A Causality Inspired Framework for Model Interpretation

Chenwang Wu
University of Science and Technology of China
Hefei, Anhui, China
wcw1996@mail.ustc.edu.cn

Xiting Wang∗
Microsoft Research Asia
Beijing, China
xitwan@microsoft.com

Defu Lian∗
University of Science and Technology of China
Hefei, Anhui, China
liandefu@ustc.edu.cn

Xing Xie
Microsoft Research Asia
Beijing, China
xing.xie@microsoft.com

Enhong Chen
University of Science and Technology of China
Hefei, Anhui, China
cheneh@ustc.edu.cn

ABSTRACT
A critical issue in eXplainable Artificial Intelligence (XAI) is determining whether explanations uncover the underlying causal factors for model behavior or merely show coincidental relationships. Failing to make this distinction can lead to incorrect understandings. To address this issue, we first understand the model interpretation through a causal lens. We find that the explanation scores of certain representative explanation methods align with the concept of average treatment effect in causal inference and evaluate their relative strengths and limitations from a unified causal perspective. Based on our observations, we outline the major challenges in applying causal inference to model interpretation, including identifying common causes that can be generalized across instances and ensuring that explanations provide a complete causal explanation of model predictions. We then present CIMI, a Causality-Inspired Model Interpreter, which addresses these challenges. CIMI has three modules: the causal sufficiency module and the causal intervention module ensure the explanations are both causally sufficient and generalizable, while the causal prior module facilitates easy learning. Our experiments show that CIMI provides superior and generalizable explanations and is useful for debugging and improving models.

CCS CONCEPTS
• Computing methodologies → Machine learning.

KEYWORDS
Interpretability, causal inference, machine learning.

ACM Reference Format:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Woodstock ’18, June 03–05, 2018, Woodstock, NY.
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06 . . $15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION
Although deep learning has played key roles in many fields, such as medical insurance [15], financial predictive analysis [12], and fault detection [13], deep models are mostly black boxes (complex functions that humans cannot understand) and cannot be predicted in a way that humans can understand. This opacity could have serious consequences, especially in high-stakes decisions [36]. For example, deep learning-based pollution models claim that highly polluted air poses no threat to human health [28]. Imperfect deep models are not meaningless. If the reason why the model makes a specific decision can be explained, the risk of model failure can be effectively reduced and avoided. In addition, interpretation has the potential to uncover potential errors in the model (e.g., the logic of the model does not match domain knowledge), which can also help to further improve the model [44]. Therefore, it is very necessary to study eXplainable Artificial Intelligence (XAI).

A fundamental question in XAI is: do explanations reveal important root causes of the model’s behavior or merely spurious correlations? The inability to distinguish correlation from causality can result in erroneous explanations for decision-makers. The importance of causality is further highlighted by prominent research in human-computer interaction [35], in which extensive user studies reveal that in XAI, causality increases user trust and helps evaluate the quality of explanations. This result echoes major theories in cognitive science which present that humans build mental models of the world by using causal relationships [27, 40].

XAI provides an ideal environment for causality studies due to its adherence to fundamental causality assumptions, which are usually difficult to verify in other settings. For example, in XAI, we can obtain a set of variables (e.g., input data and model parameters) that construct a complete set of possible causes for model prediction, which ensures the satisfaction of the essential causal sufficiency assumption [30, 37]. In addition, the black-box models to be studied can be easily intervened, allowing the vital do-operator to be performed directly without any further assumptions such as ignorability or exchangeability. In contrast, the inability to perform different do-operators in the same instance is the fundamental problem of causality inference in more general scenarios [30].
Due to its importance and applicability, causality has attracted increasing attention in XAI. Multiple explanation methods [23, 34, 37] utilize causal analysis techniques such as interventions (e.g., input data perturbation), and some have achieved noteworthy success in delivering more trustworthy explanations. Despite this, a formal and unified causal perspective for explainability remains lacking and some key research questions remain challenging to answer, for example:

- **RQ1**: Can the existing explanation methods be framed within a theoretical causal framework? If so, what are the causal models employed, and what distinguishes them from each other?
- **RQ2**: What are the major challenges in leveraging causal inference for model interpretation and what benefits we may achieve by solving these challenges?
- **RQ3**: How can the causal model be improved to overcome these challenges?

In this paper, we aim to bridge the gap between causality and explainability by studying these issues.

We first provide a causal theoretical interpretation for explanation methods including LIME [34], Shapley values [23], and CXPlain [37] (RQ1). Our analysis shows that their explanation scores correspond to (average) treatment effect [30] in causal inference, and they share the same causal graph, with only small differences such as the choices of the treatment (i.e., the perturbed features). This provides a unified view for understanding the precise meaning of their explanations and provides theoretical evidence about their advantages and limitations.

These observations allow us to summarize the core challenge in applying causal inference for model interpretation (RQ2). While it is easy for explanation methods to compute individual causal effects, e.g., understanding how much the model prediction will change when one input feature changes, the core challenge is how to efficiently discover prominent common causes that can be generalized to different instances from a large number of features and data points. Addressing this issue requires ensuring that the explanations can (1) generalize to different instances and are (2) causal sufficiency for understanding model predictions.

