}}^{(l),m}$ and $redundancy_{\text{or}}^{(l),m}$ in the similar way.

Results & Analysis. Fig. 3 (a) reports the average strength[8] of the overlapped AND interactions $overlap_{\text{and}}^{(l),m}$, the forgotten AND interactions $forget_{\text{and}}^{(l),m}$, and newly emerged AND interactions $new_{\text{and}}^{(l),m}$. Fig. 3 (b) tracks the completeness and redundancy of the learned interactions through layers. We discovered that even among different models on different tasks, most DNNs still tended to follow similar information-processing behaviors, as follows.

- Fig. 3 (a) shows that $all_{\text{and}}^{(l),m}$ and $all_{\text{and}}^{(L),m}$ of low order have higher strength than those of high order, which indicates that DNNs usually encode stronger low-order (simple) interactions than high-order (complex) interactions.

- Fig. 3 (b) shows that $completeness_{\text{and}}^{(l),m}$ values of most interactions start increasing at early layers. Unlike low-order interactions, high-order interactions do not reach high completeness even in later layers. These indicate that the early and middle layers usually had already learned most target interactions that were finally used by DNNs. Morever, extremely high-order interactions are learned in later layers, but the learning is unstable.

- Fig. 3 (b) shows that $redundancy_{\text{and}}^{(l),m}$ values of most interactions first rise and then fall through layers, which indicates that the utility of later layers of DNNs was mainly to remove redundant interactions encoded by earlier layers.

- **Using our results to analyze the generalization power.** We further use the layerwise change of interactions on each specific DNN to analyze its generalization power. To this end, Section 3.3 will show a clear relationship between the order of interactions and the generalization power of interactions. In this way, Appendix K introduces how to use the distribution of interactions to analyze the generalization power of features of different layers.

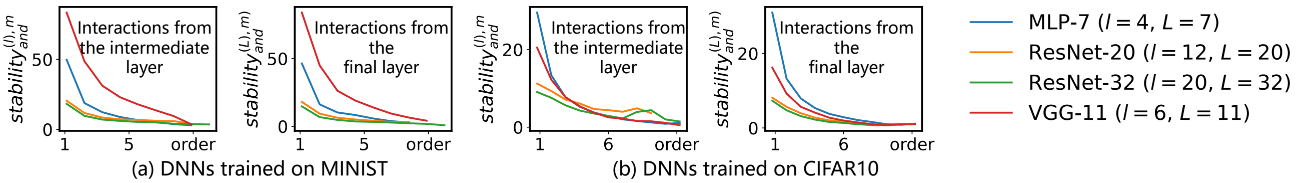
3.3. Analyzing the representation capacity of a DNN

Tracking salient interactions through layers also provides us a new perspective to understand how the representation capacity gradually changes during the forward propagation. It is because we find that the order (complexity) of interactions can well explain the generalization capacity and the instability of feature representations of a DNN.

- **Low-order interactions are more generalizable across models.** According to Lemma 3.4, we can disentangle the overall inference score based on the feature $f^{(l)}(\boldsymbol{x})$ into the sum of effects of a few salient interactions, $v^{(l)}(\boldsymbol{x}_T) \approx v(\boldsymbol{x}_\emptyset) + \sum_{S\in\Omega_{\text{and}}^{(l)}:\emptyset\neq S\subseteq T} I_{\text{and}}(S|\boldsymbol{x_T}, v^{(l)}) + \sum_{S\in\Omega_{\text{or}}^{(l)}:S\cap T\neq\emptyset} I_{\text{or}}(S|\boldsymbol{x_T}, v^{(l)})$. Thus, the generalization capacity of the feature $f^{(l)}(\boldsymbol{x})$ can be explained by the generalization capacity of salient interactions.

To this end, we consider that if multiple DNNs trained for the same task encode the same interaction, then this interaction is regarded as well-generalized. Specifically, given two DNNs, v_A and v_B, trained for the same classification task and an input sample $\boldsymbol{x}$, we follow the settings in Section 3.2.2 to extract two sets of m-order salient AND interactions from the l_a-th layer of the DNN v_A and the l_b-th

[8] We normalized each AND interaction $I_{\text{and}}(S|\boldsymbol{x}, v^{(l)})$ extracted from the l-th layer of the target DNN as $I_{\text{and}}(S|\boldsymbol{x}, v^{(l)}) \leftarrow I_{\text{and}}(S|\boldsymbol{x}, v^{(l)}) / \mathbb{E}_{\boldsymbol{x}}[|v^{(l)}(\boldsymbol{x}_N) - v^{(l)}(\boldsymbol{x}_\emptyset)|]$ for fair comparison. Each OR interaction was normalized in the similar way.



(a) DNNs trained on MINIST

(b) DNNs trained on CIFAR10

Figure 5. The relative stability (*stability*$_{\text{and}}^{(l),m}$) of AND interactions decreased along with the order m. Low-order interactions were more stable to inevitable noises in data. See Appendix I for results of OR interactions and Appendix M.3 for the selected intermediate layer.

layer of the DNN v_B, respectively, which are denoted by $A_{\text{and}}^{(l_a),m} = \{S \subseteq N : |S| = m, |I_{\text{and}}(S|\boldsymbol{x}, v_A^{(l_a)})| > \tau^7\}$ and $B_{\text{and}}^{(l_b),m}$. Accordingly, let $A_{\text{or}}^{(l_a),m}$ and $B_{\text{or}}^{(l_b),m}$ represent sets of salient OR interactions of m-th order, respectively. Then, we use the IoU metric to measure the generalization capacity of m-order interactions across different models.

$$\begin{aligned} IoU(A_{\text{and}}^{(l_a),m}, B_{\text{and}}^{(l_b),m}) &= \frac{|A_{\text{and}}^{(l_a),m} \cap B_{\text{and}}^{(l_b),m}|}{|A_{\text{and}}^{(l_a),m} \cup B_{\text{and}}^{(l_b),m}|}, \\ IoU(A_{\text{or}}^{(l_a),m}, B_{\text{or}}^{(l_b),m}) &= \frac{|A_{\text{or}}^{(l_a),m} \cap B_{\text{or}}^{(l_b),m}|}{|A_{\text{or}}^{(l_a),m} \cup B_{\text{or}}^{(l_b),m}|}. \end{aligned} \tag{9}$$

Large values of $IoU(A_{\text{and}}^{(l_a),m}, B_{\text{and}}^{(l_b),m})$ and $IoU(A_{\text{or}}^{(l_a),m}, B_{\text{or}}^{(l_b),m})$ mean that most m-order interactions encoded by a DNN can be well generalized to another DNN.

Experiments. Here, we examined the generalization capacity of interactions of different orders. We used DNNs introduced in Section 3.2.1, *i.e.,* MLP-7, VGG-11, ResNet-20, and ResNet-32 (He et al., 2016) trained on the CIFAR-10 dataset for image classification, and DistilBERT, BERT$_{\text{BASE}}$, and XLNet (Yang et al., 2019) fine-tuned on the SST-2 dataset for binary sentiment classification.

Fig. 4 reports the average IoU value of AND interactions extracted from two DNNs over different input samples, $\mathbb{E}_{\boldsymbol{x}}[IoU(A_{\text{and}}^{(l_a),m}, B_{\text{and}}^{(l_b),m})]$, given each pair of DNNs trained for the same task. We discovered low-order interactions extracted from different DNNs usually exhibited higher IoU values, *i.e.,* different DNNs trained for the same task usually encoded similar sets of salient low-order interactions. This demonstrated low-order interactions could be better generalized across DNNs. Notably, for each $m < n/2$, there are the same number $\binom{n}{m}$ of potential combinations for both m-order interactions and $(n-m)$-order interactions. Low-order (m-order) interactions are more generalizable than high-order interactions.

- **Low-order interactions are more stable to small noises.** We discover that the order of interactions can also be used to explain the instability of feature representations of a DNN. According to Lemma 3.4, the overall inference score based on the feature $f^{(l)}(\boldsymbol{x})$ can be disentangled into the sum of the effects of a few salient AND-OR interactions. Thus, the instability of the feature $f^{(l)}(\boldsymbol{x})$ can be explained by the instability of salient interactions.

