# DO NEURAL NETWORKS ENCODE GENERALIZABLE PATTERNS IN SPECIFIC FEATURE DIRECTIONS?

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

This paper focuses on the scientific problem of testing whether DNNs encode generalizable patterns in specific feature directions in a verifiable manner. We build on the recent finding that the output score of a neural network can be decomposed into numerical effects of different interaction patterns, some of which can generalize across DNNs and samples, whereas others cannot. We empirically observe that a DNN typically uses only a few feature directions in an intermediate-layer feature space to encode such generalizable interactions. We refer to these directions as reliable feature directions, as they capture generalizable interaction patterns. We identify such reliable feature directions and empirically examine their properties. The identified reliable feature directions usually play a dominant role in classification and are transferable to different samples.

## 1 INTRODUCTION

This study focuses on the scientific problem of explaining the features or mechanisms that contribute to generalization or overfitting in deep neural networks (DNNs). In this direction, mechanistic interpretability studies (Olah et al., 2020; Mu & Andreas, 2020) analyze the semantic meaning of individual neurons or circuits within a DNN. However, it is difficult to rigorously verify whether each feature/mechanism encoded by a DNN represents a generalizable pattern or an overfitted pattern.

To this end, interaction-based explanations provide a verifiable perspective for studying DNN generalization. A growing body of empirical evidence (Ren et al., 2023; Li & Zhang, 2023; Chen et al., 2024) and theoretical analyses (Ren et al., 2024a;b) suggests that *the output score of a DNN for a given input can be faithfully decomposed into the numerical effects of a few interaction patterns*.

> For example, as illustrated in Figure 1, each interaction encoded by a DNN for image classification typically corresponds to a local visual pattern composed of multiple image patches. The co-occurrence of these patches (*e.g.* "eye + snout" patches) triggers the interaction and contributes a numerical effect $0.71$ to the prediction score of the category "corgi".

Thus, interactions provide an interpretable metric for examining the internal factors that affect DNN generalization. Specifically, some interactions are found to generalize across different DNNs and are widely regarded as internal factors governing generalization (Chen et al., 2024; Ren et al., 2024b), whereas non-generalizable interactions are considered overfitted patterns.

In this paper, we find empirical evidence that generalizable interaction patterns are concentrated in specific feature directions. These feature directions encode mainly interactions that are consistently shared across different DNNs and samples. Thus, they are referred to as *reliable feature directions*.

The interaction perspective circumvents a central challenge in conventional feature analysis (*e.g.*, canonical correlation analysis, CCA; Andrew et al., 2013; Bykhovskaya & Gorin, 2024), which typically requires aligning feature spaces across models to identify shared feature components. In contrast, interactions encoded by different feature directions are naturally defined in the same interaction space. Therefore, if two feature directions encode similar interactions, their shared interaction patterns provide a basis for examining generalizability across different DNNs.

Thus, we identify a set of feature directions that primarily encode generalizable interactions, namely reliable feature directions, while other directions primarily encode non-generalizable interactions.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

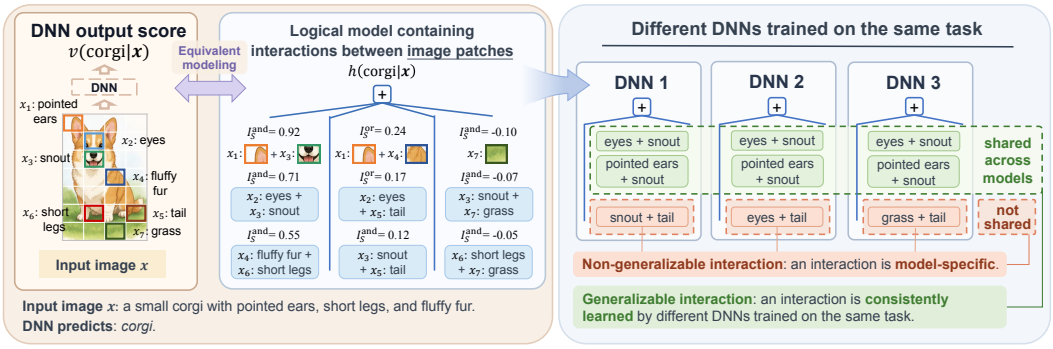



Figure 1: (1) Illustration of interaction-based explanation. (Left) For a given input $x$, the complex inference logic of a DNN can be rewritten as a small set of interactions between different image patches with theoretically guaranteed faithfulness. (Right) Some interactions between patches are shared by different DNNs for the same task and thus are termed generalizable interactions, whereas others are non-generalizable interactions. We find that generalizable interactions are concentrated in a few specific feature directions, which constitute reliable feature components. See Appendix C.3 for extracted interactions on ImageNet images.

The learned directions tend to encode low-order interactions, *i.e.*, simple inference patterns, most of which can generalize to other DNNs. High-order interactions are less likely to generalize across DNNs and are mainly learned after the DNN begins to overfit.

Based on the new interaction-analysis paradigm, our experiments yield richer empirical insights into the hypothesis that generalizable patterns are encoded along specific directions. Specifically, (1) Reliable feature directions learned from a small set of samples can transfer to many unseen samples. (2) Notably, the features along the learned directions play a dominant role in classification, whereas residual features along other directions contribute marginally.

## 2 Algorithm

### 2.1 Preliminaries: Interactions

Consider a DNN and an input sample $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ consisting of $n$ input variables[1], and let $N = \{1, 2, \ldots, n\}$ denote the set of indices for these input variables. Following (Deng et al., 2022; Shen et al., 2023), the scalar output of the DNN, $v(\boldsymbol{x}) \in \mathbb{R}$, is typically defined as the classification confidence score[2] of the sample $\boldsymbol{x}$:

$$v(\boldsymbol{x}) \overset{\text{def}}{=} \log \frac{p(y = y^{\text{truth}}|\boldsymbol{x})}{1 - p(y = y^{\text{truth}}|\boldsymbol{x})} \in \mathbb{R}, \tag{1}$$

where $p(y = y^{\text{truth}}|\boldsymbol{x})$ denotes the probability[2] of classifying $\boldsymbol{x}$ into the ground-truth category $y^{\text{truth}}$.

Ren et al. (2024a); Chen et al. (2024) have proposed to use the following logical model $h(\cdot)$ based on AND-OR interactions to explain the inference logic of the DNN $v(\cdot)$ on the input sample $\boldsymbol{x}$.

$$h(\boldsymbol{x}) \overset{\text{def}}{=} \sum\nolimits_{S \in \Omega^{\text{and}}} \mathbb{1}_{\text{and}}(S \mid \boldsymbol{x}) \cdot I_S^{\text{and}} + \sum\nolimits_{S \in \Omega^{\text{or}}} \mathbb{1}_{\text{or}}(S \mid \boldsymbol{x}) \cdot I_S^{\text{or}} + b, \tag{2}$$

where each interaction represents a specific inference pattern used by the DNN to classify $\boldsymbol{x}$. $\mathbb{1}_{\text{and}}(S \mid \boldsymbol{x})$ and $\mathbb{1}_{\text{or}}(S \mid \boldsymbol{x})$ are binary trigger functions for AND interaction and OR interaction, respectively, and will be introduced later. Accordingly, $I_S^{\text{and}}$ and $I_S^{\text{or}}$ denote the numerical effects of these interactions, which are computed by using all DNN outputs $\boldsymbol{v} = [v(\boldsymbol{x}_{S_1}), v(\boldsymbol{x}_{S_2}), \ldots, v(\boldsymbol{x}_{S_{2^n}})]^\top$ (see Appendix C.1 for details). The output vector $\boldsymbol{v}$ collects the DNN outputs over all $2^n$ masked states[3] of $\boldsymbol{x}$, where $\boldsymbol{x}_S$ denotes a masked state of the input sample. $\Omega^{\text{and}}$ and $\Omega^{\text{or}}$ are the set of AND interactions and the set of OR interactions, respectively, and the bias term $b$ is set to $v(\boldsymbol{x}_\emptyset)$.

[1] See Appendix B.1 for details on treating image patches or tabular fields as individual input variables.

[2] See Appendix B.1 for the probability in image and tabular classification tasks.

[3] The masked sample $\boldsymbol{x}_S$ is constructed by retaining the input variables in $S$ while masking the other input variables in $N \backslash S$. Each variable $i$ is masked by setting $\boldsymbol{x}_i$ to a baseline value $b_i$. For image classification,

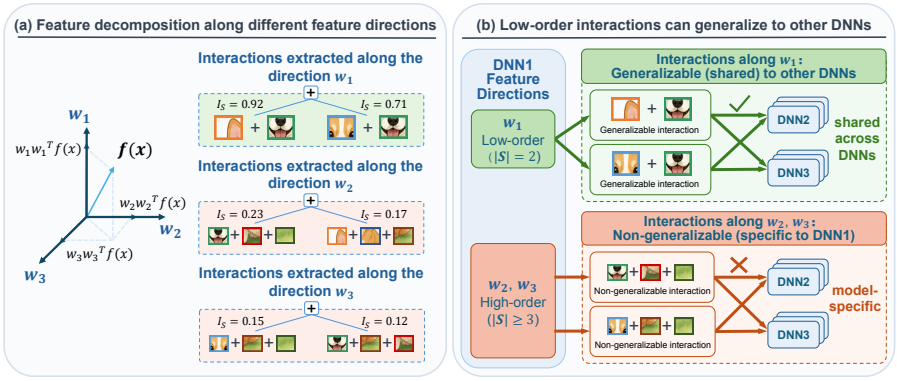



Figure 2: (a) The intermediate-layer feature $\boldsymbol{f}(\boldsymbol{x})$ can be decomposed along feature directions $\boldsymbol{w}_k$, each associated with a set of interactions. (b) Generalizable interactions can also be encoded by other DNNs, potentially along different feature directions.

• **Proved fidelity.** Chen et al. (2024) have proven that above AND-OR logical model $h(\cdot)$ can precisely represent the output score of the DNN $v(\cdot)$ across exponentially many samples.

$$\forall \boldsymbol{x}' \in \mathcal{D}, \quad |h(\boldsymbol{x}') - v(\boldsymbol{x}')| < \epsilon, \tag{3}$$

where $\mathcal{D} = \{\boldsymbol{x}_T \mid T \subseteq N\}$ is the set of all $2^n$ masked states[3] of the input sample $\boldsymbol{x}$, and $\epsilon$ is a small constant quantifying the fidelity of the logical model.

• **Proved sparsity.** Ren et al. (2024a) have proven that a well-trained DNN[4] typically extracts only a strictly bounded[5] number of interactions from a single sample. Empirically, a DNN usually encodes around 50-150 interactions.

The sparsity and the fidelity of interactions have been broadly verified on diverse DNN architectures and tasks (see Appendix C.2 and recent studies (Li & Zhang, 2023; Zhou et al., 2024; Cheng et al., 2024)). This guarantees the faithfulness of the explanation. *The supplementary **video demo** provides a detailed illustration of interaction-based explanation.*

**How to understand AND-OR interactions $\mathbb{1}_{\text{and}}(S \mid \boldsymbol{x})$ and $\mathbb{1}_{\text{or}}(S \mid \boldsymbol{x})$.** $\mathbb{1}_{\text{AND}}(S|\boldsymbol{x}_{\text{mask}}) \in \{0, 1\}$ is a binary trigger function for an AND interaction among a subset $S \subseteq N$ of input variables, and $\mathbb{1}_{\text{OR}}(S|\boldsymbol{x}_{\text{mask}}) \in \{0, 1\}$ is the trigger function for the OR interaction among $S \subseteq N$ of input variables.

> **AND interactions** represent visual patterns that capture synergistic effects. In Figure 1, the eye and snout patches jointly activate an AND interaction supporting the "corgi" prediction; masking either patch removes its contribution. By contrast, **OR interactions** capture redundancy effects, where the presence of any patch in $S$ is sufficient to activate the interaction.

**Efficient Computation.** Direct interaction extraction incurs exponential time complexity with respect to the number of input variables. Nevertheless, prior work has developed approximation algorithms (Kang et al., 2024; 2025; Butler et al., 2025) and practical engineering techniques (Shen et al., 2023; Cheng et al., 2024; Chen et al., 2024) that enable efficient interaction extraction. Interaction-based explanations have also been used in industrial settings(SymTrust AI, 2026). See Appendix C.4 for the details of computational techniques.

### 2.2 Identifying feature directions related to generalizable interactions

Defining the representational quality of a feature direction remains an intriguing open problem in AI theory. We offer a verifiable exploration from the interaction perspective. Specifically, we ask whether a DNN uses only a small number of feature directions to encode generalizable interactions, while the

following Zhang et al. (2021), $b_i$ is set to the dataset-wide average pixel value. For tabular classification, continuous variables are set to empirical means and categorical variables are set to their most frequent categories.

[4]The sparsity of interactions can be guaranteed by three common conditions for the DNN's smooth inferences on randomly masked samples. See Appendix B.3 for the detailed conditions.

[5]Specifically, given a sample with $n$ input variables, among the $2^n$ types of variable combinations, the number of interactions generally falls within $O(n^p)$, where $p \in [1.5, 2]$ in most cases.

remaining directions primarily encode non-generalizable interactions. To study this question, we first decompose the DNN output into components along different feature directions.

**Decomposing the DNN output into components along different feature directions.** Given a DNN $v(\cdot)$ and an input sample $\boldsymbol{x}$, let $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_K$ denote a set of mutually orthogonal unit vectors in the feature space of an intermediate layer $\boldsymbol{f}(\cdot) \in \mathbb{R}^K$. We decompose the intermediate-layer feature along different directions (Figure 2(a)) as $\boldsymbol{f}(\boldsymbol{x}) = \sum_{k=1}^{K} \boldsymbol{w}_k \boldsymbol{w}_k^\top \boldsymbol{f}(\boldsymbol{x})$. Consequently, the output score $v(\boldsymbol{x})$ of the DNN can also be decomposed along these feature directions as follows.