To solve the above challenges (RQ3), we follow important causal principles, and propose Causality Inspired Model Interpreter (CIMI). Specifically, we first discuss different choices of causal graphs for model interpretation and identify the one that can address the aforementioned challenges. Based on the selected causal graph, we devise training objectives and desirable properties of our neural interpreters following important causal principles. We then show how these training objectives and desirable properties can be achieved through our CIMI framework. Specifically, CIMI consists of three modules: Causal Sufficiency Module and Causal Intervention Module address the challenges of causal sufficiency and generalizability respectively, while the Causal Prior Module facilitates easier learning for the interpreter.

Finally, we conduct extensive experiments on four datasets and nine explanation methods. The results consistently show that CIMI significantly outperforms baselines on both causal sufficiency and generalizability metrics on all datasets. Notably, we also show the potential of the explanation to remove shortcuts and improve the model.

### 2 REVISITING XAI FROM CAUSAL PERSPECTIVE

#### 2.1 Preliminary about Causal Inference

We follow the common terminologies in causal inference [30] to discuss existing and our explanation methods.

**Causal graph** is used to formally depict causal relations. In the graph, each node is a random variable, and each direct edge represents a causal relation, which means that the target node (child) can change in response to the change of the source node (parent).

**Do-operator** is a mathematical operator for intervention. In general, applying a do-operator $do(E = e)$ on a random variable $E$ means that we set the random variable to value $e$. For example,

- $P(Y = y | do(E = e))$ is the probability that $Y$ is $y$ when in every instance, $E$ is assigned to $e$. This is a global intervention that happens to the whole population. In comparison, $P(Y = y | E = e)$ denotes the probability that $Y$ is $y$ on the subpopulation where $E$ is observed to be $e$.
- $P(Y = y | do(E = e), C = c)$ applies do-operator to the subpopulation where the random variable $C$ has value $c$.

**Treatment effect** is an important method to quantify how much causal effect a random variable $E$ has on $Y$. Suppose that $E$ is a binary value, the average treatment effect $T$ of $E$ on $Y$ is:

$$T(Y | do(E)) = \mathbb{E}_c(Y | do(E = e), C = c)$$

$$= \mathbb{E}_c(Y | do(E = 1), C = c) - \mathbb{E}(Y | do(E = 0), C = c),$$

where $Y | do(E = e), C = c$ represents the value of $Y$ when $E$ is set to $e$ and all other causes $C$ is fixed to $c$.

#### 2.2 Causal Graph of Existing XAI Methods

![Figure 1: Comparison of causal graphs for model interpretation](image)

(a) The causal graph for existing methods, in which explanation $E$ is not the sole cause for model prediction $Y$; (b) Another causal graph, in which explanations are causally sufficient for prediction, but not generalizable; (c) Our proposed causal model which explanation $E$ is generalizable and modeled as the only cause for $Y$. Observed variables are shaded.

Revisiting existing methods from the causal perspective allows us to show that many well-known perturbation-based methods such as LIME [34], Shapley values [23], and CXPlain [37] actually compute or learn the treatment effect, and that their causal graph corresponds to the one shown in Fig. 1(a). While a formal theoretical analysis is given in Appendix A.1, here we briefly summarize the commonalities and differences among these XAI methods by presenting the main intuition behind the mathematical analysis.

In the causal graph shown in Fig. 1(a), $E$ corresponds to the specific treatment, characterized by one feature (or a set of features) to
be perturbed. By \( \text{do}(E = 1) \), these methods include the feature in the input, while \( \text{do}(E = 0) \) does the opposite. Then, they obtain the model’s outcome \( \hat{Y} \) when \( E \) is changed and compute the treatment effect \( T(\hat{Y}(\text{do}(E))) = \hat{Y}(\text{do}(E = 1), C = c) - \hat{Y}(\text{do}(E = 0), C = c) \), where \( C \) denotes the context concerning \( E \), or more intuitively, the features that remain unchanged after changing \( E \). The treatment effect then composes (or is equal to) the explanation weight, revealing the extent to which the feature can be considered in the explanation, or “contribution” for each feature in the model prediction.

Although all three methods can be summarized using the framework in Fig. 1(a), they differ a little in terms of the following aspects. It is worth emphasizing that we will see how this unified view allows us to easily compare the pros and cons of each work.

- **Intervened features** \( E \). CXPlain and Shapley value only consider one feature as \( E \) while \( \text{CF} \) uses a set of features as \( E \) for testing. Thus, the former two methods cannot measure the causal contribution of a set of features without further extension or assumptions.
- **Context** \( C \). Shapley values consider all subsets of features as possible context, while the other methods take the input instance \( x \) as the major context. Accordingly, Shapley values compute the average treatment effect on all contexts (i.e., all possible subsets of features) while others consider individual treatment effects. While individual treatment effects may be computed more efficiently and have a more precise meaning, their ability to generalize to similar inputs may be significantly reduced.
- **Model output** \( \hat{Y} \). Most methods track changes in model predictions, while CXPlain observes how input changes the error of the model prediction. Thus, CXPlain may be more useful for debugging, while the others may be more suitable for understanding model behavior.

More details about revisiting perturbation-based XAI methods from a causal lens can be found in Appendix A.1, including the relationship between their learning objectives and (average) treatment effect, the correspondence between their causal graph and Fig. 1(a), and their comparison from a unified causal view (Table 3).

