Interpreting Multivariate Shapley Interactions in DNNs

Hao Zhang,[*] Yichen Xie,[*] Longjie Zheng, Die Zhang, Quanshi Zhang[†]
Shanghai Jiao Tong University, China
{1603023-zh,xieyichen,bugatti,zizhan52,zqs1022}@sjtu.edu.cn

Abstract

This paper aims to explain deep neural networks (DNNs) from the perspective of multivariate interactions. In this paper, we define and quantify the significance of interactions among multiple input variables of the DNN. Input variables with strong interactions usually form a coalition and reflect prototype features, which are memorized and used by the DNN for inference. We define the significance of interactions based on the Shapley value, which is designed to assign the attribution value of each input variable to the inference. We have conducted experiments with various DNNs. Experimental results have demonstrated the effectiveness of the proposed method.

Introduction

Deep neural networks (DNNs) have exhibited significant success in many tasks, and the interpretability of DNNs has received increasing attention in recent years. Most previous studies of post-hoc explanation of DNNs either explain DNN semantically/visually (Lundberg and Lee 2017; Ribeiro, Singh, and Guestrin 2016), or analyze the representation capacity of DNNs (Higgins et al. 2017; Achille and Soatto 2018; Fort et al. 2019; Liang et al. 2020).

In this paper, we propose a new perspective to explain a trained DNN, *i.e.* quantifying interactions among input variables that are used by the DNN during the inference process. Each input variable of a DNN usually does not work individually. Instead, input variables may cooperate with other variables to make inferences. We can consider the strongly interacted input variables to form a prototype feature (or a coalition), which is memorized by the DNN. For example, the face is a prototype feature for person detection, which is comprised of the eyes, nose, and mouth. Each compositional part inside a face does not contribute to the inference individually, but they collaborate with each other to make the combination of all parts a meaningful feature for person detection.

Previous studies mainly focused on the interaction between two variables (Lundberg, Erion, and Lee 2018; Singh, Murdoch, and Yu 2018; Murdoch, Liu, and Yu 2018; Janizek, Sturmfels, and Lee 2020). Grabisch and Roubens (1999) measured 2^n different interaction values for all the 2^n combinations of n variables, and the cost of computing each interaction value is NP-hard. In comparison to the unaffordable computational cost, our research summarizes all 2^n interactions into a single metric, which provides a global view to explain potential prototype features encoded by DNNs.

In this study, we define the significance of interactions among multiple input variables based on game theory. In the game theory, each input variable can be viewed as a player. All players (variables) are supposed to obtain a high reward in a game. Let us consider interactions *w.r.t.* the scalar output $y = f(I)$ of the DNN (or interactions *w.r.t.* one dimension of the vectorized network output), given the input I with overall n players. The absolute change of y *w.r.t.* an empty input $\mathbf{0}$, *i.e.* $|f(I) - f(\mathbf{0})|$, can be used as the total reward of all players (input variables). The reward can be allocated to each player, which indicates the contribution made by the player (input variable), denoted by $\phi_1, \phi_2, \cdots, \phi_n$, satisfies $|f(I) - f(\mathbf{0})| = \sum_{i=1}^{n} \phi_i$.

A simple definition of the interaction is given as follows. If a set of m players S always participates in the game together, these players can be regarded to form a coalition. The coalition can obtain a reward, denoted by ϕ_S. ϕ_S is usually different from the sum of rewards when players in the set S participate in the game individually. The additional reward $\phi_S - \sum_{i\in S} \phi_i$ obtained by the coalition can be quantified as the interaction. If $\phi_S - \sum_{i\in S} \phi_i > 0$, we consider players in the coalition have a positive effect. Whereas, $\phi_S - \sum_{i\in S} \phi_i < 0$ indicates an negative/adversarial effect among the set of variables in S.

However, $\phi_S - \sum_{i\in S} \phi_i$ mainly measures the interaction in a single coalition, whose interaction is either purely positive or purely negative. In real applications, for the set S with m players, players may form at most $2^m - m - 1$ different coalitions. Some coalitions have positive interaction effects, while others have negative interaction effects. Thus, interactions of different coalitions can counteract each other.

In order to objectively measure the significance of the overall interactions within a specific set S with m players (variables), we propose a new metric, which reflects both positive and negative interaction effects. We develop a method to divide the m players into a few coalitions, and ensures that each coalition mainly contains positive interaction effects.

[*]Contribute equally to this paper.
[†]Corresponding author. This work was done under the supervision of Dr. Quanshi Zhang.

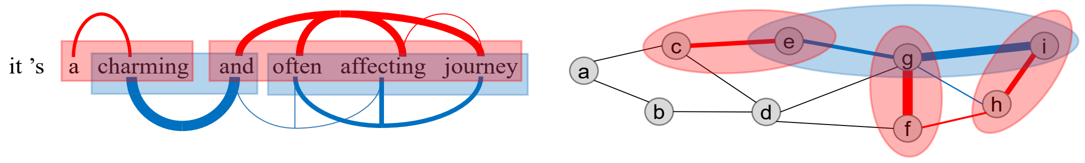


Figure 1: Overview of the proposed method. In this paper, we aim to quantify the significance of interactions among a set of input variables. We divide the set of input variables into various coalitions, which mainly encode either positive or negative interaction effects. In this figure, each red region represents a coalition that mainly encodes positive interaction effects, and each blue region represents a coalition that mainly encodes negative interaction effects. Thickness of the edge indicates the strength of the interaction.

Similarly, we can also divide the m players into another few coalitions, each of which mainly encodes negative interaction effects. In this way, we can quantify the significance of both positive and negative interaction effects, as shown in Figure 1.

In experiments, we have applied our method to five DNNs with different architectures for various tasks. Experimental results have provided new insights to understand these DNNs. Besides, our method can be used to mine prototype features towards correct and incorrect predictions of DNNs.

Contributions of this study can be summarized as follows. (1) In this paper, we define and quantify the significance of interactions among multiple input variables in the DNN, which can reflect both positive and negative interaction effects. (2) Our method can extract prototype features, which provides new perspectives to understand the DNN, and mines prototype features towards correct and incorrect predictions of the DNN. (3) We develop a method to efficiently approximate the significance of interactions.