To this end, let us add a small Gaussian perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$ to the input sample $\boldsymbol{x}$, in order to mimic inevitable noises/variations in data. Although there may exist other noises in data, we just use Gaussian perturbation to represent noises/variations in data, which may still provide insights into real-world applications. Thus, we use the following metrics to measure the relative stability of AND-OR interactions of each order m.

$$\begin{aligned} \textit{stability}_{\text{and}}^{(l),m} &= \mathbb{E}_{\boldsymbol{x}} \mathop{\mathbb{E}}_{S \in \Omega_{\text{and}}^{(l),m}} \left[|E_{\text{and}}^{(l)}(S, \boldsymbol{x})| \Big/ \sqrt{\textit{Var}_{\text{and}}^{(l)}(S, \boldsymbol{x})} \right], \\ \textit{stability}_{\text{or}}^{(l),m} &= \mathbb{E}_{\boldsymbol{x}} \mathop{\mathbb{E}}_{S \in \Omega_{\text{or}}^{(l),m}} \left[|E_{\text{or}}^{(l)}(S, \boldsymbol{x})| \Big/ \sqrt{\textit{Var}_{\text{or}}^{(l)}(S, \boldsymbol{x})} \right]. \end{aligned} \tag{10}$$

where $E_{\text{and}}^{(l)}(S, \boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\epsilon}}[I_{\text{and}}(S|\boldsymbol{x} + \boldsymbol{\epsilon}, v^{(l)})]$ and $\textit{Var}_{\text{and}}^{(l)}(S, \boldsymbol{x}) = \textit{Var}_{\boldsymbol{\epsilon}}[I_{\text{and}}(S|\boldsymbol{x} + \boldsymbol{\epsilon}, v^{(l)})]$ denote the mean and variance of the AND interaction $I_{\text{and}}(S|\boldsymbol{x} + \boldsymbol{\epsilon}, v^{(l)})$ *w.r.t.* Gaussian perturbations $\boldsymbol{\epsilon}$, which are encoded by the l-th layer of the DNN. Similarly, $E_{\text{or}}^{(l)}(S, \boldsymbol{x})$ and $Var_{\text{or}}^{(l)}(S, \boldsymbol{x})$ represent the mean and variance of the OR interaction $I_{\text{or}}(S|\boldsymbol{x} + \boldsymbol{\epsilon}, v^{(l)})$ *w.r.t.* noises $\boldsymbol{\epsilon}$. Large values of *stability*$_{\text{and}}^{(l),m}$ and *stability*$_{\text{or}}^{(l),m}$ indicates that m-order interactions are stable to inevitable noises.

Towards normalization of the noise. As a common understanding, people usually think that higher-order interactions contain more noise than low-order interactions, because they involve more input variables. However, it is noteworthy that high-order interactions also obtain more input signals, and the signal-to-noise ratio of each interaction is relatively consistent over interactions of different orders. Thus *stability*$_{\text{and}}^{(l),m}$ and *stability*$_{\text{or}}^{(l),m}$ ensures a fair comparison of interactions of different orders m. Please see Appendix L for details.

Experiments. We conducted experiments to check the instability of AND-OR interactions of each order. To this end, we added Gaussian perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.02^2\boldsymbol{I})$ to each training sample. Then, for each order m, we computed metrics *stability*$_{\text{and}}^{(l),m}$ based on DNNs, and the DNNs for testing have been introduced in Section 3.2.1. Fig. 5 shows that the relative stability $\textit{stability}_{\text{and}}^{(l),m}$ decreased along with the order m, which indicated that low-order interactions were more stable to inevitable noises in data than high-order interactions. In other words, low-order interactions usually exhibited consistent effects $I_{\text{and}}(S|\boldsymbol{x}+\boldsymbol{\epsilon}, v^{(l)})$ on the network output/intermediate-layer feature *w.r.t.* different noises $\boldsymbol{\epsilon}$ than high-order interactions. This indicated that low-order interactions were more likely to be generalized to similar samples (*e.g.,* samples with small intra-class variations).

Thus, according to Figs. 3, 4, 5, we discovered that for ResNet-20 trained on both the MNIST dataset and the CIFAR-10 dataset, their later layers usually exclusively forgot redundant high-order interactions without encoding new interactions, which were non-generalizable and unstable. Besides, later layers of DistilBERT and $\text{BERT}_{\text{BASE}}$ trained on the SST-2 dataset usually forgot redundant and non-generalizable high-order interactions.

4. Conclusion and Discussion

In this paper, we use interaction primitives to represent knowledge encoded by the DNN. The sparsity and the universal-matching property of interactions ensure the trustworthiness of taking interactions as symbolic primitive inference patterns encoded by a DNN. Thus, we further quantify and track the newly emerged interaction primitives and the forgotten interaction primitives in each layer during the forward propagation, which provides new insights into the learning behavior of DNNs. The layer-wise change of interactions potentially reveals the change of the generalization capacity and instability of feature representations of a DNN.

Acknowledgements

This work is partially supported by the National Science and Technology Major Project (2021ZD0111602), the National Nature Science Foundation of China (92370115, 62276165). This work is also partially supported by Huawei Technologies Inc.

Impact Statement

This paper uses interaction primitives to represent knowledge encoded by the DNN, and further quantify and track the layer-wise change of interaction primitives in the DNN during the forward propagation. The layer-wise change of interactions also reveals the change of the generalization capacity and instability of feature representations of the DNN. There are no ethical issues with this paper, and there are no potential societal consequences of this paper that need to be specifically highlighted.

References

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

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Belinkov, Y. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

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

Dabkowski, P. and Gal, Y. Real time image saliency for black box classifiers. *Advances in neural information processing systems*, 30, 2017.

Deng, H., Ren, Q., Zhang, H., and Zhang, Q. DISCOVERING AND EXPLAINING THE REPRESENTATION BOTTLENECK OF DNNS. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=iRCUlgmdfHJ.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Fong, R., Patrick, M., and Vedaldi, A. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2950–2958, 2019.

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

Jain, S., Salman, H., Wong, E., Zhang, P., Vineet, V., Vemprala, S., and Madry, A. Missingness bias in model debugging. *arXiv preprint arXiv:2204.08945*, 2022.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Kolchinsky, A., Tracey, B. D., and Kuyk, S. V. Caveats for information bottleneck in deterministic scenarios. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rke4HiAcY7.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, M. and Zhang, Q. Does a neural network really encode symbolic concepts? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 20452–20469. PMLR, 23–29 Jul 2023.

Liang, R., Li, T., Li, L., Wang, J., and Zhang, Q. Knowledge consistency between neural networks and beyond. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJeS62EtwH.

Liu, D., Deng, H., Cheng, X., Ren, Q., Wang, K., and Zhang, Q. Towards the difficulty for a deep neural network to learn concepts of different complexities. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Pimentel, T., Valvoda, J., Maudslay, R. H., Zmigrod, R., Williams, A., and Cotterell, R. Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*, 2020.

Ren, J., Li, M., Chen, Q., Deng, H., and Zhang, Q. Defining and quantifying the emergence of sparse concepts in dnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023a. URL https://arxiv.org/pdf/2111.06206v5.pdf.

Ren, J., Zhou, Z., Chen, Q., and Zhang, Q. Can we faithfully represent absence states to compute shapley values on a DNN? In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=YV8tP7bW6Kt.

Ren, Q., Gao, J., Shen, W., and Zhang, Q. Where we have arrived in proving the emergence of sparse interaction primitives in AI models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3pWSL8My6B.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ry_WPG-A-.

Shen, W., Wei, Z., Huang, S., Zhang, B., Fan, J., Zhao, P., and Zhang, Q. Interpretable compositional convolutional neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 2971–2978, 2021.

Shen, W., Cheng, L., Yang, Y., Li, M., and Zhang, Q. Can the inference logic of large language models be disentangled into symbolic concepts?, 2023.

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

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

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.

Sundararajan, M., Dhamdhere, K., and Agarwal, A. The shapley taylor interaction index. In *International conference on machine learning*, pp. 9259–9268. PMLR, 2020.

Wang, Z., Huang, S.-L., Kuruoglu, E. E., Sun, J., Chen, X., and Zheng, Y. PAC-bayes information bottleneck. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=iLHOIDsPv1P.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Zhang, Q., Wang, X., Wu, Y. N., Zhou, H., and Zhu, S.-C. Interpretable cnns for object classification. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3416–3431, 2020.

Zhou, H., Tang, H., Li, M., Zhang, H., Liu, Z., and Zhang, Q. Explaining how a neural network play the go game and let people learn. *arXiv preprint arXiv:2310.09838*, 2023.

Zhou, H., Zhang, H., Deng, H., Liu, D., Shen, W., Chan, S.-H., and Zhang, Q. Explaining generalization power of a dnn using interactive concepts. In *Thirty-Eight AAAI Conference on Artificial Intelligence*, 2024. URL arXivpreprintarXiv:2302.13091.

