A REPRESENTATION BOTTLENECK OF BAYESIAN NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unlike standard deep neural networks (DNNs), Bayesian neural networks (BNNs) formulate network weights as probability distributions, which results in distinctive representation capacities from standard DNNs. In this paper, we explore the representation bottleneck of BNNs from the perspective of conceptual representations. It is proven that the logic of a neural network can be faithfully mimicked by a specific sparse causal graph, where each causal pattern can be considered as a concept encoded by the neural network. Then, we formally define the complexity of concepts, and prove that compared to standard DNNs, it is more difficult for BNNs to encode complex concepts. Extensive experiments verify our theoretical proofs. *The code will be released when the paper is accepted.*

1 INTRODUCTION

Unlike standard deep neural networks (DNNs), Bayesian neural networks (BNNs) represent network weights as probability distributions. Therefore, BNNs exhibit distinctive representation capacities from standard DNNs. Existing studies (Blundell et al., 2015; Gal & Smith, 2018; Kristiadi et al., 2020; Carbone et al., 2020; Wenzel et al., 2020; Krishnan et al., 2020; Zhang et al., 2022) usually analyzed BNNs in terms of generalization power, adversarial robustness, and optimization.

Unlike the above studies, this paper focuses on a new perspective to investigate the representation capacity of BNNs, *i.e.*, which types of concepts are more likely or less likely to be encoded by a BNN. Specifically, we discover and theoretically prove that BNNs are less likely to encode complex concepts than standard DNNs.

Representing concepts encoded by a neural network. Mathematically formulating concepts encoded by a neural network has been considered as a big problem for decades. Fortunately, Ren et al. (2021a) have proven that a specific sparse causal graph can faithfully explain the inference logic of a neural network. In the causal graph, each intermediate node is a causal pattern, which encodes the AND relationship between a set of input variables. For example, in face recognition, when eyes, nose, and mouth appear together, they form a *causal pattern* $=$ {eyes, nose, mouth}. In this way, **each causal pattern can be considered as *an interactive concept* encoded by the neural network**.

More importantly, the following *faithfulness* and *sparsity* of using the causal graph to explain a neural network further guarantee the trustworthiness of using causal patterns to represent concepts encoded by the neural network. Given an input sample with n variables, there are 2^n different ways to randomly mask input variables. It is proven (Ren et al., 2021a) that we can usually construct a sparse enough causal graph, which only contains as few as tens of causal patterns (interactive concepts), such that the causal graph can accurately mimic the output of the neural network on as many as 2^n randomly masked samples.

Specifically, we can further understand an interactive concept (causal pattern) encoded by a neural network as follows. In fact, the neural network usually does not consider each input variable (*e.g.*, a patch in an image) to work independently. Instead, the neural network encodes an AND relationship between a set S of input variables to form an interactive concept for inference, which can be logically represented as $I(S = \{\text{eyes, nose, mouth}\}) = U_S \cdot \text{exist}(\text{eyes}) \cdot \text{exist}(\text{nose}) \cdot \text{exist}(\text{mouse})$, in the above example of face recognition. If any image patch in the set $S = \{\text{eyes, nose, mouth}\}$ is masked, the face concept will be deactivated, and a causal effect U_S is removed from the inference score.

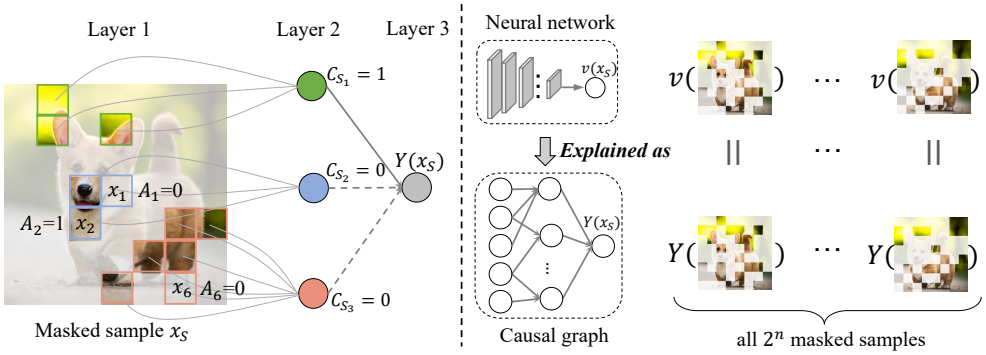

Figure 1: (Left) The causal graph explanation of a neural network. (Right) The faithfulness of the causal graph explanation. Specifically, for all 2^n masked input samples $\boldsymbol{x}_S$, all outputs $v(\boldsymbol{x}_S)$ of a neural network can be well mimicked by the outputs $Y(\boldsymbol{x}_S)$ of the corresponding causal graph.

Conceptual complexity. We define the complexity of an interactive concept S as the number of variables in the set S, which is also termed *the order of the interactive concept*, *i.e.*, $\text{order}(S) = |S|$. Then, a low-order interactive concept represents a relatively simple collaboration among a few variables, and a high-order interactive concept reflects a relatively complex collaboration among a large number of variables.

We discover and theoretically prove that **compared to standard DNNs, it is more difficult for BNNs to encode high-order (complex) interactive concepts.** Specifically, we progressively prove this conclusion through the following three steps.

First, it is difficult to theoretically analyze interactive concepts encoded by BNNs, because BNNs represent network weights as probability distributions. To this end, we demonstrate that we can use a *surrogate DNN model*, which is constructed by adding perturbations to both the input and low-layer features of a standard DNN, to mimic feature representations of a BNN. In this way, we can directly analyze the surrogate DNN model with feature uncertainty, instead of investigating the BNN with weight uncertainty.

Second, we theoretically prove that in the surrogate DNN model, high-order interactive concepts are more sensitive to random perturbations than low-order interactive concepts.

Third, we theoretically prove that the sensitivity makes high-order interactive concepts difficult to be learned when features are perturbed. In this way, we can conclude that high-order interactive concepts are also less likely to be learned by the BNN when its weights are perturbed.

In addition, extensive experiments showed that the strength of high-order (complex) interactive concepts encoded by BNNs was weaker than those encoded by standard DNNs, which verified the above theoretical conclusion.

Potential values of our theoretical proof. This study clarifies the shortcoming of the BNN in encoding complex concepts, which may explain the inferior performance of the BNN from a new perspective. Furthermore, it has been found that the complexity (order) of interactive concepts encoded by a neural network is closely connected with the generalization power (Lengerich et al., 2022) and adversarial robustness (Ren et al., 2021b) of the neural network. Therefore, our study sheds new light on further explaining the representation capacity of BNNs.

2 Representation Bottleneck of BNNs

Unlike standard DNNs, a BNN represents each weight in the network as a probability distribution, instead of a fixed value. In this paper, let us limit our study to the scope of classical BNNs, in which all weights $\boldsymbol{W}$ are formulated as a Gaussian distribution $\mathcal{N}(\boldsymbol{W}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ (Blundell et al., 2015), and the covariance matrix $\boldsymbol{\Sigma}$ is a diagonal matrix. All extended versions (*e.g.*, Gal & Ghahramani (2016)) are not discussed. The BNN learns parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and we use $q_{\boldsymbol{\theta}}(\boldsymbol{W})$ to represent the weight distribution. Let us consider a classification task given the training data $\mathcal{D} = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})\}$. A typical way of training a BNN (Blundell et al., 2015) is to

minimize the Kullback-Leibler (KL) divergence between the distribution $q_{\boldsymbol{\theta}}(\boldsymbol{W})$ and the posterior distribution $p(\boldsymbol{W}|\mathcal{D})$.

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathrm{KL}[q_{\boldsymbol{\theta}}(\boldsymbol{W})\|p(\boldsymbol{W}|\mathcal{D})] = \arg\min_{\boldsymbol{\theta}} -\mathbb{E}_{\boldsymbol{W}\sim q_{\boldsymbol{\theta}}(\boldsymbol{W})}[\log p(\mathcal{D}|\boldsymbol{W})] + \mathrm{KL}[q_{\boldsymbol{\theta}}(\boldsymbol{W})\|p(\boldsymbol{W})], \quad (1)$$

where the first term is the classification loss, and the second term is the KL divergence between $q_{\boldsymbol{\theta}}(\boldsymbol{W})$ and the prior distribution of weights $p(\boldsymbol{W})$, which is usually formulated as a Gaussian distribution $\mathcal{N}(\boldsymbol{W}; \mathbf{0}, \boldsymbol{I})$. In addition, given a testing sample $\boldsymbol{x}$, the inference of the BNN is conducted as follows. First, network weights are sampled from the weight distribution $q_{\boldsymbol{\theta}}(\boldsymbol{W})$ to construct multiple neural networks. Then, each network is used to conduct inference on the sample $\boldsymbol{x}$, and the final inference result $p(y|\boldsymbol{x})$ is computed as the average classification probability of all these networks.

$$p(y|\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{W}\sim q_{\boldsymbol{\theta}}(\boldsymbol{W})}[p(y|\boldsymbol{x}, \boldsymbol{W})] \quad (2)$$

2.1 Representing a neural network by interactive concepts

• **Explaining a neural network as a causal graph.** In fact, Ren et al. (2021a) have proven that the inference logic of a neural network can be represented as a specific sparse causal graph. Given a trained neural network v and an input sample $\boldsymbol{x} = [x_1, \cdots, x_n]$ with n input variables indexed by $N = \{1, \ldots, n\}$, the inference logic of the neural network can be well mimicked by the corresponding causal graph. As illustrated in Figure 1, the causal graph consists of three layers. The *first layer* contains n nodes. Each node indicates whether each input variable x_i is present or masked, and the masking state of x_i is denoted by an indicator variable $A_i \in \{0, 1\}$. The *second layer* is composed of all causal patterns in the set $\Omega \subseteq 2^N = \{S|S \subseteq N\}$. Each causal pattern encodes an AND relationship between input variables in the subset $S \subseteq N$. Accordingly, each node $C_S \in \{0, 1\}$ in the *second layer* represents the triggering state of each causal pattern. For example, in face recognition, only when input variables in the set $S = \{\text{eyes, nose, mouth}\}$ all appear together, the face pattern will be triggered ($C_S = 1$); otherwise, the face pattern will be deactivated ($C_S = 0$). The *third layer* contains a single node $Y \in \mathbb{R}$ as the output of the causal graph. Therefore, the transition probability of the specific causal graph can be calculated as

$$P(C_S = 1|A_1, A_2, \ldots, A_n) = \prod\nolimits_{k\in S} A_k, \quad P(Y|\{C_S|S \in \Omega\}) = \mathbb{1}(Y = \sum\nolimits_{S\in\Omega} C_S \cdot U_S). \quad (3)$$

Here, $P(C_S = 0|A_1, A_2, \ldots, A_n) = 1 - P(C_S = 1|A_1, A_2, \ldots, A_n)$. In addition, $U_S \in \mathbb{R}$ and $\mathbb{1}(\cdot)$ denotes the indicator function.

Theorem 1. *Given an input sample $\boldsymbol{x}$ with n input variables, let $\boldsymbol{x}_S$ denote a masked input sample, where variables in $N \setminus S$ are masked and variables in S keep unchanged. It is proven that for each neural network v, there exists a specific causal graph (parameterized by $\{U_{S'}|S' \in \Omega\}$), such that for any arbitrarily masked input sample $\boldsymbol{x}_S$, the output $v(\boldsymbol{x}_S)$ of the neural network can be well mimicked by the output $Y(\boldsymbol{x}_S)$ of the causal graph, i.e.,*

$$\exists\, \Omega \subseteq 2^N, \exists\{U_{S'}|S' \in \Omega\},\ s.t., \quad \forall S \subseteq N,\ v(\boldsymbol{x}_S) = Y(\boldsymbol{x}_S) \quad (4)$$

where $Y(\boldsymbol{x}_S)$ denotes the output of causal graph (see Eq. (5)) on the masked sample $\boldsymbol{x}_S$ by setting $A_i = \mathbb{1}(i \in S)$. In particular, a special solution of the causal effects U_S (the causal graph) satisfying the above equation is the Harsanyi dividend (Harsanyi, 1963), $U_S = \sum_{S'\subseteq S}(-1)^{|S|-|S'|} v(\boldsymbol{x}_{S'})$.

Faithfulness of the causal graph. Given a neural network v, Theorem 1 shows the faithfulness of the corresponding causal graph explanation. Theoretically, given an input sample $\boldsymbol{x}$ with n input variables, there are as many as 2^n different ways of masking input variables in all potential subsets $S \subseteq N$, so as to obtain 2^n masked samples $\boldsymbol{x}_S$. Theorem 1 proves that for all the exponential number of arbitrarily masked samples $\boldsymbol{x}_S$, diverse outputs $v(\boldsymbol{x}_S)$ of the neural network can be well mimicked by outputs $Y(\boldsymbol{x}_S)$ of the corresponding causal graph. In this way, it is theoretically supported that the inference logic of the neural network can be faithfully explained as a specific causal graph.

Sparsity of the causal graph. Remark 1 shows that the inference logic of a neural network can usually be explained by a very sparse causal graph.

Remark 1. Causal effects U_S of most causal patterns are actually negligible, *i.e.*, $|U_S| \approx 0$, and only a few salient causal patterns have significant causal effects. Thus, instead of enumerating all subsets of input variables ($\forall S \subseteq N$), we can find a sparse subgraph, which contains a few causal patterns with top-ranked causal effects in the set $\Omega' \subseteq \Omega$, $|\Omega'| \ll 2^n$, such that the output of the neural network can be approximated by the sparse causal graph, *i.e.*, $v(\boldsymbol{x}_S) \approx Y(\boldsymbol{x}_S|\Omega')$. The sparsity of the causal graph is empirically verified in Appendix B.