3 METHODOLOGY

3.1 Causal Graph

Causally insufficiency of explanations in Fig. 1(a). From the previous section, we have seen that existing work adopts the causal graph in Fig. 1(a). The major issue of this framework is that the model prediction \( \hat{Y} \) is determined by both the explanation and the non-explanation, where \( E = x_\mathcal{U} \) means that the explanation for \( X = x \) is \( x_\mathcal{U} \), and \( U = x_\mathcal{U} \) means that the explanation for \( X = x \) is \( x_\mathcal{U} \). In both causal graphs, \( \hat{Y} \) has the only parent (cause), which is the explanation, making the explanation sufficient for understanding the model prediction.

**Issue of explanations’ generalizability.** While both causal graphs allow explanations to be the only cause of model predictions, Fig. 1(b) fails to model the explanation’s generalizability: in this causal graph, the explanation may change in arbitrary ways when \( X \) changes. Since a core for deep models to perform well is their generalizability, it would be problematic if our XAI method cannot model such generalizability or capture these more generalizable features. Also, it would be more user-friendly if we can capture the explanations that are common and invariant to changes, as they allow users to quickly identify prominent common causes.

**Our choice.** Considering the above, we choose the causal graph in Fig. 1(c), which resembles the Domain Generalization causal graph [24] (see Appendix A.2 for a detailed comparison) and follows its common cause principle to build a shared parent node (in our case \( E \)) for two statistically dependent variables (in our case \( X \) and \( \hat{Y} \)). In the causal graph, it is evident that alterations to non-explanatory variable \( U \) have no impact on the explanation \( E \) or the prediction \( \hat{Y} \), only resulting in slight variations in \( X \). This demonstrates the stability of the explanation across different instances of \( X \) and its sufficiency as a cause for the model prediction \( \hat{Y} \), as \( E \) is the only determining factor (parent) for \( \hat{Y} \).

3.2 Problem Formulation with Causal Inference

Given the causal graph in Fig. 1(c), we aim to learn unobserved causal factors \( E \) and \( U \), where \( E \) denotes the generalizable causal explanation for model prediction, and \( U \) denotes the non-explanations.

Following the common assumption of existing feature-attribution-based explanations, we assume that \( E \) and \( U \) could be mapped into the input space of \( X \). More specifically, we assume that \( E \) is the set of features in \( X \) that influences \( \hat{Y} \), while \( U = X \setminus E \) is the other features in \( X \) that are not included in \( E \). Equivalently, \( E \) and \( U \) can be represented by learning masks \( M \) over \( X \):

- \( E = M \odot X \), where \( \odot \) is element-wise multiplication, and \( M_i = 1 \) means that the \( i \)-th feature in \( X \) is included in the explanation.
- \( U = (1 - M) \odot X \), where \( M_i = 0 \) means the \( i \)-th feature in \( X \) is included in the non-explanation.

Our goal is to learn a function \( g : X \rightarrow M \) that inputs an instance \( X = x \) and outputs the masks representing the causal factors \( E \) and non-causal factor \( U \). Function \( g \) is called the interpreter in this paper.

3.3 Optimization Principles and Modules

It is impractical to directly reconstruct the causal mechanism in the causal graph of Fig. 1(c) since important causal factors are
unobservable and ill-defined [24]. However, causal factors in causal graphs need to follow clear principles. We use the following two main principles in causality inference to devise training objectives and desirable properties.

Principle 1. Humean’s Causality Principle [10]: There exists a causal $x_i \rightarrow y$ if the use of all available information results in a more precise prediction of $y$ than using information excluding $x_i$, all causes for $y$ are available, and $x_i$ occurs prior to $y$.

Principle 2. Independent Causal Mechanisms Principle [32]: The conditional distribution of each variable given its causes (i.e., its mechanism) does not inform or influence the other mechanisms.

Accordingly, we design three modules to ensure that the extracted explanations (causal factors) satisfy the basic properties required by Principles 1 and 2.

- **Causal Sufficiency Module.** Following Principle 1, we desire to discover $E$ that is causally sufficient for $\hat{Y}$ by ensuring that $E$ contains all the information to predict $Y$ and explaining the dependency between $X$ and $\hat{Y}$. Similarly, we also ensure that $U$ is causally insufficient for predicting $\hat{Y}$.

- **Causal Intervention Module.** Following Principle 2, we ensure that $U$ and $E$ are independent by intervening $U$ and guarantee that the learned $g(X) = E$ will not change accordingly. This also allows us to find explanations that can better be generalized to varying cases.

- **Causal Prior Module.** Following Principle 1, we facilitate the learning of explanations by using potential causal hints as inputs to the interpreter and weakly supervise over its output causal masks $M$. These learning priors enable faster and easier learning, as well as avoiding trivial solutions that cannot be avoided by the first two modules.

3.3.1 Causal Sufficiency Module. According to Principle 1, to ensure that $E$ is a sufficient cause of $\hat{Y}$, it is necessary to guarantee that $E$ is the most suitable feature for predicting $\hat{Y} = f(X)$, rather than other features $U$. In other words, $x_e$ can always predict $f(x)$ through an optimal function $f'$ that maps explanation $x_e$ to $f(x)$, while non-explanation $x_u$ cannot give meaning information for predicting $f(x)$. Accordingly, the causal sufficiency loss can be modeled as follows

$$L_{c'} = \min_{f'} \mathbb{E}_{x}(\ell(f(x), f'(x_e)) - \ell(f(x), f'(x_u))), \quad (2)$$

where $\ell(\cdot)$ is the mean squared error loss, $x_e = g(x) \odot x$, $x_u = (1 - g(x)) \odot x$, and $x$ is sampled from the entire model input space.