A. Detailed Analysis for Previous Studies Using Knowledge to Explain DNNs

Explaining and quantifying the exact knowledge encoded by a DNN presents a significant challenge to explainable AI. So far, there has not existed a widely accepted definition of knowledge that enables us to accurately disentangle and quantify knowledge encoded by intermediate layers of a DNN, because it covers various aspects of cognitive science, neuroscience, and mathematics. To this end, previous works have employed different methods to quantify knowledge encoded by a DNN. Then, let us revisit previous studies from the perspective of three challenges mentioned in Section 1.

First, Bau et al. (2017); Kim et al. (2018) associated neurons with manually annotated semantics/concepts (knowledge). However, these works could not quantify the exact amount of knowledge in the DNN, or discover new concepts emerged in intermediate layers. Second, learning interpretable neural networks with meaningful features in intermediate layers was another classic direction in explainable AI (Zhang et al., 2020; Shen et al., 2021; Chen et al., 2019). Although these studies automatically learned meaningful concepts without human annotations, they did not provide a mathematically guaranteed boundary for each concept/knowledge. Thus, these works could not quantify the exact amount of newly emerged/forgotten/unexplainable knowledge in each layer.

Third, the information-bottleneck theory (Shwartz-Ziv & Tishby, 2017; Saxe et al., 2018) used the mutual information between inputs and intermediate-layer features to quantify knowledge encoded by the DNN. However, the mutual information could only measure the overall information contained in each feature, but could not accurately quantify exact knowledge represented by the newly emerged information and the forgotten information. Besides, Kolchinsky et al. (2019) showed the mutual information was difficult to measure accurately, and Wang et al. (2022); Saxe et al. (2018) discovered the mutual information had mathematical flaws in explaining the generalization power of a DNN.

Fourth, Liang et al. (2020) disentangled feature components from each layer, which could be reconstructed by features in other layers, so as to evaluate the changes of features in different layers. However, the changes of features in different layers could not be aligned to the same feature space for fair comparison, and could not be employed to explain the generalization capacity of the DNN.

B. Comparison between Interaction-based Explanation and Attribution Interpretability Methods

We compare our interaction-based explanation with attribution interpretability methods from the following three perspectives.

- **Perspective 1: Whether the method can explain the detailed inference logic of a DNN.** We have conducted a new experiment to show explanation results of other interpretability methods for comparison. Figure 6 shows that the main difference between our interaction-based explanation and traditional attribution methods (such as Integrated Gradient (**?**), Shapley value (**?**), and *etc*.) is that the interaction-based explanation precisely shows the detailed inference logic of the DNN, while attribution methods can only provide the importance score of each input variable to the network output. Thus, our method can provide a more precise explanation than attribution methods, and the faithfulness of our method is theoretically ensured by the universal matching property in Theorem 3.3.

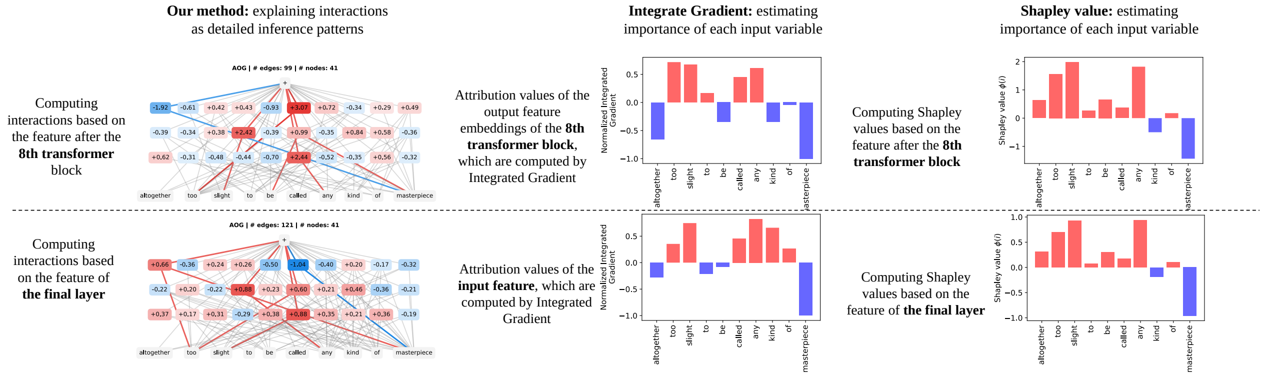


Figure 6. The comparison of explanation results between our interaction-based method, Integrated Gradient, and Shapley value.

- **Perspective 2: Theoretical connections between AND-OR interactions and attribution explanation methods.**

Besides, our AND-OR interactions can be considered as the elementary factors that determine the Shapley value (**?**). The Shapley value is a widely accepted standard attribution method, which assigns the numerical attribution of each input variable to the network output score. The Shapley value satisfies *linearity, nullity, symmetry*, and *efficiency* axioms. The Shapley value $\phi(i)$ of each input variable $i \in N$ can be rewritten as a re-allocation of AND-OR interactions, *i.e.,*, $\phi(i) = \sum_{S \subseteq N, S \ni i} \frac{1}{|S|} I_{\text{and}}(S) + \sum_{S \subseteq N, S \ni i} \frac{1}{|S|} I_{\text{or}}(S)$. It means that the computation of the Shapley value $\phi(i)$ can be explained as uniformly allocating the effect of each interaction $I(S)$ to its compositional input variables $i \in S$.

- **Perspective 3: If each feature dimension in an intermediate layer does not have a clear receptive field, the explanation based on this feature dimension's attribution will not have clear semantic meaning. In this case, we can only use interactions to explain the DNN.** For example, in an MLP, each feature dimension in an intermediate layer does not have a clear receptive field on input variables, *i.e.,*, this feature dimension does not have a clear physical meaning. In this case, visualizing the attribution of the feature dimension does not provide an intuitive explanation. so that it is meaningless to compute the attribution/importance of these features. In comparison, Figure. 6 shows that interactions extracted from a high layer still have clear physical meaning, and interactions used for computing a high-layer feature can also be aligned with interactions for computing a low-layer feature.

C. Proving the OR Interaction Can Be Considered A Specific AND Interaction

The OR interaction $I_{\text{or}}(S|\boldsymbol{x})$ can be considered as a specific AND interaction interaction $I_{\text{and}}(S|\boldsymbol{x})$, when we we inverse the definition of masked states and unmasked states of the input variable.

Specifically, given an input sample $\boldsymbol{x} \in \mathbb{R}^n$, let $\boldsymbol{x}_{N\setminus T}$ denote the masked sample obtained by masking input variables in T, while leaving variables in $N \setminus T$ unaltered. Here, we mask the input variable $i \in T$ to the baseline value b_i to represent its masked state, as follows.

$$(\boldsymbol{x}_{N\setminus T})_i = \begin{cases} x_i, & i \in N \setminus T \\ b_i, & i \in T \end{cases} \tag{11}$$

Then, let us consider the masked sample $\boldsymbol{x}'_T$, where we inverse the definition of the masked state and the unmasked state of each input variable to obtain this masked sample. That is, we mask input variables in the set $N \setminus T$ to baseline values, and keep variables in T unchanged, as follows.

$$(\boldsymbol{x}'_T)_i = \begin{cases} x_i, & i \in T \\ b_i, & i \in N \setminus T \end{cases} \tag{12}$$

Thus, the OR interaction $I_{\text{and}}(S|\boldsymbol{x})$ in Eq. 2 in main paper can be represented by the specific AND interaction $I_{\text{and}}(S|\boldsymbol{x}')$, as follows.

$$\begin{aligned} I_{\text{or}}(S|\boldsymbol{x}) &= -\sum\nolimits_{T \subseteq S} (-1)^{|S|-|T|} v(\boldsymbol{x}_{N\setminus T}), \\ &= -\sum\nolimits_{T \subseteq S} (-1)^{|S|-|T|} v(\boldsymbol{x}'_T), \\ &= -I_{\text{and}}(S|\boldsymbol{x}'). \end{aligned} \tag{13}$$

In this way, based on Eq. (13), the proven sparsity of AND interactions in (Ren et al., 2024) also proves the sparsity of OR interactions, *i.e.,* most well-trained DNNs usually encode a small number of OR interactions.