**Theorem 2.1 (Decomposing DNN Output $v(\boldsymbol{x})$ into Components $\{u_k(\boldsymbol{x})\}$).** *Given an input sample $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ composed of $n$ input variables[1], the DNN output can be expressed via a Taylor series expansion around the reference feature[6] $\boldsymbol{f}_\emptyset$, as follows.*

$$\begin{aligned} v(\boldsymbol{x}) &= v_{\boldsymbol{f}_\emptyset} + \boldsymbol{g}^\top \big(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}_\emptyset\big) + O\big(\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}_\emptyset\|_2^2\big) \\ &= \underbrace{\boldsymbol{g}^\top [\boldsymbol{w}_1 \boldsymbol{w}_1^\top] \boldsymbol{f}(\boldsymbol{x})}_{u_1(\boldsymbol{x})} + \underbrace{\boldsymbol{g}^\top [\boldsymbol{w}_2 \boldsymbol{w}_2^\top] \boldsymbol{f}(\boldsymbol{x})}_{u_2(\boldsymbol{x})} + \cdots + \underbrace{\boldsymbol{g}^\top [\boldsymbol{w}_K \boldsymbol{w}_K^\top] \boldsymbol{f}(\boldsymbol{x})}_{u_K(\boldsymbol{x})} + c \\ \text{s.t.} \quad & \forall i \in \{1, 2, \ldots, K\} : \boldsymbol{w}_i^\top \boldsymbol{w}_i = 1, \quad \forall i \neq j : \boldsymbol{w}_i^\top \boldsymbol{w}_j = 0 \end{aligned} \tag{4}$$

*where $u_k(\boldsymbol{x})$ denotes the output component attributed to the $k$-th feature direction $\boldsymbol{w}_k$, with $\forall k \in \{1, 2, \ldots, K\}, u_k(\boldsymbol{x}) = \boldsymbol{g}^\top [\boldsymbol{w}_k \boldsymbol{w}_k^\top] \boldsymbol{f}(\boldsymbol{x})$. Here, $\boldsymbol{g} = \nabla_{\boldsymbol{f}} v_{\boldsymbol{f}_\emptyset}$ denotes the gradient of the DNN output $v(\boldsymbol{x})$ w.r.t. $\boldsymbol{f}$ evaluated at the reference feature $\boldsymbol{f}_\emptyset$, and $v_{\boldsymbol{f}_\emptyset}$ is the network output when the intermediate-layer feature is set to $\boldsymbol{f}_\emptyset$. The term $c = v_{\boldsymbol{f}_\emptyset} - \boldsymbol{g}^\top \boldsymbol{f}_\emptyset + O\big(\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}_\emptyset\|_2^2\big)$ collects the bias and the higher-order remainder in the Taylor expansion.*[7]

**Defining interactions corresponding to each feature direction.** The above decomposition of output score $v(\boldsymbol{x})$ also induces an overlapping partition of interactions attributed to different feature directions[8]. Each feature direction $\boldsymbol{w}_k$ is then associated with a specific set of AND-OR interactions. Therefore, for each feature direction $\boldsymbol{w}_k$, we follow Equation (2) to construct the following logical model $h_k(\boldsymbol{x}; \boldsymbol{w}_k)$ to interpret each output component $u_k(\boldsymbol{x})$.

$$\forall \boldsymbol{x}' \in \mathcal{D}, \quad |h_k(\boldsymbol{x}'; \boldsymbol{w}_k) - u_k(\boldsymbol{x}')| < \epsilon, \tag{5}$$

$$\text{with} \quad h_k(\boldsymbol{x}'; \boldsymbol{w}_k) = \sum\nolimits_{S \in \Omega_k^{\text{and}}} \mathbb{1}_{\text{and}}(S \mid \boldsymbol{x}') \cdot J_S^{\text{and}}(\boldsymbol{w}_k) + \sum\nolimits_{S \in \Omega_k^{\text{or}}} \mathbb{1}_{\text{or}}(S \mid \boldsymbol{x}') \cdot J_S^{\text{or}}(\boldsymbol{w}_k) \; + \; b_k.$$

Following Chen et al. (2024), the logical model $h_k(\boldsymbol{x}; \boldsymbol{w}_k)$ with properly learned interaction weights $\{J_S^{\text{and}}(\boldsymbol{w}_k), J_S^{\text{or}}(\boldsymbol{w}_k)\}$ can accurately match $u_k(\boldsymbol{x})$ across an exponential number of masked samples[3] in $\mathcal{D}$, where $\mathcal{D} = \{\boldsymbol{x}_T \mid T \subseteq N\}$ denotes the set of all $2^n$ masked states of the input sample $\boldsymbol{x}$. $\epsilon$ is a small constant that bounds the matching error of the logical model $h_k(\cdot; \boldsymbol{w}_k)$. $\Omega_k^{\text{and}}$ and $\Omega_k^{\text{or}}$ denote the sets of AND and OR interactions extracted from $u_k(\cdot)$, respectively. $J_S^{\text{and}}(\boldsymbol{w}_k)$ and $J_S^{\text{or}}(\boldsymbol{w}_k)$ are scalar weights determined once $\boldsymbol{w}_k$ is specified.

### 2.2.1 HYPOTHESIS

Theorem 2.1 enables us to formulate the hypothesis tested in this study: *that there exists a set $\mathcal{G}$ of feature directions that primarily encode generalizable interactions, while the complementary set primarily encodes non-generalizable interactions.*

Under this hypothesis, the output score $v(\boldsymbol{x})$ can be decomposed into two output components: $U^{\text{rel}}$ attributed to generalizable interactions, and $U^{\text{non-rel}}$ attributed to non-generalizable interactions.

**Hypothesis 1** (Existence of Feature Directions Primarily Encoding Generalizable Interactions)**.** *There exists a set $\mathcal{G}$ of feature directions such that the directions in $\mathcal{G}$ primarily encode generalizable*

[6]The reference feature $\boldsymbol{f}_\emptyset$ is computed by feeding the reference sample $\boldsymbol{x}_\emptyset$ in which all input variables are masked into the DNN. We can also use other reference features, such as the average feature across all masked states of a given sample. The detailed discussion on the choice of reference feature is provided in Appendix D.3.

[7]For notational simplicity, we suppress the input dependence of $c$, which generally varies across masked states. See Appendix D.2 for a detailed discussion and an empirical analysis of the Taylor remainder.

[8]The interaction decomposition in Section 2.1 is not restricted to the original DNN output. It can be applied to any scalar function defined on masked inputs. In this way, once a feature direction $\boldsymbol{w}_k$ is fixed, the projection of the intermediate feature onto this direction defines such a scalar function $u_k(\boldsymbol{x})$, and we can therefore apply the interaction decomposition to $u_k(\cdot)$.

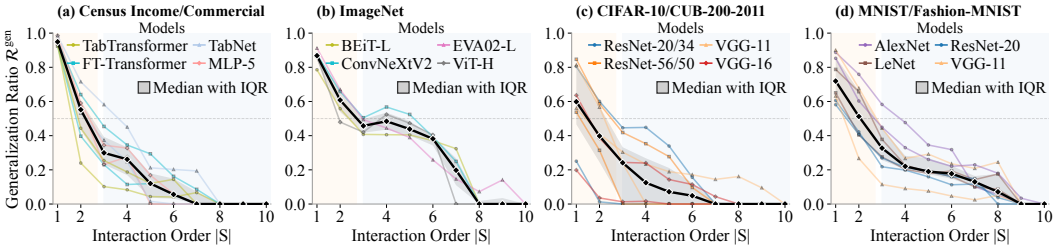



Figure 3: Curves of the generalization ratio $\mathcal{R}^{\text{gen}}(m)$ across interaction orders $|S|$. Each light curve represents the generalization ratio of an individual DNN. The bold line denotes the median across models, and the shaded band indicates the IQR[10]. The results show that low-order interactions, especially orders 1-2, are substantially more generalizable than high-order interactions at orders 3–10.

*interactions, whereas the complementary directions primarily encode non-generalizable ones. For such a set $\mathcal{G}$, the output score $v(\boldsymbol{x}')$ for any masked sample $\boldsymbol{x}' \in \mathcal{D}$ can be decomposed as follows.*

$$\forall \boldsymbol{x}' \in \mathcal{D}, \quad v(\boldsymbol{x}') = U^{rel}(\boldsymbol{x}') + U^{non\text{-}rel}(\boldsymbol{x}') + c, \tag{6}$$

$$U^{rel}(\boldsymbol{x}') := \sum\nolimits_{k\in\mathcal{G}} u_k(\boldsymbol{x}') \approx \sum\nolimits_{k\in\mathcal{G}} h_k(\boldsymbol{x}'; \boldsymbol{w}_k) \quad \text{// } \begin{array}{l}\textit{most interactions } \{J_S^{and}(\boldsymbol{w}_k), J_S^{or}(\boldsymbol{w}_k)\} \\ \textit{in } h_k(\boldsymbol{x}'; \boldsymbol{w}_k) \textit{ are generalizable}^9\end{array}$$

$$U^{non\text{-}rel}(\boldsymbol{x}') := \sum\nolimits_{k'\notin\mathcal{G}} u_{k'}(\boldsymbol{x}') \approx \sum\nolimits_{k'\notin\mathcal{G}} h_{k'}(\boldsymbol{x}'; \boldsymbol{w}_{k'}) \quad \text{// } \begin{array}{l}\textit{most interactions } \{J_S^{and}(\boldsymbol{w}_{k'}), J_S^{or}(\boldsymbol{w}_{k'})\} \\ \textit{in } h_{k'}(\boldsymbol{x}'; \boldsymbol{w}_{k'}) \textit{ are non-generalizable}\end{array}$$

*This hypothesis states that such a set $\mathcal{G}$ exists, so that $U^{rel}(\boldsymbol{x}')$ is primarily attributed to generalizable interactions, while $U^{non\text{-}rel}(\boldsymbol{x}')$ is primarily attributed to non-generalizable interactions.*

**Understanding the output components $U^{\text{rel}}(\boldsymbol{x})$ and $U^{\text{non-rel}}(\boldsymbol{x})$.** We notice that the feature $\boldsymbol{f}(\boldsymbol{x})$ can be decomposed along different feature directions, with $\boldsymbol{f}(\boldsymbol{x}) = \sum_{k=1}^{K} \boldsymbol{w}_k \boldsymbol{w}_k^\top \boldsymbol{f}(\boldsymbol{x})$. Thus, $U^{\text{rel}}(\boldsymbol{x})$ can be interpreted as the classification utility attributed to the decomposed feature component $\boldsymbol{f}^{\text{rel}}(\boldsymbol{x})$ along the feature directions in $\mathcal{G}$, while $U^{\text{non-rel}}(\boldsymbol{x})$ analogously represents the classification utility attributed to the decomposed feature component $\boldsymbol{f}^{\text{non-rel}}(\boldsymbol{x})$ along other feature directions.

$$U^{\text{rel}}(\boldsymbol{x}) = \sum_{k\in\mathcal{G}} u_k(\boldsymbol{x}) = \boldsymbol{g}^\top \underbrace{\sum\nolimits_{k\in\mathcal{G}} \boldsymbol{w}_k \boldsymbol{w}_k^\top \boldsymbol{f}(\boldsymbol{x})}_{\boldsymbol{f}^{\text{rel}}(\boldsymbol{x})}, \quad U^{\text{non-rel}}(\boldsymbol{x}) = \sum_{k\notin\mathcal{G}} u_k(\boldsymbol{x}) = \boldsymbol{g}^\top \underbrace{\sum\nolimits_{k\notin\mathcal{G}} \boldsymbol{w}_k \boldsymbol{w}_k^\top \boldsymbol{f}(\boldsymbol{x})}_{\boldsymbol{f}^{\text{non-rel}}(\boldsymbol{x})}. \tag{7}$$

### 2.2.2 Defining and distinguishing generalizable interactions.

To test the above hypothesis that there exist specific feature directions that primarily encode generalizable interactions, we first need to define and distinguish generalizable interactions and non-generalizable interactions.

**Defining the generalizability of each specific interaction.** We follow He et al. (2025) to define interaction generalizability based on generalizability to baseline DNNs $v^{\text{base}}(\cdot)$, which are trained for the same task but on a different training set $\mathcal{X}$. *If an interaction encoded by the target DNN $v$ is also universally encoded by baseline DNNs $v^{base}(\cdot)$ trained on different datasets, then this interaction can be considered to be generalizable to unseen samples in $\mathcal{X}$.* We thus introduce the following binary indicators $\rho_S^{\text{and}}$ and $\rho_S^{\text{or}}$ to identify these generalizable interactions.

$$\rho_S^{\text{and}} = \mathbb{1}(S \in \Omega^{\text{and,base}}) \cdot \mathbb{1}(I_S^{\text{and}} \cdot I_S^{\text{and,base}} > 0), \quad \rho_S^{\text{or}} = \mathbb{1}(S \in \Omega^{\text{or,base}}) \cdot \mathbb{1}(I_S^{\text{or}} \cdot I_S^{\text{or,base}} > 0), \tag{8}$$

where $\Omega^{\text{and,base}}$ and $\Omega^{\text{or,base}}$ denote the sets of AND interactions and OR interactions extracted from the baseline DNN $v^{\text{base}}(\cdot)$, respectively. We require $I_S^{\text{and,base}}$ and $I_S^{\text{and}}$ to have consistent signs to ensure the AND interaction $S$ has similar utilities in both the target DNN and baseline DNN. The same requirement applies to the OR interaction.