In practice, finding the optimal $f'$ directly is very difficult due to the vast and sometimes even continuous input space. The interaction between optimizing $f'$ and the interpreter $g$ may also easily lead to unstable training and difficulty in converging to an optimal solution [45]. To address this issue, we approximate the optimal $f'$ by using $f$, under the assumption that the difference between $f'$ and $f$ is minimum, considering that explanation $x_e$ is in the same space with the origin model inputs $X$. By setting $f'$ to $f$, we are actually minimizing each individual treatment effect, which has a precise causal meaning. Besides, since we do not have to optimize $f'$, it may allow us to sample much fewer samples $x'$ to optimize $L_{c'}$ and learn an interpreter $g$. In summary, the causal sufficiency loss $L_{c}$ is rewritten as follows

$$L_{c} = \mathbb{E}_{x}(\ell(f(x), f(x_e)) - \ell(f(x), f(x_u))), \quad (3)$$

where $x_e = g(x) \odot x$, and $x_u = (1 - g(x)) \odot x$. Details about different methods for sampling $x$ and a comparison of them is presented in Appendix A.4.5.

3.3.2 Causal Intervention Module. Following Principle 2, we desire $U$ and $E$ to be independent, which makes it possible to find the invariant explanations of neighboring instances and improve the interpreter’s generalizability. Despite the lack of true explanations for supervised training, we have the prior knowledge that the learned interpreter $g$ should be invariant to the intervention of $U$, that is the $do(U)$ does not affect $E$. Based on this prior knowledge, we design a causal intervention loss to separate explanations.

First, we describe how to intervene on $U$. Following the common practice [24] in causal inference, we perturb the non-explanation $x_u$ via a linear interpolation between the non-explanation positions of the original instance $x$ and another instance $x'$ sampled randomly from $X$. The intervention paradigm is shown as follows:

$$x_{int} = g(x) \odot x + (1 - g(x)) \odot ((1 - \lambda) \cdot x + \lambda \cdot x'),$$

where $\lambda \sim U(0, \epsilon)$, and $\epsilon$ limits the magnitude of perturbation. Furthermore, we can optimize the following causal intervention loss to ensure that $U$ and $E$ are independent.

$$L_{I} = \mathbb{E}_{x} \ell(g(x), g(x_{int})). \quad (5)$$

This loss ensures that the generated explanations do not change before and after intervening in non-explanations. This invariant property guarantees local consistency of explanations. i.e., interpreters should generate consistent explanations with neighboring (or similar) data points. This coincides with the smooth landscape assumption of the loss function of the deep learning model [21], which may help to capture more generalizable features and improve the generalizability of the interpreter.

3.3.3 Causal Prior Module. To facilitate the learning of the interpreter, we 1) inject potential causal hints into the neural network of the interpreter, and 2) design a weakly supervision loss on the output causal masks $M$.

**Interpreter neural network design.** A core challenge in XAI is that there lack of prior knowledge about which architecture should be used for the interpreter [37]. When we learn an interpreter with the neural network, it is difficult to decide which neural network structure should be used. If the architecture of $g$ is not as expressive and complex as the black-box model $f$, then how we can be sure that $g$ has the ability to understand the original black-box $f$? If $g$ could be more complicated than $f$, then it is prone to slow training efficiency and overfitting.

Our solution to this problem is inspired by Principle 1, which states that causes (model $f$) are more effective in predicting the effects (explanation $x_e$). Hence, we generate the explanation $x_e$ by directly utilizing the parameters of the black-box model $f$. To achieve this, we use the encoding part of the black-box model $f$ (denoted as $f_d$) as the encoder in our interpreter model $g$. The decoder of $g$ is a simple neural network, denoted as $\phi$. The ease of learning
Figure 2: The framework of CIMI. The only trainable component is the decoder \( \phi \), which is a simple neural network that can be trained with a relatively small number of samples. The causal sufficiency loss \( L_\phi \) ensures that the generated explanations are causally sufficient for model prediction. In the interpreter, the causal intervention loss \( L_\gamma \) ensures the independence of explanations and non-explanations, and the weakly supervising loss \( L_p \) makes learning easier.

is supported by information bottleneck theory, which states that information in each layer decreases as we progress through the model [11]. Therefore, the input \( x \) contains the most information, while \( f(x) \) contains less information as the information deemed unnecessary for prediction has been removed. The final prediction and ground-truth explanation use the least amount of information. Consequently, compared with \( X \), the last embedding layer output \( f_e(X) \) is a better indicator to find the explanation.