D. Discussion on Techniques and Limitations of Classifier Probe

Probing classifiers have become one of the prominent methodologies for interpreting and analyzing deep neural network models. This approach involves training a classifier to predict a specific linguistic property based on the representations generated by a model. In our study, we utilize this technique by selecting features from the mid-layers of a neural network. These selected features are then used to train a linear classifier, enabling us to assess and understand the knowledge contained within the mid-layer features of the neural network. However, it's important to acknowledge that the probing classifiers approach is not without its drawbacks. Belinkov (2022) have outlined the limitations of this framework, and we will explore these challenges in the following discussion.

- A primary challenge arises in how we interpret the performance from the probing classifier, particularly in selecting an appropriate baseline for comparison. Our study diverges from the traditional method of directly comparing overall model performance. Instead, we concentrate on the knowledge contained within the middle layer features, specifically their direct applicability to classification tasks. In fact, our findings reveal a high degree of similarity in the knowledge interpreted across adjacent layers, which partly reflects the faithfulness of our work.

- The second challenge concerns the selection of the classifier's structure. Pimentel et al. (2020) contend that to obtain the most accurate estimate of the information a model possesses about a given property, it is advisable to use the most complex probe available. However, our research is not focused on the ultimate classifiability of the intermediate layer. Instead, we are interested in determining the extent to which the features of this layer can be directly applied to the classification task. In light of this, we opt for one of the simplest classifier structures available: the linear classifier. This choice is driven by our specific objective of evaluating the direct applicability of middle layer features, rather than maximizing the classification potential of the probe.

- The third challenge addresses the disconnect between the probing classifier g and the original model f. This implies that the knowledge inferred from the classifier may not always align with what is actually utilized by the original model. This is a good question. In fact, this perspective is central to our research. Our findings indicate that neural networks tend to learn numerous redundant features in the middle layers, which are subsequently forgotten in the later layers. Our methodology offers a quantifiable analysis of the variation in knowledge across different layers, shedding light on how information is processed and transformed within the network.

- The fourth challenge concerns the imperfection of the dataset used for training. Specifically, the classifier is unable to exhaustively uncover all the knowledge present due to the limitations inherent in the dataset. This is a real drawback, as it is not feasible to use an all-encompassing dataset for perfect training. Our approach mitigates this issue by training the classifier on the same dataset as the original model. This strategy aims to ensure as fair and balanced a training process as possible, while acknowledging the constraints of the dataset while striving.

In summary, while employing probing classifiers in the interpretation of neural network models does introduce specific challenges, our approach aims to maximize the potential and ensure a thorough and insightful analysis of neural network models.

E. Discussion on the Bias Introduced by Masking Input Variables

In attribution method research, a prevalent approach involves utilizing a designated baseline value to obscure input variables in a DNN(Lundberg & Lee, 2017; Ancona et al., 2019; Fong et al., 2019). This technique measures the impact of these masked inputs on the network's output, thereby estimating the significance of each input variable. Nevertheless, research by Jain et al. (2022) indicates that this current method of masking may introduce substantial bias in the model's predictions. Specifically, it has been observed that the DNN tends to make errors influenced more by the areas subjected to masking than by the unmasked features.

In efforts to mitigate this bias, Ren et al. (2023b) have proposed utilizing causal patterns to scrutinize the reliability of baseline values. More importantly, they have established that causal patterns can be interpreted as the fundamental logic behind the concept of the Shapley value. Building upon this, they have proposed a novel methodology for determining optimal baseline values. The efficacy of this approach is underscored by the positive outcomes observed in various experimental settings.

F. Proof of Theorem 3.3

Theorem 3.3 *Given an input sample $\boldsymbol{x} \in \mathbb{R}^n$, the network output score $v(\boldsymbol{x}_T)$ on each masked input samples $\{\boldsymbol{x}_T | T \subseteq N\}$ can be decomposed into effects of AND interactions and OR interactions, subject to $I_{and}(\emptyset|\boldsymbol{x}) = v_{and}(\boldsymbol{x}_\emptyset) = v(\boldsymbol{x}_\emptyset)$ and $I_{or}(\emptyset|\boldsymbol{x}) = v_{or}(\boldsymbol{x}_\emptyset) = 0$.*

$$\begin{aligned} v(\boldsymbol{x}_T) &= v_{and}(\boldsymbol{x}_T) + v_{or}(\boldsymbol{x}_T) \\ &= \sum\nolimits_{S \subseteq T} I_{and}(S|\boldsymbol{x}_T) + \sum\nolimits_{S \cap T \neq \emptyset} I_{or}(S|\boldsymbol{x}_T). \end{aligned} \tag{14}$$

Proof. Let us first focus on the sum of AND interactions, as follows.

$$\begin{aligned} \sum\nolimits_{S \subseteq T} I_{\text{and}}(S|\boldsymbol{x}_{\boldsymbol{T}}) &= \sum\nolimits_{S \subseteq T} \sum\nolimits_{L \subseteq S} (-1)^{|S|-|L|} v_{\text{and}}(\boldsymbol{x}_L) \\ &= \sum\nolimits_{L \subseteq T} \sum\nolimits_{S: L \subseteq S \subseteq T} (-1)^{|S|-|L|} v_{\text{and}}(\boldsymbol{x}_L) \\ &= \underbrace{v_{\text{and}}(\boldsymbol{x}_T)}_{L=T} + \sum\nolimits_{L \subseteq T, L \neq T} v_{\text{and}}(\boldsymbol{x}_L) \cdot \underbrace{\sum\nolimits_{m=0}^{|T|-|L|} (-1)^m}_{=0} \\ &= v_{\text{and}}(\boldsymbol{x}_T). \end{aligned} \tag{15}$$

Then, let us concentrate on the the sum of OR interactions, as follows.

$$\begin{aligned} \sum\nolimits_{S \cap T \neq \emptyset} I_{\text{or}}(S|\boldsymbol{x}_{\boldsymbol{T}}) &= -\sum\nolimits_{S \cap T \neq \emptyset, S \neq \emptyset} \sum\nolimits_{L \subseteq S} (-1)^{|S|-|L|} v_{\text{or}}(\boldsymbol{x}_{N \setminus L}) \\ &= -\sum\nolimits_{L \subseteq N} \sum\nolimits_{S: S \cap T \neq \emptyset, S \supseteq L} (-1)^{|S|-|L|} v_{\text{or}}(\boldsymbol{x}_{N \setminus L}) \\ &= -\underbrace{v_{\text{or}}(\boldsymbol{x}_\emptyset)}_{L=N} - \underbrace{v_{\text{or}}(\boldsymbol{x}_T)}_{L=N \setminus T} \cdot \underbrace{\sum_{|S_2|=1}^{|T|} C_{|T|}^{|S_2|} (-1)^{|S_2|}}_{=-1} \\ &\quad - \sum_{L \cap T \neq \emptyset, L \neq N} v_{\text{or}}(\boldsymbol{x}_{N \setminus L}) \cdot \sum_{S_1 \subseteq N \setminus T \setminus L} \underbrace{\sum_{|S_2|=|T \cap L|}^{|T|} C_{|T|-|T \cap L|}^{|S_2|-|T \cap L|} (-1)^{|S_1|+|S_2|}}_{=0} \\ &\quad - \sum_{L \cap T = \emptyset, L \neq N \setminus T} v_{\text{or}}(\boldsymbol{x}_{N \setminus L}) \cdot \underbrace{\sum_{S_2 \subsetneqq T} \sum_{S_1 \subseteq N \setminus T \setminus L} (-1)^{|S_1|+|S_2|}}_{=0} \\ &= v_{\text{or}}(\boldsymbol{x}_T) - v_{\text{or}}(\boldsymbol{x}_\emptyset) \end{aligned} \tag{16}$$

.