Related work

Interpretability of DNNs in NLP

The interpretability of DNNs is an emerging direction in machine learning, and many methods have been proposed to explain the DNN in natural language processing (NLP). Some methods computed importance/attribution/saliency values for input variables *w.r.t.* the output of the DNN (Lundberg and Lee 2017; Ribeiro, Singh, and Guestrin 2016; Binder et al. 2016; Simonyan, Vedaldi, and Zisserman 2013; Shrikumar et al. 2016; Springenberg et al. 2014; Li et al. 2015; Li, Monroe, and Jurafsky 2016). Another kind of methods to understand DNNs are to measure the representation capacity of DNNs (Higgins et al. 2017; Achille and Soatto 2018; Fort et al. 2019; Liang et al. 2020; Cheng et al. 2020). Guan et al. (2019) proposed a metric to measure how much information of an input variable was encoded in an intermediate-layer of the DNN. Some studies designed a network to learn interpretable feature representations. Chung, Ahn, and Bengio (2017) modified the architecture of recurrent neural network (RNN) to capture the latent hierarchical structure in the sentence. Shen et al. (2019) revised the LSTM to learn interpretable representations. Other studies explained the feature processing encoded in a DNN by embedding hierarchical structures into the DNN (Tai, Socher, and Manning 2015; Wang, Lee, and Chen 2019). Wang et al. (2019) used structural position representations to model the latent structure of the input sentence by embedding a dependency tree into the DNN. Dyer et al. (2016) proposed a generative model to model hierarchical relationships among words and phrases.

In comparison, this paper focuses on the significance of interactions among multiple input variables, which provides a new perspective to understand the behavior of DNNs.

Interactions

Several studies explored the interaction between input variables. Bien, Taylor, and Tibshirani (2013) developed an algorithm to learn hierarchical pairwise interactions inside an additive model. Sorokina et al. (2008) proposed a method to detect the statistical interaction via an additive model-based ensemble of regression trees. Some studies defined different types of interactions between variables to explain DNNs from different perspectives. Murdoch, Liu, and Yu (2018) proposed to use the contextual decomposition to extract the interaction between gates in the LSTM. Singh, Murdoch, and Yu (2018); Jin et al. (2019) extended the contextual decomposition to produce a hierarchical clustering of input features and contextual independence of input words, respectively. Janizek, Sturmfels, and Lee (2020) quantified the pairwise feature interaction in DNNs by extending the explanation method of Integrated Gradients (Sundararajan, Taly, and Yan 2017). Tsang, Cheng, and Liu (2018) measured the pairwise interaction by applying the learned weights of the DNN.

Unlike above studies, Lundberg, Erion, and Lee (2018) defined the interaction between two variables based on the Shapley value for tree ensembles. Because Shapley value was considered as the unique standard method to estimate contributions of input words to the prediction score with solid theoretical foundations (Weber 1988), this definition of the interaction can be regarded to objectively reflect the collaborative/adversarial effects between variables *w.r.t.* the prediction score. Furthermore, Grabisch and Roubens (1999); Dhamdhere, Agarwal, and Sundararajan (2019) extended this definition to the elementary interaction component and the Shapley-Taylor index among various variables, respectively. These studies quantified the significance of interaction for each possible specific combination of variables, instead of providing a single metric to quantify interactions among all combinations of variables. In comparison, this paper aims to use a single metric to quantify the significance of interactions among all combinations of input variables, and reveal prototype features modeled by the DNN.

Shapley values

The Shapley value was initially proposed in the game theory (Shapley 1953). Let us consider a game with multiple players. Each player can participate in the game and receive a reward individually. Besides, some players can form a coalition and play together to pursue a higher reward. Different players in a coalition usually contribute differently to the game, thereby being assigned with different proportions of the coalition's reward. The Shapley value is considered as a unique method that fairly allocates the reward to players with certain desirable properties (Weber 1988; Ancona, Oztireli, and Gross 2019). Let $N = \{1, 2, \cdots, n\}$ denote the set of all players, and 2^N represents all potential subsets of N. A game $v : 2^N \to \mathbb{R}$ is implemented as a function that maps from a subset to a real number. When a subset of players $S \subseteq N$ plays the game, the subset can obtain a reward $v(S)$. Specifically, $v(\emptyset) = 0$. The Shapley value of the i-th player $\phi_v(i|N)$ can be considered as an unbiased contribution of the i-th player.

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

Weber (1988) have proved that the Shapley value is the only reward with the following axioms.

Linearity axiom: If the reward of a game u satisfies $u(S) = v(S) + w(S)$, where v and w are another two games. Then the Shapley value of each player $i \in N$ in the game u is the sum of Shapley values of the player i in the game v and w, *i.e.* $\phi_u(i|N) = \phi_v(i|N) + \phi_w(i|N)$.

Dummy axiom: The dummy player is defined as the player that satisfies $\forall S \subseteq N \setminus \{i\}, v(S \cup \{i\}) = v(S) + v(\{i\})$. In this way, the dummy player i satisfies $v(\{i\}) = \phi_v(i|N)$, *i.e.* the dummy player has no interaction with other players in N.

Symmetry axiom: If $\forall S \subseteq N \setminus \{i, j\}, v(S \cup \{i\}) = v(S \cup \{j\})$, then $\phi_v(i|N) = \phi_v(j|N)$.

Efficiency axiom: $\sum_{i\in N} \phi_v(i|N) = v(N)$. The efficiency axiom can ensure the overall reward can be distributed to each player in the game.

Algorithm

Multivariate interactions in game theory

Given a game with n players $N = \{1, 2, \cdots, n\}$, in this paper, we define the interaction among multiple players based on game theory. Specifically, we focus on the significance of interactions of a set of m players selected from all players. Let 2^N denote all potential subsets of N, *e.g.* if $N = \{a, b\}$, then $2^N = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$. The game $v : 2^N \to \mathbb{R}$ is a set function that maps a set of players to a scalar value. We consider this scalar value as the reward for the set of players. For example, we consider the DNN with a scalar output as a game, and consider the input variables as a set of players. In this way, $v(S)$, $S \subseteq N$, represents the output of the DNN when all input variables in $N \setminus S$ are replaced by the baseline value. In this paper, the baseline value is set as zero, which is similar to (Ancona, Oztireli, and Gross 2019). If the DNN is trained for multi-category classification, $v(S)$ is implemented as the score of the true category before the softmax layer. The overall reward can be represented as $v(N) - v(\emptyset)$, *i.e.* the score obtained by all input variables subtract the score obtained by the zero input. The overall reward can be allocated to each player as $\sum_{i=1}^{n} \phi_v(i|N) = v(N) - v(\emptyset)$, where $\phi_v(i|N)$ denotes the allocated reward of the player i over the set N. Each player is supposed to obtain a high reward in the game. In this paper, we use $\phi(i|N)$ to represent $\phi_v(i|N)$ in the following manuscript for simplicity without causing ambiguity. Specifically, $\phi(i|N)$ can be computed as the Shapley value of i based on Equation (1). The Shapley value is an unbiased method to allocate the reward, so we use the Shapley value to define the interaction.