**Finding that interaction order can distinguish generalizable interactions from non-generalizable interactions.** The order (or complexity) of an interaction $S$ is defined as the number of input variables

[9] See the detailed definition of generalizable interactions in Section 2.2.2.

[10] IQR is the range between the 25th and 75th percentiles of the data, providing a measure of variability.

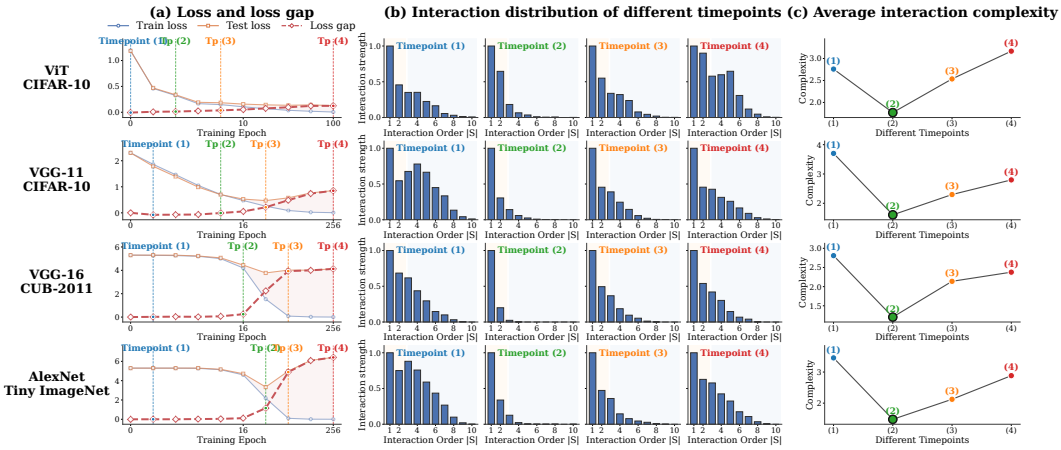



Figure 4: Evolution of interaction strength distributions during DNN training. Curves of the training loss (blue), testing loss (orange), and loss gap (red) in **(a)** are temporally aligned with the evolution of the interaction distribution $\boldsymbol{a}$ at different timepoints in **(b)**. Changes in the average interaction complexity, defined as $\sum_{m=1}^{n} m a^{(m)} / \sum_{m=1}^{n} a^{(m)}$, demonstrate a close correlation between DNN overfitting and the emergence of high-order interactions. Specifically, the loss gap in **(a)** does not increase until the DNN begins to learn interactions of increasing orders at Timepoint 2 in **(c)**.

it involves, *i.e., order*$(S) = |S|$. Many empirical studies (Zhou et al., 2024; Ren et al., 2024b) have reported that low-order interactions are more likely to be generalizable than high-order interactions. To quantify this trend, we define the ratio of generalizable interactions for each $m$-th order, as follows.

$$\forall m \in \{1, 2, \ldots, n\}, \quad \mathcal{R}^{\text{gen}}(m) = \frac{\sum_{|S|=m}\left(\sum_{S\in\Omega^{\text{and}}} \rho_S^{\text{and}} \cdot |I_S^{\text{and}}| + \sum_{S\in\Omega^{\text{or}}} \rho_S^{\text{or}} \cdot |I_S^{\text{or}}|\right)}{\sum_{|S|=m}\left(\sum_{S\in\Omega^{\text{and}}} |I_S^{\text{and}}| + \sum_{S\in\Omega^{\text{or}}} |I_S^{\text{or}}|\right)} \tag{9}$$

We conduct the following two experiments, which show that low-order interactions are much more likely to be generalizable than high-order interactions.

*Experiment 1: Low-order interactions are more generalizable.* We conduct experiments to verify that low-order interactions are more generalizable than high-order ones. Specifically, we measure the ratio of generalizable interactions $\mathcal{R}^{\text{gen}}(m)$ defined in Equation (9) across different orders. The evaluation spans a diverse set of architectures and datasets, including ResNet-20/34/50/56 (He et al., 2016), VGG-11/16 (Simonyan & Zisserman, 2014), AlexNet (Krizhevsky et al., 2012), and LeNet (LeCun et al., 1998) on CIFAR-10 (Krizhevsky & Hinton, 2009), MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), and CUB-200-2011 (Wah et al., 2011); FT-Transformer (Gorishniy et al., 2021), TabTransformer (Huang et al., 2020), TabNet (Arik & Pfister, 2021), and MLP-5 on the Census (Becker & Kohavi, 1996) and Commercial tabular datasets; and EVA-02-L (Fang et al., 2023), ViT-H (Dosovitskiy et al., 2021), BEiT-L (Bao et al., 2022), and ConvNeXtV2 (Woo et al., 2023) on ImageNet (Deng et al., 2009). Detailed experimental settings are provided in Appendix E.1.

Figure 3 shows $\mathcal{R}^{\text{gen}}(m)$ across interaction orders. Low-order interactions (e.g., orders 1–2) exhibit high generalizability, whereas high-order interactions (e.g., orders 3–10) show substantially lower ratios, indicating that low-order interactions are more consistently shared across DNNs. This trend is consistent across architectures and datasets.

*Experiment 2: Overfitted DNNs tend to learn high-order interactions.* We examine the relationship between overfitting and interaction complexity by tracking the gap between training and testing losses together with the interaction strength distribution $\boldsymbol{a}$ during training. We train a ViT model, VGG-11/16 models, and an AlexNet model on the CIFAR-10, CUB-200-2011, and TinyImageNet datasets. Detailed experimental settings are provided in Appendix E.1. We visualize $\boldsymbol{a}$ at different training stages, where $\boldsymbol{a}$ aggregates the total interaction strength at each order:

$$\boldsymbol{a} \overset{\text{def}}{=} [a^{(1)}, a^{(2)}, \ldots, a^{(n)}]^\top, \quad a^{(m)} = \sum\nolimits_{|S|=m} \left(\sum\nolimits_{S\in\Omega^{\text{and}}} |I_S^{\text{and}}| + \sum\nolimits_{S\in\Omega^{\text{or}}} |I_S^{\text{or}}|\right). \tag{10}$$

Figure 4 shows the loss gap and the interaction strength distribution $\boldsymbol{a}$ during training until the DNN is overfitted. By comparing the loss gap with the interaction distributions at different timepoints, we

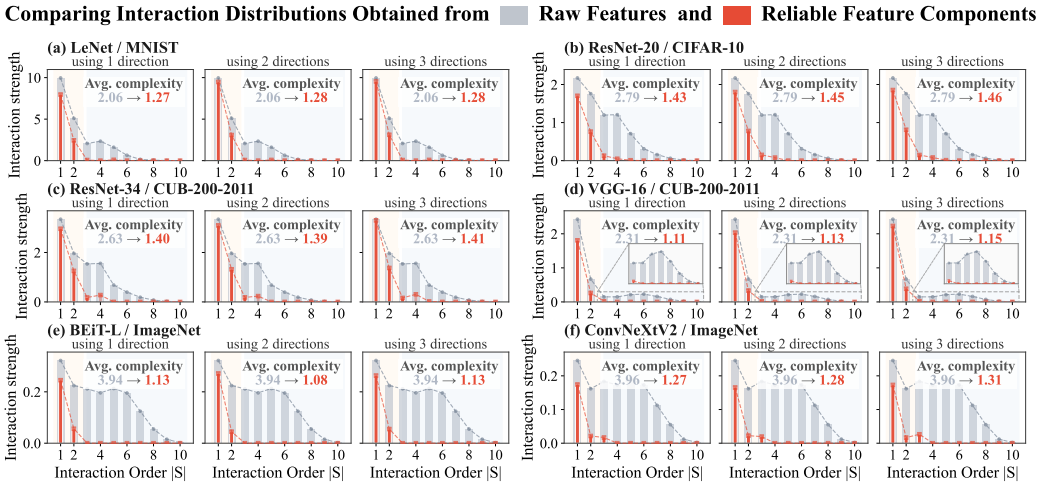



Figure 5: Interaction strength distribution $\boldsymbol{a}$ across different orders $m$. The interactions extracted from the reliable feature components $\boldsymbol{f}^{\text{rel}}(\boldsymbol{x})$ (denoted by red bars) are mainly of low orders, compared with interactions extracted from the raw feature $\boldsymbol{f}(\boldsymbol{x})$ (denoted by gray bars). We find that using different numbers of feature directions yields similar interaction distributions, indicating that DNNs utilize only a few feature directions to encode generalizable interactions.

find that the loss gap begins to increase only after the DNN starts to learn interactions of increasing orders (Timepoint 2). This finding suggests a close correlation between DNN overfitting and the emergence of high-order interactions, which are empirically less likely to be generalizable.

### 2.2.3 Identifying feature directions for generalizable interactions

In this subsection, we examine the core question of this paper, *i.e.*, whether there exist specific feature directions that primarily encode generalizable interactions.

In the previous subsection, we empirically observe that low-order interactions are more likely to generalize, whereas high-order interactions are less likely to generalize and are associated with overfitting. Since directly calculating the generalizability criterion in Equation (8) requires repeatedly extracting interactions from multiple baseline DNNs, we use interaction order as a tractable surrogate for learning feature directions. Therefore, according to Equation (5), the learning of feature directions for generalizable interactions can be roughly reformulated as learning a direction $\boldsymbol{w}$, such that the corresponding decomposed output component $u_k(\boldsymbol{x})$ predominantly encodes low-order interactions $J_S^{\text{and}}(\boldsymbol{w})$ and $J_S^{\text{or}}(\boldsymbol{w})$, which are empirically more likely to be generalizable. Accordingly, we use the following loss function to identify such feature directions.

$$\min_{\boldsymbol{w}} \ \mathcal{L}(\boldsymbol{w}) \overset{\text{def}}{=} \sum\nolimits_{\text{type}\in\{\text{and,or}\}} \sum\nolimits_{S\in\Omega^{\text{type}}} \ell^{\text{type}}(S), \quad \text{subject to} \quad \|\boldsymbol{w}\|_2 = 1,$$
$$\text{with} \ \ \ell^{\text{type}}(S) := \mathbb{1}(|S|\le M)\cdot\left[J_S^{\text{type}}(\boldsymbol{w}) - I_S^{\text{type}}\right]^2 + \lambda\cdot\mathbb{1}(|S|>M)\cdot\left[J_S^{\text{type}}(\boldsymbol{w})\right]^2. \tag{11}$$

where $I_S^{\text{and}}$ and $I_S^{\text{or}}$ denote the interaction effects extracted from the raw feature $\boldsymbol{f}(\boldsymbol{x})$. The loss $\mathcal{L}(\boldsymbol{w})$ is designed to retain low-order interactions ($|S|\le M$) while suppressing high-order ones ($|S|>M$). Specifically, the loss encourages the low-order interaction extracted from a single feature direction $J_S^{\text{type}}(\boldsymbol{w})$ to match the corresponding interaction in the raw feature $I_S^{\text{type}}$. The hyperparameter $\lambda$ balances these two terms. Based on the empirical results in Figure 3, we use $M=2$ as the order threshold in our experiments[11], since interactions of orders 1–2 exhibit substantially higher generalizability than higher-order interactions.

**Extension to multiple directions.** The above method of learning a single feature direction $\boldsymbol{w}$ can be extended to learn multiple feature directions $\{\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_k\}$ that capture different sets of generalizable interactions. Specifically, starting with $\boldsymbol{w}_1$, when we have learned a set of $k-1$ feature

[11]The choice of $M=2$ in this paper is motivated by the observed separation in generalizability between orders 1–2 and higher orders in the evaluated settings, rather than an arbitrary a priori assumption; the precise boundary may vary across models and tasks.

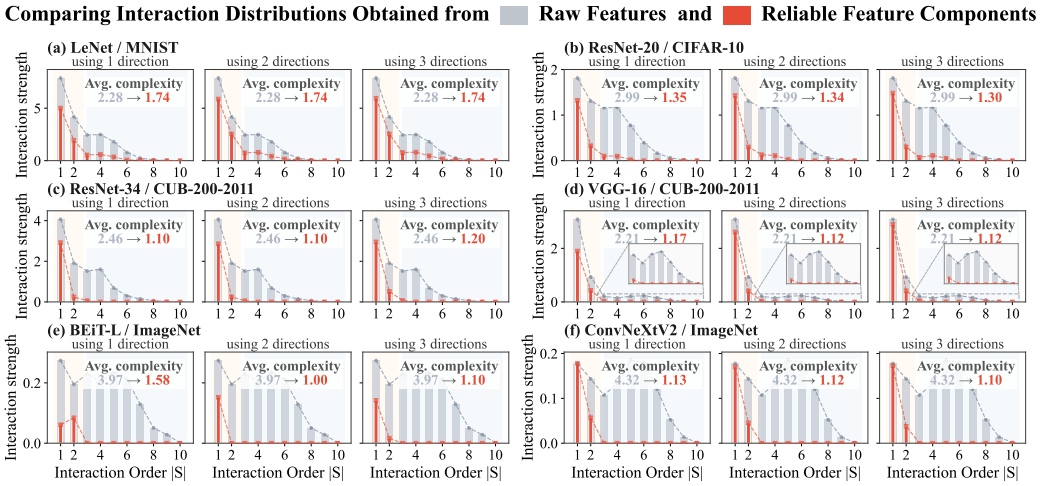



Figure 6: Interaction strength distribution a across different orders m in unseen samples. On unseen samples, the learned feature directions still encode low-order interactions while suppressing highorder ones, indicating that the learned feature directions can transfer to unseen samples.

directions $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{k-1}\}$, we remove the subspace spanned by these directions from the feature space and learn the next direction $\boldsymbol{w}_k$ from the residual feature component. In this manner, we can further capture additional feature directions to progressively capture additional generalizable interactions. Please see the pseudocode and details in Appendix D.4.