• **Understanding causal patterns in the causal graph as interactive concepts.** The faithfulness and sparsity of the causal graph guarantee that the logic of a neural network can be well explained by a relatively small number of causal patterns. Specifically, the transition probability in Eq. (3) of the causal graph can be rewritten as the following structural causal model (SCM) (Pearl, 2009)

$$Y(\boldsymbol{x}_S) = \sum\nolimits_{S'\in\Omega} I(S'), \quad \text{where } I(S') = U_{S'} \cdot C_{S'}(\boldsymbol{x}_S) = U_{S'} \cdot \prod\nolimits_{i\in S'} A_i \tag{5}$$

The above equation indicates that the output $Y(\boldsymbol{x}_S)$ of the causal graph is the sum of causal effects of all causal patterns, where U_S denotes the causal effect of the causal pattern S. Each causal pattern, in particular, indicates an AND relationship between multiple input variables, which can be considered as an *interactive concept* memorized by the neural network. For example, the causal pattern $S = \{\text{eyes, nose, mouth}\}$ can be understood as an interactive concept. Only when eyes, nose, and mouth co-appear, the interactive concept will be triggered ($C_S = 1$) and make a causal effect $I(S) = U_S$ on the output of the causal graph. In contrast, the absence of any input variables will deactivate the interactive concept and remove the causal effect, *i.e.*, $C_S = 0$ and $I(S) = 0$.

• **Conceptual complexity**. Given an interactive concept S, its complexity is defined as the number of input variables in the set S, which is also termed *the order of the interactive concept*, *i.e.*, $\text{order}(S) = |S|$. In this way, a low-order interactive concept in the causal graph encodes a relatively simple collaboration among a small number of input variables, while a high-order interactive concept represents a relatively complex collaboration among a large number of input variables.

2.2 Approximating weight uncertainty by adding input perturbations

As proven above, a sparse causal graph can well mimic the network output on all 2^n arbitrarily masked input samples, which guarantees the trustworthiness of using interactive concepts to explain a network. In the following subsections, we further prove that compared to standard DNNs, it is more difficult for BNNs to encode high-order (complex) interactive concepts, which represent intricate collaborations between a large number of input variables.

However, unlike standard DNNs, a BNN formulates each weight as a probability distribution, which boosts the difficulty of theoretically analyzing a BNN. Therefore, the first step of our work is to add random perturbations to both input variables and low-layer features of a standard DNN, and to demonstrate that such a perturbed DNN can well approximate feature representations of a BNN, as a *surrogate DNN model*. In other words, we demonstrate that introducing uncertainty to weights in the BNN can be approximated by adding perturbations to input variables and low-layer features.

Let us consider a feed-forward BNN, which has L cascaded linear layers and ReLU layers. Given an input sample $\boldsymbol{x} \in \mathbb{R}^{D_0}$ ($D_0 = n$), the feature of the l-th layer $\boldsymbol{h}^{(l)} \in \mathbb{R}^{D_l}$ is computed as follows.

$$\forall\, 1 \le l \le L, \quad \boldsymbol{h}^{(l)} = \boldsymbol{W}^{(l)}(\cdots \boldsymbol{\Phi}^{(1)}(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})\cdots) + \boldsymbol{b}^{(l)}, \tag{6}$$

where $\boldsymbol{W}^{(l)} \in \mathbb{R}^{D_l\times D_{l-1}}$ and $\boldsymbol{b}^{(l)} \in \mathbb{R}^{D_l}$ denote the weight matrix and bias of the l-th linear layer, respectively. In the BNN, $W_{ij}^{(l)} \sim \mathcal{N}(\overline{W}_{ij}^{(l)}, (\sigma_{ij}^{(l)})^2)$ is independently sampled from Gaussian distributions. We use $\boldsymbol{\mu}_{\boldsymbol{W}^{(l)}} = [\overline{W}_{ij}^{(l)}] \in \mathbb{R}^{D_l\times D_{l-1}}$ to denote the mean of the weight matrix. Besides, $\boldsymbol{b}^{(l)} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{b}^{(l)}}, \boldsymbol{\Sigma}_{\boldsymbol{b}^{(l)}})$, where $\boldsymbol{\Sigma}_{\boldsymbol{b}^{(l)}}$ is a diagonal matrix. The diagonal matrix $\boldsymbol{\Phi}^{(l)} = \text{diag}(\phi_1^{(l)}, \cdots, \phi_{D_l}^{(l)}) \in \{0,1\}^{D_l\times D_l}$ denotes binary gating states of the l-th ReLU layer.

Therefore, given an input sample $\boldsymbol{x}$, let us focus on the feature $\boldsymbol{h}^{(l)}$ of the l-th layer in the BNN, which follows a specific distribution $p(\boldsymbol{h}^{(l)})$. Then, we construct the surrogate DNN model with the same architecture as the BNN, and the parameters of this surrogate DNN model $\boldsymbol{\psi}$ are set as the mean of the weight distribution and the mean of the bias distribution in the BNN, *i.e.*, $\boldsymbol{\psi} = \{\boldsymbol{\mu}_{\boldsymbol{W}^{(l)}}, \boldsymbol{\mu}_{\boldsymbol{b}^{(l)}}\}_{l=1}^{L}$. Then, we add perturbations $\Delta\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\Delta\boldsymbol{x}})$ to input variables and perturbations $\Delta\boldsymbol{h}^{(l')} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\Delta\boldsymbol{h}^{(l')}})$ to features between the first layer and the $(l-1)$-th layer, in order to let the feature distribution $p(\tilde{\boldsymbol{h}}^{(l)}|\boldsymbol{\psi}, \boldsymbol{\Delta})$ of the l-th layer in the surrogate DNN model to mimic the feature distribution $p(\boldsymbol{h}^{(l)})$ in the BNN. In this way, the objective function of approximation is formulated as minimizing the following KL divergence.

$$\forall\, 1 \le l \le L, \quad \min_{\boldsymbol{\Delta}} \text{KL}(p(\boldsymbol{h}^{(l)}) \| p(\tilde{\boldsymbol{h}}^{(l)}|\boldsymbol{\psi}, \boldsymbol{\Delta})). \tag{7}$$

where $\boldsymbol{\Delta} = \{\boldsymbol{\Sigma}_{\Delta\boldsymbol{x}}, \boldsymbol{\Sigma}_{\Delta\boldsymbol{h}^{(1)}}, \ldots, \boldsymbol{\Sigma}_{\Delta\boldsymbol{h}^{(l-1)}}\}$ denotes covariance matrices of perturbations added to input variables and intermediate-layer features of the surrogate DNN model. In addition, $\boldsymbol{\Sigma}_{\Delta\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{\Delta\boldsymbol{h}^{(l')}}$ are diagonal matrices.

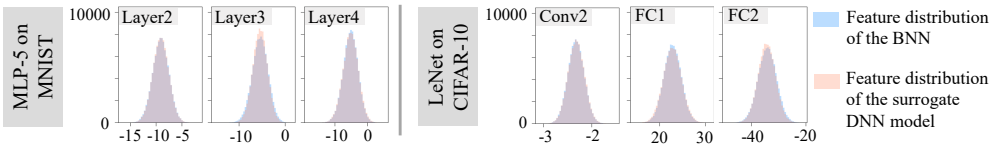

Figure 2: Comparison between the feature distribution of the BNN and the feature distribution of the surrogate DNN model. We randomly selected a feature dimension from each layer of the network. Each sub-figure compares feature distributions between the BNN and the surrogate DNN model in the selected dimension. Please see Appendix F.1 for comparison results on tabular datasets.

However, it is difficult to directly optimize Eq. (7). Therefore, we learn $\boldsymbol{\Delta}$ in a layer-wise manner, as follows. First, we learn the covariance matrix $\boldsymbol{\Sigma}_{\Delta \boldsymbol{x}}$ on input variables to match the first-layer feature of the surrogate DNN model to the first-layer feature of the BNN, *i.e.*, $\min_{\boldsymbol{\Sigma}_{\Delta \boldsymbol{x}}} \mathrm{KL}(p(\boldsymbol{h}^{(1)}) \| p(\tilde{\boldsymbol{h}}^{(1)} | \boldsymbol{\psi}, \boldsymbol{\Sigma}_{\Delta \boldsymbol{x}}))$. Then, we keep $\boldsymbol{\Sigma}_{\Delta \boldsymbol{x}}$ fixed, and learn the covariance matrix $\boldsymbol{\Sigma}_{\Delta \boldsymbol{h}^{(1)}}$ on the first-layer feature to fit feature distributions of the second layer by minimizing $\mathrm{KL}(p(\boldsymbol{h}^{(2)}) \| p(\tilde{\boldsymbol{h}}^{(2)} | \boldsymbol{\psi}, \boldsymbol{\Sigma}_{\Delta \boldsymbol{x}}, \boldsymbol{\Sigma}_{\Delta \boldsymbol{h}^{(1)}}))$. We recursively learn the covariance matrix of the upper layer by fixing covariance matrices in all lower layers, until the last layer.

• **Experiments.** We trained BNNs on image datasets and tabular datasets to verify the quality of using the surrogate DNN model to approximate the feature distribution of the BNN. For image datasets, we tested BNNs with two architectures. For the MNIST dataset (LeCun et al., 1998), we constructed a BNN with the architecture of a 5-layer MLP. We also tested a BNN with the LeNet architecture (LeCun et al., 1998), which was trained on the CIFAR-10 dataset (Krizhevsky et al., 2009). We used two tabular datasets, including the UCI TV news dataset (termed *TV news*) and the UCI census income dataset (termed *Census*) (Dua & Graff, 2017). We constructed BNNs with an 8-layer MLP architecture for these tabular datasets. All MLPs contained 100 neurons in each hidden layer. For each BNN, we constructed a corresponding surrogate DNN model. Please see Appendix E for training details and experimental details. Figure 2 shows that the feature distribution of the surrogate DNN model well matched the feature distribution of the BNN.

Experimental results showed that the weight uncertainty in a BNN could be well approximated by adding random perturbations to both input variables and low-layer features.

2.3 High-order interactive concepts are sensitive to perturbations

In the above subsection, we have demonstrated that introducing the weight uncertainty in a BNN is approximately equivalent to adding random perturbations to both input variables and features of different layers. However, simultaneously adding perturbations to features of multiple layers significantly boosts the difficulty of analysis. Fortunately, adding perturbations to output features of the l-th layer can be considered as perturbing input variables of the $(l+1)$-th layer. Hence, we can analyze interactive concepts in a much simpler case that we perturb input variables of a certain layer, instead of investigating the complex case of simultaneously perturbing features of different layers.

In this subsection, we theoretically prove that high-order interactive concepts are more sensitive to input perturbations than low-order interactive concepts. To facilitate the proof, given the function of a network $v(\boldsymbol{x})$, we first derive the analytical form of causal effect $I(S)$ of an interactive concept.

Lemma 1 (Proof in Appendix A.2)**.** *Originally, the causal effect in Eq. (5) is defined as a binary variable in the causal graph, $I(S) \in \{U_S, 0\}$. Given a continuous network function $v(\boldsymbol{x})$, we can use the following Taylor expansion to decompose the network output, which extends the causal effect $I(S)$ to a continuous function. This continuous function well fits the binary states of $I(S)$ on all the 2^n masked samples $\boldsymbol{x}'$ with different masking states, i.e., $\boldsymbol{x}' \in \{\boldsymbol{x}_T | \forall T \subseteq N\}$.*

$$v(\boldsymbol{x}') = \sum\nolimits_{S \subseteq N} \sum\nolimits_{\boldsymbol{\pi} \in Q_S} U_{S,\boldsymbol{\pi}} \cdot J(S, \boldsymbol{\pi} | \boldsymbol{x}') \;\Rightarrow\; I(S|\boldsymbol{x}') = \sum\nolimits_{\boldsymbol{\pi} \in Q_S} U_{S,\boldsymbol{\pi}} \cdot J(S, \boldsymbol{\pi} | \boldsymbol{x}'), \quad (8)$$

where $J(S, \boldsymbol{\pi} | \boldsymbol{x}') = \prod_{i \in S} \left(\mathrm{sign}(x'_i - r_i) \cdot \frac{x'_i - r_i}{\tau} \right)^{\pi_i}$ denotes a Taylor expansion term of the degree $\boldsymbol{\pi}$. Here, $\boldsymbol{\pi} \in Q_S = \{[\pi_1, \ldots, \pi_n] | \forall i \in S, \pi_i \in \mathbb{N}^+; \forall i \notin S, \pi_i = 0\}$. In addition, $U_{S,\boldsymbol{\pi}} = \frac{\tau^m}{\prod_{i=1}^n \pi_i!} \frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1} \cdots \partial x_n^{\pi_n}} \cdot \prod_{i \in S} [\mathrm{sign}(x'_i - r_i)]^{\pi_i}$, s.t. $m = \sum_{i=1}^n \pi_i$. $v(\boldsymbol{x}_\emptyset)$ indicates the network output when we mask all input variables to reference values r_i. Moreover, $C_S(\boldsymbol{x}') = I(S|\boldsymbol{x}')/U_S$.

In Lemma 1, the reference value r_i of the input variable x_i is set as follows. Let $\mathbb{E}_{\boldsymbol{x}}[x_i]$ denote the average value of the input variable x_i over all input samples, which is usually regarded as a no-information state of this input variable (Ancona et al., 2019). In this paper, we remove the information from the input variable x_i by pushing x_i by a large enough distance τ towards its mean value. In other words, if $x_i > \mathbb{E}_{\boldsymbol{x}}[x_i]$, we set the reference value $r_i = x_i - \tau$; otherwise, $r_i = x_i + \tau$. Here, $\tau \in \mathbb{R}$ is a pre-defined constant. Furthermore, compared to directly setting $r_i = \mathbb{E}_{\boldsymbol{x}}[x_i]$, the above setting ensures comparable perturbation magnitudes over different input dimensions.