Based on this observation, we design \( \phi \) so its input concatenates the encoded embedding \( f_e(x) \in \mathbb{R}^{|x| \times d} \) and the original instance embedding \( e_x \in \mathbb{R}^{|x| \times d} \) along the axis \( 1 \), i.e., \( \{f_e(x); e_x\}_1 \in \mathbb{R}^{|x| \times 2d} \), where \( d \) is the dimension of embedding, and the operator \([a; b]\) denotes the axis \( i \) along which matrix \( a \) and \( b \) will be joined. Therefore, the decoder \( \phi \) maps input \( \{f_e(x); e_x\}_1 \in \mathbb{R}^{|x| \times 2d} \) to \([0, 1]^{|x| \times 1}\), and the \( i \)-th dimension of the output represents the probability that the token \( i \) is used for explanation. \( \phi \) can be any neural network, and we use the structure of 1-layer LSTM+MLP in this paper. In summary, the interpreter \( g \) can be reformulated as

\[
g(x) = \phi([f_e(x); e_x]_1).
\]  

(6)

It is worth emphasizing that by setting the encoder in \( g \) as \( f_e \), the architecture of \( g \) could be as complicated as \( f_e \), and such a complex structure helps to fully understand the model’s decision-making mechanism. \( g \) can also be considered simple, because the parameters of \( f_e \) in \( g \) are fixed and only the decoder \( \phi \) is learnable, while only requiring a few additional parameters (1-layer LSTM+MLP used in the paper), avoiding the issues of overfitting and large training cost. We also find in the experiment (Section 4.4) that this design of interpreter is more effective.

Weakly supervising loss. Without a further regularization loss on the causal factors, there exists a trivial solution (i.e., all explanation masks set to 1) that makes the interpreter collapse. A common regularization in causal discovery is sparsity loss which requires the number of involved causal factors to be small [4]. However, this sparsity loss may fail to adapt to the different requirements of different instances, as the constraints are the same for complicated sentences and simple sentences. Therefore, this poses difficulty in tuning the hyper-parameters for different datasets.

To tackle this issue, we leverage noisy ground-truth labels as a prior for the causal factor \( E \) to guide the learning process. Our approach is based on the intuition that the explanation for \( x \) should contain more information about \( x \) itself than information about another instance \( x' \). Using this, we derive a weakly supervision loss by maximizing the probability that the token in instance \( x \) is included in \( x_e \) while minimizing the probability that a token not in \( x \) (noise) is predicted to be the explanation:

\[
L_p = \mathbb{E}_{x, x'} \log \sigma(g(x) - g(x')) ,
\]  

(7)

where \( g(x) \) means to map \( f_e(x) \) to \( x' \) in \( g(x) \), refer to Eq. 6, that is, \( g(x) = \phi([f_e(x); e_x]_1) \). Correspondingly, \( g(x) = \phi([f_e(x); e_x]_1) = g(x) \), and the subscript in \( g(x) \) are omitted for simplicity.

This weakly supervising loss prevents the interpreter from overly optimistically predicting all tokens as explanations, which helps alleviate trivial solutions. We verify its effectiveness in Section 4.4.

3.3.4 Overall Framework and Optimization.

Overall loss function. Combining the above three modules, the overall optimization objective of CIMI is summarized as follows and the framework is shown in Fig. 2.

\[
\min_{\phi} L_{\phi} + \alpha L_{\gamma} + L_p ,
\]  

(8)

where \( \alpha \) is the trade-off parameter. Notably, the introduction of weakly supervising loss is to avoid the difficulty of tuning regularization parameters, so this term does not require trade-off parameters.

Analysis of the framework. As shown in Fig. 2, the only trainable parameter in our framework is the simple decoder in the interpreter \( g \), which uses a 1-layer LSTM (hidden size is 64) and 1-layer
MLP (input features are 64, and the output features are 2). This enables us to learn the interpreter efficiently with a small number of forward propagations through \( f \). The validity of our framework can be further verified by considering the information bottleneck theory, which says that during the forward propagation, a neural network gradually focuses on the most important parts in the input by filtering information that is not useful for prediction through the layer [11]. Let us denote information in a random variable \( Z \) as \( I(Z) \), then we have \( I(X) \geq I(f(X)) \geq I(X_e) \geq I(f(X_e)) \) according to the architecture in the framework, meaning that the explanations cannot have more information than \( I(f(X)) \). Then, we enforce that \( I(X_e) \geq I(f(X_e)) = I(f(X)) \) by using the causal sufficiency loss \( L_s \), thus, \( I(f(X)) \geq I(X_e) \geq I(f(X)) \). The possible noise in \( I(f(X_e)) \) is then removed from \( I(X_e) \) by using the intervention loss \( L_i \) and probing loss \( L_p \), allowing us to approximately have \( I(f(X)) \geq I(X_e) \geq I(f(X)) \) and thus \( I(X_e) \approx I(f(X)) = I(\hat{Y}) \).

**Differentiable training.** The output of the interpreter is a probability vector \( g(x) \in [0,1]^{\|X\|} \) that each token belongs to the explanation. In order to perform end-to-end training \( X \rightarrow E \rightarrow \hat{Y} \), we do not discretize the probability vector to obtain the explanation, but use the element-wise multiplication of the mask vector \( g(x) \) and the token embeddings of the input as \( E \).

### 4 EXPERIMENTS

#### 4.1 Experimental Setup

##### 4.1.1 Datasets.**

We use four datasets from the natural language processing domain, including Clickbait [2], Hate [7], Yelp [30], and IMDB [25]. See Appendix A.3.1 for their details.

##### 4.1.2 Black-box Models.

Although pre-training models are brilliant in many fields, it is difficult to answer what information they encode. For this reason, we choose two well-known pre-training models, BERT [8] and RoBERTa [22] (See Appendix A.3.2 for the specific details), as the black box models to be explained. Notably, the main body shows the experimental results on BERT, and the results of RoBERTa can be found in Appendix A.4.1 and A.4.2.

##### 4.1.3 Baselines.**

We compare the proposed CIMI with nine baselines, including Gradient [41], Attention [3], LIME [34], KernelSHAP [23], Rationale [20], Probing [1], CXPlain [37], AXAI [33], Smask [21]. Their details can be found in Appendix A.3.3.