# 3 Experiments

## 3.1 Verifying that learned feature directions capture low-order interactions

In this subsection, we evaluate whether the learned feature directions predominantly encode low-order interactions while suppressing high-order interactions. Given a set of learned feature directions, we compute the distribution of interaction strength $\boldsymbol{a}$ in Equation (10) across different interaction orders $m$. Specifically, we evaluate LeNet, ResNet-20, ResNet-34, and VGG-16-BN on MNIST, CIFAR-10, and CUB-200-2011, together with BEiT (Bao et al., 2022) and ConvNeXtV2 (Woo et al., 2023) on ImageNet. We conduct experiments on overfitted samples (those containing lots of non-generalizable interactions). As an exploratory study, this work focuses on identifying reliable feature directions for a specific classification category. Language generation tasks (*e.g.*, with LLMs) do not involve fixed-category classification and are therefore outside the scope of this evaluation. Detailed experimental settings are provided in Appendix E.4.

In this experiment, we observe that the evaluated DNNs use only a small number (1–3) of feature directions to encode low-order interactions. Figure 5 shows the distribution of interactions $\mathbf{a} = [a^{(1)}, a^{(2)}, \ldots, a^{(n)}]^\top$ across different orders. We compare the interaction distributions extracted using different numbers of feature directions. We observe that these distributions remain relatively stable as the number of directions varies, suggesting that the reliable interaction structure is concentrated in a small number of feature directions. Across the displayed model–dataset pairs, the learned directions predominantly capture low-order interactions. Using the interaction generalizability criterion defined in Eq. (8), we find that the learned reliable feature directions retain 82.7–93.8% of generalizable interaction strength and remove 34.3–59.9% of non-generalizable interaction strength. See Appendix F for details.

## 3.2 Problem 1: Are the feature directions transferable to unseen samples?

We investigate whether the learned feature directions remain effective on unseen samples. Specifically, we use these directions to extract interactions from unseen samples and examine whether the extracted interactions remain concentrated at low orders, which were shown above to be more likely to be generalizable. We use the same experimental settings as in the previous subsection, and evaluate the interaction distribution on unseen samples that are not used for learning the feature directions.

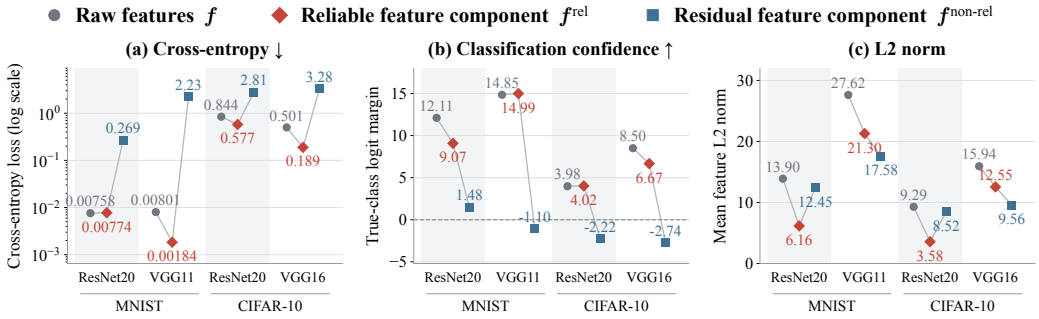



Figure 7: Classification utility and magnitude of feature components. We compare the raw features $\boldsymbol{f}$, the reliable component $\boldsymbol{f}^{\text{rel}}$, and the residual component $\boldsymbol{f}^{\text{non-rel}}$. Panels show (a) mean cross-entropy on a logarithmic axis, (b) classification confidence[12], and (c) mean feature $L_2$ norm.

Figure 6 shows the distribution of interaction strength $\boldsymbol{a}$ across orders $m$ on unseen samples. Across the displayed unseen samples, the learned feature directions remain concentrated at low orders while suppressing high-order interactions. These findings provide evidence that the learned directions can transfer beyond the samples used to learn them, while retaining interaction patterns that are empirically more likely to be generalizable.

### 3.3 Problem 2: Can the reliable feature directions serve as backbone features?

The previous subsections have shown that the learned feature directions primarily encode low-order interactions, which are empirically more likely to be generalizable. This naturally raises the question of whether these directions can serve as the backbone features for classification, *i.e.*, whether a DNN's classification performance can be largely attributed to the features along these learned directions.

Based on the output decomposition in Equation (7), the output component $U^{\text{rel}}(\boldsymbol{x})$ can be roughly interpreted as the classification utility of features $\boldsymbol{f}^{\text{rel}}(\boldsymbol{x}) = \sum_{k\in\mathcal{G}} \boldsymbol{w}_k \boldsymbol{w}_k^\top \boldsymbol{f}(\boldsymbol{x})$ along the learned directions in $\mathcal{G}$. To quantify this utility explicitly, we replace the DNN's intermediate feature $\boldsymbol{f}(\boldsymbol{x})$ with the feature component $\boldsymbol{f}^{\text{rel}}(\boldsymbol{x})$, and evaluate the resulting classification performance. This provides a direct measure of the classification utility attributable to the learned feature directions.

As illustrated in Figure 7, we compare the classification loss and confidence obtained using the raw features $\boldsymbol{f}(\boldsymbol{x})$, the reliable components $\boldsymbol{f}^{\text{rel}}(\boldsymbol{x})$, and the non-reliable components $\boldsymbol{f}^{\text{non-rel}}(\boldsymbol{x})$. Results show that the reliable components achieve performance comparable to that of the raw features and substantially outperform the non-reliable components. The reliable components also exhibit higher classification confidence than the residual components across all four model–dataset pairs. This result indicates that features along the learned directions play a dominant role in classification, supporting their role as backbone representations, whereas the residual components along other directions contribute marginally. Additional comparisons with other rank-three subspaces and paired uncertainty estimates are provided in Appendix G.

## 4 Conclusion and Discussion

This paper tests the hypothesis that DNNs encode generalizable patterns in specific feature directions. Our study is motivated by the finding of (Chen et al., 2024) that the output score of a DNN can be faithfully decomposed into numerical effects attributable to different interactions, some of which are generalizable whereas others are non-generalizable. We identify a set of feature directions that primarily encode generalizable interactions. We find that the interactions encoded by the learned feature directions are predominantly low-order. We further show that these reliable feature directions are transferable to unseen samples. Additionally, the feature components along these directions play a dominant role in classification, compared with the residual components along other directions. Taken together, these findings provide empirical evidence for the existence of specific feature directions that primarily encode generalizable interactions in the evaluated DNNs.

[12]Classification confidence is measured by the mean true-class logit margin $z_y - \max_{c\neq y} z_c$, where $y$ is the true class and $z_c$ is the original unscaled logit for class $c$.

## AI USE STATEMENT

Generative AI tools were used solely to improve the clarity, grammar, and readability of the manuscript. They were not used to generate research ideas, develop theoretical analyses, design methods or experiments, perform data analysis, or interpret results or scientific conclusions. All AI-assisted edits were carefully reviewed by the authors, who take full responsibility for the final content of this work.

## REFERENCES

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

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pp. 1247–1255. PMLR, 2013.

Sercan O. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):6679–6687, 2021.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

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

Barry Becker and Ronny Kohavi. Adult, 1996. URL `https://archive.ics.uci.edu/dataset/2/adult`. Also known as the Census Income dataset.

Akhilan Boopathy, Kevin Liu, Jaedong Hwang, Shu Ge, Asaad Mohammedsaleh, and Ila R Fiete. Model-agnostic measure of generalization difficulty. In *International Conference on Machine Learning*, pp. 2857–2884. PMLR, 2023.

Landon Butler, Abhineet Agarwal, Justin Singh Kang, Yigit Efe Erginbas, Bin Yu, and Kannan Ramchandran. Proxyspex: Inference-efficient interpretability via sparse feature interactions in llms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Anna Bykhovskaya and Vadim Gorin. Canonical correlation analysis: review, 2024. URL `https://arxiv.org/abs/2411.15625`.

Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Exploring neural networks with activation atlases. *Distill.*, 2019.

Lu Chen, Siyu Lou, Benhao Huang, and Quanshi Zhang. Defining and extracting generalizable interaction primitives from dnns. In *International Conference on Learning Representations*, 2024.

Xu Cheng, Lei Cheng, Zhaoran Peng, Yang Xu, Tian Han, and Quanshi Zhang. Layerwise change of knowledge in neural networks. In *International Conference on Machine Learning*, pp. 8038–8059. PMLR, 2024.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Luciano Dyballa, Evan Gerritz, and Steven W Zucker. A separability-based approach to quantifying generalization: which layer is best? *arXiv preprint arXiv:2405.01524*, 2024.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18932–18943, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Yuxuan He, Junpeng Zhang, Lei Cheng, Hongyuan Zhang, and Quanshi Zhang. Technical report: Quantifying and analyzing the generalization power of a dnn, 2025. URL https://arxiv.org/abs/2505.06993.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936. doi: 10.1093/biomet/28.3-4.321. URL https://www.jstor.org/stable/2333955.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020. URL https://arxiv.org/abs/2012.06678.

Justin S Kang, Yigit E Erginbas, Landon Butler, Ramtin Pedarsani, and Kannan Ramchandran. Learning to understand: Identifying interactions via the möbius transform. *Advances in Neural Information Processing Systems*, 37:46160–46202, 2024.

Justin Singh Kang, Landon Butler, Abhineet Agarwal, Yigit Efe Erginbas, Ramtin Pedarsani, Bin Yu, and Kannan Ramchandran. Spex: Scaling feature interaction explanations for llms. In *Forty-second International Conference on Machine Learning*, 2025.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1106–1114, 2012.

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.

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

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Leonardo Petrini, Francesco Cagnetta, Eric Vanden-Eijnden, and Matthieu Wyart. Learning sparse features can lead to overfitting in neural networks. *Advances in Neural Information Processing Systems*, 35:9403–9416, 2022.

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

Qihan Ren, Jiayang Gao, Wen Shen, and Quanshi Zhang. Where We Have Arrived in Proving the Emergence of Sparse Interaction Primitives in DNNs. In *The Twelfth International Conference on Learning Representations*, 2024a.

Qihan Ren, Junpeng Zhang, Yang Xu, Yue Xin, Dongrui Liu, and Quanshi Zhang. Towards the dynamics of a DNN learning symbolic interactions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=dIHXwKjXRE.

Wen Shen, Lei Cheng, Yuxiao Yang, Mingjie Li, and Quanshi Zhang. Can the inference logic of large language models be disentangled into symbolic concepts?, 2023.

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

SymTrust AI. Symtrust AI. https://www.symtrustai.com/en/, 2026. Accessed: 2026-05-05.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16133–16142, 2023.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Hao Zhang, Sen Li, Yinchao Ma, Mingjie Li, Yichen Xie, and Quanshi Zhang. Interpreting and Boosting Dropout from a Game-Theoretic View. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Jacdvfjicf7.

Huilin Zhou, Hao Zhang, Huiqi Deng, Dongrui Liu, Wen Shen, Shih-Han Chan, and Quanshi Zhang. Explaining generalization power of a dnn using interactive concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17105–17113, 2024.

# A Related work

**Mechanistic interpretability and interaction-based explanations.** Mechanistic interpretability aims to explain how DNNs implement inference through interpretable computational units and their compositions. Existing studies have analyzed individual neurons, activation patterns, and circuits formed by interacting features and connections (Bau et al., 2017; Olah et al., 2018; Carter et al., 2019; Olah et al., 2020; Mu & Andreas, 2020). These methods provide important insights into the internal mechanisms of DNNs, but they usually explain representations in terms of local units or manually identified circuits. A complementary line of work explains model predictions through interaction effects among input variables. It shows that the output score of a DNN can often be faithfully approximated by a sparse set of AND-OR interactions, some of which are consistently shared across samples and even across independently trained models (Ren et al., 2023; Li & Zhang, 2023; Chen et al., 2024; Ren et al., 2024a). This interaction perspective provides a verifiable description of inference patterns, but most existing analyses remain at the output level. In contrast, our work studies how these interactions are encoded in intermediate representations. We treat feature directions that encode specific interaction patterns as primitive explanatory units, thereby connecting output-level interaction explanations with the geometry of the feature space.

**Feature-space views of DNN generalization.** A separate line of research studies generalization through the geometry and quality of learned representations, including sparse features, task difficulty, and layerwise separability (Petrini et al., 2022; Boopathy et al., 2023; Dyballa et al., 2024). Closely related methods analyze shared structure between representations using canonical correlation analysis (CCA), originally introduced by Hotelling (1936) and later extended to nonlinear neural representations through deep CCA (Andrew et al., 2013). Recent reviews further connect CCA with high-dimensional statistics and random matrix theory (Bykhovskaya & Gorin, 2024). These methods are useful for measuring or learning aligned subspaces across views, models, or modalities. However, they mainly compare feature coordinates or their correlations, and therefore require the alignment problem to be defined directly in feature space. By contrast, interactions define a model-independent space of inference patterns. Two feature directions can be compared by the interactions they encode, even when their coordinate systems are not directly aligned.