Based on Lemma 1, we analyze the sensitivity of the causal effect $I(S)$, when we add a small Gaussian perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ to the input sample $\boldsymbol{x}$. To simplify the proof, we can ignore the extremely low probability of large perturbations $|\epsilon_i| \geq \tau$, because of the small variance σ^2.

Let us start with the simplest case in Lemma 1. Since people usually adopt low-order Taylor expansion for approximation in real implementations, we first approximate the causal effect $I(S|\boldsymbol{x}')$ using the expansion term of the lowest degree (corresponds to $\hat{\boldsymbol{\pi}}$ satisfying $\forall i \in S, \hat{\pi}_i = 1; \forall i \notin S, \hat{\pi}_i = 0$). In this case, the causal effect $I(S|\boldsymbol{x}')$ is given by $I(S|\boldsymbol{x}') \approx U_{S,\hat{\boldsymbol{\pi}}} \cdot J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}')$, according to Eq. (8).

Theorem 2 (Proof in Appendix A.3)**.** *If we only consider the approximation based on the lowest degree $\hat{\boldsymbol{\pi}}$, then the mean and variance of $I(S|\boldsymbol{x}+\boldsymbol{\epsilon})$ over different perturbations $\boldsymbol{\epsilon}$ are given as*

$$\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})] = U_{S,\hat{\boldsymbol{\pi}}}, \quad \mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})] = U_{S,\hat{\boldsymbol{\pi}}}^2((1+(\sigma/\tau)^2)^{|S|}-1) \tag{9}$$

Theorem 2 proves that *the variance* $\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]$ *increases along with the order* $|S|$ *of the interactive concept in an exponential manner*. It indicates that high-order interactive concepts are much more sensitive to input perturbations than low-order concepts. Furthermore, as mentioned in Section 2.2, since we can add perturbations to a surrogate DNN model to well mimic feature representations of a BNN, **we can consider that high-order interactive concepts encoded by the BNN are much more sensitive to weight uncertainty in the BNN than low-order concepts.**

Furthermore, we can extend Theorem 2 to a **more general case**, where we use a higher-order Taylor expansion to represent $I(S|\boldsymbol{x}')$.

Theorem 3 (Proof in Appendix A.4)**.** *Let $\boldsymbol{\pi} \in Q_S = \{[\pi_1, \ldots, \pi_n] | \forall i \in S, \pi_i \in \mathbb{N}^+; \forall i \notin S, \pi_i = 0\}$ denote an arbitrary degree. Then, the mean and the variance of $J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})$ over $\boldsymbol{\epsilon}$ are given as*

$$\mathbb{E}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] = \mathbb{E}_{\boldsymbol{\epsilon}}[\prod_{i\in S}(1+\frac{\epsilon_i}{\tau})^{\pi_i}], \quad \mathrm{Var}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] = \mathrm{Var}_{\boldsymbol{\epsilon}}[\prod_{i\in S}(1+\frac{\epsilon_i}{\tau})^{\pi_i}] \tag{10}$$

Theorem 4 (Proof in Appendix A.5)**.** *Let S' be an interactive concept extended from the concept S, i.e., $S \subsetneq S'$. Let us consider expansion terms $J(S, \boldsymbol{\pi})$ and $J(S', \boldsymbol{\pi}')$, where the term $J(S', \boldsymbol{\pi}')$ can be considered being extended from the term $J(S, \boldsymbol{\pi})$ with $\boldsymbol{\pi} \prec \boldsymbol{\pi}'$. I.e., (1) $\forall i \in S', \pi'_i \in \mathbb{N}^+$; otherwise, $\pi'_i = 0$. (2) Given $\boldsymbol{\pi}'$, $\forall j \in S, \pi_j = \pi'_j$; otherwise, $\pi_j = 0$. Then, we have*

$$\begin{aligned} \frac{\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S',\boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]}{\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]} &> \prod\nolimits_{i\in S'\setminus S} \mathbb{E}^2_{\epsilon_i}[(1+\tfrac{\epsilon_i}{\tau})^{\pi'_i}], \\ \frac{\mathbb{E}_{\boldsymbol{\epsilon}}[J(S',\boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S',\boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]}{\mathbb{E}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]} &< \frac{1}{\prod_{i\in S'\setminus S}\mathbb{E}_{\epsilon_i}[(1+\frac{\epsilon_i}{\tau})^{\pi'_i}]}, \end{aligned} \tag{11}$$

and it is easy to obtain $\mathbb{E}_{\epsilon_i}[(1+\frac{\epsilon_i}{\tau})^{\pi'_i}] \geq 1$.

Remark 2. According to Theorem 4, we can obtain that for an arbitrary degree $\boldsymbol{\pi}$ of the interactive concept $\boldsymbol{S}$, $\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S',\boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]$ increases in an exponential manner along with $|S' \setminus S| = |S'| - |S|$. Therefore, we can roughly consider that $\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]$ increases exponentially *w.r.t.* the order $|S|$. Furthermore, according to Lemma 1, $I(S|\boldsymbol{x}+\boldsymbol{\epsilon})$ can be re-written as the weighted sum of $J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})$. Since coefficients $U_{S,\boldsymbol{\pi}}$ *w.r.t.* different S and $\boldsymbol{\pi}$ are usually chaotic, we can consider that the sensitivity of $I(S|\boldsymbol{x}+\boldsymbol{\epsilon})$ also grows exponentially along with the order $|S|$ of the interactive concept S. In addition, Theorem 4 also proves the approximately exponential decrease of $\frac{\mathbb{E}_{\boldsymbol{\epsilon}}[J(S',\boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S',\boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]}{\mathbb{E}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]}$ along with $|S'|-|S|$. Similarly, we can obtain that the relative stability $\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]$ decreases along with the order $|S|$.

• **Conclusions.** Both Theorem 2 and Remark 2 tell us that high-order interactive concepts are much more sensitive to input perturbations. Furthermore, combined with the conclusion in Section 2.2, **we can conclude that high-order interactive concepts encoded by the BNN are much more sensitive to the weight uncertainty in the BNN than low-order concepts.**

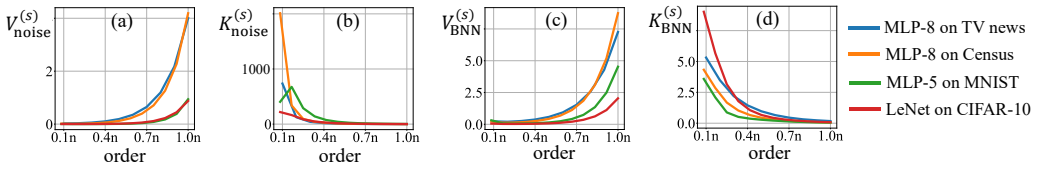

Figure 3: (a) The exponential increase of the average variance $V^{(s)}_{\text{noise}}$ and (b) the roughly exponential decrease of the average relative stability $K^{(s)}_{\text{noise}}$ along with the order s, under perturbations from a prior distribution[1] $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.05^2 \cdot \boldsymbol{I})$. (c) The exponential increase of the average variance $V^{(s)}_{\text{BNN}}$ and (d) the roughly exponential decrease of the average relative stability $K^{(s)}_{\text{BNN}}$ along with the order s, under weight uncertainty in the BNN.

• **Experimental verification.** We conducted experiments to verify the above conclusions. To verify the sensitivity to input perturbations, we added a random perturbation $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ to a given input sample $\boldsymbol{x}$, where $\sigma^2 = 0.05^2$. Then, we used two metrics, $V^{(s)}_{\text{noise}} = \mathbb{E}_{\boldsymbol{x}}[\mathbb{E}_{|S|=s}[\text{Var}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]]]$ and $K^{(s)}_{\text{noise}} = \mathbb{E}_{\boldsymbol{x}}[\mathbb{E}_{|S|=s}[|\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]|/\text{Var}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]]]$, to measure the average variance and the average relative stability of the s-order interactive concepts *w.r.t.* the input perturbation $\boldsymbol{\epsilon}$. Then, a large value of $V^{(s)}_{\text{noise}}$ or a small value of $K^{(s)}_{\text{noise}}$ indicated that the s-order interactive concepts were sensitive to input perturbations.

Similarly, to verify the sensitivity to weight uncertainty, we sampled different weights $\boldsymbol{W}$ from the distribution $q_{\boldsymbol{\theta}}(\boldsymbol{W})$ of the BNN. Then, we used $V^{(s)}_{\text{BNN}} = \mathbb{E}_{\boldsymbol{x}}[\mathbb{E}_{|S|=s}[\text{Var}_{\boldsymbol{W}\sim q_{\boldsymbol{\theta}}(\boldsymbol{W})}[I(S|\boldsymbol{x},\boldsymbol{W})]]]$ and $K^{(s)}_{\text{BNN}} = \mathbb{E}_{\boldsymbol{x}}[\mathbb{E}_{|S|=s}[|\mathbb{E}_{\boldsymbol{W}\sim q_{\boldsymbol{\theta}}(\boldsymbol{W})}[I(S|\boldsymbol{x},\boldsymbol{W})]|/\text{Var}_{\boldsymbol{W}\sim q_{\boldsymbol{\theta}}(\boldsymbol{W})}[I(S|\boldsymbol{x},\boldsymbol{W})]]]$ to measure the average variance and the average relative stability of the s-order interactive concepts *w.r.t.* the weight uncertainty in the BNN. Therefore, a large value of $V^{(s)}_{\text{BNN}}$ or a small value of $K^{(s)}_{\text{BNN}}$ indicated that the s-order interactive concepts were sensitive to the weight uncertainty. We followed experimental settings in the *experiments* paragraph in Section 2.2 to train BNNs. Specifically, we trained BNNs with the MLP architecture on the MNIST dataset, the TV news dataset, and the Census dataset. We trained BNNs with the LeNet architecture on the CIFAR-10 dataset. Appendix E introduces how to efficiently compute $I(S|\boldsymbol{x})$ on images.

Figure 3 shows that the average variance $V^{(s)}_{\text{noise}}$ and $V^{(s)}_{\text{BNN}}$ increased exponentially along with the order s, while the relative stability $K^{(s)}_{\text{noise}}$ and $K^{(s)}_{\text{BNN}}$ both decreased along with the order. This demonstrated that high-order interactive concepts were much more sensitive to input perturbations and the weight uncertainty in the BNN, thereby verifying Theorem 2 and Remark 2.

2.4 Sensitive interactive concepts are difficult to learn

In the above subsection, we have proven that high-order interactive concepts were much more sensitive to weight uncertainty in the BNN. Then, the SCM in Eq. (5) and Theorem 1 allow us to roughly consider a neural network v as a linear function of different interactive concepts, *i.e.*, $v(\boldsymbol{x}) = Y(\boldsymbol{x}) = \sum_{S\in\Omega} U_S \cdot C_S(\boldsymbol{x})$. Then, $C_S(\boldsymbol{x})$ can be considered an input dimension of the linear function, which indicates whether the input sample $\boldsymbol{x}$ contains the interactive concept S. The coefficient U_S can be considered as the strength of the neural network in encoding the interactive concept S. Because most interactive concepts have negligible coefficients $|U_S| \approx 0$, we can consider that the neural network only encodes a few interactive concepts S with large absolute values $|U_S|$.

Based on the conclusion in Section 2.2, we can roughly consider that training a BNN on normal samples is equivalent to training a surrogate DNN model on perturbed input samples. Let us consider a regression problem for analysis. Then, according to Eq. (5), the learning of the BNN on a certain input sample can be roughly represented as $\min_{\{U_S|S\in\Omega\}} L(\{U_S\})$, and the loss is given by

$$L(\{U_S\}) = \mathbb{E}_{\boldsymbol{\epsilon}}\left[(y^* - v(\boldsymbol{x}+\boldsymbol{\epsilon}))^2\right] = \mathbb{E}_{\boldsymbol{\epsilon}}\left[(y^* - Y(\boldsymbol{x}+\boldsymbol{\epsilon}))^2\right] = \mathbb{E}_{\boldsymbol{\epsilon}}[(y^* - \sum_{S\in\Omega} U_S \cdot C_S(\boldsymbol{x}+\boldsymbol{\epsilon}))^2] \quad (12)$$

where $\boldsymbol{x}$ and y^* denote the input sample and the ground-truth output, respectively. The continuous version of $C_S(\boldsymbol{x}+\boldsymbol{\epsilon})$ is formulated in Lemma 1.

[1]The prior distribution is manually set, rather than being learned using Eq. (7).

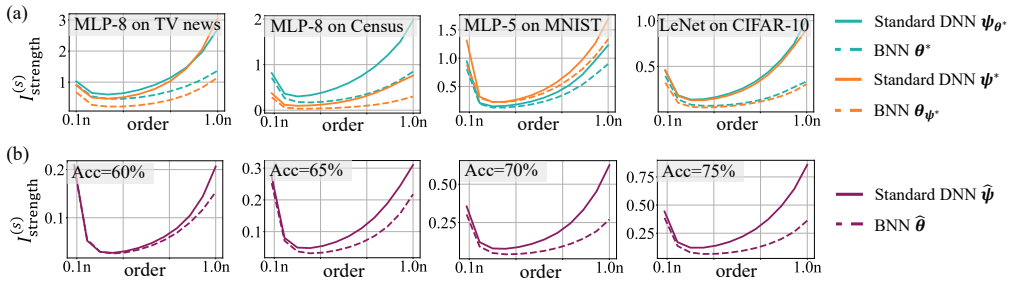

Figure 4: (a) Comparison of the strength of interactive concepts (i) between a trained BNN $\boldsymbol{\theta}^*$ and the constructed standard DNN $\psi_{\boldsymbol{\theta}^*}$, (ii) between a trained standard DNN ψ^* and the constructed BNN $\boldsymbol{\theta}_{\psi^*}$. (b) We trained a standard DNN $\hat{\psi}$ and a BNN $\hat{\boldsymbol{\theta}}$ with the LeNet architecture on the CIFAR-10 dataset, and compared the strength of interactive concepts between the two networks when the two networks were trained to have the same training accuracy.