##### 4.1.4 Evaluation Metrics.

First, we evaluate the causal sufficiency using three faithful metrics (see Appendix A.3.4 for more details).

- **Decision Flip-Fraction of Tokens (DFOTT) [38],** which measures the minimum fraction of important tokens that need to be erased in order to change the model prediction.
- **Comprehensiveness (COMP) [9],** which measures the faithfulness score by the change in the output probability of the original prediction class after the important tokens are removed.
- **Sufficiency (SUFF) [9],** in contrast to COMP, it only keeps important tokens and compares the changes in output probabilities over the original predicted class. Notably, this metric is not equivalent to the causal sufficiency we focus on.

The number of important tokens is selected from \( \{1, 5, 10, 20, 50\} \) and the average performance is taken. Notably, SUFF and COMP have been proven to be more faithful to the model prediction than other metrics [5]. In addition to the above metrics, we use AvgSen to measure the explanation’s generalizability.

**Average Sensitivity (AvgSen) [46],** which measures the average sensitivity of an explanation when the input is perturbed. In our experiments, we replace 5 tokens per instance and calculate the sensitivity of top-10 important tokens.

#### 4.1.5 Parameter Settings.

For the two pre-training models used, BERT and RoBERTa, we both add two-layer MLP as decoders for downstream tasks. In all four datasets, the optimization is based on Adam with a learning rate of \( 1 \times 10^{-5} \). The training epoch is 20, and the batch size is 8. For the proposed CIMI, without special instructions, we train 100 epochs on Clickbait and Hate, and 50 epochs on the other two larger datasets to improve efficiency. For the trade-off parameter \( \alpha \), set it to 1, 1, 0.1 on Clickbait, Hate, Yelp, and IMDB respectively. In addition, the perturbation magnitude \( \epsilon \) in the causal intervention module is set to 0.2. The source code of CIMI is available at https://anonymous.4open.science/r/CIMI.

### 4.2 Faithfulness Comparison

In this section, we evaluate the causal sufficiency of the explanations using faithfulness metrics. Table 1 summarizes the average results of 10 independent repeated experiments.

First, it can be seen that the proposed method achieves the best or comparable results compared with the baselines on various datasets. In particular, this improvement is more pronounced on more complex datasets (from Clickbait \( \rightarrow \) IMDB). For example, the improvement over the best baselines reaches 119% on IMDB w.r.t. DFOTT metric. Such invaluable property could adapt to the complex trend of black-box models. This gratifying result verifies that CIMI can generate explanations that are more faithful to the model. Second, Gradient has impressive performance in some cases, which indicates that their linear assumption can reflect the model’s decision-making process to some extent. Third, among the perturbation-based methods, LIME, KernelSHAP, and CXPlain all show satisfactory performance. Especially LIME based on local linear approximation, which once again verifies the rationality of the first finding, the model linear assumption.

In addition, we also illustrate the performance w.r.t. COMP under different explanation lengths, as shown in Fig. 3 (similar findings can be obtained when concerning SUFF, see Appendix A.4.3). The experimental results show that regardless of explanation length, CIMI exhibits significant competitiveness. The above results demonstrate the power of causal principle constraints in CIMI for understanding model predictions.

### 4.3 Generalizability Comparison

This section uses AvgSen metric to evaluate the generalizability of the generated explanations to neighboring (similar) samples. It is undeniable that for AvgSen, some important tokens included in the explanation may be replaced, but the probability is low, especially in Yelp and IMDB which have more tokens. The results are summarized in Table 2. It can be found that the explanations generated by CIMI are the most generalizable. Specifically, in the four datasets, at least 8 of the top-10 important tokens before and after perturbation are consistent, which is impossible for other methods. Besides, as the dataset becomes more complex, our performance remains
Table 1: Faithfulness comparison when explaining BERT. *** indicate that our method’s improvements over the best results of baselines results are statistically significant for \( p < 0.001 \).

<table>
<thead>
<tr>
<th>Method</th>
<th>Clickbait</th>
<th>Hate</th>
<th>Yelp</th>
<th>IMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DFFOT</td>
<td>COMP</td>
<td>SUFF</td>
<td>DFFOT</td>
</tr>
<tr>
<td>Gradient</td>
<td>0.5139</td>
<td>0.3651</td>
<td>0.1308</td>
<td>0.2776</td>
</tr>
<tr>
<td>Attention</td>
<td>0.5247</td>
<td>0.3655</td>
<td>0.1213</td>
<td>0.5933</td>
</tr>
<tr>
<td>AXAI</td>
<td>0.5234</td>
<td>0.3641</td>
<td>0.1245</td>
<td>0.4738</td>
</tr>
<tr>
<td>Probing</td>
<td>0.5278</td>
<td>0.3606</td>
<td>0.1249</td>
<td>0.5679</td>
</tr>
<tr>
<td>LIME</td>
<td>0.3994</td>
<td>0.4374</td>
<td>0.0778</td>
<td>0.2800</td>
</tr>
<tr>
<td>KernelSHAP</td>
<td>0.4447</td>
<td>0.4183</td>
<td>0.0725</td>
<td>0.4012</td>
</tr>
<tr>
<td>Rationale</td>
<td>0.5250</td>
<td>0.3651</td>
<td>0.1226</td>
<td>0.5937</td>
</tr>
<tr>
<td>CXPlain</td>
<td>0.4505</td>
<td>0.4092</td>
<td>0.0952</td>
<td>0.3796</td>
</tr>
<tr>
<td>Smask</td>
<td>0.5268</td>
<td>0.3561</td>
<td>0.1320</td>
<td>0.6121</td>
</tr>
<tr>
<td>CMI</td>
<td>0.3826</td>
<td><strong>0.4612</strong></td>
<td><strong>0.0416</strong></td>
<td><strong>0.2761</strong></td>
</tr>
</tbody>
</table>

Figure 3: Performance comparison concerning COMP under different length explanations.