Although feature-space analyses quantify representation quality, they do not directly identify the inference primitives that make a representation reliable. Related studies suggest that reliable behavior is often associated with simpler or more stable structures, whereas unstable behavior tends to involve complex interactions, bottlenecks, or noise-sensitive features (Deng et al., 2022; Achille & Soatto, 2018; Ren et al., 2024b). Our work follows this intuition but operates at a finer granularity. We take interactions as the primitive unit of explanation and learn a low-dimensional orthogonal subspace whose directions are explicitly constrained by interaction structure. This allows us to examine whether there exist specific feature directions that primarily encode generalizable interactions and whether residual directions are more likely to encode non-generalizable interactions.

# B Interaction decomposition

## B.1 Masked samples and utility functions

Table A1 summarizes the shared hyper-parameter settings for interaction extraction. The same settings are used for the output-level utility in Section 2.1 and the feature-direction utility in Eq. 14. We also use a unified masking rule to construct all masked samples $\boldsymbol{x}_S$.

**Image data.** For image models, we treat image patches as input variables. Masking a subset of variables means replacing the corresponding image regions by the dataset-wide average pixel value. When interactions are computed on a selected feature map, we use the feature of the fully masked input as $\boldsymbol{f}_\emptyset$.

**Tabular data.** For tabular models, we treat each input field as one input variable. Masking a subset of variables means replacing the corresponding fields with baseline values computed from the training set. Specifically, continuous variables are replaced by their empirical means, and categorical variables are replaced by their most frequent categories. The feature representation of the fully masked input is used as $\boldsymbol{f}_\emptyset$.

**Masking baseline.** Replacing image patches with the dataset-wide average pixel value can produce inputs outside the natural image distribution. This is a limitation of the perturbation-based interaction analysis. The reconstruction results in Appendix C.2 check fidelity over the masked states generated with this baseline; they do not show that the results would be unchanged under a different masking rule.

**Probability of the ground-truth class.** For both image and tabular classification tasks, $p(y = y^{\text{truth}} \mid \boldsymbol{x})$ is the predicted probability of the ground-truth label.

Table A1: Mathematical setting of hyper-parameters for interaction extraction.

| | |
|---|---|
| Output function $v(\cdot)$ | $v(\boldsymbol{x}) = \log \frac{p(y^{\text{truth}}\mid \boldsymbol{x})}{1-p(y^{\text{truth}}\mid \boldsymbol{x})}$ |
| Threshold $\tau$ | $\tau = 0.02\, \mathbb{E}_{\boldsymbol{x}}[\lvert v(\boldsymbol{x}) - v(\boldsymbol{x}_\emptyset)\rvert]$ |
| Baseline value $\boldsymbol{b}$ | Image data: dataset-wide average pixel value |
| | Tabular data: empirical mean or mode |

**Salience threshold.** The threshold $\tau$ in Table A1 scales with the expected utility change between the full and fully masked inputs. This gives the same relative criterion for selecting salient interactions across the evaluated model–dataset pairs. The experiments in Appendix C.2 assess reconstruction with this fixed rule; sensitivity to other thresholds is outside the scope of those results.

### B.2 Universal matching property

**Theorem B.1.** *Given an input sample $\boldsymbol{x}$, if the interaction weights are defined by $I_S^{and} = \sum_{T\subseteq S}(-1)^{|S|-|T|}\cdot u_T^{and}$ and $I_S^{or} = -\sum_{T\subseteq S}(-1)^{|S|-|T|}\cdot u_{N\setminus T}^{or}$ for all $S \subseteq N$, then:*

$$\forall\, T \subseteq N, \quad v(\boldsymbol{x}_T) = h(\boldsymbol{x}_T),$$

*where $\boldsymbol{x}_T$ denotes the masked sample that keeps variables in $T$ and masks variables in $N\setminus T$. The output $v(\boldsymbol{x}_T)$ is decomposed as $u_T^{and} = 0.5\cdot v(\boldsymbol{x}_T) + \gamma_T$ and $u_T^{or} = 0.5\cdot v(\boldsymbol{x}_T) - \gamma_T$, with learnable parameters $\{\gamma_T\}$.*

*Proof.* **(1) Universal matching property of AND interactions.**

We show that the AND component on all masked samples $\{\boldsymbol{x}_S : S \subseteq N\}$ is exactly represented by AND interactions: $\forall \emptyset \neq S \subseteq N, u_{\text{and}}(\boldsymbol{x}_S) = \sum_{\emptyset\neq T\subseteq S} I_T^{\text{and}} + v(\boldsymbol{x}_\emptyset)$.

Specifically, the AND interaction is defined as

$$I_T^{\text{and}} = \sum\nolimits_{L\subseteq T}(-1)^{|T|-|L|}u_L^{\text{and}}.$$

To compute the sum of AND interactions, we exchange the order of summation:

$$\begin{aligned}
&\sum\nolimits_{\emptyset\neq T\subseteq S} I_T^{\text{and}} \\
=&\sum\nolimits_{\emptyset\neq T\subseteq S}\sum\nolimits_{L\subseteq T}(-1)^{|T|-|L|}u_L^{\text{and}} \\
=&\sum\nolimits_{L\subseteq S}\sum\nolimits_{T:L\subseteq T\subseteq S}(-1)^{|T|-|L|}u_L^{\text{and}} - u_\emptyset^{\text{and}} \\
=&u_S^{\text{and}} + \sum\nolimits_{L\subseteq S, L\neq S} u_L^{\text{and}}\cdot \underbrace{\sum\nolimits_{m=0}^{|S|-|L|} C_{|S|-|L|}^{m}(-1)^m}_{=0} - u_\emptyset^{\text{and}} \\
=&u_S^{\text{and}} - u_\emptyset^{\text{and}} \\
=&u_S^{\text{and}} - v(\boldsymbol{x}_\emptyset).
\end{aligned}$$

This gives $\forall \emptyset \neq S \subseteq N, u_{\text{and}}(\boldsymbol{x}_S) = \sum_{\emptyset\neq T\subseteq S} I_T^{\text{and}} + v(\boldsymbol{x}_\emptyset)$.

**(2) Universal matching property of OR interactions.**

Using the OR definition, we similarly derive $\forall S \subseteq N, u_S^{\text{or}} = \sum_{T:T\cap S\neq\emptyset} I_T^{\text{or}}$, with $u_\emptyset^{\text{or}} = 0$. The OR interaction is

$$I_S^{\text{or}} = -\sum\nolimits_{L\subseteq T}(-1)^{|T|-|L|}u_{N\setminus L}^{\text{or}}.$$

By exchanging the order of summation and grouping terms by the choice of $L \subseteq N$, we obtain

$$\begin{aligned}\sum\nolimits_{T:T\cap S\neq\emptyset} I_T^{\text{or}} &= \sum\nolimits_{T:T\cap S\neq\emptyset}\left[-\sum\nolimits_{L\subseteq T}(-1)^{|T|-|L|}u_{N\setminus L}^{\text{or}}\right]\\ &= -\sum\nolimits_{L\subseteq N}\sum\nolimits_{T:T\cap S\neq\emptyset, T\supseteq L}(-1)^{|T|-|L|}u_{N\setminus L}^{\text{or}}\\ &= u_S^{\text{or}}.\end{aligned}$$

**(3) Universal matching property of AND-OR interactions.**

Combining the two identities,

$$v(\boldsymbol{x}_S) = u_S^{\text{and}} + u_S^{\text{or}} = v(\boldsymbol{x}_\emptyset) + \sum_{\emptyset\neq T\subseteq S} I_T^{\text{and}} + \sum_{T:T\cap S\neq\emptyset} I_T^{\text{or}},$$

which establishes universal matching for AND-OR interactions. □

**Truncation error.** Theorem B.1 concerns the full set of interactions. Retaining only interactions whose absolute effects exceed $\tau$ introduces a reconstruction error. Figure 8(b) shows that this error is small for the evaluated models and masked states; exact matching does not follow for the truncated model.

### B.3 Sparse interaction conditions

Ren et al. (2024a) identify three common conditions under which interactions become sparse.

**Condition 1.** *The DNN does not encode interactions above the $M$-th order: $\forall\, S \in \{S \subseteq N \mid |S| \geq M+1\},\ I_S^{and} \approx 0$.*

Condition 1 states that the DNN avoids extremely high-order interactions, which typically correspond to overly complex and overfitted patterns.

**Condition 2.** *Let $\bar{u}^{(k)} \overset{\text{def}}{=} \mathbb{E}_{|S|=k}[v(\boldsymbol{x}_S) - v(\boldsymbol{x}_\emptyset)]$ denote the average network output over all masked samples $\boldsymbol{x}_S$ with $k$ unmasked input variables. This average output monotonically increases with $k$: $\forall\, k' \leq k,\ \bar{u}^{(k')} \leq \bar{u}^{(k)}$.*

Condition 2 states that a well-trained DNN tends to produce higher confidence when more input variables remain unmasked.

**Condition 3.** *Given the average network output $\bar{u}^{(k)}$ for samples with $k$ unmasked input variables, the average network output for samples with $k' \leq k$ unmasked input variables has a polynomial lower bound: $\forall\, k' \leq k,\ \bar{u}^{(k')} \geq (\frac{k'}{k})^p\, \bar{u}^{(k)}$, where $p > 0$ is a constant.*

Condition 3 states that the model confidence does not collapse under masking; the model can still use local information from the remaining unmasked variables.

**Relation to the empirical results.** The rapid interaction-strength decay in Figure 8(a) is consistent with sparse interactions in the evaluated settings. The reconstruction result in Figure 8(b) checks whether the retained interactions approximate outputs on masked states. Neither panel directly verifies the monotonicity or polynomial lower-bound assumptions in Conditions 2 and 3.

## C Interaction extraction and validation

### C.1 Extracting sparse AND-OR interactions

Recent studies (Li & Zhang, 2023; Chen et al., 2024) introduced a method to jointly extract AND interactions $I_S^{\text{and}}$ and OR interactions $I_S^{\text{or}}$. For a masked sample $\boldsymbol{x}_T$, the output $v(\boldsymbol{x}_T)$ is decomposed as

$$v(\boldsymbol{x}_T) = u_T^{\text{and}} + u_T^{\text{or}},$$

where $u_T^{\text{and}}$ and $u_T^{\text{or}}$ are the AND and OR contributions. The decomposition is

$$u_T^{\text{and}} = 0.5 \cdot v(\boldsymbol{x}_T) + \gamma_T, \quad u_T^{\text{or}} = 0.5 \cdot v(\boldsymbol{x}_T) - \gamma_T,$$

where $\{\gamma_T : T \subseteq N\}$ are learnable decomposition parameters. By Theorem B.1, interactions are computed as

$$I_S^{\text{and}} = \sum_{T \subseteq S} (-1)^{|S|-|T|} u_T^{\text{and}}, \quad I_S^{\text{or}} = -\sum_{T \subseteq S} (-1)^{|S|-|T|} u_{N \setminus T}^{\text{or}}.$$

To promote sparsity, $\{\gamma_T\}$ are optimized via a LASSO-like objective:

$$\min_{\{\gamma_T\}} \sum_{S \subseteq N} \left| I_S^{\text{and}} \right| + \left| I_S^{\text{or}} \right|. \tag{12}$$

**Output perturbations.** Small output perturbations $\delta_S$ may affect extracted interactions, especially at high orders. Following Li & Zhang (2023), we incorporate $\delta_S$ into the decomposition:

$$u_T^{\text{and}} = 0.5 \cdot (v(\boldsymbol{x}_T) - \delta_T) + \gamma_T, \quad u_T^{\text{or}} = 0.5 \cdot (v(\boldsymbol{x}_T) - \delta_T) - \gamma_T.$$

Here, $\{\delta_T\}$ and $\{\gamma_T\}$ are jointly optimized under Eq. (12), with $\delta_T \in [-\zeta, \zeta]$ and $\zeta = 0.02 \cdot \mathbb{E}_{\boldsymbol{x}}|v(\boldsymbol{x}) - v(\boldsymbol{x}_\emptyset)|$.

**Algorithm 1** Extraction of AND-OR Interactions

**Require:** Input sample $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$, DNN output function $v(\boldsymbol{x})$, threshold $\tau$, noise bound $\zeta$
**Ensure:** AND interactions $I_S^{\text{and}}$, OR interactions $I_S^{\text{or}}$, salient interaction sets $\Omega^{\text{and}}$, $\Omega^{\text{or}}$

```
Initialize learnable parameters {γ_T} and {δ_T} for all T ⊆ N
ζ ← 0.02 · E_x|v(x) − v(x_∅)|
while not converged do
    for each subset T ⊆ N do
        Compute masked sample x_T
        u_T^and ← 0.5 · (v(x_T) − δ_T) + γ_T
        u_T^or ← 0.5 · (v(x_T) − δ_T) − γ_T
    end for
    for each subset S ⊆ N do
        I_S^and ← Σ_{T⊆S} (−1)^{|S|−|T|} u_T^and
        I_S^or ← − Σ_{T⊆S} (−1)^{|S|−|T|} u_{N\T}^or
    end for
    Optimize {γ_T} and {δ_T} by minimizing Σ_{S⊆N} |I_S^and| + |I_S^or|
    Constrain δ_T ∈ [−ζ, ζ] for all T ⊆ N
end while
Ω^and ← {S ⊆ N : |I_S^and| > τ}
Ω^or ← {S ⊆ N : |I_S^or| > τ}
return I_S^and, I_S^or, Ω^and, Ω^or
```

### C.2 Empirical validation of sparsity and universal matching

**In this section, we empirically examine the faithfulness of interactions.** Specifically, the following two experiments are designed to validate the sparsity property and the universal matching property. We follow the interaction-computation setting in Appendix B.1: image patches or tabular fields are treated as input variables, masked samples are generated with the baseline values in Table A1, and the scalar utility is the log-odds of the predicted probability of the ground-truth class.