Theorem 5 (Proof in Appendix A.6)**.** *Given two random interactive concepts S and S', let us assume that $C_S(\boldsymbol{x}+\boldsymbol{\epsilon})$ is independent of $C_{S'}(\boldsymbol{x}+\boldsymbol{\epsilon})$. Let $\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]$ and $\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]$ denote the mean and the variance of $C_S(\boldsymbol{x}+\boldsymbol{\epsilon})$ w.r.t. $\boldsymbol{\epsilon}$, respectively. Then, the solution to Eq. (12) satisfies the following property:*

$$\forall\, S \in \Omega, \quad |U_S^*| \propto |\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]| \tag{13}$$

Theorem 5 proves that the strength of a network in encoding an interactive concept S, measured by $|U_S^*|$, is proportional to the relative stability of the interactive concept $|\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]|$ *w.r.t.* perturbations $\boldsymbol{\epsilon}$. This indicates that sensitive interactive concepts are more difficult to learn. The experimental verification of this theorem is shown in Appendix D.

Theorem 6 (Proof in Appendix A.7)**.** *Let $A_S^{min} = \min_S |U_S|$ and $A_S^{max} = \max_S |U_S|$ denote the lower bound and the upper bound of $|U_S|$ over all interactive concepts S. Then, we have*

$$A_S^{min} \cdot \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]} \le \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]} \le A_S^{max} \cdot \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]} \tag{14}$$

Theorem 6 proves that high-order (complex) interactive concepts have low relative stability *w.r.t.* perturbations $\boldsymbol{\epsilon}$. In fact, both Remark 2 and Figure 3 have told us that $|\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]|$ significantly decreases along with the order $s = |S|$ of the interactive concept S. Therefore, both the lower bound and the upper bound of $|\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]|$ in Eq. (14) decrease along with the order s significantly. In this way, we can approximately consider that the strength of encoding a concept $|U_S^*| \propto |\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]|$ also decreases along with the order of interactive concepts. In other words, we prove that high-order interactive concepts are more difficult to be learned under perturbations $\boldsymbol{\epsilon}$. Combining the conclusion in Section 2.2, we also prove that high-order interactive concepts are more difficult to be learned by the BNN.

3 Experiments

In this section, we experimentally verified that compared to standard DNNs, BNNs were less likely to encode high-order (complex) interactive concepts. Specifically, we constructed three pairs of baseline networks for comparison.

(1) Given a trained BNN $\boldsymbol{\theta}^*$, we constructed a standard DNN by setting its weights to the mean value of the weight distribution of the BNN. The standard DNN was denoted by $\psi_{\boldsymbol{\theta}^*}$. Then, we compared the strength of all high-order interactive concepts between the BNN $\boldsymbol{\theta}^*$ and the standard DNN $\psi_{\boldsymbol{\theta}^*}$ without weight/feature uncertainty.

(2) Similarly, given a trained standard DNN ψ^*, we constructed a BNN $\boldsymbol{\theta}_{\psi^*}$ by setting the mean value of its weight distribution to the weights of the standard DNN. We set all weight dimensions in the l-th layer of the BNN to share the same variance σ_l^2, where σ_l^2 was computed as the average of variances of all weight dimensions in the l-th layer of the previous BNN $\boldsymbol{\theta}^*$. Then, we compared the strength of high-order interactive concepts between the standard DNN ψ^* and the BNN $\boldsymbol{\theta}_{\psi^*}$.

(3) We trained a standard DNN and a BNN with the same architecture. Then, we compared the strength of high-order interactive concepts between each pair of standard DNN $\hat{\psi}$ and the BNN $\hat{\boldsymbol{\theta}}$ when these two networks were trained to have the same training accuracy. We used the training accuracy to align the learning progress of the two networks for fair comparison.

Specifically, the average strength of the s-order interactive concepts was measured as $I^{(s)}_{\text{strength}} = \mathbb{E}_{\boldsymbol{x}}[\mathbb{E}_{S\subseteq N,|S|=s}[|I(S|\boldsymbol{x})|]]$. To compute the causal effect $I(S|\boldsymbol{x})$, we set $v(\boldsymbol{x}_S) = \log \frac{p(y=y^*|\boldsymbol{x}_S)}{1-p(y=y^*|\boldsymbol{x}_S)} \in (-\infty, \infty)$, which reflected the confidence of classifying the masked input sample $\boldsymbol{x}_S$ into the ground-truth category y^*. For standard DNNs, $p(y = y^*|\boldsymbol{x}_S)$ referred to the classification probability of the ground-truth category on the masked sample $\boldsymbol{x}_S$. For BNNs, $p(y = y^*|\boldsymbol{x}_S)$ was computed according to Eq. (2), where we sampled ten neural networks from the weight distribution $q_{\boldsymbol{\theta}}(\boldsymbol{W})$ of the BNN, and computed the average classification probability over all these networks.

We followed experimental settings in the *experiments* paragraph in Section 2.2 to train the networks. Specifically, we trained standard DNNs and BNNs with the MLP architecture on the TV news dataset, the Census dataset, and the MNIST dataset. We trained standard DNNs and BNNs with the LeNet architecture on the CIFAR-10 dataset. Appendix E introduces how to efficiently compute $I(S|\boldsymbol{x})$ on images. Figure 4 shows that the strength of high-order interactive concepts of BNNs was much weaker than that of standard DNNs in all comparisons. This verified that BNNs were less likely to encode high-order (complex) interactive concepts than standard DNNs.

4 Related work

Analyzing the representation capacity of BNNs. Many studies investigated the representation capacity of BNNs from different perspectives. Gal & Smith (2018) and Carbone et al. (2020) proved that BNNs were robust to adversarial attacks. Kristiadi et al. (2020) proved that BNNs could mitigate the over-confidence problem in standard ReLU networks. Wenzel et al. (2020) considered that the poor performance of BNNs was due to the inappropriate prior distribution of weights in the BNN, and a series of studies (Wu et al., 2019; Krishnan et al., 2020; Fortuin et al., 2022) found that using carefully-designed prior distributions of weights could improve the performance of the BNN. Zhang et al. (2022) also showed that adding adversarial perturbations to weights during training could improve the performance of the BNN. Besides, Foong et al. (2020) proved that using either fully-factorized Gaussian distributions or dropout operations to approximate the posterior distribution of a BNN would lead to inaccurate uncertainty estimation of the network prediction. Unlike previous studies, we focus on the conceptual representation of BNNs, and theoretically prove that BNNs are less likely to encode complex interactive concepts than standard DNNs.

Using causality to explain neural networks. Causality was first proposed to investigate the causal structure between a set of variables (Pearl, 2009; Hoyer et al., 2008). Then, in recent years, many studies used causality as a new perspective to explain neural networks. Some studies proposed to improve existing attribution methods by considering manually defined causal relationship between input variables (Frye et al., 2020; Heskes et al., 2020; Wang et al., 2021). Similarly, Alvarez-Melis & Jaakkola (2017) proposed a causal framework to study the causal relationship between inputs and outputs of a sequence-to-sequence model, and Harradon et al. (2018) used a causal model to identify salient features in a CNN. Unlike previous studies, Ren et al. (2021a) first proved the faithfulness of using a sparse causal graph to explain the inference logic of a neural network. Thus, we further use causal patterns in the causal graph to investigate interactive concepts encoded by a BNN.

5 Conclusions

In this paper, we have investigated the bottleneck of the BNN in representing interactive concepts of different complexities. We have shown that the inference logic of a neural network can be faithfully represented using the causal structure between the network output and many interactive concepts. Then, we have theoretically proven that BNNs are less likely to encode complex interactive concepts than standard DNNs. This study has provided a new perspective of explaining the inferior performance of a BNN, and has clarified the shortcoming of the BNN in encoding complex concepts. Furthermore, we have conducted experiments to verify our proofs.

REPRODUCIBILITY STATEMENT

We have provided the proof for all theoretical results in Appendix A. We have also provided experimental details in the *experiments* paragraph in Section 2.2 and also Appendix E. The code will be released when the paper is accepted.

REFERENCES

David Alvarez-Melis and Tommi S Jaakkola. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *EMNLP*, 2017.

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

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Ginevra Carbone, Matthew Wicker, Luca Laurenti, Andrea Patane, Luca Bortolussi, and Guido Sanguinetti. Robustness of bayesian neural networks to gradient-based attacks. *Advances in Neural Information Processing Systems*, 33:15602–15613, 2020.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Andrew Foong, David Burt, Yingzhen Li, and Richard Turner. On the expressiveness of approximate inference in bayesian neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15897–15908. Curran Associates, Inc., 2020.

Vincent Fortuin, Adrià Garriga-Alonso, Sebastian W. Ober, Florian Wenzel, Gunnar Ratsch, Richard E Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited. In *International Conference on Learning Representations*, 2022.

Christopher Frye, Colin Rowat, and Ilya Feige. Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1229–1239. Curran Associates, Inc., 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Yarin Gal and Lewis Smith. Sufficient conditions for idealised models to have no adversarial examples: a theoretical and empirical study with bayesian neural networks, 2018.

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

Michael Harradon, Jeff Druce, and Brian Ruttenberg. Causal learning and explanation of deep neural networks via autoencoded activations. *arXiv preprint arXiv:1802.00541*, 2018.

John C. Harsanyi. A simplified bargaining model for the n-person cooperative game. *International Economic Review*, 4(2):194–220, 1963. ISSN 00206598, 14682354.

Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 4778–4789. Curran Associates, Inc., 2020.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, Bernhard Schölkopf, et al. Nonlinear causal discovery with additive noise models. In *NIPS*, 2008.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. URL http://arxiv.org/abs/1412.6980.

Ranganath Krishnan, Mahesh Subedar, and Omesh Tickoo. Specifying weight priors in bayesian deep neural networks with empirical bayes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4477–4484, 2020.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International conference on machine learning*, pp. 5436–5446. PMLR, 2020.

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

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

Benjamin J. Lengerich, Eric Xing, and Rich Caruana. Dropout as a regularizer of interaction effects. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 7550–7564. PMLR, 28–30 Mar 2022.

Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1754–1763, 2018.

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

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

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

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

Jie Ren, Die Zhang, Yisen Wang, Lu Chen, Zhanpeng Zhou, Yiting Chen, Xu Cheng, Xin Wang, Meng Zhou, Jie Shi, and Quanshi Zhang. Towards a unified game-theoretic view of adversarial perturbations and robustness. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 3797–3810. Curran Associates, Inc., 2021b.

Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1161–1170, 2019.

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

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

Jiaxuan Wang, Jenna Wiens, and Scott Lundberg. Shapley flow: A graph-based approach to interpreting model predictions. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 721–729. PMLR, 13–15 Apr 2021.

Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? In *International conference on machine learning*, 2020.

R. Willink. Normal moments and hermite polynomials. *Statistics & Probability Letters*, 73(3): 271–275, 2005. ISSN 0167-7152. doi: https://doi.org/10.1016/j.spl.2005.03.015.

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E. Turner, Jose Miguel Hernandez-Lobato, and Alexander L. Gaunt. Deterministic variational inference for robust bayesian neural networks. In *International Conference on Learning Representations*, 2019.

Jiaru Zhang, Yang Hua, Tao Song, Hao Wang, Zhengui Xue, Ruhui Ma, and Haibing Guan. Improving bayesian neural networks by adversarial sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

A PROOF OF THEOREMS

A.1 PROOF OF THEOREM 1 IN THE MAIN PAPER

Theorem 1. *Given an input sample $\boldsymbol{x}$ with n input variables, let $\boldsymbol{x}_S$ denote a masked input sample, where variables in $N \setminus S$ are masked and variables in S keep unchanged. It is proven that for each neural network v, there exists a specific causal graph (parameterized by $\{U_{S'}|S' \in \Omega\}$), such that for any arbitrarily masked input sample $\boldsymbol{x}_S$, the output $v(\boldsymbol{x}_S)$ of the neural network can be well mimicked by the output $Y(\boldsymbol{x}_S)$ of the causal graph, i.e.,*

$$\exists\, \Omega \subseteq 2^N, \exists \{U_{S'}|S' \in \Omega\},\ s.t.,\quad \forall S \subseteq N,\ v(\boldsymbol{x}_S) = Y(\boldsymbol{x}_S)$$

where $Y(\boldsymbol{x}_S)$ denotes the output of causal graph (see Eq. (5)) on the masked sample $\boldsymbol{x}_S$ by setting $A_i = \mathbb{1}(i \in S)$. In particular, a special solution of the causal effects U_S (the causal graph) satisfying the above equation is the Harsanyi dividend (Harsanyi, 1963), $U_S = \sum_{S' \subseteq S} (-1)^{|S|-|S'|} v(\boldsymbol{x}_{S'})$.