Interaction between two players: Let us first consider the interaction between two players in the game v. Given two players i and j, $\phi(i|N)$ and $\phi(j|N)$ represent their Shapley values, respectively. If players i and j always participate in the game together and always be absent together, then they can be regarded to form a coalition. The reward obtained by the coalition is usually different from the sum of rewards when players i and j participate in the game individually. This coalition can be considered as one singleton player, denoted by S_{ij}. Note that the Shapley value of the coalition can also be computed using Equation (1) by replacing the player i with S_{ij}. In this way, we can consider there are only $n-1$ players in the game, including S_{ij}, and excluding players i and j, $N' = N\setminus\{i, j\}\cup S_{ij}$. The interaction between players i and j is defined as the additional reward brought by the coalition S_{ij} *w.r.t.* the sum of rewards of each player playing individually, *i.e.*

$$B(S_{ij}) = \phi(S_{ij}|N') - [\phi(i|N_i) + \phi(j|N_j)], \quad (2)$$

where $\phi(i|N_i)$, $\phi(j|N_j)$, and $\phi(S_{ij}|N')$ are computed over the set of players $N_i = N\setminus\{j\}$, $N_j = N\setminus\{i\}$, and N', respectively. **(1)** If $B(S_{ij}) > 0$, then players i and j cooperate with each other for a higher reward, *i.e.* the interaction is positive. **(2)** If $B(S_{ij}) < 0$, then the interaction between players i and j leads to a lower reward, *i.e.* the interaction is negative.

Interaction among multiple players: We can extend the definition of interactions to multiple players. Let us focus on a subset of players $A \subsetneq N$. If players in A always participate in the game together, then players in A form a coalition, which can be considered as a singleton player, denoted by $[A]$. The interaction in A can be measured as

$$B([A]) = \phi([A]|N_{[A]}) - \sum_{i\in A} \phi(i|N_i), \quad (3)$$

where $\phi([A]|N_{[A]})$ is the Shapley value of the coalition $[A]$. We compute the Shapley value over the player set $N_{[A]} = N \setminus A \cup \{[A]\}$. Similarly, $\phi(i|N_i)$ represents the Shapley value computed over the set $N_i = N \setminus A \cup \{i\}$, where players in $A \setminus \{i\}$ are supposed not to attend the game. In this way, $B([A])$ reflects all interactions inside A, including both positive and negative interactions, which counteract each other. For example, let the set $A = \{i, j, k, l\}$, we assume that only the interaction between players i and j and the interaction between players k and l are positive. Interactions of other coalitions are negative. Then, the positive interaction effects are counteracted by negative interaction effects in $B([A])$. We will introduce more details in Equation (6).

However, the significance of interactions is supposed to reflect both positive and negative interaction effects. Therefore, we hope to propose a new metric to measure the significance of interactions, which contains both positive and negative effects among a set of players $A \subsetneqq N$. We aim to divide all players in A into a number of coalitions $\Omega = \{C_1, C_2, \cdots, C_k\}$, which are ensured to mainly encode positive interaction effects. Ω is a partition of the set A. In the above example, the four players $\{i, j, k, l\}$ is supposed to be divided into $\Omega = \{C_1 = \{i, j\}, C_2 = \{k, l\}\}$. In this case, the following equation can better reflect positive interaction effects than Equation (3), when we measure the reward by taking each coalition C_i as a singleton player.

$$B_{\text{max}}([A]) = \max_\Omega \sum_{C\in\Omega} \phi(C|N_C) - \sum_{i\in A} \phi(i|N_i) \quad (4)$$

where $\phi(C|N_C)$ is computed over $N_C = N \setminus A \cup \{[C]\}$, and $\phi(i|N_i)$ is computed over $N_i = N \setminus A \cup \{i\}$. Similarly, we can use $B_{\text{min}}([A]) = \min_\Omega \sum_{C\in\Omega} \phi(C|N_C) - \sum_{i\in A} \phi(i|N_i)$ to roughly quantify the negative interaction effects inside the set A. Thus, we define the metric $T([A])$ to measure the significance of both positive and negative interaction effects, as follows.

$$\begin{aligned} T([A]) &= B_{\text{max}}([A]) - B_{\text{min}}([A]) \\ &= \max_\Omega \sum_{C\in\Omega} \phi(C|N_C) - \min_\Omega \sum_{C\in\Omega} \phi(C|N_C) \end{aligned} \quad (5)$$

Relationship between $T([A])$ and $B([A])$: Theoretically, the interaction $B([A])$ in Equation (3) can be decomposed into elementary interaction components $I(A)$, which was firstly proposed by (Grabisch and Roubens 1999). $I(A)$ quantifies the marginal reward of A, which removes all marginal rewards from all combinations of A. We derive the following equation to encode the relationship between $B([A])$ and $I(A)$.

$$B([A]) = \sum_{A' \subseteq A, |A'|>1} I(A'). \quad (6)$$

According to Equation (6), there are a total of $2^m - m - 1$ different elementary interaction components inside the interaction $B([A])$, where m represents the number of players inside A. Some of these interaction components are positive, and others are negative, which leads to a counteraction among all possible interaction components. In order to quantify the significance of all interaction components, a simple idea is to sum up the absolute values of all elementary interaction components, as follows. $B'([A]) = \sum_{A'\subseteq A, |A'|>1} |I(A')| = B^+ - B^-$, where $B^+ = \sum_{A'\subseteq A, I(A')>0} I(A')$ and $B^- = \sum_{A'\subseteq A, I(A')<0} I(A')$, subject to $|A'| > 1$. In Equation (4), $B_{\text{max}}([A])$ is supposed to mainly encode most positive interaction effects within A, which is highly related to B^+. Similarly, $B_{\text{min}}([A])$ is highly related to B^- for mainly encoding negative interaction effects within A. Therefore, $T([A]) = B_{\text{max}}([A]) - B_{\text{min}}([A])$ is highly related to $B'([A]) = B^+ - B^-$.

Why we need a single metric? Grabisch and Roubens (1999) designed $I(A)$, which computed 2^m different values of $I(A')$ for all possible $A' \subseteq A$. In comparison, the proposed $B([A])$ is a single metric to represent the overall interaction significance among all m input variables, which provides a global view to explain DNNs.