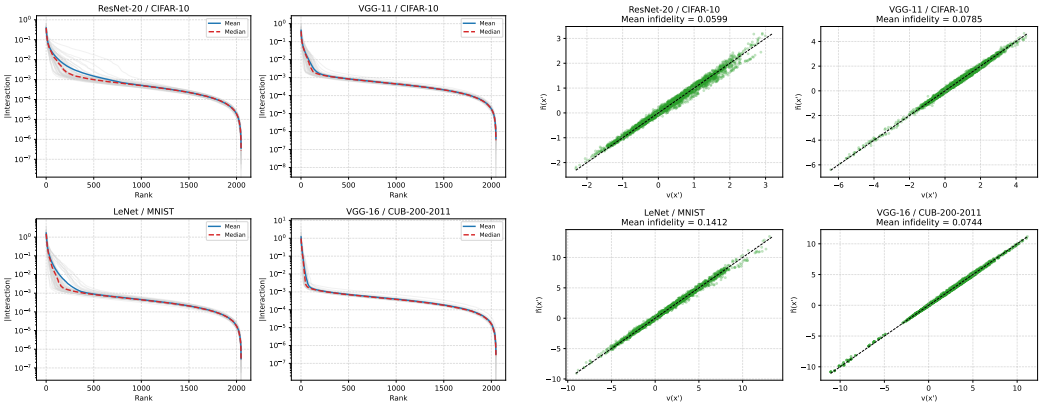
**Experiment 1.** As defined by Eq. (2), the AND-OR logical model consists of a set of AND interactions and a set of OR interactions, denoted by $\Omega^{\text{and}}$ and $\Omega^{\text{or}}$. The goal of this experiment is to validate the sparsity property, *i.e.*, both $\Omega^{\text{and}}$ and $\Omega^{\text{or}}$ are small relative to all possible coalitions. Following Chen et al. (2024), we select salient interactions by $\Omega^{\text{and}} = \{S \subseteq N : |I_S^{\text{and}}| > \tau\}$ and $\Omega^{\text{or}} = \{S \subseteq N : |I_S^{\text{or}}| > \tau\}$, where $\tau$ is defined in Table A1. For each evaluated sample, we compute $I_S^{\text{and}}$ and $I_S^{\text{or}}$ over all $2^n$ subsets and sort their absolute strengths in descending order. Figure 8(a)

shows that these sorted strengths decay rapidly across different models and datasets, indicating that only a small number of interactions remain salient after thresholding.

**Experiment 2.** To validate the universal matching property, we evaluate whether a small-size logical model reconstructed from salient interactions can approximate the DNN output on masked states. We use the following infidelity metric:

$$\epsilon_{\text{inf}}(\boldsymbol{x}') = \left| v(\boldsymbol{x}') - \hat{h}(\boldsymbol{x}') \right|,$$
$$\hat{h}(\boldsymbol{x}') = \sum\nolimits_{S\in\Omega^{\text{and}}} I_S^{\text{and}} \cdot \mathbb{1}_{\text{and}}(S|\boldsymbol{x}') + \sum\nolimits_{S\in\Omega^{\text{or}}} I_S^{\text{or}} \cdot \mathbb{1}_{\text{or}}(S|\boldsymbol{x}') + v(\boldsymbol{x}_\emptyset), \tag{13}$$

where $\boldsymbol{x}'$ denotes a masked state of the same input sample, and $\Omega^{\text{and}}$ and $\Omega^{\text{or}}$ are selected using the same threshold $\tau$. We compute the matching error in Eq. (13) over the masked states. Figure 8(b) shows that the reconstructed outputs closely match the DNN outputs across masked states, and the mean infidelity remains small across models. This supports the use of AND-OR interactions as faithful explanations.




(a) Sparsity of salient interactions (b) Universal matching of the logical model

Figure 8: Empirical validation of interaction faithfulness. (a) The magnitudes of AND-OR interactions decay rapidly after sorting, indicating that only a small fraction of all possible coalitions have non-negligible effects. (b) Logical models reconstructed from salient interactions accurately approximate DNN outputs over masked states, supporting the universal matching property.

### C.3 Case studies of extracted interactions on ImageNet

Figure 9 shows AND and OR interactions extracted from VGG-16 on four ImageNet validation samples. For each image, we select a small set of patches and rank the extracted interactions by their effects on the scalar utility. These examples provide a qualitative check that the extracted coalitions can be related to image regions; they do not establish that the corresponding feature directions are reliable across models or samples.

### C.4 Computational efficiency of interaction extraction

A key limitation of interaction-based explanation is its computational cost. By the universal matching property in Theorem B.1, exact extraction of all AND-OR interaction effects for a sample with $n$ input variables requires evaluating the model on all $2^n$ masked states. This exhaustive computation provides a theoretically complete description of the model output over the masked sample space, but its cost grows exponentially with the number of variables. Therefore, direct exhaustive extraction is mainly suitable when the input can be represented by a moderate number of variables, such as image patches or tabular fields, and becomes challenging when the number of candidate variables is very large.

In practice, this cost can be partially mitigated by reducing the effective number of variables before interaction extraction. Prior studies have developed engineering pipelines for interaction analysis that first identify salient regions or variables and then compute interactions over the selected units (Shen

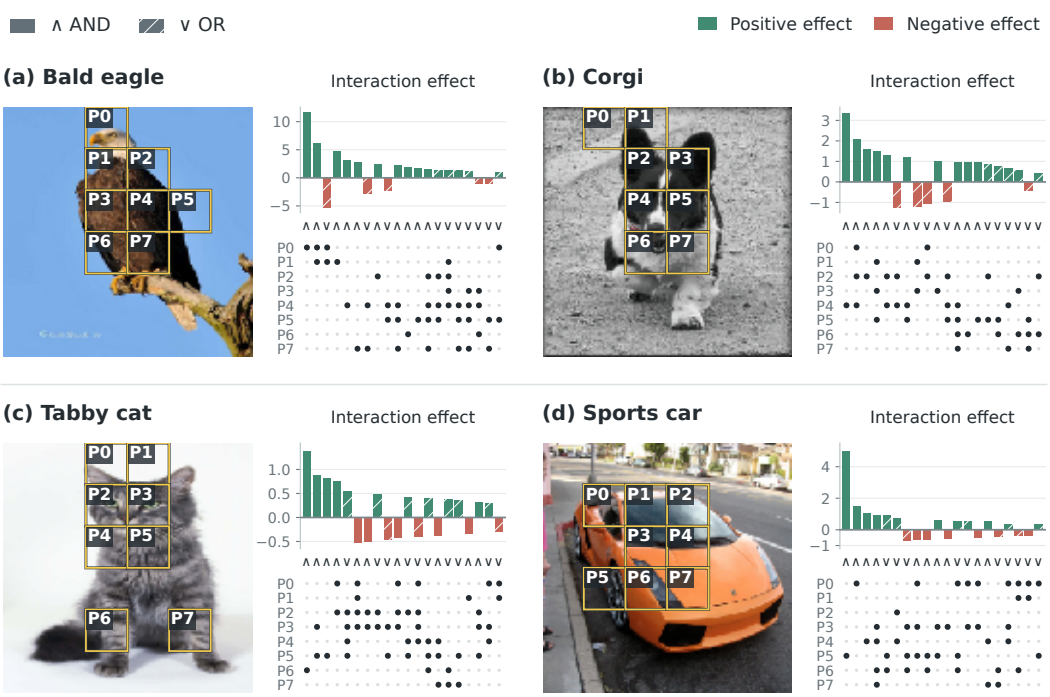



Figure 9: Case studies of AND and OR interactions extracted from VGG-16 on four ImageNet validation samples. Each sample shows the ten strongest AND and ten strongest OR interactions, jointly ordered by absolute effect. Each column corresponds to one interaction, with filled dots indicating its constituent patches. AND and OR interactions are marked by $\wedge$ (solid bars) and $\vee$ (hatched bars), respectively. Green and red indicate positive and negative effects.

et al., 2023; Cheng et al., 2024; Chen et al., 2024). This is consistent with the image and tabular settings in this paper, where variables are defined as image patches or tabular fields rather than raw pixels or all low-level activations. Empirically, using a moderate number of selected variables is often sufficient for practical interaction analysis, because salient evidence is usually concentrated in a small number of regions or fields.

Another direction is to use approximate extraction algorithms. Recent methods approximate interaction utilities without enumerating all masked states (Kang et al., 2024; 2025; Butler et al., 2025). For example, SPEX-style estimators can approximate interaction effects among many input variables within a reasonable time budget. These approximation methods are complementary to our study: this paper focuses on defining reliable feature directions and analyzing the distribution of generalizable and non-generalizable interactions, rather than on designing a new approximation algorithm.

Finally, the theoretical claims in this paper do not rely on a particular acceleration technique. The decomposition and generalizability analyses characterize the interaction structure that would be obtained from the model outputs over the masked sample space. Approximate or dimension-reduced extraction changes the practical estimator used to recover this structure, but not the conceptual distinction between generalizable and non-generalizable interactions. For example, interaction-based explanation has been used in an industrial setting to explain, evaluate, and improve DNN behavior (SymTrust AI, 2026).

# D Feature-direction interaction analysis

## D.1 Proof of Theorem 2.1

*Proof.* Let $F(\boldsymbol{f})$ denote the scalar output of the downstream part of the DNN when the intermediate-layer feature is set to $\boldsymbol{f}$. Thus, $v(\boldsymbol{x}) = F(\boldsymbol{f}(\boldsymbol{x}))$ and $v_{\boldsymbol{f}_\emptyset} = F(\boldsymbol{f}_\emptyset)$. Applying the first-order Taylor

expansion of $F$ around the reference feature $\boldsymbol{f}_\emptyset$ gives

$$F(\boldsymbol{f}(\boldsymbol{x})) = F(\boldsymbol{f}_\emptyset) + \boldsymbol{g}^\top \big(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}_\emptyset\big) + O\big(\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}_\emptyset\|_2^2\big),$$

where $\boldsymbol{g} = \nabla_{\boldsymbol{f}} F(\boldsymbol{f})|_{\boldsymbol{f}=\boldsymbol{f}_\emptyset}$. This proves the first line of Theorem 2.1.

Because $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$ are mutually orthogonal unit vectors in the $K$-dimensional feature space, they form an orthonormal basis. Therefore,

$$\boldsymbol{I} = \sum_{k=1}^{K} \boldsymbol{w}_k \boldsymbol{w}_k^\top, \qquad \boldsymbol{f}(\boldsymbol{x}) = \sum_{k=1}^{K} \boldsymbol{w}_k \boldsymbol{w}_k^\top \boldsymbol{f}(\boldsymbol{x}).$$

Substituting this identity into the linear term of the Taylor expansion yields

$$\boldsymbol{g}^\top \boldsymbol{f}(\boldsymbol{x}) = \sum_{k=1}^{K} \boldsymbol{g}^\top [\boldsymbol{w}_k \boldsymbol{w}_k^\top] \boldsymbol{f}(\boldsymbol{x}).$$

We define each directional output component as

$$u_k(\boldsymbol{x}) = \boldsymbol{g}^\top [\boldsymbol{w}_k \boldsymbol{w}_k^\top] \boldsymbol{f}(\boldsymbol{x}).$$

The remaining terms in the Taylor expansion do not depend on the decomposition of $\boldsymbol{f}(\boldsymbol{x})$ into feature directions, and can be collected as

$$c = v_{\boldsymbol{f}_\emptyset} - \boldsymbol{g}^\top \boldsymbol{f}_\emptyset + O\big(\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}_\emptyset\|_2^2\big).$$

Combining the above equations gives

$$v(\boldsymbol{x}) = \sum_{k=1}^{K} u_k(\boldsymbol{x}) + c = \sum_{k=1}^{K} \boldsymbol{g}^\top [\boldsymbol{w}_k \boldsymbol{w}_k^\top] \boldsymbol{f}(\boldsymbol{x}) + c,$$

which is the claimed decomposition.

The remainder term reflects the local approximation error of the first-order Taylor expansion around the reference feature $\boldsymbol{f}_\emptyset$. We do not attribute this term to any specific feature direction, because the directional components $u_k(\boldsymbol{x})$ are defined by the linear term $\boldsymbol{g}^\top \boldsymbol{f}(\boldsymbol{x})$. Thus, the decomposition should be understood as separating the first-order directional contributions while collecting the higher-order residual into $c$. This treatment does not affect the subsequent interaction analysis, which is applied to the scalar utility induced by each direction $\boldsymbol{w}_k$. □

### D.2 Discussion and case study of the Taylor remainder

The decomposition in Theorem 2.1 defines a first-order directional attribution. Its higher-order remainder depends on the masked input and on the local nonlinearity around the reference feature $\boldsymbol{f}_\emptyset$. Thus, $c$ collects a sample-dependent residual and should not be interpreted as a constant shared exactly across masked states. Our interaction analysis instead applies to the scalar first-order utility induced by each feature direction; it does not require the directional utilities to reconstruct the nonlinear DNN output exactly.