Thus, we obtain $v_{\text{or}}(\boldsymbol{x}_T) = \sum_{S \cap T \neq \emptyset} I_{\text{or}}(S) + v_{\text{or}}(\boldsymbol{x}_\emptyset)$, according to Eq. (16). Thus, the output score $v(\boldsymbol{x}_T)$ of the DNN on the masked sample $\boldsymbol{x}_T$ can be represented as the sum of effects of AND-OR interactions.

$$\begin{aligned} v(\boldsymbol{x}_T) &= v_{\text{and}}(\boldsymbol{x}_T) + v_{\text{or}}(\boldsymbol{x}_T) \\ &= \sum\nolimits_{S \subseteq T} I_{\text{and}}(S|\boldsymbol{x}_{\boldsymbol{T}}) + \sum\nolimits_{S \cap T \neq \emptyset, S \neq \emptyset} I_{\text{or}}(S|\boldsymbol{x}_{\boldsymbol{T}}) + v_{\text{or}}(\boldsymbol{x}_\emptyset) \\ &= \sum\nolimits_{S \subseteq T, S \neq \emptyset} I_{\text{and}}(S|\boldsymbol{x}_{\boldsymbol{T}}) + v_{\text{and}}(\boldsymbol{x}_\emptyset) + \sum\nolimits_{S \cap T \neq \emptyset} I_{\text{or}}(S|\boldsymbol{x}_{\boldsymbol{T}}) + v_{\text{or}}(\boldsymbol{x}_\emptyset) \\ &= v(\boldsymbol{x}_\emptyset) + \sum\nolimits_{S \subseteq T, S \neq \emptyset} I_{\text{and}}(S|\boldsymbol{x}_{\boldsymbol{T}}) + \sum\nolimits_{S \cap T \neq \emptyset} I_{\text{or}}(S|\boldsymbol{x}_{\boldsymbol{T}}) \\ &= \sum\nolimits_{S \subseteq T} I_{\text{and}}(S|\boldsymbol{x}_T) + \sum\nolimits_{S \cap T \neq \emptyset} I_{\text{or}}(S|\boldsymbol{x}_T). \end{aligned} \tag{17}$$

Thus, Theorem 3.3 is proven.

□

G. Proof of Lemma 3.4

Lemma 3.4 (**Proving interactions as primitive inference patterns**) *Given an input sample $\boldsymbol{x} \in \mathbb{R}^n$, the network output on all 2^n masked input samples $\{\boldsymbol{x}_S | S \subseteq N\}$ can be universally matched by a small set of salient interactions.*

$$\begin{aligned} v(\boldsymbol{x}_T) &= v_{\textit{and}}(\boldsymbol{x}_T) + v_{\textit{or}}(\boldsymbol{x}_T) = \sum_{S \subseteq T} I_{\textit{and}}(S|\boldsymbol{x}_T) + \sum_{S \cap T \neq \emptyset} I_{\textit{or}}(S|\boldsymbol{x}_T) \\ &\approx v(\boldsymbol{x}_\emptyset) + \sum\nolimits_{S \in \Omega^{\textit{and}}_{\textit{salient}} : \emptyset \neq S \subseteq T} I_{\textit{and}}(S|\boldsymbol{x}_{\boldsymbol{T}}) + \sum\nolimits_{S \in \Omega^{\textit{or}}_{\textit{salient}} : S \cap T \neq \emptyset} I_{\textit{or}}(S|\boldsymbol{x}_{\boldsymbol{T}}). \end{aligned} \tag{18}$$

Proof. Ren et al. (2023a) have proven that under some common conditions[1], the output $v_{\text{and}}(\boldsymbol{x}_T)$ of a well-trained DNN on all 2^n masked samples $\{\boldsymbol{x}_T | T \subseteq N\}$ can be universally approximated by a small number of AND interactions $T \in \Omega^{\text{and}}_{\text{salient}}$ with salient effects $I_{\text{and}}(T|\boldsymbol{x})$ on the network output, subject to $|\Omega^{\text{and}}_{\text{salient}}| \ll 2^n$.

Besides, as proven in Appendix C, the OR interaction can be considered as a specific AND interaction. Thus, the output $v_{\text{or}}(\boldsymbol{x}_T)$ of a well-trained DNN on all 2^n masked samples $\{\boldsymbol{x}_T | T \subseteq N\}$ can be universally approximated by a small number of OR interactions $T \in \Omega^{\text{or}}_{\text{salient}}$ with salient effects $I_{\text{or}}(T|\boldsymbol{x})$ on the network output, subject to $|\Omega^{\text{or}}_{\text{salient}}| \ll 2^n$.

In this way, Eq. (17) can be further approximated as

$$\begin{aligned} v(\boldsymbol{x}_T) &= v_{\text{and}}(\boldsymbol{x}_T) + v_{\text{or}}(\boldsymbol{x}_T) \\ &= v(\boldsymbol{x}_\emptyset) + \sum\nolimits_{S \subseteq T} I_{\text{and}}(S|\boldsymbol{x}_{\boldsymbol{T}}) + \sum\nolimits_{S \cap T \neq \emptyset} I_{\text{or}}(S|\boldsymbol{x}_{\boldsymbol{T}}) \\ &\approx v(\boldsymbol{x}_\emptyset) + \sum\nolimits_{S \in \Omega^{\text{and}}_{\text{salient}} : \emptyset \neq S \subseteq T} I_{\text{and}}(S|\boldsymbol{x}_{\boldsymbol{T}}) + \sum\nolimits_{S \in \Omega^{\text{or}}_{\text{salient}} : S \cap T \neq \emptyset} I_{\text{or}}(S|\boldsymbol{x}_{\boldsymbol{T}}). \end{aligned} \tag{19}$$

Thus, Lemma 3.4 is proven.

□

H. Table of Metrics Used in the Paper

For a better understanding of our paper, we conclude all used metrics into the following table, where we clarify the formulation and the physical meaning of each metric.

Metric	Formulation	Physical Meaning
$I_{\text{and}}(S\mid\boldsymbol{x})$	$I_{\text{and}}(S\mid\boldsymbol{x}) \overset{\text{def}}{=} \sum_{T\subseteq S}(-1)^{\lvert S\rvert-\lvert T\rvert}\cdot v(\boldsymbol{x}_T)$	the AND relationship of input variables in S encoded by the DNN
$I_{\text{or}}(S\mid\boldsymbol{x})$	$I_{\text{or}}(S\mid\boldsymbol{x}) \overset{\text{def}}{=} -\sum_{T\subseteq S}(-1)^{\lvert S\rvert-\lvert T\rvert}\cdot v(\boldsymbol{x}_{N\setminus T})$	the OR relationship of input variables in S encoded by the DNN
$\mathit{all}_{\text{and}}^{(l),m}$	$\mathit{all}_{\text{and}}^{(l),m} \overset{\text{def}}{=} \sum_{S\in\Omega_{\text{and}}^{(l),m}} \lvert I_{\text{and}}(S\mid\boldsymbol{x},v^{(l)})\rvert$	the overall strength of all m-order salient AND interactions encoded by the l-th layer
$\mathit{all}_{\text{and}}^{(L),m}$	$\mathit{all}_{\text{and}}^{(L),m} \overset{\text{def}}{=} \sum_{S\in\Omega_{\text{and}}^{(L),m}} \lvert I_{\text{and}}(S\mid\boldsymbol{x},v^{(L)})\rvert$	the overall strength of all m-order salient AND interactions encoded by the final layer
$\mathit{overlap}_{\text{and}}^{(l),m}$	$\mathit{overlap}_{\text{and}}^{(l),m} \overset{\text{def}}{=} \sum_{S\in\Omega_{\text{and}}^{(l),m}\bigcap\Omega_{\text{and}}^{(L),m}} \left\lvert I_{\text{and, shared}}^{(l,L)}(S\mid\boldsymbol{x})\right\rvert$	the overall strength of AND interactions shared by both the l-th layer and the final layer
$\mathit{forget}_{\text{and}}^{(l),m}$	$\mathit{forget}_{\text{and}}^{(l),m} \overset{\text{def}}{=} \sum_{S\in\Omega_{\text{and}}^{(l),m}} \left\lvert I_{\text{and}}(S\mid\boldsymbol{x},v^{(l)}) - I_{\text{and, shared}}^{(l,L)}(S\mid\boldsymbol{x})\right\rvert$	the overall strength of interactions encoded by the l-th layer but later forgotten in the final layer
$\mathit{new}_{\text{and}}^{(l),m}$	$\mathit{new}_{\text{and}}^{(l),m} \overset{\text{def}}{=} \sum_{S\in\Omega_{\text{and}}^{(L),m}} \left\lvert I_{\text{and}}(S\mid\boldsymbol{x},v^{(L)}) - I_{\text{and, shared}}^{(l,L)}(S\mid\boldsymbol{x})\right\rvert$	the overall strength of interactions that are encoded in the final layer, but are not encoded by the l-th layer
$\mathit{IoU}(A_{\text{and}}^{(l_a),m}, B_{\text{and}}^{(l_b),m})$	$\mathit{IoU}(A_{\text{and}}^{(l_a),m}, B_{\text{and}}^{(l_b),m}) \overset{\text{def}}{=} \frac{\lvert A_{\text{and}}^{(l_a),m}\cap B_{\text{and}}^{(l_b),m}\rvert}{\lvert A_{\text{and}}^{(l_a),m}\cup B_{\text{and}}^{(l_b),m}\rvert}$	the IoU metric to measure the generalization capacity of m-order AND interactions across two different models A and B
$\mathit{stability}_{\text{and}}^{(l),m}$	$\mathit{stability}_{\text{and}}^{(l),m} \overset{\text{def}}{=} \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{S\in\Omega_{\text{and}}^{(l),m}}\left[\frac{\lvert E_{\text{and}}^{(l),m}(S,\boldsymbol{x})\rvert}{\sqrt{\mathit{Var}_{\text{and}}^{(l),m}(S,\boldsymbol{x})}}\right]$	the relative stability of m-order AND interactions to small noises