Explanation of DNNs using multivariate interactions in NLP

In this paper, we use the interaction based on game theory to explain DNNs for NLP tasks. Given an input sentence with n words $N = \{1, 2, \cdots, n\}$, each word in the sentence can be considered as a player, and the output score of DNN can be taken as the game score $v(S)$. $v(S)$ represents the output of DNN when we mask all words in the set $N \setminus S$, *i.e.* setting word vectors of masked words to zero vectors. For the DNN with a scalar output, $v(S)$ denotes the output of the DNN. If the DNN was trained for the multi-category classification task, we take the output of the true category before the softmax layer as $v(S)$. Strongly interacted words usually cooperate with each other and form a prototype feature, which is memorized by the DNN for inference. Thus, the multivariate interaction can be used to analyze prototype features in DNNs.

Among all input words in N, we aim to quantify the significance of interactions among a subset of m sequential words $A \subsetneqq N$. Chen et al. (2019) showed that non-successive words usually contain much fewer interactions than successive words. Thus, we require each coalition consists of several sequential words to simplify the implementation. Although $T([A])$ in Equation (5) can be computed by enumerating all possible partitions in A, Ω, the computational cost of such a method is too unaffordable.

Therefore, we develop a sampling-based method to efficiently approximate $T([A])$ in Equation (5). The computation of $\max_\Omega \sum_{C\in\Omega} \phi(C|N_C)$ requires us to enumerate all potential partitions to determine the maximal value. In order to avoid such computational expensive enumeration, we propose to use $\mathbf{p} = \{p_1, p_2, \cdots, p_{m-1}\}$ to represent all possible partitions. p_i $(0 \le p_i \le 1)$ denotes the probability of the i-th word and the $(i+1)$-th word belonging to the same coalition. We can sample $\mathbf{g} = \{g_1, g_2, \cdots g_{m-1}\}$ based on the $\mathbf{p}$, *i.e.* $g_i \in \{0, 1\}$, $g_i \sim \text{Bernoulli}(p_i)$, to represent a specific partition Ω. $g_i = 1$ indicates the i-th word and the $(i+1)$-th word belong to the same coalition. $g_i = 0$ represents these words belong to different coalitions. In this way, we can use $\max_{\mathbf{p}} \mathbb{E}_{\mathbf{g}\sim\text{Bernoulli}(\mathbf{p})} \sum_{C\in\Omega_{\mathbf{g}}} \phi(C|N_C)$ to approximate $\max_\Omega \sum_{C\in\Omega} \phi(C|N_C)$, where $\Omega_{\mathbf{g}}$ denotes the partition determined by $\mathbf{g}$.

In addition, $\phi(C|N_C)$ can be approximated using a sampling-based method (Castro, Gómez, and Tejada 2009).

$$\begin{aligned} \phi(C|N_C) &= \mathbb{E}_r\Big[\mathbb{E}_{\substack{S\subseteq N_C,\\ |S|=r}}[v(S\cup C) - v(S)]\Big] \\ &= \mathbb{E}_r\Big\{\mathbb{E}_{\substack{|S|=r+1,\\ S\subseteq N_C, S\ni[C]}}[v(S)] - \mathbb{E}_{\substack{|S|=r,\\ S\subseteq N_C, S\not\ni[C]}}[v(S)]\Big\} \end{aligned} \quad (7)$$

where r represents the number of players fed into the DNN, $N_C = N \setminus A \cup \{[C]\}$. In this way, we can approximate $\max_\Omega \sum_{C\in\Omega} \phi(C|N_C)$ as

Table 1: Accuracy of the estimated partition.

	Add-Multiple Dataset	AND-OR Dataset	Exponential Dataset
Baseline 1	0.500	0.503	0.506
Baseline 2	1.000	0.996	1.000
Baseline 3	1.000	0.523	0.846
Our method	**1.000**	**0.999**	**1.000**

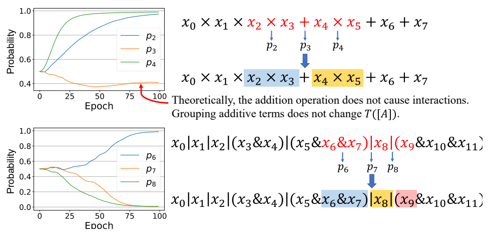


Figure 2: Convergence of p_i during the training process.

$\max_{\mathbf{p}} \mathbb{E}_{\mathbf{g}\sim\text{Bernoulli}(\mathbf{p})} \sum_{C\in\Omega_{\mathbf{g}}} \phi(C|N_C)$, as follows.

$$\mathcal{L}=\mathop{\mathbb{E}}_{\mathbf{g}\sim\text{Bernoulli}(\mathbf{p})}\sum_{C\in\Omega_{\mathbf{g}}}\phi(C|N_C),\ \max_{\mathbf{p}}\mathcal{L}=\max_{\mathbf{p}}\sum_{i\in A}$$
$$\mathop{\mathbb{E}}_{r}\left\{\mathop{\mathbb{E}}_{\mathbf{g}\sim\text{Bernoulli}(\mathbf{p})}\lambda_i(\mathbf{g})\left[\mathop{\mathbb{E}}_{\substack{S\in\text{Sam}(\Omega_{\mathbf{g}}),\\ S\ni i,|S|=r+1}}[v(S)]-\mathop{\mathbb{E}}_{\substack{S\in\text{Sam}(\Omega_{\mathbf{g}}),\\ S\not\ni i,|S|=r}}[v(S)]\right]\right\} \tag{8}$$

$$\frac{\partial\mathcal{L}}{\partial p_i}=\sum_{j\in A}\mathbb{E}_r\left\{\mathop{\mathbb{E}}_{\substack{\mathbf{g}\sim\text{Bernoulli}(\mathbf{p})\\ g_i=1}}\lambda_i(\mathbf{g})\left[\mathop{\mathbb{E}}_{\substack{S\in\text{Sam}(\Omega_{\mathbf{g}}),\\ S\ni j,|S|=r+1}}[v(S)]-\mathop{\mathbb{E}}_{\substack{S\in\text{Sam}(\Omega_{\mathbf{g}}),\\ S\not\ni j,|S|=r}}[v(S)]\right]\right.$$
$$\left.-\mathop{\mathbb{E}}_{\substack{\mathbf{g}\sim\text{Bernoulli}(\mathbf{p})\\ g_i=0}}\lambda_i(\mathbf{g})\left[\mathop{\mathbb{E}}_{\substack{S\in\text{Sam}(\Omega_{\mathbf{g}}),\\ S\ni j,|S|=r+1}}[v(S)]-\mathop{\mathbb{E}}_{\substack{S\in\text{Sam}(\Omega_{\mathbf{g}}),\\ S\not\ni j,|S|=r}}[v(S)]\right]\right\} \tag{9}$$