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

Proof. According to the SCM in Eq. (5), we have $Y(\boldsymbol{x}_S) = \sum_{S' \in \Omega} U_{S'} \cdot C_{S'}(\boldsymbol{x}_S) = \sum_{S' \subseteq S} U_{S'}$. Hence, we only need to prove that $\forall\, S \subseteq N, v(\boldsymbol{x}_S) = \sum_{S' \subseteq S} U_{S'}$. Specifically,

$$\begin{aligned}
\sum_{S' \subseteq S} U_{S'} &= \sum_{S' \subseteq S} \sum_{L \subseteq S'} (-1)^{|S'|-|L|} v(\boldsymbol{x}_L) \\
&= \sum_{L \subseteq S} \sum_{S' \subseteq S: S' \supseteq L} (-1)^{|S'|-|L|} v(\boldsymbol{x}_L) \\
&= \sum_{L \subseteq S} \sum_{s'=1}^{|S'|} \sum_{S' \subseteq S: S' \supseteq L, |S'|=s'} (-1)^{s'-|L|} v(\boldsymbol{x}_L) \\
&= \sum_{L \subseteq S} v(\boldsymbol{x}_L) \sum_{m=0}^{|S|-|L|} \tbinom{|S|-|L|}{m} (-1)^m = v(\boldsymbol{x}_S)
\end{aligned} \tag{15}$$

□

A.2 PROOF OF LEMMA 1 IN THE MAIN PAPER

Lemma 1 *Originally, the causal effect in Eq. (5) is defined as a binary variable in the causal graph, $I(S) \in \{U_S, 0\}$. Given a continuous network function $v(\boldsymbol{x})$, we can use the following Taylor expansion to decompose the network output, which extends the causal effect $I(S)$ to a continuous function. This continuous function well fits the binary states of $I(S)$ on all the 2^n masked samples $\boldsymbol{x}'$ with different masking states, i.e., $\boldsymbol{x}' \in \{\boldsymbol{x}_T|\forall T \subseteq N\}$.*

$$I(S|\boldsymbol{x}') = \sum\nolimits_{\boldsymbol{\pi} \in Q_S} U_{S,\boldsymbol{\pi}} \cdot J(S, \boldsymbol{\pi}|\boldsymbol{x}'), \tag{16}$$

where $J(S, \boldsymbol{\pi}|\boldsymbol{x}') = \prod_{i \in S} \left(\text{sign}(x'_i - r_i) \cdot \frac{x'_i - r_i}{\tau}\right)^{\pi_i}$ denotes a Taylor expansion term of the degree $\boldsymbol{\pi}$. Here, $\boldsymbol{\pi} \in Q_S = \{[\pi_1, \ldots, \pi_n] | \forall i \in S, \pi_i \in \mathbb{N}^+; \forall i \notin S, \pi_i = 0\}$. In addition, $U_{S,\boldsymbol{\pi}} = \frac{\tau^m}{\prod_{i=1}^n \pi_i!} \frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1} \cdots \partial x_n^{\pi_n}} \cdot \prod_{i \in S} [\text{sign}(x'_i - r_i)]^{\pi_i}$, s.t. $m = \sum_{i=1}^n \pi_i$. $v(\boldsymbol{x}_\emptyset)$ indicates the network output when we mask all input variables to reference values r_i. Moreover, $C_S(\boldsymbol{x}') = I(S|\boldsymbol{x}')/U_S$.

Proof. Let us denote the continuous function on the right of Eq.(16) by $\tilde{I}(S|\boldsymbol{x}')$, *i.e.*,

$$\tilde{I}(S|\boldsymbol{x}') = \sum\nolimits_{\pi \in Q_S} U_{S,\pi} J(S, \pi|\boldsymbol{x}') \tag{17}$$

We need to prove that the continuous function $\tilde{I}(S|\boldsymbol{x}')$ well fits the binary variable $I(S|\boldsymbol{x}')$ on all the 2^n masked samples $\boldsymbol{x}'$. *i.e.*, $\tilde{I}(S|\boldsymbol{x}') = I(S|\boldsymbol{x}') \in \{U_S, 0\}, \forall \boldsymbol{x}' \in \{\boldsymbol{x}_T|\forall T \subseteq N\}$.

We prove this theorem by two steps. (i) In the first step, we prove that $\tilde{U}_S \overset{\text{def}}{=} \tilde{I}(S|\boldsymbol{x})$ on the given sample $\boldsymbol{x}$ also satisfies the faithfulness requirement in Theorem 1. Furthermore, Grabisch & Roubens (1999) and Ren et al. (2021b) has proven that the Harsanyi dividend $U_S = I(S|\boldsymbol{x})$ is the unique metric to satisfy Theorem 1. Therefore, we can obtain that $\tilde{U}_S = U_S$. (ii) In the second step, we prove that on all the 2^n masked samples $\boldsymbol{x}' \in \{\boldsymbol{x}_T|\forall T \subseteq N\}$, $\tilde{I}(S|\boldsymbol{x}') = I(S|\boldsymbol{x}') \in \{U_S, 0\}$.

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

$$\forall T \subseteq N, \quad v(\boldsymbol{x}_T) = \sum_{\pi_1=0}^{\infty}\sum_{\pi_2=0}^{\infty}\cdots\sum_{\pi_n=0}^{\infty} \frac{1}{\prod_{i=1}^n \pi_i!} \frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1}\cdots\partial x_n^{\pi_n}} \cdot \prod_{i=1}^n [(\boldsymbol{x}_T)_i - r_i]^{\pi_i} \tag{18}$$

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

According to the definition of the masked sample $\boldsymbol{x}_T$, we have that $\forall i \in T$, $(\boldsymbol{x}_T)_i = x_i$ and $\forall i \notin T$, $(\boldsymbol{x}_T)_i = r_i$. Hence, $\forall i \notin T, [(\boldsymbol{x}_T)_i - r_i]^{\pi_i} = 0$. Then, among all Taylor expansion terms, only terms corresponding to degrees $\boldsymbol{\pi}$ in the set $P = \{[\pi_1,\ldots,\pi_n]|\forall i \in T, \pi_i \in \mathbb{N}; \forall i \notin T, \pi_i = 0\}$ may not be zero. Therefore, Eq. (18) can be re-written as follows.

$$\forall T \subseteq N, \quad v(\boldsymbol{x}_T) = \sum_{\pi\in P} \frac{1}{\prod_{i=1}^n \pi_i!} \frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1}\cdots\partial x_n^{\pi_n}} \cdot \prod_{i\in T} (x_i - r_i)^{\pi_i} \tag{19}$$

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

$$\begin{aligned}\forall T \subseteq N, \quad v(\boldsymbol{x}_T) &= \sum_{S\subseteq T}\sum_{\pi\in Q_S} \frac{1}{\prod_{i=1}^n \pi_i!} \frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1}\cdots\partial x_n^{\pi_n}} \cdot \prod_{i\in S} (x_i - r_i)^{\pi_i} \\ &= \sum_{S\subseteq T}\sum_{\pi\in Q_S} \underbrace{\frac{\tau^m}{\prod_{i=1}^n \pi_i!} \frac{\partial^m v(\boldsymbol{x}_\emptyset)}{\partial x_1^{\pi_1}\cdots\partial x_n^{\pi_n}} \prod_{i\in S} (\delta_i)^{\pi}}_{\text{termed } U_{S,\pi}} \cdot \underbrace{\prod_{i\in S'} (\delta_i \frac{x_i - r_i}{\tau})^{\pi_i}}_{\text{termed } J(S,\pi|\boldsymbol{x})},\end{aligned} \tag{20}$$

where $\tau \in \mathbb{R}$ is a pre-defined constant and $\delta_i = \text{sign}(x_i - r_i)$. Then, Eq. (20) can be re-written as,

$$\forall T \subseteq N,\ v(\boldsymbol{x}_T) = \sum\nolimits_{S\subseteq T} \tilde{U}_S \tag{21}$$

i.e., $\{\tilde{U}_S|S \subseteq N\}$ also satisfies the faithfulness requirement in Theorem 1. Moreover, Grabisch & Roubens (1999) and Ren et al. (2021b) has proven that the Harsanyi dividend $U_S = I(S|\boldsymbol{x})$ is the unique metric to satisfy Theorem 1. Therefore, we can obtain that $\tilde{U}_S = U_S$.

Proof of Step 2. We aim to prove that for a specific interactive concept S, $I(S|\boldsymbol{x}') = \tilde{I}(S|\boldsymbol{x}')$ holds for all the 2^n masked samples $\boldsymbol{x}' \in \{\boldsymbol{x}_T|\forall T \subseteq N\}$. Specifically, for the interactive concept S, let us divided all masked samples $\boldsymbol{x}_T$ into two groups, (i) $\{\boldsymbol{x}_T|S \subseteq T\}$ and (ii) $\{\boldsymbol{x}_T|S \not\subseteq T\}$. According to the SCM in Eq. (5), we can obtain that

$$I(S|\boldsymbol{x}_T) = U_S \cdot \mathbb{1}(S \subseteq T) = \begin{cases} U_S, & \text{if } S \subseteq T; \\ 0, & \text{if } S \not\subseteq T. \end{cases} \tag{22}$$

According to the definition of $\tilde{I}(S|\boldsymbol{x}')$, it is easy to obtain that when $S \subseteq T$, $\tilde{I}(S|\boldsymbol{x}_T) = \tilde{U}_S = U_S$; otherwise, $\tilde{I}(S|\boldsymbol{x}_T) = 0$. Then, Lemma 1 holds. □

A.3 Proof of Theorem 2 in the main paper

Theorem 2. *If we only consider the approximation based on the lowest degree $\hat{\boldsymbol{\pi}}$, then the mean and variance of $I(S|\boldsymbol{x}+\boldsymbol{\epsilon})$ over different perturbations $\boldsymbol{\epsilon}$ are given as*

$$\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})] = U_{S,\hat{\boldsymbol{\pi}}}, \quad \text{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})] = U_{S,\hat{\boldsymbol{\pi}}}^2((1+(\sigma/\tau)^2)^{|S|} - 1) \tag{23}$$

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

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

$$\begin{aligned} I(S|\boldsymbol{x}+\boldsymbol{\epsilon}) &\approx U_{S,\hat{\boldsymbol{\pi}}} \cdot J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon}) \\ J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon}) &= \prod_{i \in S} \operatorname{sign}(x_i + \epsilon_i - r_i) \cdot \frac{x_i + \epsilon_i - r_i}{\tau} \\ &= \prod_{i \in S} \left(\operatorname{sign}(x_i + \epsilon_i - r_i) \cdot \frac{x_i - r_i}{\tau} + \operatorname{sign}(x_i + \epsilon_i - r_i) \cdot \frac{\epsilon_i}{\tau} \right) \end{aligned} \tag{24}$$

According to the setting of the reference value in Section 2.3, we have $\forall i \in S, x_i - r_i \in \{-\tau, \tau\}$. In Section 2.3, we have assumed that the variance of the perturbation $\boldsymbol{\epsilon}$ is small, so that we can ignore the extremely low probability that the perturbation is large such that $|\epsilon_i| \geq \tau$. In this way, we have $\operatorname{sign}(x_i + \epsilon_i - r_i) = \operatorname{sign}(x_i - r_i)$, and we can obtain

$$\begin{aligned} J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon}) &= \prod_{i \in S} \left(\operatorname{sign}(x_i - r_i) \cdot \frac{x_i - r_i}{\tau} + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau} \right) \\ &= \prod_{i \in S} \left(1 + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau} \right) \end{aligned} \tag{25}$$

$$\begin{aligned} \Rightarrow \mathbb{E}_{\boldsymbol{\epsilon}}[J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \mathbb{E}_{\boldsymbol{\epsilon}} \left[\prod_{i \in S} \left(1 + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau} \right) \right] \\ \operatorname{Var}_{\boldsymbol{\epsilon}}[J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \operatorname{Var}_{\boldsymbol{\epsilon}} \left[\prod_{i \in S} \left(1 + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau} \right) \right] \end{aligned} \tag{26}$$

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

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

According to the above proposition, we have

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\epsilon}}[J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \prod_{i \in S} 1 = 1 \\ \operatorname{Var}_{\boldsymbol{\epsilon}}[J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \prod_{i \in S} \left(1^2 + (\sigma/\tau)^2 \right) - \prod_{i \in S} 1^2 \\ &= \left(1 + (\sigma/\tau)^2 \right)^{|S|} - 1 \end{aligned} \tag{27}$$

Therefore,

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})] &\approx \mathbb{E}_{\boldsymbol{\epsilon}}[U_{S,\hat{\boldsymbol{\pi}}} \cdot J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})] = U_{S,\hat{\boldsymbol{\pi}}} \\ \operatorname{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})] &\approx \operatorname{Var}_{\boldsymbol{\epsilon}}[U_{S,\hat{\boldsymbol{\pi}}} \cdot J(S, \hat{\boldsymbol{\pi}}|\boldsymbol{x}+\boldsymbol{\epsilon})] = U_{S,\hat{\boldsymbol{\pi}}}^2 \left(\left(1 + (\sigma/\tau)^2 \right)^{|S|} - 1 \right) \end{aligned} \tag{28}$$

□

A.4 Proof of Theorem 3 in the main paper

Theorem 3. *Let $\boldsymbol{\pi} \in Q_S = \{[\pi_1, \ldots, \pi_n] | \forall i \in S, \pi_i \in \mathbb{N}^+; \forall i \notin S, \pi_i = 0\}$ denote an arbitrary degree. Then, the mean and the variance of $J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})$ over $\boldsymbol{\epsilon}$ are given as*

$$\mathbb{E}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] = \mathbb{E}_{\boldsymbol{\epsilon}}[\prod_{i \in S} (1 + \frac{\epsilon_i}{\tau})^{\pi_i}], \quad \operatorname{Var}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] = \operatorname{Var}_{\boldsymbol{\epsilon}}[\prod_{i \in S} (1 + \frac{\epsilon_i}{\tau})^{\pi_i}] \tag{29}$$