I. More Experimental Results

I.1. Experimental Results of OR Interactions

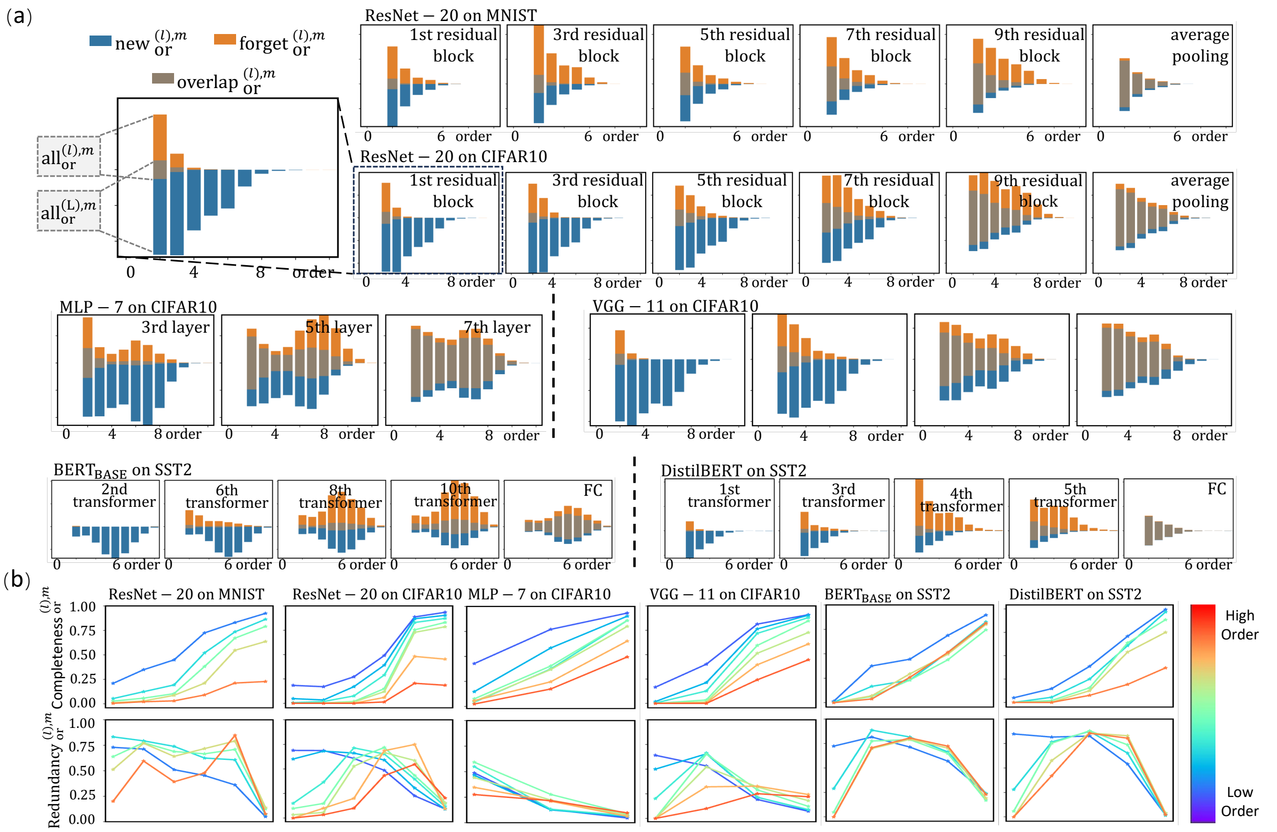
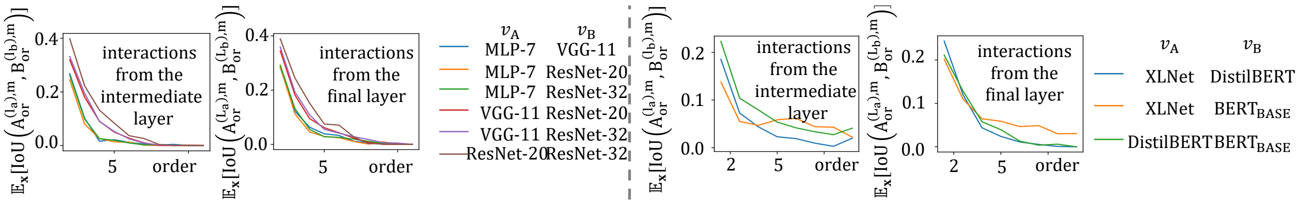


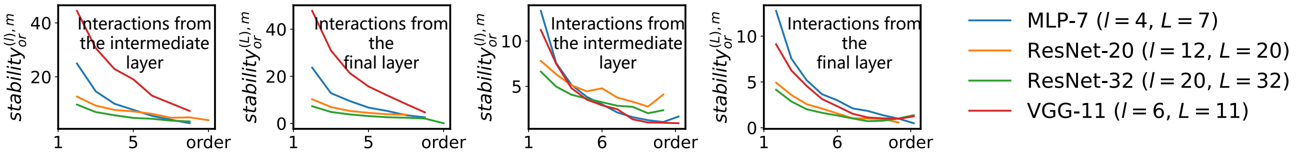
Figure 7. (a) Tracking the change of the average strength of the overlapped $\mathit{overlap}_{\mathrm{or}}^{(l),m}$, forgotten $\mathit{forget}_{\mathrm{or}}^{(l),m}$, and newly emerged interactions $\mathit{new}_{\mathrm{or}}^{(l),m}$ through different layers. For each subfigure, the total length of the orange bar and the grey bar equals to $\mathit{all}_{\mathrm{or}}^{(l),m}$, and the total length of the blue bar and the grey bar equals to $\mathit{all}_{\mathrm{or}}^{(L),m}$ (b) Tracking the change of $\mathit{completeness}_{\mathrm{or}}^{(l),m}$ and $\mathit{redundancy}_{\mathrm{or}}^{(l),m}$ through different layers. We do not show interactions of the highest four orders, because almost no interactions of extremely high orders were learned.



(a) DNNs trained on CIFAR10

(b) DNNs trained on SST2

Figure 8. Average IoU values of OR interactions extracted from two DNNs trained for the same task over different input samples. Low-order interactions usually exhibited higher IoU values, which indicated that low-order interactions could be better generalized across DNNs than high-order interactions. Appendix M.3 introduces the selected intermediate layer for each DNN.



(a) DNNs trained on MINIST

(b) DNNs trained on CIFAR10

Figure 9. The relative stability $stability_{\text{or}}^{(l),m}$ of OR interactions decreased along with the order m. It indicated that low-order interactions were more stable to inevitable noises in data. Appendix M.3 introduces the selected intermediate layer for each DNN.

I.2. Experimental Results on Tabular Datasets

We also trained the MLP-7 model on two tabular datasets (the UCI census dataset and the commercial dataset), and tracked the layer-wise change of interactions during the forward propagation in the MLP-7 model. Specifically, we calculated metrics $overlap_{\text{and}}^{(l),m}$, $forget_{\text{and}}^{(l),m}$, and $new_{\text{and}}^{(l),m}$ to quantify the overlapped AND interactions, forgotten AND interactions, and newly emerged AND interactions, respectively. We also calculated the completeness$_{\text{and}}^{(l),m}$ metric and the redundance$_{\text{and}}^{(l),m}$ metric to evaluate the progress of learning target AND interactions and removing redundant AND interactions.

Figure 10 (a) reports the average strength of the overlapped, forgotten, and newly emerged interactions through different layers, and Figure 10 (b) tracks the completeness and redundancy of the learned interactions through layers. It shows that (1) DNNs encode stronger low-order (simple) interactions than high-order (complex) interactions. (2) The early and middle layers usually had already learned most target interactions that were finally used by DNNs. Moreover, extremely high-order interactions are learned in later layers, but the learning is unstable. (3) DNNs quickly learn all target interactions without learning many redundant interactions. These conclusions are similar to the conclusions obtained on models trained on MNIST, CIFAR-10, SST-2 in the paper.