where $\lambda_i(\mathbf{g})=\frac{1}{|C_i|}$, and $S\in\text{Sam}(\Omega_{\mathbf{g}})$ represents the sampled set of words, which contains all words in $N\setminus A$ and the coalition C_i. $C_i\in\Omega_{\mathbf{g}}$ denotes the coalition determined by $\mathbf{g}$ that contains the word i. We learn $\mathbf{p}$ to maximize the above equation. For example, let $A=\{x_1,x_2,x_3,x_4,x_5,x_6\}$, and the sampled $\mathbf{g}=\{g_1=1,g_2=1,g_3=0,g_4=0,g_5=1\}$. We consider the first three words in A as a coalition, and the last two words in A as another coalition, *i.e.* $\Omega_{\mathbf{g}}=\{C_1=\{x_1,x_2,x_3\},C_2=\{x_4\},C_3=\{x_5,x_6\}\}$. The set S is supposed to be sampled over these coalitions in $\Omega_{\mathbf{g}}$, instead of over individual words. In this case, we have $\lambda_1(\mathbf{g})=\lambda_2(\mathbf{g})=\lambda_3(\mathbf{g})=1/3$, $\lambda_4(\mathbf{g})=1$, $\lambda_5(\mathbf{g})=\lambda_6(\mathbf{g})=1/2$. In order to learn p_i, we compute $\partial\mathcal{L}/\partial p_i$, which is given in Equation (9). In this way, we can use $\max_{\mathbf{p}}\mathcal{L}$ in Equation (8) to approximate the solution to $\max_\Omega\sum_{C\in\Omega}\phi(C|N_C)$. Similarly, we can use $\min_{\mathbf{p}}\mathcal{L}$ to approximate the solution to $\min_\Omega\sum_{C\in\Omega}\phi(C|N_C)$, thereby obtaining $T([A])=\max_\Omega\sum_{C\in\Omega}\phi(C|N_C)-\min_\Omega\sum_{C\in\Omega}\phi(C|N_C)$.

Comparison of computational cost: Given a set of m words in a sentence with n words, the computation of $T([A])$ based on its definition in Equation (5) and Equation (1) is NP-hard, *i.e.* $O(2^n m)$[1]. Instead, we propose a polynomial method to approximate $T([A])$ with computational cost $O(K_1K_2K_3nm)$[1], where K_1 denotes the number of updating the probability $\mathbf{p}$, K_2 and K_3 represent the sampling number of $\mathbf{g}\sim\text{Bernoulli}(\mathbf{p})$ and $S\in\text{Sam}(\Omega_{\mathbf{g}})$, respectively. In addition, we conducted experiments to show the accuracy of the estimation of $T([A])$ increases along with the increase of the sampling number.

Experiments

Evaluation of the correctness of the estimated partition of coalitions: The core challenge of evaluating the correctness of the partition of coalitions was that we had no ground-truth annotations of inter-word interactions, which were encoded in DNNs. To this end, we constructed three new datasets with ground-truth partitions between input variables, *i.e. the Addition-Multiplication Dataset*, *the AND-OR Dataset*, and *the Exponential Dataset*.

[1]Please see supplementary materials for the proof.

• *The Addition-Multiplication Dataset* contained 10000 addition-multiplication models, each of which only consisted of addition operations and multiplication operations. For example, $y=f(\mathbf{x})=x_1+x_2\times x_3+x_4\times x_5+x_6+x_7$. Each variable x_i was a binary variable, *i.e.* $x_i\in\{0,1\}$. For each model, we selected a number of sequential input variables to construct A, *e.g.* $A=\{x_2,x_3,x_4,x_5\}$. We applied our method to extract the partition of coalitions *w.r.t.* y, which maximized $\sum_{C\in\Omega}\phi(C|N_C)$.

• *The AND-OR Dataset* contained 10000 AND-OR models, each of which only contained AND operations and OR operations. For simplicity, we used & to denote the AND operation, and used | to denote the OR operation. For example, $y=f(\mathbf{x})=x_1\mid(x_2\ \&\ x_3)\mid(x_4\ \&\ x_5)\mid x_6\mid x_7$. Each variable x_i was a binary variables, *i.e.* $x_i\in\{0,1\}$. For each model, we selected a number of sequential input variables to construct A, *e.g.* $A=\{x_2,x_3,x_4,x_5\}$. We used our method to extract the partition of coalitions, which maximized $\sum_{C\in\Omega}\phi(C|N_C)$.

• *The Exponential Dataset* contained 10000 models, each of which contained exponential operations and addition operations. For example, $y=f(\mathbf{x})=x_1^{x_2}+x_3^{x_4}+x_5+x_6$. Each variable x_i was a binary variables, *i.e.* $x_i\in\{0,1\}$. For each model, we selected a number of sequential input variables to construct A, which always contained the exponential operations, *e.g.* $A=\{x_1,x_2,x_3,x_4\}$. We used our method to extract the partition of coalitions, which maximized $\sum_{C\in\Omega}\phi(C|N_C)$.

We compared the extracted partition of coalitions with the ground-truth partition of coalitions. Variables in multiplication operations and AND operations usually had positive interactions, which were supposed to be allocated into one coalition during $\max_\Omega\sum_{C\in\Omega}\phi(C|N_C)$. Variables in OR operations had negative interactions, which were supposed to be allocated to different coalitions during $\max_\Omega\sum_{C\in\Omega}\phi(C|N_C)$. Note that the addition operation did not cause any interactions. Therefore, we did not consider interactions of addition operations when we evaluated the correctness of the partition. In the above example of addition-multiplication model, if we considered the set $A=\{x_2,x_3,x_4,x_5\}$, then the ground-truth par-

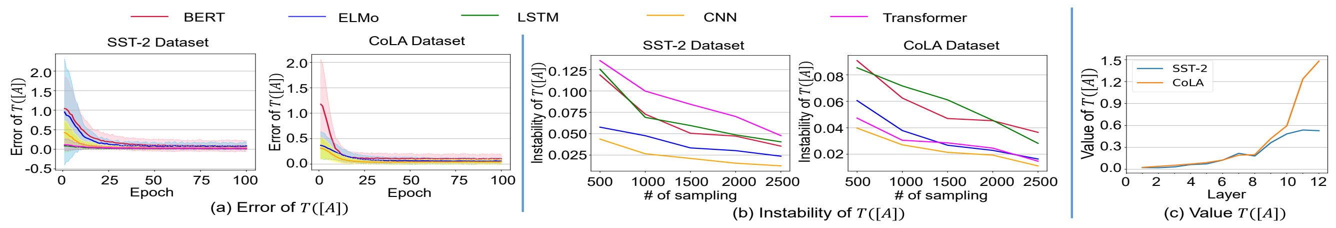


Figure 3: (a) The decrease of error of $T([A])$ during the learning process. We obtained accurate $T([A])$ after 100 epochs. (b) The instability of interactions with different numbers of sampling. When the number of sampling was larger than 1000, the instability of $T([A])$ was less than 0.1 on all models and datasets. (c) The significance of interactions in the computation of intermediate-layer features gradually increased during the forward propagation in the BERT.