Table 2: Generalizability comparison under BERT. IMP indicates the improvement of our method compared to baselines.

<table>
<thead>
<tr>
<th>Method</th>
<th>Clickbait</th>
<th>Hate</th>
<th>Yelp</th>
<th>IMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AvgSen</td>
<td>IMP(%)</td>
<td>AvgSen</td>
<td>IMP(%)</td>
</tr>
<tr>
<td>Gradient</td>
<td>0.2182</td>
<td>43.18</td>
<td>0.4530</td>
<td>123.68</td>
</tr>
<tr>
<td>Attention</td>
<td>0.2155</td>
<td>41.42</td>
<td>0.5413</td>
<td>167.29</td>
</tr>
<tr>
<td>Lime</td>
<td>0.2036</td>
<td>33.59</td>
<td>0.4689</td>
<td>131.56</td>
</tr>
<tr>
<td>KernelSHAP</td>
<td>0.2022</td>
<td>32.66</td>
<td>0.4989</td>
<td>146.38</td>
</tr>
<tr>
<td>Rationale</td>
<td>0.2163</td>
<td>41.93</td>
<td>0.5440</td>
<td>168.64</td>
</tr>
<tr>
<td>Probing</td>
<td><strong>0.1460</strong></td>
<td>-4.23</td>
<td>0.2093</td>
<td>3.34</td>
</tr>
<tr>
<td>CXPlain</td>
<td>0.2101</td>
<td>37.86</td>
<td>0.5066</td>
<td>150.19</td>
</tr>
<tr>
<td>AXAI</td>
<td>0.2146</td>
<td>40.79</td>
<td>0.5127</td>
<td>153.17</td>
</tr>
<tr>
<td>Smask</td>
<td>0.2179</td>
<td>42.98</td>
<td>0.5552</td>
<td>173.20</td>
</tr>
</tbody>
</table>

Our 0.1524 0.2025 0.1200 0.1135

stable while the baselines decrease significantly, thus, the average improvement on IMDB reaches a staggering 629%. Additionally, we report the distribution of explanation overlaps before and after input perturbations, as shown in Fig. 4. It intuitively conveys the stability of the explanations generated by CMI against neighboring data. In summary, these results demonstrate the outstanding ability of the proposed method to capture invariant generalizable features.

4.4 Effectiveness of Causal Modules

4.4.1 Effectiveness w.r.t. Faithfulness. In this section, we verify the effectiveness of the proposed three causal modules concerning faithfulness. We define versions that remove causal sufficiency loss, causal intervention loss, weakly supervising loss, and interpreter’s encoder \( f_i \) as CMI-s, CMI-i, CMI-p, and CMI-f, respectively. Their sufficient effects are shown in Fig. 5. Overall, removing any module leads to performance degradation, which justifies the design of three modules. Specifically, first, we found that the removal of the causal sufficient module (CMI-s) has an impact on sufficient performance, which reasonably explains the original intention of our design of this module, ensuring that the explanations \( E \) are causally sufficient for the model predictions \( \hat{Y} \). Second, the sufficient impact of the causal intervention module (CMI-i) is marginal, since this module is designed primarily for the explanation’s generalizability. Finally, both the weakly supervising loss and the interpreter’s encoder design in the causal prior module can assist the model to learn more easily.
Figure 4: The distribution of explanation overlaps (top-10 tokens) before and after the input change of five words.

Figure 5: The faithful effect of the causal modules concerning COMP and SUFF.

Figure 6: The generalizable effect of the causal modules concerning AvgSen.

4.4.2 Effectiveness w.r.t. Generalizability. In this section, we discuss the generalizable effect of the three causal modules. Keeping the experimental settings consistent with Section 4.3, the results are illustrated in Fig. 6. First, we find that the causal sufficiency module helps improve generalizability against perturbations on Clickbait and Hate, but significantly degrades performance on Yelp and IMDB. We suspect that in the latter two datasets, CIMI-s explanations are not faithful to model prediction (Fig. 5 (c)(d)), then it is difficult to capture the invariant explanations of model decisions from similar instances, resulting in a decrease in generalizability. Second, only CIMI-i’s performance decreases consistently across the four datasets. This is because the causal intervention module aims to make $U$ and $E$ independent to ensure that the explanation can be generalizable to similar instances, that is, generalizability.
Removing it resulted in performance degradation justifying the rationality of this module design.