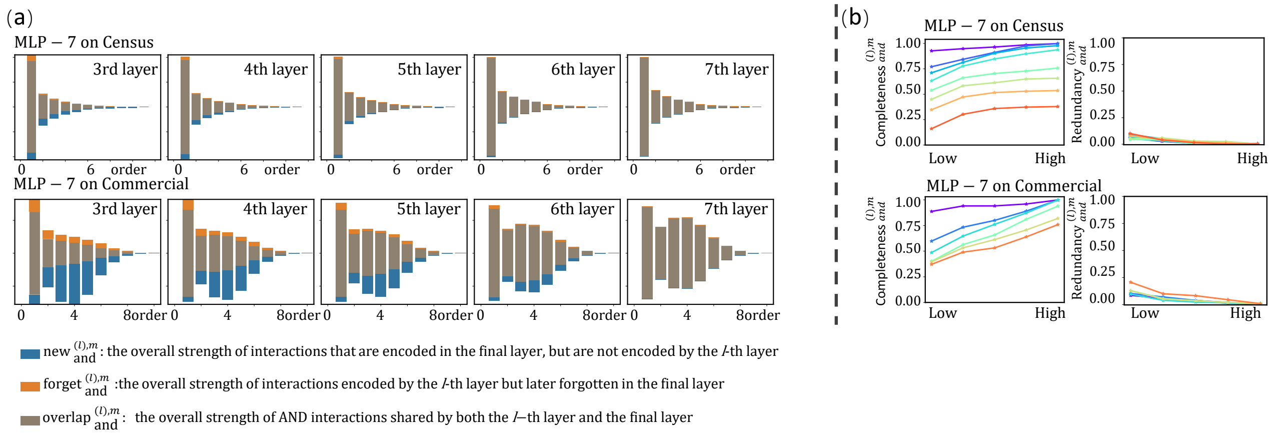


Figure 10. (a) Tracking the change of the average strength of the overlapped $overlap_{\text{or}}^{(l),m}$, forgotten $forget_{\text{or}}^{(l),m}$, and newly emerged interactions $new_{\text{or}}^{(l),m}$ through different layers. For each subfigure, the total length of the orange bar and the grey bar equals to $all_{\text{or}}^{(l),m}$, and the total length of the blue bar and the grey bar equals to $all_{\text{or}}^{(L),m}$ (b) Tracking the change of $completeness_{\text{or}}^{(l),m}$ and $redundancy_{\text{or}}^{(l),m}$ through different layers. We do not show interactions of the highest four orders, because almost no interactions of extremely high orders were learned.

J. Ablation Study of κ Value in Section 3.2.2

We find that the small noises in the output can significantly change the interaction effect. To remove the tiny noise in the model output and then extract relatively clean interactions, we define the new model output score $v(\boldsymbol{x})$.

$$\begin{aligned} v(\boldsymbol{x}) &= \log \frac{p(y = y^{\text{truth}}|\boldsymbol{x})}{1 - p(y = y^{\text{truth}}|\boldsymbol{x})} - \delta_N \\ v(\boldsymbol{x}_T) &= \log \frac{p(y = y^{\text{truth}}|\boldsymbol{x}_T)}{1 - p(y = y^{\text{truth}}|\boldsymbol{x}_T)} - \delta_T \end{aligned} \tag{20}$$

where $\delta_T, s.t.\ \forall T \subseteq N, |\delta_T| < \kappa$ is a learnable residual proposed to model and remove the tiny noise in the output $v^{(l)}(\boldsymbol{x}_T)$, so as to extract relatively clean interactions. δ_T is constrained to a small range $\kappa = 0.04 \cdot |v^{(l)}(\boldsymbol{x}_N) - v^{(l)}(\boldsymbol{x}_\emptyset)|$.

We conducted the ablation study to verify that the extraction of interactions is relatively robust to the κ value. Given a well-trained DNN (or linear classifier learned by intermediate layer's features) and an input sample $\boldsymbol{x} \in \mathbb{R}^n$, we verify that the interactions extracted are stable in different κ value settings. To this end, we used the ResNet-20 trained on the CIFAR-10 (introduced in Section 3.2.1) and extracted the all AND-OR interactions encoded in different setting, *e.g.,* $\kappa = 0.03 \cdot |v^{(l)}(\boldsymbol{x}_N) - v^{(l)}(\boldsymbol{x}_\emptyset)|$, $\kappa = 0.04 \cdot |v^{(l)}(\boldsymbol{x}_N) - v^{(l)}(\boldsymbol{x}_\emptyset)|$ and $\kappa = 0.05 \cdot |v^{(l)}(\boldsymbol{x}_N) - v^{(l)}(\boldsymbol{x}_\emptyset)|$.

Fig 11 shows that the all AND-OR interactions encoded in three different κ value settings are almost the same. This indicates that the extraction of interactions is relatively robust to the κ value.

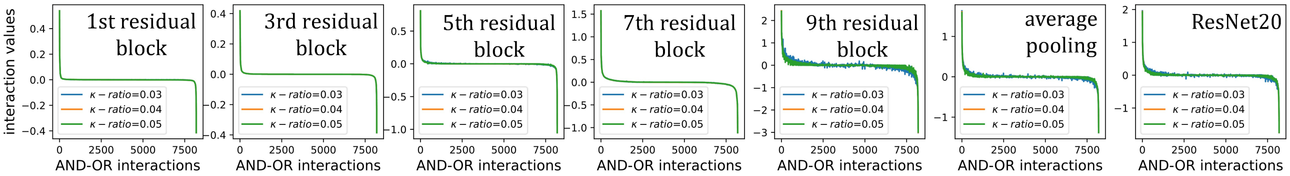


Figure 11. The extracted AND-OR interactions encoded in different setting, *e.g.,* $\kappa = 0.03 \cdot |v^{(l)}(\boldsymbol{x}_N) - v^{(l)}(\boldsymbol{x}_\emptyset)|$, $\kappa = 0.04 \cdot |v^{(l)}(\boldsymbol{x}_N) - v^{(l)}(\boldsymbol{x}_\emptyset)|$ and $\kappa = 0.05 \cdot |v^{(l)}(\boldsymbol{x}_N) - v^{(l)}(\boldsymbol{x}_\emptyset)|$ through different layers, as well as the raw ResNet-20. We rearranged all of the AND-OR interactions in the order of the interaction value strength at $\kappa = 0.04 \cdot |v^{(l)}(\boldsymbol{x}_N) - v^{(l)}(\boldsymbol{x}_\emptyset)|$.

K. Discussions on Distinctive Information-Processing Behaviors of Each Specifc DNN

We discovered that in most DNNs, low layers and middle layers usually learned to fit target interactions that were finally used by DNNs at the cost of encoding lots of redundant interactions. Such redundant interactions would be removed in high layers.

Distinctive information-processing behaviors of different DNNs. Specifically, for DNNs trained on the MNIST dataset, they usually learned the target interactions for inference quickly, because the MNIST dataset was easy to learn. Particularly, for the ResNet-20 trained on both the MNIST dataset and the CIFAR-10 dataset, its low layers and middle layers mainly learned target interactions for inference, while high layers mainly forgot high-order interactions. These high-order interactions were unstable and exhibited poor generalization capacity, as verified in Section 3.3.

For MLP-7 and VGG-11 trained on the CIFAR-10 dataset, low layers were unable to learn interactions that could be directly used for classification, due to the challenge of classification on the CIFAR dataset. Then, middle layers and high layers gradually learned the target interactions for inference without generating redundant interactions. High layers did not change the interactions significantly.

For the DistilBERT and BERTBASE trained on the SST-2 dataset, low layers usually could not encode target interactions. Then, middle layers gradually learned the target interactions for inference, but also brought in lots of redundant interactions. High layers usually forgot redundant interactions, which were mainly high-order and unstable.