tition was supposed to be either $\{\{x_2, x_3\}, \{x_4, x_5\}\}$ or $\{\{x_2, x_3, x_4, x_5\}\}$. For the above example of AND-OR model and $A = \{x_2, x_3, x_4, x_5\}$, the ground-truth partition should be $\{\{x_2, x_3\}, \{x_4, x_5\}\}$. For each operation, if the proposed method allocated variables of an operation in the same way as the ground-truth partition, then we considered the operation was correctly allocated by the proposed method. The average rate of correctly allocated operations over all operations was reported in Table 1.

We also designed three baselines. **Baseline 1** randomly combined input variables to generate coalitions. Lundberg, Erion, and Lee (2018) defined the interaction between two input variables, which was used as the **Baseline 2**. Li, Monroe, and Jurafsky (2016) proposed a method to estimate the importance of input variables, which was taken as **Baseline 3**. **Baseline 2** and **Baseline 3** merged variables whose interactions were greater than zero to generate coalitions.

Table 1 compares the accuracy of coalitions generated by our method and that of other baselines. Our method achieved a better accuracy than other baselines. Figure 2 shows the change of the probability p_i during the training process. Our method successfully merged variables into a single coalition in multiplication operations and AND operations. Besides, our method did not merge variables in OR operations. Note that the probability p_3 of the addition operation in Figure 2 did not converge. It was because the addition operation did not cause any interactions, *i.e.* arbitrary value of p_3 satisfied the ground-truth.

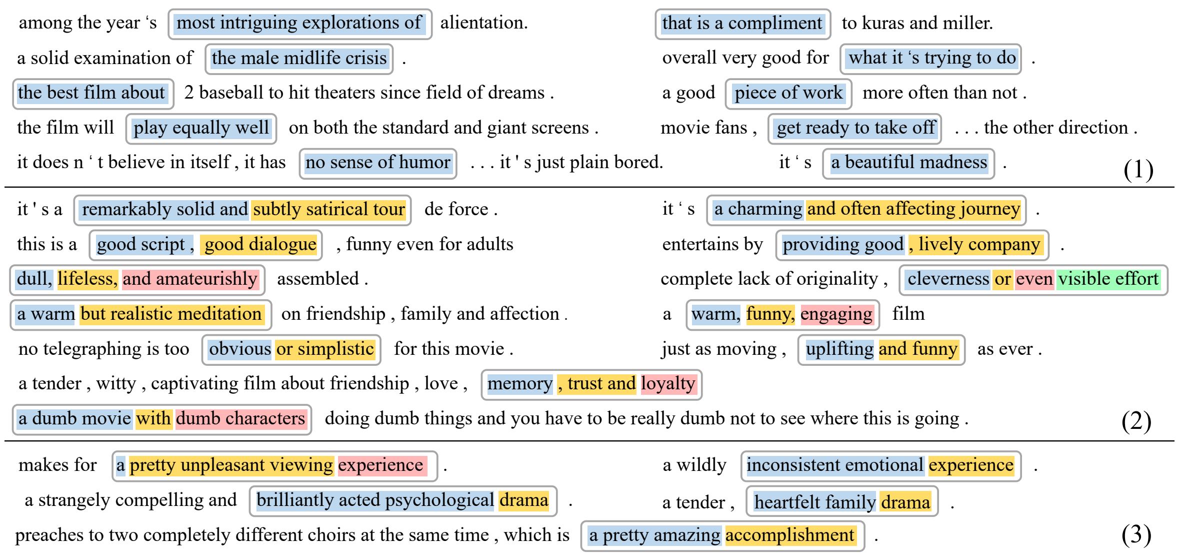
Evaluation of the accuracy of $T([A])$: We applied our method to DNNs in NLP to quantify the interaction among a set of input words. We trained DNNs for two tasks, *i.e.* binary sentiment classification based on the SST-2 dataset (Socher et al. 2013) and prediction of linguistic acceptability based on the CoLA dataset (Warstadt, Singh, and Bowman 2018). For each task, we trained five DNNs, including the BERT (Devlin et al. 2018), the ELMo (Peters et al. 2018), the CNN proposed by (Kim 2014), the two-layer unidirectional LSTM (Hochreiter and Schmidhuber 1997), and the Transformer (Vaswani et al. 2017). For each sentence, the set of successive words A was randomly selected.

We compared the extracted significance of interactions $T([A])$ with the accurate significance of interactions. The accurate significance of interactions was quantified based on the definition in Equation (5) and Equation (1), which was computed by enumerating all partitions Ω and all subsets S with an extremely high computational cost. Considering the unaffordable computational cost, such evaluation could only be applied to sentences with less than 12 words. Figure 3 (a) reports the error of $T([A])$, *i.e.* $|T_{\text{truth}}([A]) - T([A])|$, where $T_{\text{truth}}([A])$ was the true interaction significance accurately computed via massive enumerations. We found that the estimated significance of interactions was accurate enough after the training of 100 epochs.

Stability of $T([A])$: We also measured the stability of $T([A])$, when we computed $T([A])$ multiple times with different sampled sets of $\mathbf{g}$ and S. The instability of $T([A])$ was computed as instability $= \mathbb{E}_I[\frac{\mathbb{E}_{u,v:u\neq v}|T_{(u)}([A])-T_{(v)}([A])|}{\mathbb{E}_w|T_{(w)}([A])|}]$, where $T_{(u)}([A])$ denotes the u-th computation result of $T([A])$. Figure 3 (b) shows the instability of interactions in different DNNs and datasets. Experimental results showed that interactions in CNN and ELMo converged more quickly. Moreover, instability decreased quickly along with the increase of the sampling numbers. Our metric was stable enough when the number of sampling was larger than 2000.