Proof. According to Lemma 1, given an arbitrary input sample $\boldsymbol{x}'$, we have

$$J(S, \boldsymbol{\pi}|\boldsymbol{x}') = \prod\nolimits_{i\in S} \left(\operatorname{sign}(x_i' - r_i) \cdot \frac{x_i' - r_i}{\tau}\right)^{\pi_i} \tag{30}$$

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

$$\begin{aligned} J(S, \boldsymbol{\pi}|\boldsymbol{x} + \boldsymbol{\epsilon}) &= \prod_{i\in S} \left(\operatorname{sign}(x_i + \epsilon_i - r_i) \cdot \frac{x_i + \epsilon_i - r_i}{\tau}\right)^{\pi_i} \\ &= \prod_{i\in S} \left(\operatorname{sign}(x_i + \epsilon_i - r_i) \cdot \frac{x_i - r_i}{\tau} + \operatorname{sign}(x_i + \epsilon_i - r_i) \cdot \frac{\epsilon_i}{\tau}\right)^{\pi_i} \end{aligned} \tag{31}$$

According to the setting of the reference value in Section 2.3, $\forall i \in S, x_i - r_i \in \{-\tau, \tau\}$. In Section 2.3, we have assumed that the variance of the perturbation $\boldsymbol{\epsilon}$ is small, so that we can ignore the extremely low probability that the perturbation is large such that $|\epsilon_i| \geq \tau$. In this way, $\operatorname{sign}(x_i + \epsilon_i - r_i) = \operatorname{sign}(x_i - r_i)$, and we can obtain

$$\begin{aligned} J(S, \boldsymbol{\pi}|\boldsymbol{x} + \boldsymbol{\epsilon}) &= \prod_{i\in S} \left(\operatorname{sign}(x_i - r_i) \cdot \frac{x_i - r_i}{\tau} + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau}\right)^{\pi_i} \\ &= \prod_{i\in S} \left(1 + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau}\right)^{\pi_i} \end{aligned} \tag{32}$$

$$\begin{aligned} \Rightarrow \mathbb{E}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x} + \boldsymbol{\epsilon})] &= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S} \left(1 + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau}\right)^{\pi_i}\right] \\ \operatorname{Var}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x} + \boldsymbol{\epsilon})] &= \operatorname{Var}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S} \left(1 + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau}\right)^{\pi_i}\right] \end{aligned} \tag{33}$$

Since $\forall i \in S$, ϵ_i is independent of each other, according to Proposition 1 and Eq. (33), we have

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x} + \boldsymbol{\epsilon})] &= \prod_{i\in S} \mathbb{E}_{\epsilon_i}\left[\left(1 + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau}\right)^{\pi_i}\right] \\ \operatorname{Var}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x} + \boldsymbol{\epsilon})] &= \prod_{i\in S} \mathbb{E}_{\epsilon_i}\left[\left(1 + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau}\right)^{2\pi_i}\right] - \prod_{i\in S} \left(\mathbb{E}_{\epsilon_i}\left[\left(1 + \operatorname{sign}(x_i - r_i) \cdot \frac{\epsilon_i}{\tau}\right)^{\pi_i}\right]\right)^2 \end{aligned} \tag{34}$$

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

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

□

A.5 PROOF OF THEOREM 4 IN THE MAIN PAPER

Theorem 4. *Let S' be an interactive concept extended from the concept S, i.e., $S \subsetneq S'$. Let us consider expansion terms $J(S, \boldsymbol{\pi})$ and $J(S', \boldsymbol{\pi}')$, where the term $J(S', \boldsymbol{\pi}')$ can be considered being extended from the term $J(S, \boldsymbol{\pi})$ with $\boldsymbol{\pi} \prec \boldsymbol{\pi}'$. I.e., (1) $\forall i \in S', \pi'_i \in \mathbb{N}^+$; otherwise, $\pi'_i = 0$. (2) Given $\boldsymbol{\pi}'$, $\forall j \in S, \pi_j = \pi'_j$; otherwise, $\pi_j = 0$. Then, we have*

$$\begin{aligned} \frac{\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S', \boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]}{\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]} &> \prod\nolimits_{i \in S' \setminus S} \mathbb{E}^2_{\epsilon_i}[(1+\tfrac{\epsilon_i}{\tau})^{\pi'_i}], \\ \frac{\mathbb{E}_{\boldsymbol{\epsilon}}[J(S', \boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S', \boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]}{\mathbb{E}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]} &< \frac{1}{\prod_{i \in S' \setminus S} \mathbb{E}_{\epsilon_i}[(1+\frac{\epsilon_i}{\tau})^{\pi'_i}]}, \end{aligned} \tag{35}$$

and it is easy to obtain $\mathbb{E}_{\epsilon_i}[(1+\frac{\epsilon_i}{\tau})^{\pi'_i}] \geq 1$.

Proof. According to Theorem 3, we have

$$\begin{aligned} \mathrm{Var}_{\boldsymbol{\epsilon}}[J(S', \boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \mathrm{Var}_{\boldsymbol{\epsilon}}\left[\prod_{i \in S'} \left(1+\frac{\epsilon_i}{\tau}\right)^{\pi'_i}\right] \\ &= \mathrm{Var}_{\boldsymbol{\epsilon}}\left[\prod_{i \in S} \left(1+\frac{\epsilon_i}{\tau}\right)^{\pi'_i} \prod_{i \in S' \setminus S} \left(1+\frac{\epsilon_i}{\tau}\right)^{\pi'_i}\right] \quad //S \subsetneq S' \\ &= \mathrm{Var}_{\boldsymbol{\epsilon}}\left[\underbrace{\prod_{i \in S} \left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i}}_{A} \underbrace{\prod_{i \in S' \setminus S} \left(1+\frac{\epsilon_i}{\tau}\right)^{\pi'_i}}_{B}\right] \quad //\forall i \in S, \pi'_i = \pi_i \\ &= \mathrm{Var}_{\boldsymbol{\epsilon}}[AB] \\ &= (\mathbb{E}^2_{\boldsymbol{\epsilon}}[A] + \mathrm{Var}_{\boldsymbol{\epsilon}}[A])(\mathbb{E}^2_{\boldsymbol{\epsilon}}[B] + \mathrm{Var}_{\boldsymbol{\epsilon}}[B]) - \mathbb{E}^2_{\boldsymbol{\epsilon}}[A]\mathbb{E}^2_{\boldsymbol{\epsilon}}[B] \\ &\quad //A \text{ and } B \text{ are independent; Proposition 1} \\ &= \mathbb{E}^2_{\boldsymbol{\epsilon}}[A]\mathrm{Var}_{\boldsymbol{\epsilon}}[B] + \mathbb{E}^2_{\boldsymbol{\epsilon}}[B]\mathrm{Var}_{\boldsymbol{\epsilon}}[A] + \mathrm{Var}_{\boldsymbol{\epsilon}}[A]\mathrm{Var}_{\boldsymbol{\epsilon}}[B] \\ &> \mathbb{E}^2_{\boldsymbol{\epsilon}}[B]\mathrm{Var}_{\boldsymbol{\epsilon}}[A] + \mathrm{Var}_{\boldsymbol{\epsilon}}[A]\mathrm{Var}_{\boldsymbol{\epsilon}}[B] \end{aligned} \tag{36}$$

Therefore, we can prove the first equality as follows.

$$\begin{aligned} \frac{\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S', \boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]}{\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S, \boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]} &= \frac{\mathrm{Var}_{\boldsymbol{\epsilon}}[AB]}{\mathrm{Var}_{\boldsymbol{\epsilon}}[A]} \\ &> \mathbb{E}^2_{\boldsymbol{\epsilon}}[B] + \mathrm{Var}_{\boldsymbol{\epsilon}}[B] \\ &> \mathbb{E}^2_{\boldsymbol{\epsilon}}[B] \\ &= \mathbb{E}^2_{\boldsymbol{\epsilon}}\left[\prod_{i \in S' \setminus S} \left(1+\frac{\epsilon_i}{\tau}\right)^{\pi'_i}\right] \\ &= \prod_{i \in S' \setminus S} \mathbb{E}^2_{\epsilon_i}\left[\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi'_i}\right] \\ &\quad //\epsilon_i \text{ is independent of each other; Proposition 1} \end{aligned} \tag{37}$$

Furthermore, we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\epsilon}}[J(S',\boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})] &= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S'}\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i'}\right] \\ &= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S}\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i'}\prod_{i\in S'\setminus S}\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i'}\right] \quad //S\subsetneq S' \\ &= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S}\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i}\prod_{i\in S'\setminus S}\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i'}\right] \quad //\forall i\in S,\pi_i'=\pi_i \\ &= \mathbb{E}_{\boldsymbol{\epsilon}}[AB]\end{aligned} \tag{38}$$

and also

$$\mathbb{E}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})] = \mathbb{E}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S}\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i}\right] = \mathbb{E}_{\boldsymbol{\epsilon}}[A]. \tag{39}$$

Therefore, we have

$$\frac{\mathbb{E}_{\boldsymbol{\epsilon}}[J(S',\boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]}{\mathbb{E}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]} = \frac{\mathbb{E}_{\boldsymbol{\epsilon}}[AB]}{\mathbb{E}_{\boldsymbol{\epsilon}}[A]} = \mathbb{E}_{\boldsymbol{\epsilon}}[B]. \tag{40}$$

Then, we can prove the second inequality as follows.

$$\begin{aligned}&\frac{\mathbb{E}_{\boldsymbol{\epsilon}}[J(S',\boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S',\boldsymbol{\pi}'|\boldsymbol{x}+\boldsymbol{\epsilon})]}{\mathbb{E}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[J(S,\boldsymbol{\pi}|\boldsymbol{x}+\boldsymbol{\epsilon})]} \\ =&\frac{\mathbb{E}_{\boldsymbol{\epsilon}}[B]}{\mathrm{Var}_{\boldsymbol{\epsilon}}[AB]/\mathrm{Var}_{\boldsymbol{\epsilon}}[A]} \\ <&\frac{\mathbb{E}_{\boldsymbol{\epsilon}}[B]}{\mathbb{E}^2_{\boldsymbol{\epsilon}}[B]} \\ =&\frac{1}{\mathbb{E}_{\boldsymbol{\epsilon}}[B]} \\ =&\frac{1}{\mathbb{E}_{\boldsymbol{\epsilon}}\left[\prod_{i\in S'\setminus S}\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i'}\right]} \\ =&\frac{1}{\prod_{i\in S'\setminus S}\mathbb{E}_{\epsilon_i}\left[\left(1+\frac{\epsilon_i}{\tau}\right)^{\pi_i'}\right]}\end{aligned} \tag{41}$$

Moreover, we can prove that $\mathbb{E}_{\epsilon_i}[(1+\frac{\epsilon_i}{\tau})^k] \geq 1, \forall k \in \mathbb{N}^+$, *i.e.*, $\mathbb{E}[X^k] \geq 1$, where $X \sim \mathcal{N}(1,(\sigma/\tau)^2)$.

For a random variable following a Gaussian distribution $\tilde{X}\sim\mathcal{N}(\tilde{\mu},\tilde{\sigma}^2)$, Willink (2005) proved the following property:

$$\mathbb{E}\left[\tilde{X}^{k+1}\right] = \tilde{\mu}\mathbb{E}\left[\tilde{X}^k\right] + k\tilde{\sigma}^2\mathbb{E}\left[\tilde{X}^{k-1}\right] \tag{42}$$

Now let us consider $X\sim\mathcal{N}(1,(\sigma/\tau)^2)$. We have $\mathbb{E}\left[X^{k+1}\right] = \mathbb{E}\left[X^k\right] + k(\sigma/\tau)^2\mathbb{E}\left[X^{k-1}\right]$. By induction, it is easy to prove that $\mathbb{E}[X^k]\geq\mathbb{E}[X]=1$. □

A.6 Proof of Theorem 5

Theorem 5. *Let us assume that $C_S(\boldsymbol{x}+\boldsymbol{\epsilon})$ is independent of $C_{S'}(\boldsymbol{x}+\boldsymbol{\epsilon})$ for each pair of (S,S'). Let $\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]$ and $\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]$ denote the mean and the variance of $C_S(\boldsymbol{x}+\boldsymbol{\epsilon})$ w.r.t. $\boldsymbol{\epsilon}$, respectively. Then, the solution to Eq. (12) satisfies the following property:*

$$\forall\, S\in\Omega,\quad |U_S^*| \propto |\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]| \tag{43}$$

Proof. Let $p = |\Omega|$. Let $\boldsymbol{C}(\boldsymbol{x}+\boldsymbol{\epsilon}) = [C_{S_1}(\boldsymbol{x}+\boldsymbol{\epsilon}), \cdots, C_{S_p}(\boldsymbol{x}+\boldsymbol{\epsilon})]^\top$ denote the vector of all $C_S(\boldsymbol{x}+\boldsymbol{\epsilon}), S \in \Omega$, and let $\boldsymbol{U} = [U_{S_1}, \cdots, U_{S_p}]^\top$ denote the vector of all coefficients $U_S, S \in \Omega$. To further simplify the notation, we simply use $\boldsymbol{C}$ to denote the random vector $\boldsymbol{C}(\boldsymbol{x}+\boldsymbol{\epsilon})$. Besides, since we assume that each dimension of the vector $\boldsymbol{C}(\boldsymbol{x}+\boldsymbol{\epsilon})$ is independent of each other, we can use $\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}] = [\alpha_1, \cdots, \alpha_p]^\top \in \mathbb{R}^p$ and $\mathrm{Var}_{\boldsymbol{\epsilon}}[\boldsymbol{C}] = \mathrm{diag}(\beta_1^2, \cdots, \beta_p^2) \in \mathbb{R}^{p\times p}$ to denote the mean vector and covariance matrix of the random vector $\boldsymbol{C}(\boldsymbol{x}+\boldsymbol{\epsilon})$, respectively. We prove this theorem in three steps.