L. Discussion on the Noise Ratio of Interactions over Different Orders

In section 3.3 of the main paper, we add a small Gaussian perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ to the input sample $\boldsymbol{x}$, in order to mimic inevitable noises/variations in data. And then we use the new metrics to measure the relative stability of AND-OR interactions of each order m as follow.

$$\begin{aligned} \textit{stability}_{\text{and}}^{(l),m} &= \mathbb{E}_{\boldsymbol{x}} \mathop{\mathbb{E}}_{S \in \Omega_{\text{and}}^{(l),m}} \left[|E_{\text{and}}^{(l),m}(S, \boldsymbol{x})| \Big/ \sqrt{\textit{Var}_{\text{and}}^{(l),m}(S, \boldsymbol{x})} \right] \\ \textit{stability}_{\text{or}}^{(l),m} &= \mathbb{E}_{\boldsymbol{x}} \mathop{\mathbb{E}}_{S \in \Omega_{\text{or}}^{(l),m}} \left[|E_{\text{or}}^{(l),m}(S, \boldsymbol{x})| \Big/ \sqrt{\textit{Var}_{\text{or}}^{(l),m}(S, \boldsymbol{x})} \right] \end{aligned} \tag{21}$$

where $E_{\text{and}}^{(l),m}(S, \boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\epsilon}}[I_{\text{and}}(S|\boldsymbol{x} + \boldsymbol{\epsilon}, v^{(l)})]$ and $\textit{Var}_{\text{and}}^{(l),m}(S, \boldsymbol{x}) = \textit{Var}_{\boldsymbol{\epsilon}}[I_{\text{and}}(S|\boldsymbol{x} + \boldsymbol{\epsilon}, v^{(l)})]$ denote the mean and variance of the AND interaction $I_{\text{and}}(S|\boldsymbol{x} + \boldsymbol{\epsilon}, v^{(l)})$ *w.r.t.* Gaussian perturbations $\boldsymbol{\epsilon}$, which are encoded by the l-th layer of the DNN. In fact, higher-order interactions comprise more input variables, which means it would obtain more gaussian perturbations. However, the signal strength of higher order interaction also increases linearly with order m. Fig.5 in the main paper shows that low-order interactions are more stable to small noises. Here we verify that the noise ratio of each interaction is relatively consistent over interactions of different orders, so that stability enables a fair comparison of interactions of different orders.

To this end, we used the images in CIFAR-10 dataset and annotated semantic parts in each image, following Appendix M.1. Then we added the small Gaussian perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ to the image x, and obtained the noise image x_ϵ. For each interaction pattern S, we calculate the ratio of noise intensity to interaction pattern signal intensity, *e.g.*, $Radio_\epsilon^S = \|x^s - x_\epsilon^s\|_2 / \|x^s\|_2$.

Fig. 12 shows the distribution of $Radio_\epsilon^S$ under different interaction order. It can be found that the median $Radio_\epsilon^S$ values for different orders interactions is basically the same, and there is no phenomenon that the noise proportion of high-order interactions is significantly higher than that of low-order interactions. Therefore, the poor stability of the high order interaction is not due to its noise ratio.

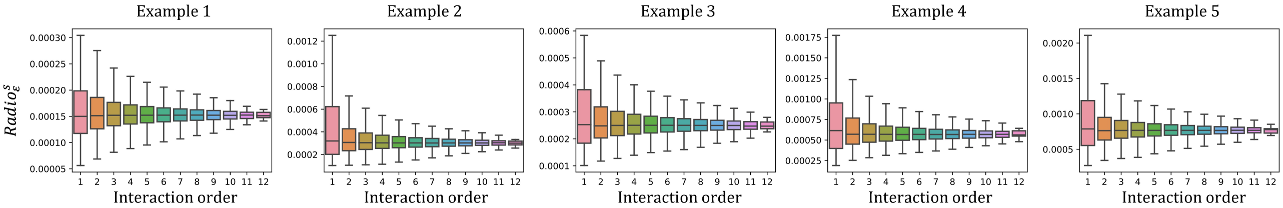


Figure 12. Box-and-whisker diagram of $Radio_\epsilon^S$ for interaction pattern S of different interaction order.

M. Experimental details

M.1. Annotating Semantics Parts

We followed (Li & Zhang, 2023) to annotate semantic parts in MNIST dataset and CIFAR-10 dataset. Given an input sample $\boldsymbol{x}\mathbb{R}^n$, the DNN may encode at most 2^n interactions. The computational cost for extracting salient interactions is high, when the number of input variables n is large. In order to overcome this issue, we simply annotate 10–12 semantic parts in each input sample, such that the annotated semantic parts are aligned over different samples in the same dataset. Then, each semantic part in an input sample is taken as a "single" input variable to the DNN.

- For images in the MNIST dataset, we followed settings in (Li & Zhang, 2023) to annotate semantic parts for 100 samples. Specifically, given an image, we divided the whole image into small patches of size 3×3. Considering the DNN mainly used the digit in the foreground to make inference, we selected $n = 10$ patches in the foreground as input variables to calculate interactions, in order to reduce the computational cost.

- For images in the CIFAR-10 dataset, we followed settings in (Ren et al., 2023a) to annotate semantic parts for 30 samples. Specifically, given an image, we divided the whole image into small patches of size 4×4, thereby obtaining 8×8 image patches in total. Considering the DNN mainly used information contained in the foreground to make inference, we randomly selected $n = 12$ patches from 6×6 image patches located in the center of the image, in order to reduce the computational cost.

- For the SST-2 dataset, we followed settings in (Ren et al., 2023a) to select sentences with a length of 10 words without unclear semantics, such as stop words. For each selected sentence, we considered each word as an input variable, thereby obtaining $n = 10$ input variables in sum. We used 50 sentences to calculate interactions in Section 3.

M.2. Training Linear Classifier in Section 3.2.2

Generally, the training parameters of the intermediate layers classifier are consistent with those of the original model.

- For MLP-7 trained on the MNIST dataset, we used SGD with learning rate 0.01, and set the batch size to 256 to train the intermediate layers. For VGG-11 trained on both the MNIST dataset, we used SGD with learning rate 0.001, and set the batch size to 256 to train the intermediate layers. For ResNet-20 trained on both the MNIST dataset, we used SGD with learning rate 0.001, and set the batch size to 256 to train the intermediate layers.

- For MLP-7 trained on the CIFAR-10 dataset, we used SGD with learning rate 0.001, and set the batch size to 256 to train the intermediate layers. For VGG-11 trained on both the CIFAR-10 dataset, we used SGD with learning rate 0.001, and set the batch size to 256 to train the intermediate layers. For ResNet-20 trained on both the CIFAR-10 dataset, we used SGD with learning rate 0.001, and set the batch size to 100 to train the intermediate layers.

- For DistilBERT finetuned on the SST-2 dataset, we used SGD with learning rate 2E-5, and set the batch size to 32 to train the intermediate layers. For BERTBASE finetuned on the SST-2 dataset, we used SGD with learning rate 1E-5, and set the batch size to 16 to train the intermediate layers.

M.3. Intermediate Layers Selected to Calculate Interactions in Section 3.3

- For DNNs trained on both the MNIST dataset and the CIFAR-10 dataset, we used intermediate layers close to the output to compute interactions. Specifically, the MLP-7 model contained 7 linear layers, and we used features of the 4-th linear layer. For the VGG-11 model, we employed features of $conv4\_2$. The ResNet-20 mdoel contained 9 residual blocks, and we used features after the 6-th residual block. The ResNet-32 model contained 15 residual blocks, and we used used features after the 10-th residual block.

- For DNNs trained on the SST2 dataset, we also used intermediate layers close to the output to compute interactions. Specifically, the DistilBERT model contained 6 transformers, and we employed features after the 4-th transformer. The BERT$_{\text{BASE}}$ model contained 12 transformers, and we employed features after the 8-th transformer. The XLNet model contained 12 transformer-XLs, we employed features after the 8-th transformer-XL.

M.4. Experimental Details for Verifying the Sparsity of Interactions in Section 3.2.

For each sample in the MNIST dataset, as introduced in Appendix M.1, we set $n = 10$. For each sample in the CIFAR-10 dataset, as introduced in Appendix M.1, we set $n = 12$. We randomly selected 100 images in the MNIST dataset and 30 images in the CIFAR-10 dataset to verify the sparsity of interactions. We set $\tau = 0.05 \cdot \max_{\boldsymbol{x}} \max_S(\max\{|I_{\text{and}}(S|\boldsymbol{x}, v^{(l)})|, |I_{\text{or}}(S|\boldsymbol{x}, v^{(l)})|\})$ for each target layer of the target DNN to determine its salient interactions. Note that in experiments, we concluded first-order OR interactions to the first-order AND interactions for convenience. In other words, the first-order AND interactions were the sum of first-order OR interactions and the first-order AND interactions, because one single input variable could be considered as either OR relationship or AND relationship with itself.