We examine the approximation at the global-average-pooling layer of a ResNet-20 trained on CIFAR-10 (seed 1029), using 50 images of class 2. Each image has 10 players and 1,024 masked states employed for interaction extraction. For each masked state $\boldsymbol{x}_S$, we compare the true log-odds utility with its first-order approximation and measure the normalized residual

$$r(\boldsymbol{x}_S) = \frac{\big|v(\boldsymbol{x}_S) - \big[v_{\boldsymbol{f}_\emptyset} + \boldsymbol{g}^\top \big(\boldsymbol{f}(\boldsymbol{x}_S) - \boldsymbol{f}_\emptyset\big)\big]\big|}{\big|v(\boldsymbol{x}_S) - v_{\boldsymbol{f}_\emptyset}\big|},$$

defined for states with $|v(\boldsymbol{x}_S) - v_{\boldsymbol{f}_\emptyset}| > \epsilon$, where $\epsilon = 10^{-12}$. This excludes the 50 fully masked states, leaving 51,150 states for the residual distribution. Figure 10 shows that the linearized utility follows the trend of the true utility, while the residual has a non-negligible tail. This case study supports using the Taylor expansion as a local attribution for examining the feature-direction hypothesis, with approximation error retained as a limitation.

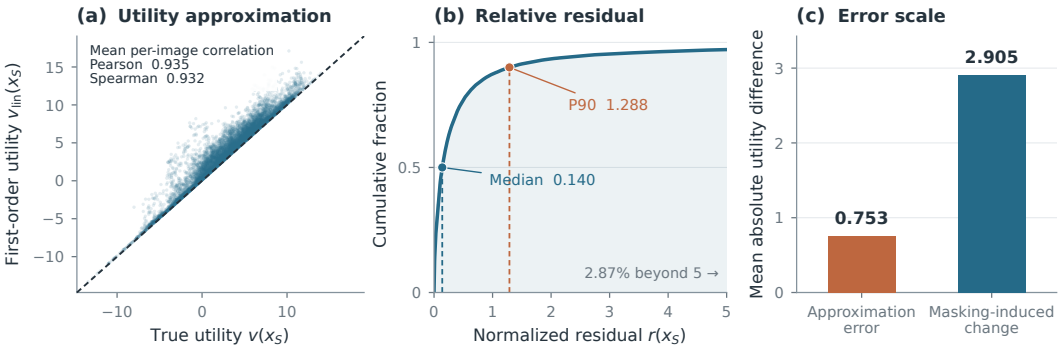



Figure 10: First-order Taylor attribution in a ResNet-20/CIFAR-10 case study. (a) True versus first-order utility, with a dashed identity line. Correlations are averaged over 50 images; 10,000 of the 51,200 masked states are displayed. (b) The empirical cumulative distribution over 51,150 eligible states has median 0.140 and P90 1.288; 2.87% lie beyond the displayed range of 5. (c) Mean absolute approximation error (0.753) and mean absolute utility change from the fully masked reference (2.905), both averaged over all masked states. The ratio of these means is 25.93%.

### D.3 Choice of reference feature

By default, the reference feature $\boldsymbol{f}_\emptyset$ is computed from the fully masked input $\boldsymbol{x}_\emptyset$. This choice is consistent with the output-level interaction decomposition, where the bias term is defined by the model output on the fully masked sample. In practice, one may also use the average feature over all masked states of a given sample as the reference feature. We use the fully masked feature in the experiments unless otherwise specified, because it provides a fixed expansion point for Theorem 2.1 and determines the bias component and sample-dependent remainder collected in $c$ in the output decomposition.

### D.4 Detailed learning algorithm for reliable directions

#### D.4.1 Projected interactions on a single direction

Given a masked state $\boldsymbol{x}_S$, let $\boldsymbol{f}(\boldsymbol{x}_S)$ denote the intermediate-layer feature of the DNN on this masked input. Following Theorem 2.1, a unit feature direction $\boldsymbol{w}$ induces the projected feature component

$$\boldsymbol{f}_S(\boldsymbol{w}) = \boldsymbol{w}\boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{x}_S), \qquad \boldsymbol{f}_S^{\perp}(\boldsymbol{w}) = (\boldsymbol{I} - \boldsymbol{w}\boldsymbol{w}^\top)\boldsymbol{f}(\boldsymbol{x}_S).$$

The scalar output component induced by direction $\boldsymbol{w}$ on the masked state $\boldsymbol{x}_S$ is defined in the same way as the main text:

$$u_S(\boldsymbol{w}) = \boldsymbol{g}^\top \boldsymbol{f}_S(\boldsymbol{w}) = \boldsymbol{g}^\top [\boldsymbol{w}\boldsymbol{w}^\top] \boldsymbol{f}(\boldsymbol{x}_S), \tag{14}$$

where $\boldsymbol{g}$ is the same output-gradient direction used in Theorem 2.1. We compute AND-OR interactions from the scalar utility values $\{u_T(\boldsymbol{w})\}_{T\subseteq N}$ using the same extraction procedure as in Section 2.1. Let $J_S^{\text{and}}(\boldsymbol{w})$ and $J_S^{\text{or}}(\boldsymbol{w})$ denote the interactions extracted from this projected output component. These projected interactions are compared with the original interaction effects $I_S^{\text{and}}$ and $I_S^{\text{or}}$ in the direction-learning loss in Eq. 11. Here, the original interaction effects are extracted from the raw feature utility $\boldsymbol{g}^\top \boldsymbol{f}(\boldsymbol{x}_S)$, so Eq. 11 compares interactions before and after projecting the same feature representation onto $\boldsymbol{w}$.

#### D.4.2 Sequential learning of multiple directions

Suppose that $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{k-1}$ have already been learned. As stated in the main text, we remove the subspace spanned by these directions from the feature space and learn the next direction from the residual feature component:

$$\boldsymbol{f}^{(k-1)}(\boldsymbol{x}_S) \overset{\text{def}}{=} \Big(\boldsymbol{I} - \sum\nolimits_{i=1}^{k-1} \boldsymbol{w}_i \boldsymbol{w}_i^\top\Big) \boldsymbol{f}(\boldsymbol{x}_S). \tag{15}$$

The projected scalar utility for the $k$-th direction is therefore $u_S(\boldsymbol{w}_k) = \boldsymbol{g}^\top[\boldsymbol{w}_k\boldsymbol{w}_k^\top]\boldsymbol{f}^{(k-1)}(\boldsymbol{x}_S)$. In practice, $\boldsymbol{w}_k$ is updated by projected gradient descent. After each gradient step, we remove the component in the span of previously learned directions and renormalize:

$$\tilde{\boldsymbol{w}}_k \leftarrow \boldsymbol{w}_k - \eta\nabla_{\boldsymbol{w}_k}\mathcal{L}(\boldsymbol{w}_k), \qquad \boldsymbol{w}_k \leftarrow \frac{\left(\boldsymbol{I} - \sum_{i=1}^{k-1}\boldsymbol{w}_i\boldsymbol{w}_i^\top\right)\tilde{\boldsymbol{w}}_k}{\left\|\left(\boldsymbol{I} - \sum_{i=1}^{k-1}\boldsymbol{w}_i\boldsymbol{w}_i^\top\right)\tilde{\boldsymbol{w}}_k\right\|_2}. \tag{16}$$

This update ensures that each newly learned direction remains orthogonal to all previously learned directions.

### D.4.3 Pseudo-code

**Algorithm 2** Sequential learning of reliable feature directions

1: **Input:** learning samples $\mathcal{D}_{\text{learn}}$, masked features $\{\boldsymbol{f}(\boldsymbol{x}_S)\}_{S\subseteq N}$, output-gradient direction $\boldsymbol{g}$, number of directions $K$
2: Extract the original interaction effects $I_S^{\text{and}}$ and $I_S^{\text{or}}$ from the raw feature utility $\boldsymbol{g}^\top\boldsymbol{f}(\boldsymbol{x}_S)$
3: **for** $k = 1, \ldots, K$ **do**
4: Initialize a unit vector $\boldsymbol{w}_k$ orthogonal to $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{k-1}$
5: **while** not converged **do**
6: Compute the residual feature component $\boldsymbol{f}^{(k-1)}(\boldsymbol{x}_S)$ using Eq. 15
7: Compute the projected utility $u_S(\boldsymbol{w}_k) = \boldsymbol{g}^\top[\boldsymbol{w}_k\boldsymbol{w}_k^\top]\boldsymbol{f}^{(k-1)}(\boldsymbol{x}_S)$
8: Extract projected interactions $J_S^{\text{and}}(\boldsymbol{w}_k)$ and $J_S^{\text{or}}(\boldsymbol{w}_k)$
9: Compute $\mathcal{L}(\boldsymbol{w}_k)$ using Eq. 11
10: Update $\boldsymbol{w}_k$ by Eq. 16
11: **end while**
12: **end for**
13: **Return:** $\boldsymbol{W}^{\text{rel}} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$

## E Experimental protocols

### E.1 Generalizability evaluation

For the order-wise generalizability analysis in Section 2.2.2, we use separate protocols for models trained from scratch and pretrained ImageNet-scale models. In the controlled trained-from-scratch setting, each target DNN and its corresponding baseline DNN are trained for the same classification task, with the same architecture and training recipe, but on disjoint halves of the original training set. Specifically, we randomly shuffle the training set with a fixed seed, use the first half as the target split and the second half as the baseline split, and evaluate both networks on the original test set. The target network is trained with seed 1029 and the baseline network with seed 1031. For each target model, we select same-class samples from the target split for interaction extraction.

We then extract AND-OR interactions from both the target and baseline DNNs. Interactions are matched by exact coalition-index matching under the same variable partition. For image data, this means the same subset of selected image patches; for tabular data, this means the same subset of input fields. We do not use spatial-shift matching or semantic-equivalence matching. An interaction is counted as generalizable only if the corresponding AND or OR interaction is salient in both the target and baseline DNNs and has the same sign. AND and OR interactions are evaluated separately. The order-wise generalization ratio $\mathcal{R}^{\text{gen}}(m)$ is strength-weighted: for each order $m$, the numerator is the total absolute interaction strength of generalizable target interactions, and the denominator is the total absolute interaction strength of salient target interactions at the same order.

For ImageNet-scale pretrained models, the original training splits, recipes, and random seeds are not controlled. We therefore assess whether salient interactions transfer across pretrained architectures or checkpoints on the same ImageNet validation samples under the same input-variable partition. A coalition is counted as transferable when it is salient in the comparison model and has the same sign. This setting measures cross-model transferability and is distinct from the controlled disjoint-data evaluation above.

Table A2: Experimental scope for the order-wise generalizability analysis. Models trained from scratch use the controlled disjoint-split protocol; ImageNet-scale pretrained models use cross-model transferability on the same validation samples.

| Task | Models | Datasets | Samples | Variables |
|---|---|---|---|---|
| Image classification | ResNet-20/34/50/56, VGG-11/16, AlexNet, LeNet | CIFAR-10, MNIST, Fashion-MNIST, CUB-200-2011 | 50 | 10 patches |
| Tabular classification | FT-Transformer, TabTransformer, TabNet | Census, Commercial | 50 | 10 fields |
| ImageNet classification | EVA-02-L, ViT-H, BEiT-L, ConvNeXtV2 | ImageNet validation set | 50 | 10 patches |

### E.2 Overfitting dynamics

For the overfitting analysis in Section 2.2.2, we track the training loss, testing loss, and the order-wise interaction strength distribution $\boldsymbol{a}$ in Eq. 10 at four training stages. Table A3 lists the image-classification runs and the checkpoints corresponding to these stages.

Table A3: Dynamic-training experiments used to analyze the relation between overfitting and interaction order. Each stage contains exact AND-OR interaction extraction over 10 selected variables, corresponding to $2^{10}$ masked states per sample.

| Model | Dataset | Samples | Stage A | Stage B | Stage C | Stage D |
|---|---|---|---|---|---|---|
| AlexNet | TinyImageNet | 50 | iteration 50 | epoch 32 | epoch 64 | epoch 256 |
| VGG-11 | CIFAR-10 | 50 | iteration 57 | epoch 8 | epoch 32 | epoch 256 |
| VGG-16 | CUB-200-2011 | 50 | iteration 12 | epoch 16 | epoch 64 | epoch 256 |
| ViT | CIFAR-10 | 20 | iteration 39 | epoch 2 | epoch 6 | epoch 100 |

Across these runs, the train-test loss gap grows as training proceeds. For example, the gap increases from approximately 0.002 to 6.407 for AlexNet on TinyImageNet, from approximately -0.000 to 0.848 for VGG-11 on CIFAR-10, from approximately 0.011 to 4.123 for VGG-16 on CUB-200-2011, and from approximately -0.006 to 0.124 for ViT on CIFAR-10. We use the corresponding interaction artifacts at the four timepoints to compare how interaction strength shifts across orders during training.

### E.3 Training and interaction-extraction details

Table A4 reports the main training settings for the image models used in the reported image experiments. For target-baseline generalizability experiments, the same recipe is used for the target and baseline models, except for the disjoint training split and random seed.

For tabular generalizability experiments, FT-Transformer, TabTransformer, and TabNet are trained on Census and Commercial datasets for 100 epochs with batch size 256 and learning rate 0.001 using SGD. Census uses 12 input fields and Commercial uses 10 input fields; in the interaction analysis we use 10 selected fields.

All reported interaction analyses use exact enumeration over the selected variables. In most experiments, 10 variables are selected, yielding $2^{10} = 1024$ masked states for each sample. For image models, variables are selected feature-map patches; for tabular models, variables are input fields. Salient interactions are selected by thresholding the absolute AND and OR interaction effects separately.