Prototype features modeled by DNNs: We analyzed results obtained by the proposed method to explore prototype features modeled by DNNs. Figure 4 shows several results on the BERT. We found that: (1) For an entire constituent in the sentence, such as a short clause or a noun phrase, the BERT usually took the whole constituent as a single coalition, *i.e.* a prototype feature. (2) For the set of words that contained conjunctions or punctuations, such as "and", "or", ",", *etc.*, the BERT divided the set at conjunctions or punctuations to generate different coalitions (prototype features). (3) For the constituent modified by multiple adjectives and adverbs, the BERT usually merged all adjectives and adverbs into one coalition, and took the modified constituent as another coalition. Such phenomena fit the syntactic structure parsing according to human knowledge.

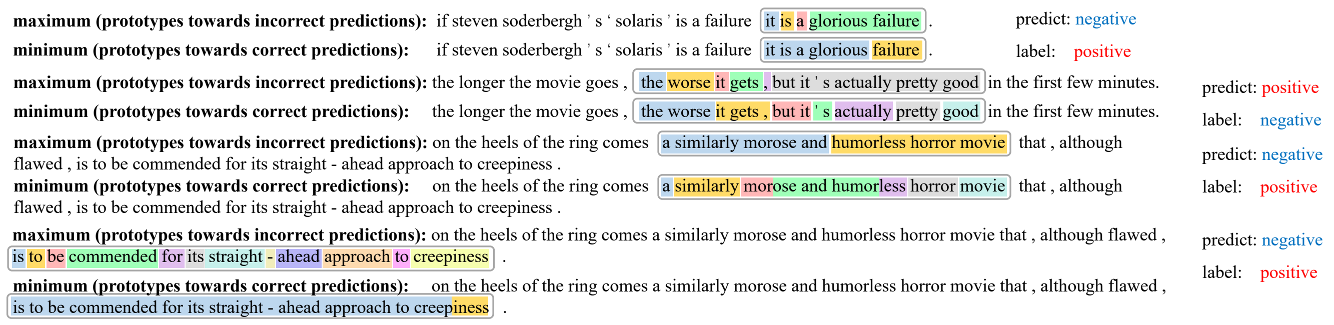
Interactions *w.r.t.* the intermediate-layer feature: We could also compute the significance of interactions among a set of input words *w.r.t.* the computation of an intermediate-layer feature $\boldsymbol{f}$. The reward was computed using the intermediate-layer feature. Let $\boldsymbol{f}_N$ and $\boldsymbol{f}_S$ represent intermediate-layer features obtained by the set of input variables N and S, respectively. Since the intermediate-layer feature usually can be represented as high dimensional vertors, instead of a scalar value. We computed the output $v(S)$ as $v_{\boldsymbol{f}}(S) = \langle \boldsymbol{f}_N, \boldsymbol{f}_S \rangle / \|\boldsymbol{f}_N\|_2$, where $\|\boldsymbol{f}_N\|_2$ was the L2-


among the year 's most intriguing explorations of alientation.
a solid examination of the male midlife crisis .
the best film about 2 baseball to hit theaters since field of dreams .
the film will play equally well on both the standard and giant screens .
it does n ' t believe in itself , it has no sense of humor . . . it ' s just plain bored.
that is a compliment to kuras and miller.
overall very good for what it 's trying to do .
a good piece of work more often than not .
movie fans , get ready to take off . . . the other direction .
it ' s a beautiful madness . (1)

it ' s a remarkably solid and subtly satirical tour de force .
this is a good script , good dialogue , funny even for adults
dull, lifeless, and amateurishly assembled .
a warm but realistic meditation on friendship , family and affection .
no telegraphing is too obvious or simplistic for this movie .
a tender , witty , captivating film about friendship , love , memory , trust and loyalty
a dumb movie with dumb characters doing dumb things and you have to be really dumb not to see where this is going .
it ' s a charming and often affecting journey .
entertains by providing good , lively company .
complete lack of originality , cleverness or even visible effort
a warm, funny, engaging film
just as moving , uplifting and funny as ever . (2)

makes for a pretty unpleasant viewing experience .
a strangely compelling and brilliantly acted psychological drama .
preaches to two completely different choirs at the same time , which is a pretty amazing accomplishment .
a wildly inconsistent emotional experience .
a tender , heartfelt family drama . (3)

Figure 4: Prototype features modeled by the BERT trained using the SST-2 dataset. Grey box indicated words in A. We used different colors to indicate the extracted coalitions. Words in the same coalition formed a prototype feature modeled by the DNN. The set of words A was randomly selected.


maximum (prototypes towards incorrect predictions): if steven soderbergh ' s ' solaris ' is a failure it is a glorious failure . predict: negative
minimum (prototypes towards correct predictions): if steven soderbergh ' s ' solaris ' is a failure it is a glorious failure . label: positive

maximum (prototypes towards incorrect predictions): the longer the movie goes , the worse it gets , but it ' s actually pretty good in the first few minutes. predict: positive
minimum (prototypes towards correct predictions): the longer the movie goes , the worse it gets , but it ' s actually pretty good in the first few minutes. label: negative

maximum (prototypes towards incorrect predictions): on the heels of the ring comes a similarly morose and humorless horror movie that , although flawed , is to be commended for its straight - ahead approach to creepiness . predict: negative
minimum (prototypes towards correct predictions): on the heels of the ring comes a similarly morose and humorless horror movie that , although flawed , is to be commended for its straight - ahead approach to creepiness . label: positive

maximum (prototypes towards incorrect predictions): on the heels of the ring comes a similarly morose and humorless horror movie that , although flawed , is to be commended for its straight - ahead approach to creepiness . predict: negative
minimum (prototypes towards correct predictions): on the heels of the ring comes a similarly morose and humorless horror movie that , although flawed , is to be commended for its straight - ahead approach to creepiness . label: positive

Figure 5: Positive and negative interaction effects of sentences, which were mistakenly classified by the DNN. These results reflect prototype features towards correct and incorrect predictions, and representation flaws of the DNN.

norm and was used for normalization. Figure 3 (c) shows the significance of interactions computed with output features of different layers. The significance of interactions among a set of input variables increased along with the layer number. This phenomenon showed that the prototype feature was gradually modeled in deep layers of the DNN.

Mining prototype features towards incorrect predictions of DNNs: The proposed method could be used to analyze sentences, which were mistakenly classified by the DNN. We used the BERT learned using the SST-2 dataset. Figure 5 shows two partitions that maximized and minimized the Shapley value of coalitions for each sentence. For the partition that maximized the Shapley value, the generated coalitions were usually used to explain prototype features toward incorrect predictions of the DNN. In comparison, coalitions generated by minimizing the Shapley value represented prototype features towards correct predictions. For example, we tested two different sets of input variables A for the sentence "on the heels of ...... to creepiness.". This sentence was incorrectly predicted to be negative by the DNN. As Figure 5 shows, the DNN used "humorless horror movie" as a prototype features, which led to the incorrect negative prediction. In comparison, if we minimized the Shapley value, we got the coalition "is to be recommanded for its straight-ahead approach to creep", which towards the correct positive prediction.