Step 1. We first prove that the optimal solution to Eq. (12) is given by

$$\forall 1 \le i \le p, \quad U_{S_i}^* = \frac{1}{\det \boldsymbol{M}} \det(\boldsymbol{M}_1, \cdots, \boldsymbol{M}_{i-1}, \boldsymbol{\rho}, \boldsymbol{M}_{i+1}, \cdots, \boldsymbol{M}_p) \tag{44}$$

where $\boldsymbol{M} = \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]^\top + \mathrm{Var}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]$, $\boldsymbol{\rho} = y^*\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]$, and $\boldsymbol{M}_j$ denotes the j-th column of the matrix $\boldsymbol{M}$.

We can rewrite the objective function in Eq. (12) as

$$\min_{\boldsymbol{U}} \mathbb{E}_{\boldsymbol{\epsilon}}[(y^* - \boldsymbol{U}^\top \boldsymbol{C}(\boldsymbol{x}+\boldsymbol{\epsilon}))^2] \tag{45}$$

To minimize the loss $L = \mathbb{E}_{\boldsymbol{\epsilon}}[(y^* - \boldsymbol{U}^\top \boldsymbol{C})^2]$, we set the gradient of the loss *w.r.t* $\boldsymbol{U}$ to zero, *i.e.*,

$$\begin{aligned}
\nabla_{\boldsymbol{U}} L &= \mathbb{E}_{\boldsymbol{\epsilon}}[2\boldsymbol{C}(\boldsymbol{U}^\top \boldsymbol{C} - y^*)] \\
&= 2\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}\boldsymbol{C}^\top \boldsymbol{U} - y^*\boldsymbol{C}] \\
&= 2\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}\boldsymbol{C}^\top]\boldsymbol{U} - 2y^*\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}] \\
&= 2(\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]^\top + \mathrm{Var}_{\boldsymbol{\epsilon}}[\boldsymbol{C}])\boldsymbol{U} - 2y^*\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}] = 0
\end{aligned} \tag{46}$$

$$\Rightarrow (\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]^\top + \mathrm{Var}_{\boldsymbol{\epsilon}}[\boldsymbol{C}])\boldsymbol{U} = y^*\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}] \tag{47}$$

Let $\boldsymbol{M} = \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]^\top + \mathrm{Var}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]$, and $\boldsymbol{\rho} = y^*\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]$. By Cramer's rule, we can obtain the solution to Eq. (47):

$$\forall 1 \le i \le p, \quad U_{S_i}^* = \frac{1}{\det \boldsymbol{M}} \det(\boldsymbol{M}_1, \cdots, \boldsymbol{M}_{i-1}, \boldsymbol{\rho}, \boldsymbol{M}_{i+1}, \cdots, \boldsymbol{M}_p)$$

where $\boldsymbol{M}_j$ denotes the j-th column of the matrix $\boldsymbol{M}$.

Step 2. We prove that for the optimal solution $\boldsymbol{U}^*$, we have

$$\forall 1 \le i, j \le p, \quad \frac{|U_{S_i}^*|}{|U_{S_j}^*|} = \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[C_{S_i}(\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[C_{S_i}(\boldsymbol{x}+\boldsymbol{\epsilon})]|}{|\mathbb{E}_{\boldsymbol{\epsilon}}[C_{S_j}(\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[C_{S_j}(\boldsymbol{x}+\boldsymbol{\epsilon})]|} \tag{48}$$

Since $\boldsymbol{M} = \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]^\top + \mathrm{Var}_{\boldsymbol{\epsilon}}[\boldsymbol{C}]$, we can obtain the j-th column of $\boldsymbol{M}$ as

$$\boldsymbol{M}_j = \alpha_j \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}] + \boldsymbol{V}_j \tag{49}$$

where $\mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}] = [\alpha_1, \cdots, \alpha_p]^\top$, and $\boldsymbol{V}_j = [0, \cdots, \beta_j^2, \cdots, 0]^\top$.

According to the conclusion in Step 1, we have

$$|U_{S_i}^*| = |\frac{1}{\det \boldsymbol{M}}| \cdot |\det(\boldsymbol{M}_1, \cdots, \boldsymbol{M}_{i-1}, \boldsymbol{\rho}, \boldsymbol{M}_{i+1}, \cdots, \boldsymbol{M}_{j-1}, \boldsymbol{M}_j, \boldsymbol{M}_{j+1}, \cdots, \boldsymbol{M}_p)| \tag{50}$$

$$|U_{S_j}^*| = |\frac{1}{\det \boldsymbol{M}}| \cdot |\det(\boldsymbol{M}_1, \cdots, \boldsymbol{M}_{i-1}, \boldsymbol{M}_i, \boldsymbol{M}_{i+1}, \cdots, \boldsymbol{M}_{j-1}, \boldsymbol{\rho}, \boldsymbol{M}_{j+1}, \cdots, \boldsymbol{M}_p)| \tag{51}$$

We know that exchanging the rows or columns of a matrix only changes the sign of the determinant of the matrix, but does not change the absolute value of the determinant. Therefore, we have

$$
\begin{aligned}
|U^*_{S_i}| &= |\frac{1}{\det \boldsymbol{M}}| \cdot |\det(\boldsymbol{M}_j, \boldsymbol{\rho}, \boldsymbol{M}_1, \cdots, \boldsymbol{M}_{i-1}, \boldsymbol{M}_{i+1}, \cdots, \boldsymbol{M}_{j-1}, \boldsymbol{M}_{j+1}, \cdots, \boldsymbol{M}_p)| \\
&= |\frac{1}{\det \boldsymbol{M}}| \cdot |\det(\boldsymbol{M}_j, \boldsymbol{\rho}, \boldsymbol{M}_{\text{others}})| \quad //\text{Let } \boldsymbol{M}_{\text{others}} \text{ denote the third to the last column} \\
&= |\frac{1}{\det \boldsymbol{M}}| \cdot |\det(\alpha_j \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}] + \boldsymbol{V}_j, y^* \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}], \boldsymbol{M}_{\text{others}})| \quad //\text{Eq. (49)} \\
&= |\frac{1}{\det \boldsymbol{M}}| \cdot |\underbrace{\det(\alpha_j \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}], y^* \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}], \boldsymbol{M}_{\text{others}})}_{=0} + \det(\boldsymbol{V}_j, y^* \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}], \boldsymbol{M}_{\text{others}})| \\
&\quad //\text{The determinant is 0 if two columns are linearly dependent} \\
&= |\frac{1}{\det \boldsymbol{M}}| \cdot |\det(\boldsymbol{V}_j, y^* \mathbb{E}_{\boldsymbol{\epsilon}}[\boldsymbol{C}], \boldsymbol{M}_{\text{others}})| \\
&= |\frac{1}{\det \boldsymbol{M}}| \cdot |\det \begin{bmatrix} 0 & y^*\alpha_1 & \alpha_1\alpha_1 + \beta_1^2 & \cdots & \alpha_1\alpha_p \\ \vdots & & & & \\ 0 & y^*\alpha_i & \alpha_i\alpha_1 & \cdots & \alpha_i\alpha_p \\ \vdots & & & & \\ \beta_j^2 & y^*\alpha_j & \alpha_j\alpha_1 & \cdots & \alpha_j\alpha_p \\ \vdots & & & & \\ 0 & y^*\alpha_p & \alpha_p\alpha_1 & \cdots & \alpha_p\alpha_p + \beta_p^2 \end{bmatrix}| \\
&= |\frac{1}{\det \boldsymbol{M}}| \cdot |det \begin{bmatrix} \beta_j^2 & y^*\alpha_j & \alpha_j\alpha_1 & \cdots & \alpha_j\alpha_p \\ 0 & y^*\alpha_i & \alpha_i\alpha_1 & \cdots & \alpha_i\alpha_p \\ 0 & y^*\alpha_1 & \alpha_1\alpha_1 + \beta_1^2 & \cdots & \alpha_1\alpha_p \\ \vdots & & & & \\ 0 & y^*\alpha_p & \alpha_p\alpha_1 & \cdots & \alpha_p\alpha_p + \beta_p^2 \end{bmatrix}| \quad //\text{Exchange rows} \\
&= |\frac{\alpha_i}{\det \boldsymbol{M}}| \cdot |\det \begin{bmatrix} \beta_j^2 & y^*\alpha_j & \alpha_j\alpha_1 & \cdots & \alpha_j\alpha_p \\ 0 & y^* & \alpha_1 & \cdots & \alpha_p \\ 0 & y^*\alpha_1 & \alpha_1\alpha_1 + \beta_1^2 & \cdots & \alpha_1\alpha_p \\ \vdots & & & & \\ 0 & y^*\alpha_p & \alpha_p\alpha_1 & \cdots & \alpha_p\alpha_p + \beta_p^2 \end{bmatrix}| \quad //\text{Extract out } \alpha_i \\
&= |\frac{\alpha_i \beta_j^2}{\det \boldsymbol{M}}| \cdot |\det \boldsymbol{M}'|,
\end{aligned} \tag{52}
$$

where

$$
\boldsymbol{M}' = \begin{bmatrix} y^* & \alpha_1 & \cdots & \alpha_p \\ y^*\alpha_1 & \alpha_1\alpha_1 + \beta_1^2 & \cdots & \alpha_1\alpha_p \\ \cdots & & & \\ y^*\alpha_p & \alpha_p\alpha_1 & \cdots & \alpha_p\alpha_p + \beta_p^2 \end{bmatrix}. \tag{53}
$$

Similarly, we can prove that

$$
|U^*_{S_j}| = |\frac{\alpha_j \beta_i^2}{\det \boldsymbol{M}}| \cdot |\det \boldsymbol{M}'|. \tag{54}
$$

Therefore, we have

$$
\forall 1 \le i, j \le p, \quad \frac{|U^*_{S_i}|}{|U^*_{S_j}|} = \frac{|\alpha_i/\beta_i^2|}{|\alpha_j/\beta_j^2|} = \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[C_{S_i}(\boldsymbol{x}+\boldsymbol{\epsilon})]/\text{Var}_{\boldsymbol{\epsilon}}[C_{S_i}(\boldsymbol{x}+\boldsymbol{\epsilon})]|}{|\mathbb{E}_{\boldsymbol{\epsilon}}[C_{S_j}(\boldsymbol{x}+\boldsymbol{\epsilon})]/\text{Var}_{\boldsymbol{\epsilon}}[C_{S_j}(\boldsymbol{x}+\boldsymbol{\epsilon})]|}.
$$

Step 3. Based on Step 2, we can directly prove that for the optimal solution U^*, we have

$$
\forall S \in \Omega, \quad |U^*_S| \propto |\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]/\text{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]| \tag{55}
$$

□

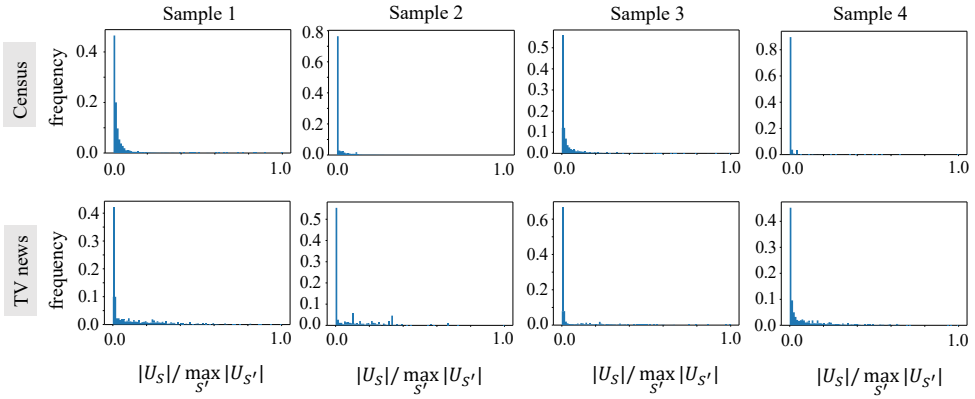

Figure 5: Histograms of the relative strength of causal effects for the 5-layer MLPs trained on the Census dataset and the TV new dataset.

A.7 Proof of Theorem 6

Theorem 6. Let $A_S^{\min} = \min_S |U_S|$ and $A_S^{\max} = \max_S |U_S|$ denote the lower bound and the upper bound of $|U_S|$ over all interactive concepts S. Then, we have

$$A_S^{\min} \cdot \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]} \le \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]} \le A_S^{\max} \cdot \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]}$$

Proof. According to the SCM in Eq. (5), we can obtain that $I(S|\boldsymbol{x}) = U_S \cdot C_S(\boldsymbol{x})$. Hence, we have

$$|\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]| = |U_S| \cdot |\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]|, \quad \mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})] = U_S^2 \cdot \mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})],$$

Therefore,

$$\frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]} = |U_S| \cdot \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]}$$

Then, let $A_S^{\min} = \min_S |U_S|$ and $A_S^{\max} = \max_S |U_S|$ denote the lower bound and the upper bound of the absolute value $|U_S|$ over all interactive concepts S, we have

$$A_S^{\min} \cdot \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]} \le \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]} \le A_S^{\max} \cdot \frac{|\mathbb{E}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]|}{\mathrm{Var}_{\boldsymbol{\epsilon}}[I(S|\boldsymbol{x}+\boldsymbol{\epsilon})]}$$

□

B Sparsity of the causal graph explanation

In this section, we empirically verified the sparsity of the causal graph explanation mentioned in Section 2.1. To this end, given an input sample, we computed causal effects U_S of all 2^n causal patterns. In the computation of the causal effects, we followed the setting of the reference value in Ren et al. (2021a). Furthermore, we computed the relative strength of causal effects as $|U_S|/\max_{S'}|U_{S'}|$ to normalize the strength of causal effects to [0,1]. We trained 5-layer MLPs on the Census dataset and the TV news dataset. Figure 5 shows the histogram of the relative strength of all causal effects on different input samples. We discovered that causal effects of most causal patterns were close to zero, and only a few causal patterns had large absolute value of causal effects. This verified the sparsity of the causal graph.