Table A4: Training settings for image models used in the reported experiments. VGG models on CIFAR-10 use a cosine learning-rate schedule with five warmup epochs. ImageNet-scale experiments use pretrained checkpoints and do not fine-tune the models.

| Model | Dataset | Epochs | Batch size | Learning rate | Optimizer |
|---|---|---|---|---|---|
| ResNet-20 | CIFAR-10 | 60 | 128 | 0.01 | SGD |
| ResNet-56 | CIFAR-10 | 60 | 128 | 0.01 | SGD |
| VGG-11 | CIFAR-10 | 100 | 128 | 0.01 | SGD |
| VGG-16 | CIFAR-10 | 100 | 128 | 0.01 | SGD |
| AlexNet | MNIST, Fashion-MNIST | 20 | 64 | 0.01 | SGD |
| LeNet | MNIST, Fashion-MNIST | 20 | 128 | 0.01 | SGD |
| ResNet-34/50 | CUB-200-2011 | 100 | 64 | 0.01 | SGD |
| VGG-11/16 | CUB-200-2011 | 100 | 64 | 0.01 | SGD |

### E.4 Learning feature directions

For the experiments on learning reliable feature directions, we first select an intermediate layer and compute masked features for the learning samples. We then optimize the directions using Eq. 11, with $M = 2$ as the empirical order threshold based on the results in Figure 3 and $\lambda = 20.0$ as the high-order suppression coefficient. The first direction is initialized from the mean full-coalition feature over the learning samples and then normalized. When learning multiple directions, each new direction is orthogonalized against previously learned directions by Gram-Schmidt projection. Table A5 reports the selected feature layer and downstream target layer for the trained-from-scratch models. Figures 5 and 6 additionally include BEiT-L (`beit_large_patch16_512`) and ConvNeXtV2-Huge (`convnextv2_huge_512`) on ImageNet, evaluated with the same projection-based analysis on frozen pretrained representations. For BEiT-L, interaction masking starts from `block5_tokens`, corresponding to the output of the sixth Transformer block (0-indexed), and the target feature is `pre_logits`; we use the `hist-peak` baseline with coalition batch size 128. For ConvNeXtV2-Huge, interaction masking starts from `stage2`, corresponding to the output of the second stage (`stages[1]`), and the target feature is `pre_logits`; we use the zero baseline with coalition batch size 64.

Table A5: Settings for learning reliable feature directions in the trained-from-scratch models. The base setting learns one direction, while the multi-direction analysis evaluates up to three sequentially learned directions. The directions are learned class-specifically: class 2 for CIFAR-10, class 3 for MNIST, and class 0 for CUB-200-2011.

| Model | Dataset | Feature layer | Target layer | Directions |
|---|---|---|---|---|
| ResNet-20 | CIFAR-10 | resnet20_layer1 | resnet20_global_pool | 1–3 |
| ResNet-34 | CUB-200-2011 | layer1 | resnet34_global_pool | 1–3 |
| LeNet | MNIST | first_relu | lenet_fc2 | 1–3 |
| VGG-16-BN | CUB-200-2011 | second_maxpooling | vgg16_bn_first_fc | 1–3 |

For transfer to unseen samples, the directions are learned on the learning samples in Table A5 and then applied without re-optimization to held-out samples. We recompute the projected interaction distribution on these samples and evaluate whether the interactions extracted from the reliable component remain concentrated at low orders.

For classification utility, we intervene on the intermediate representation at the same target layer where the directions are learned. Specifically, we compute the raw feature $\boldsymbol{f}(\boldsymbol{x})$, the reliable component $\boldsymbol{f}^{\text{rel}}(\boldsymbol{x})$ by projecting the feature onto the span of the learned directions, and the residual component $\boldsymbol{f}^{\text{non-rel}}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}^{\text{rel}}(\boldsymbol{x})$. Each component is fed into the fixed downstream layers to evaluate the resulting cross-entropy loss and feature norm. This intervention is used only for evaluation, not for training.

## F CROSS-SEED EVALUATION OF RELIABLE FEATURE DIRECTIONS

To further examine whether the learned reliable feature directions preserve generalizable interactions, we follow the criterion used in Figure 3 and examine whether salient interactions encoded by the target DNN are also encoded by a baseline DNN with consistent signs. Here, the baseline DNN is trained with a different random seed. On all direction-learning samples, the three-direction subspace retains 82.7–93.8% of generalizable interaction strength with the original sign, while the retained non-generalizable interaction strength is reduced by 34.3–59.9% relative to its full raw strength. The order-wise results in Figure 11 show strong preservation of first-order generalizable interactions and greater removal of non-generalizable strength at higher orders.




Figure 11: Reliable feature directions largely preserve low-order generalizable interaction strength, with reduced high-order non-generalizable interaction strength after projection. Retained strength is computed over salient projected interactions.

## G ADDITIONAL ANALYSES OF CLASSIFICATION UTILITY

We supplement Figure 7 using the same target-class cohorts: MNIST class 3 ($n = 1{,}010$) and CIFAR-10 class 2 ($n = 1{,}000$). W3 contains three directions learned from 40 interaction examples. All projections are unscaled and evaluated by the original frozen classifier; confidence denotes mean true-class logit margin. These are class-conditional utility comparisons (Figures 12 and 13).

**Same-rank controls.** W3 has lower mean CE and higher mean margin than PCA3, Probe3, and Random3 in all four pairs.

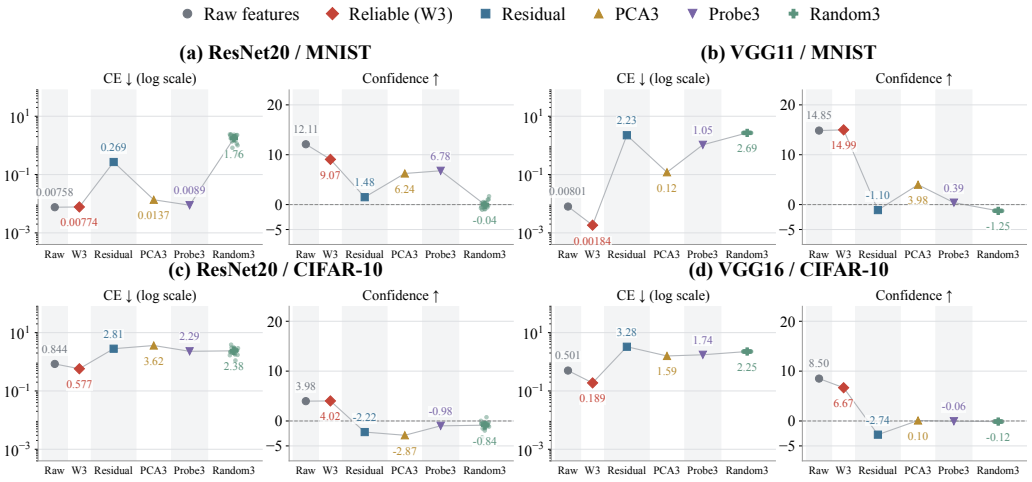



Figure 12: Same-rank controls: mean CE and true-class logit margin. Random3 shows 20 fixed bases and their mean; the spread is between bases, not a confidence interval. Raw and residual features are decomposition references.

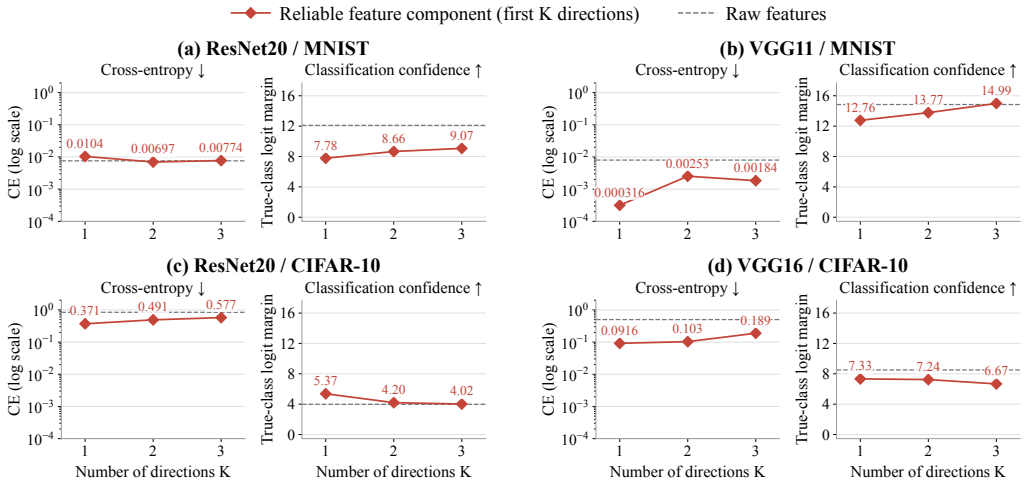



Figure 13: Effect of retaining the first $K = 1, 2, 3$ directions of the same learned sequence, without retraining. Dashed lines indicate Raw. Margin increases with $K$ on MNIST and decreases on CIFAR-10; CE does not improve monotonically.

**Control construction.** PCA3 and Probe3 use 2,000 balanced training examples; Probe3 takes the leading three right singular vectors of a multiclass logistic-regression coefficient matrix. Random3 uses 20 fixed orthonormal bases. The controls match W3's rank, with different fitting data and objectives.

**Paired comparisons.** Table A6 retains all comparisons, including intervals spanning zero. In particular, ResNet20/MNIST CE differences relative to Raw and Probe3 include zero, as does the ResNet20/CIFAR-10 margin difference relative to Raw. W3 has lower mean margin than Raw for ResNet20/MNIST and VGG16/CIFAR-10.

**Probability prediction quality and calibration.** Brier is the mean sum of squared class-probability errors; ECE uses 15 equal-width top-label confidence bins. Both use unscaled logits without temperature scaling. Brier decreases in all four pairs; ECE decreases in three and increases for ResNet20/MNIST. Brier improvement alone does not establish improved calibration.

Table A6: Paired differences (W3 minus comparator); negative CE and positive margin favor W3. Brackets show pointwise 95% percentile bootstrap intervals from 5,000 paired resamples, conditional on frozen directions. Random3 averages the 20 bases per image. Intervals capture test-sample variation, not retraining or between-basis uncertainty; no multiplicity correction is applied.

| Model / dataset | Comparator | ΔCE [95% CI] | Δmargin [95% CI] |
|---|---|---|---|
| ResNet20 / MNIST | Raw | $+0.00015$ $[-0.00315, +0.00290]$ | $-3.034$ $[-3.118, -2.953]$ |
| | Residual | $-0.26175$ $[-0.27118, -0.25247]$ | $+7.588$ $[+7.485, +7.689]$ |
| | PCA3 | $-0.00592$ $[-0.01087, -0.00182]$ | $+2.831$ $[+2.775, +2.889]$ |
| | Probe3 | $-0.00116$ $[-0.00382, +0.00133]$ | $+2.293$ $[+2.227, +2.355]$ |
| | Random3 | $-1.74884$ $[-1.75297, -1.74462]$ | $+9.111$ $[+9.000, +9.218]$ |
| VGG11 / MNIST | Raw | $-0.00617$ $[-0.01497, -0.00044]$ | $+0.143$ $[+0.106, +0.182]$ |
| | Residual | $-2.23083$ $[-2.24506, -2.21710]$ | $+16.090$ $[+15.899, +16.292]$ |
| | PCA3 | $-0.11811$ $[-0.14236, -0.09750]$ | $+11.008$ $[+10.897, +11.126]$ |
| | Probe3 | $-1.04544$ $[-1.07580, -1.01575]$ | $+14.599$ $[+14.432, +14.767]$ |
| | Random3 | $-2.68512$ $[-2.68675, -2.68232]$ | $+16.247$ $[+16.050, +16.447]$ |
| ResNet20 / CIFAR-10 | Raw | $-0.26693$ $[-0.33077, -0.20791]$ | $+0.043$ $[-0.093, +0.184]$ |
| | Residual | $-2.23244$ $[-2.34332, -2.12227]$ | $+6.243$ $[+5.944, +6.537]$ |
| | PCA3 | $-3.04087$ $[-3.14404, -2.94004]$ | $+6.886$ $[+6.636, +7.132]$ |
| | Probe3 | $-1.71221$ $[-1.77754, -1.64582]$ | $+4.995$ $[+4.737, +5.244]$ |
| | Random3 | $-1.80056$ $[-1.86699, -1.72708]$ | $+4.864$ $[+4.583, +5.138]$ |
| VGG16 / CIFAR-10 | Raw | $-0.31204$ $[-0.38750, -0.24326]$ | $-1.834$ $[-1.990, -1.670]$ |
| | Residual | $-3.09421$ $[-3.17050, -3.01651]$ | $+9.413$ $[+9.234, +9.578]$ |
| | PCA3 | $-1.39732$ $[-1.43984, -1.35239]$ | $+6.570$ $[+6.406, +6.724]$ |
| | Probe3 | $-1.55125$ $[-1.58426, -1.51566]$ | $+6.725$ $[+6.578, +6.861]$ |
| | Random3 | $-2.06365$ $[-2.11032, -2.01272]$ | $+6.787$ $[+6.598, +6.965]$ |

Table A7: Brier score and ECE on the same cohorts. Lower is better; boldface marks the lower value without implying statistical significance. Differences are descriptive; confidence intervals were not computed.

| | Brier score ↓ | | ECE ↓ | |
|---|---|---|---|---|
| Model / dataset | Raw | Projected (W3) | Raw | Projected (W3) |
| ResNet20 / MNIST | 0.00526 | **0.00284** | **0.00409** | 0.00497 |
| VGG11 / MNIST | 0.00408 | **0.00113** | 0.00181 | **0.00129** |
| ResNet20 / CIFAR-10 | 0.35893 | **0.27003** | 0.11222 | **0.05941** |
| VGG16 / CIFAR-10 | 0.17100 | **0.07760** | 0.08013 | **0.01887** |