Conclusion

In this paper, we have defined the multivariate interaction in the DNN based on game theory. We quantify the interaction among multiple input variables, which reflects both positive and negative interaction effects inside these input variables. Furthermore, we have proposed a method to approximate the interaction efficiently. Our method can be applied to various DNNs for different tasks in NLP. Note that the quantified interaction is just an approximation of the accurate interaction in Equation (5). Nevertheless, experimental results have verified high accuracy of the approximation. The proposed method can extract prototype features modeled by the DNN, which provides a new perspective to analyze the DNN.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (61906120 and U19B2043).

Ethical Impact

This study has broad impacts on the understanding of signal processing in DNNs. Our work provides researchers in the field of explainable AI with new mathematical tools to analyze DNNs. Currently, existing methods mainly focus on interactions between two input variables. Our research proposes a new metric to quantify interactions among multiple input variables, which sheds new light on the understanding of prototype features in a DNN. We also develop a method efficiently approximate such interactions. As a generic tool to analyze DNNs, we have applied our method to classic DNNs and have obtained several new insights on signal processing encoded in DNNs for NLP tasks.

References

Achille, A.; and Soatto, S. 2018. Emergence of invariance and disentanglment in deep representations. *The Journal of Machine Learning Research* 19(1): 1947–1980.

Ancona, M.; Oztireli, C.; and Gross, M. 2019. Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Values Approximation. *arXiv:1903.10992* .

Bien, J.; Taylor, J.; and Tibshirani, R. 2013. A LASSO FOR HIERARCHICAL INTERACTIONS. *Annals of Statistics* 41(3): 1111–1141.

Binder, A.; Montavon, G.; Lapuschkin, S.; Muller, K.; and Samek, W. 2016. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks (ICANN)*, 63–71.

Castro, J.; Gómez, D.; and Tejada, J. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research* 36(5): 1726–1730.

Chen, J.; Song, L.; Wainwright, M. J.; and Jordan, M. I. 2019. L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data. In *ICLR*.

Cheng, X.; Rao, Z.; Chen, Y.; and Zhang, Q. 2020. Explaining Knowledge Distillation by Quantifying the Knowledge. In *CVPR*.

Chung, J.; Ahn, S.; and Bengio, Y. 2017. Hierarchical Multiscale Recurrent Neural Networks. In *ICLR*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Dhamdhere, K.; Agarwal, A.; and Sundararajan, M. 2019. The Shapley Taylor Interaction Index. *arXiv: 1902.05622* .

Dyer, C.; Kuncoro, A.; Ballesteros, M.; and Smith, N. A. 2016. Recurrent Neural Network Grammars. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 199–209.

Fort, S.; Nowak, P. K.; Jastrzebski, S.; and Narayanan, S. 2019. Stiffness: A new perspective on generalization in neural network. *arXiv preprint arXiv: 1901.09491* .

Grabisch, M.; and Roubens, M. 1999. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory* 28: 547–565.

Guan, C.; Wang, X.; Zhang, Q.; Chen, R.; He, D.; and Xie, X. 2019. Towards A Deep and Unified Understanding of Deep Neural Models in NLP. In *ICML*, 2454–2463.

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C. P.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *In ICLR* .

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Janizek, J. D.; Sturmfels, P.; and Lee, S. 2020. Explaining Explanations: Axiomatic Feature Interactions for Deep Networks. *arXiv:2002.04138* .

Jin, X.; Du, J.; Wei, Z.; Xue, X.; and Ren, X. 2019. Towards Hierarchical Importance Attribution: Explaining Compositional Semantics for Neural Sequence Models. *arXiv:1911.06194* .

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Li, J.; Chen, X.; Hovy, E.; and Jurafsky, D. 2015. Visualizing and Understanding Neural Models in NLP. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Li, J.; Monroe, W. S.; and Jurafsky, D. 2016. Understanding Neural Networks through Representation Erasure. *arXiv: 1612:08220* .

Liang, R.; Li, T.; Li, L.; Wang, J.; and Zhang, Q. 2020. Knowledge Consistency between Neural Networks and Beyond. In *ICLR*.

Lundberg, S.; Erion, G. G.; and Lee, S. 2018. Consistent Individualized Feature Attribution for Tree Ensembles. *arXiv: Learning* .

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, 4765–4774.

Murdoch, W. J.; Liu, P. J.; and Yu, B. 2018. Beyond word importance: Contextual decomposition to extract interactions from LSTMs. *arXiv:1801.05453* .

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* .

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. *In KDD* .

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

Shen, Y.; Tan, S.; Sordoni, A.; and Courville, A. 2019. Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks. In *ICLR*.

Shrikumar, A.; Greenside, P.; Shcherbina, A.; and Kundaje, A. 2016. Not just a black box: Learning important features through propagating activation differences. *In arXiv:1605.01713* .

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *In arXiv:1312.6034* .

Singh, C.; Murdoch, W. J.; and Yu, B. 2018. Hierarchical interpretations for neural network predictions. *arXiv:1806.05337* .

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Sorokina, D.; Caruana, R.; Riedewald, M.; and Fink, D. 2008. Detecting statistical interactions with additive groves of trees. In *ICML*, 1000–1007.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2014. Striving for simplicity: The all convolutional net. *In arXiv:1412.6806* .

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *ICML*, 3319–3328.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *International Joint Conference on Natural Language Processing*, volume 1, 1556–1566.

Tsang, M.; Cheng, D.; and Liu, Y. 2018. Detecting Statistical Interactions from Neural Network Weights. In *ICLR*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, X.; Tu, Z.; Wang, L.; and Shi, S. 2019. Self-Attention with Structural Position Representations. In *International Joint Conference on Natural Language Processing*, 1403–1409.

Wang, Y.-S.; Lee, H.-Y.; and Chen, Y.-N. 2019. Tree Transformer: Integrating Tree Structures into Self-Attention. *arXiv preprint arXiv:1909.06639* .

Warstadt, A.; Singh, A.; and Bowman, S. R. 2018. Neural Network Acceptability Judgments. *arXiv preprint arXiv:1805.12471* .

Weber, R. J. 1988. Probabilistic values for games. *The Shapley Value. Essays in Honor of Lloyd S. Shapley* 101–119.