C More Discussion on Related Works

Interactions. Interactions in game theory is closely related to the quantification of causal effects of each causal pattern. Grabisch & Roubens (1999) first proposed the Shapley interaction index,

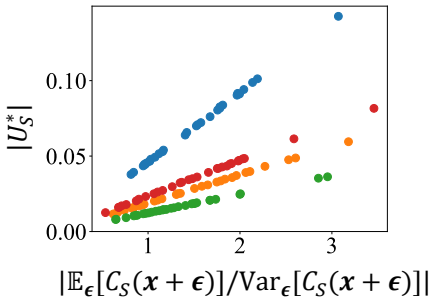

Figure 6: Experimantal verification of Theorem 5. Different colors indicate different random seeds, when we randomly sampled the mean value and the standard deviation of $C_{S_i}(\boldsymbol{x}+\boldsymbol{\epsilon}) \sim \mathcal{N}(\mu_i, \sigma_i^2)$ from the uniform distribution $\mathcal{U}(0.5, 1)$. We can see that the absolute value $|U_S^*|$ was proportional to $|\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]|$, which verified our theorem.

and Lundberg et al. (2018) later used this index to explain tree ensembles. Janizek et al. (2021) explained the pairwise feature interaction in DNNs, while Sundararajan et al. (2020) proposed the Shapley Taylor interaction index to quantify interactions among multiple input variables. Peebles et al. (2020) and Tsang et al. (2018) restricted interactions to achieve feature disentanglement. Song et al. (2019) and Lian et al. (2018) proposed special network architectures to automatically learn feature interactions. Unlike previous studies, we follow Ren et al. (2021a) to represent the inference logic of neural network as a specific sparse causal graph, and define interactive concepts based on the framework of causality.

D Experimental verification of Theorem 5

To verify Theorem 5, we conducted experiments on a linear regression problem on the following dataset. Specifically, we used a 30-dimensional vector $\boldsymbol{C} \in \mathbb{R}^{30}$ to denote the vector of all $C_S(\boldsymbol{x}+\boldsymbol{\epsilon}), S \in \Omega$, and used $\boldsymbol{U} \in \mathbb{R}^{30}$ to denote the vector of all coefficients $U_S, S \in \Omega$ to be learned. Each dimension of the vector $\boldsymbol{C}$ followed a normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$, and was independent of each other. The mean value μ_i and the standard deviation σ_i of the normal distribution was randomly sampled from a uniform distribution $\mathcal{U}(0.5, 1)$. Then, we trained this linear model according to Eq. (12). Figure 6 shows that the absolute value of the coefficient $|U_S^*|$ was proportional to $|\mathbb{E}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]/\mathrm{Var}_{\boldsymbol{\epsilon}}[C_S(\boldsymbol{x}+\boldsymbol{\epsilon})]|$, which verified Theorem 5.

E Experimental details

Training settings. We trained standard DNNs and BNNs with the same architectures on two image datasets and two tabular datasets. For image datasets, we trained standard DNNs and BNNs with two architectures. On the MNIST dataset, we trained a standard DNN and a BNN with the 5-layer MLP architecture. On the CIFAR-10 dataset, we trained a standard DNN and a BNN with the LeNet architecture. On the two tabular datasets, including the UCI TV news dataset (termed *TV news*) and the UCI census income dataset (termed *census*), we trained standard DNNs and BNNs with the 8-layer MLP architecture. All MLPs contained 100 neurons in each hidden layer. For the training of BNNs, the prior distribution of network weights was set to $\mathcal{N}(\boldsymbol{W}; \boldsymbol{0}, \boldsymbol{I})$, and the number of Monte Carlo sampling of network weights was set to 1. All standard DNNs and BNNs were trained using the Adam optimizer (Kingma & Ba, 2015) with learning rate 0.001. The 5-layer MLPs (standard DNN and BNN) on the MNIST dataset was trained for 50 epochs. The LeNet (standard DNN and BNN) on the CIFAR-10 dataset was trained for 300 epochs. The 8-layer MLPs (standard DNN and BNN) on tabular datasets were trained for 200 epochs.

Implementation details for the calculation of $I(S)$. Since the computational cost of $I(S)$ was intolerable for image datasets, we applied a sampling-based approximation method to calculate U_S. For the CIFAR-10 dataset (32×32 pixels on each image), we uniformly split each input image into 8×8 patches. Furthermore, we random sampled 12 patches from the central 6×6 region (*i.e.*, we did not sample patches that were on the edges of an image), and considered these patches as input

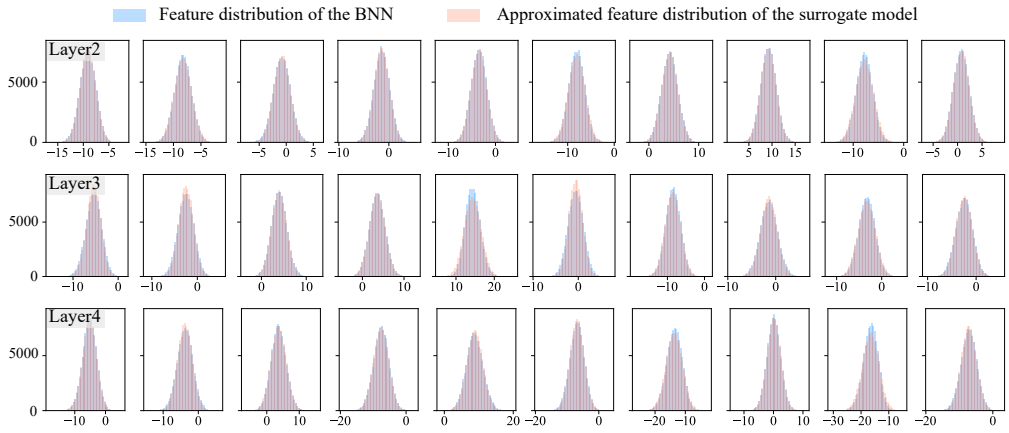

Figure 7: More visualization results of MLP-5 on the MNIST dataset.

variables for each image. The remaining 52 patches were set to the reference value. Similarly, for the MNIST dataset (28×28 pixels on each image), we uniformly split each input image into 7×7 patches, and randomly sampled 12 patches from the central 5×5 region.

Implementation details of the reference value. Let $\mathbb{E}_{\boldsymbol{x}}[x_i]$ denote the mean value of the i-th input dimension over all input samples in the dataset. Then, given an input sample $\boldsymbol{x}$, the reference value is set as follows.

$$r_i = \begin{cases} x_i - \tau, & x_i > \mathbb{E}_{\boldsymbol{x}}[x_i] \\ x_i + \tau, & x_i < \mathbb{E}_{\boldsymbol{x}}[x_i] \end{cases}$$

where $\tau \in \mathbb{R}$ is a constant. We set $\tau = 0.5$ on all datasets (including the TV news dataset, the Census dataset, the MNIST dataset, and the CIFAR-10 dataset). In our experiments, we assume that input samples have been normalized as follows. First, we subtract the mean value of each input dimension over the whole dataset from the input sample. Second, we divide each dimension of the input sample by the standard deviation of this input dimension over the whole dataset. In this way, input samples have zero mean and unit variance on each dimension over the whole dataset, *i.e.*, $\forall i \in N, \mathbb{E}_{\boldsymbol{x}}[x_i] = 0$.

Implementation details of the experiment in Section 2.2. In Section 2.2, we minimized the KL divergence between the feature distribution in the surrogate DNN model and the feature distribution in the BNN. The feature distributions in the surrogate DNN model and in the BNN were not Gaussian distributions. Therefore, the KL divergence between the feature distributions did not have a close-form formula. To facilitate the optimization, we simply used two Gaussian distributions to approximate the feature distributions in the surrogate DNN model and in the BNN, and optimized the KL divergence between the two Gaussian distributions. Besides, we did not consider the dependency between different feature dimensions to simplify the computation.

F More experimental results

F.1 More visualization results for experiments in Section 2.2

In this subsection, we provided more visualization results to show that the feature distribution of the surrogate DNN model could well approximate the feature distribution of the BNN.

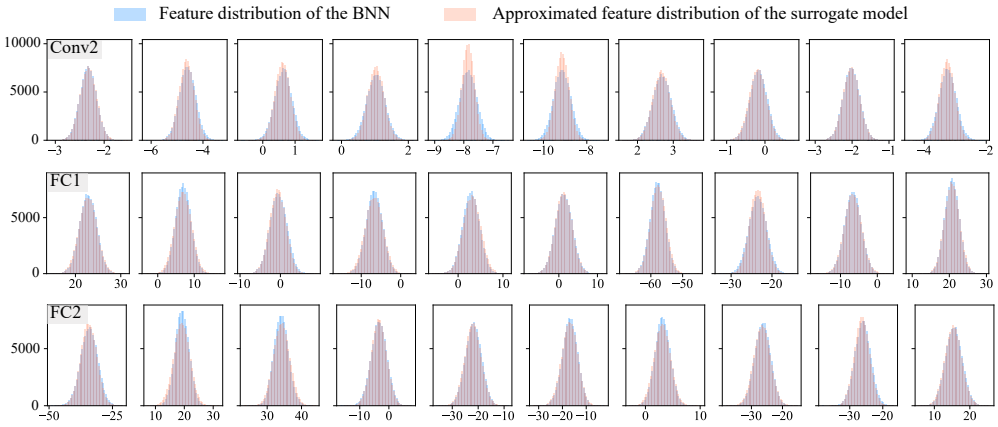

Figure 8: More visualization results of LeNet on the CIFAR-10 dataset.

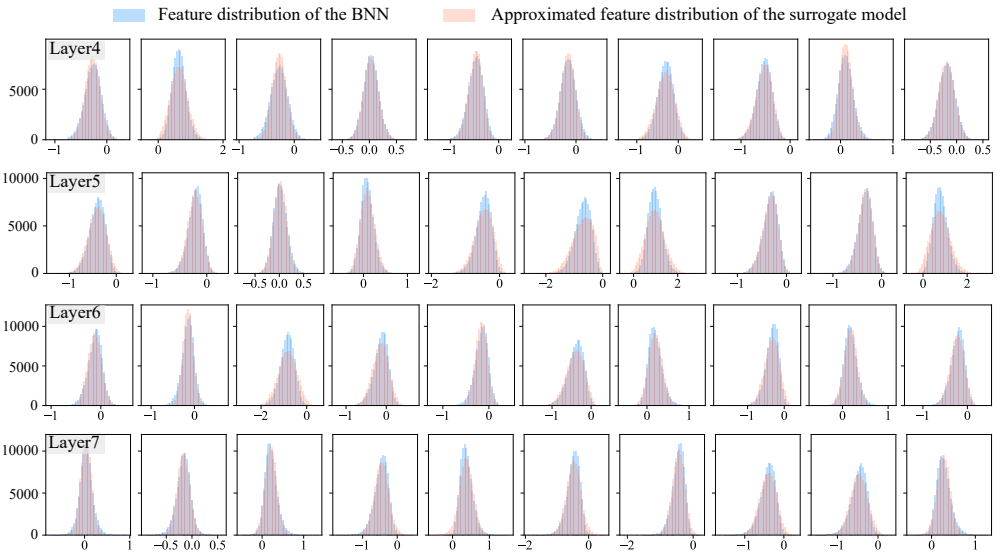

Figure 9: More visualization results of MLP-8 on the Census dataset.

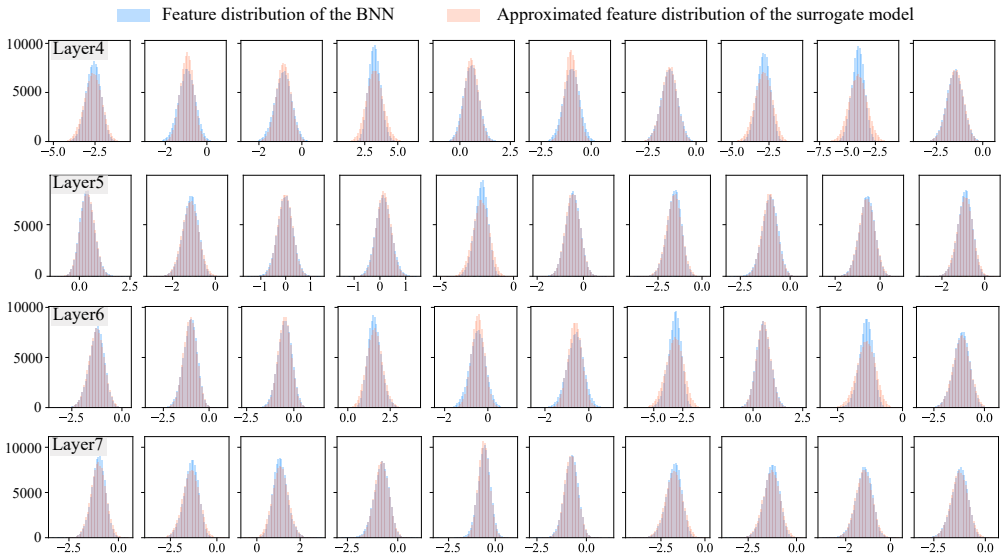

Figure 10: More visualization results of MLP-8 on the TV news dataset.