4.5 Sampling Efficiency
Fig. 7 illustrates the performance of various perturbation-based methods under the same forward propagation times to measure the sampling efficiency. First, CXPlain’s explanation mechanism makes each sample $x$ perturbed at most $|x|$ times, so it shows high efficiency on small datasets, e.g., Clickbait and Hate. However, it is meaningless to talk about efficiency without explaining quality. Second, LIME performs well on small datasets (e.g., Clickbait), however, as the dataset becomes more complex, more sampling is required to generate high-quality explanations. Rationale’s training is unstable and prone to trivial solutions, resulting in insensitivity to the number of sampling. Finally, our method significantly outperforms baselines, especially on Hate, where we only need 3 samplings to outperform baselines with 100 samplings. This benefits from the generalization of the neural network under the constraints of the causal principle, which summarizes the causal laws from a large number of data points and generalizes them to different instances, ultimately improving efficiency.

4.6 Usefulness Evaluation
In addition to allowing us to better understand the model, the explanation can also assist people in debugging the model. Noisy data collection can cause the model to learn wrong correlations during training. A common approach is feature engineering, which removes untrustworthy features to improve the model generalization. To this end, this section analyzes the effectiveness of various explanation methods in removing shortcut features. We use a subset of 20 newsgroups that classify “Christianity” and “Atheism”. The reason for choosing this dataset is that there are many shortcut features in its training set, but the test set is clean. For example, 99% of the instances in the training set where the word “posting” appears is in the category of “Atheism”, but it clearly has no correlation to the predicted class.

To test whether an explanation method can help detect shortcuts, we first train a BERT model on the noisy training set. Then, we obtain explanations of different methods and treat a token in the explanation as the potential short-cut if it does not appear in the clean test (more details in Appendix A.5). We then retrain the classification model after removing shortcuts. The metric for evaluating the quality of shortcuts is based on the retrained model’s performance (better classification performance implies that the shortcuts found are more accurate). Here we choose LIME, which has been shown to perform well in the task [34], for comparison, and the result is shown in Fig. 8. First, both explanation methods can effectively remove shortcuts to improve the model performance. Second, the improvement of the proposed CIMI is more obvious, verifying the usefulness on debugging models.

5 RELATED WORK
Existing explainable works can be divided into self-explaining methods and post-hoc methods [26, 29].

The self-explaining method focuses on building model architectures that are self-explainable and transparent [44], such as decision tree [18, 49], rule-based models [43], self-attention mechanisms, [3, 48]. In order to provide rules that are easy for humans to understand, they are often too simple to enjoy both interpretability and predictive performance [19]. Recently, methods of integrating add-on modules have received increasing attention [6, 14, 16, 19]. However, the process of generating explanations remains opaque.

Post-hoc interpretation has received more attention as models have gradually evolved into incomprehensible highly nonlinear forms [44]. Gradient-based methods [23, 39, 41, 42, 47] approximate the deep model as a linear and accordingly incorporate the gradient as feature importance. Admittedly, the gradient is only an approximation of the decision sensitivity. Influence function [17] also has been introduced to understand models, which efficiently approximates the impact of perturbations of training data through a second-order optimization strategy. Recently, causal interpretability has attracted increasing attention because it focuses on a fundamental question in XAI: whether existing explanations capture spurious correlations or remain faithful to the underlying causes of model behavior. First, many well-known perturbation-based methods such Shapley values [23], LIME [34], and Smask [21] implicitly use causal inference, and their explanatory scores correspond exactly to the (average) treatment effect [30]. From a causal point of view, the slight difference between them lies only in the number of features selected, the contextual information considered, and the model output. CXPlain [37] explicitly considered the non-informative features should have no effect on model predictions. However, many critical issues, such as how to use causal inference to generalize common causes of different instances from large amounts of data, and how to make explanations causally sufficient for model predictions, remain to be resolved.

6 CONCLUSION
We reinterpreted some classic methods from causal inference and analyzed their pros and cons from this unified view. Then, we revealed the major challenges in leveraging causal inference for interpretation: causal sufficiency and generalizability. Finally, based on a suitable causal graph and important causal principles, we devised training objectives and desirable properties of our neural interpreters and presented an efficient solution, CIMI. Through extensive experiments, we demonstrated the superiority of the proposed method in terms of the explanation’s causal sufficiency and generalizability and additionally explored the potential of explanation methods to help debug models.
Figure 7: Performance comparison with the different number of sampling (perturbation) w.r.t. SUFF.

Figure 8: Usefulness evaluation. Classification performance comparison before and after deleting shortcuts.
A APPENDIX

A.1 Revisiting XAI Through Causal Lens

In this section, we show that perturbation-based methods such as LIME [34], Shapley values [23], and CXPlain [37] actually aim to compute or learn (average) treatment effect, and that their causal graph corresponds to the one shown in Fig. 1(a). This allows us to compare them from a unified causal view (Table 3).

A.1.1 LIME [34]. LIME learns an interpretable function \( g \) to approximate the original black box \( f \) by minimizing the following objective function:

\[
L(f, g, \pi_x) = \sum_{z, z' \in Z} \pi_x(z) (f(z) - g(z'))^2, 
\]

where \( z \) is a perturbation sample created by randomly editing (e.g., deleting) features in \( x \), and \( z' \) is a binary vector that represents whether a feature is edited in \( z \). The interpretable function \( g \) is a linear model over \( z' \):

\[
g(z') = \sum w_i z'_i. 
\]

We can then define the do-operator based on \( z' \):
- \( \text{do}(z' = 0) \): Remove (edit) the features whose values in \( z' \) are 0
- \( \text{do}(z' = 1) \): Keep the features whose values in \( z' \) are 0

And the treatment effect of \( z' \) on \( f \) is:

\[
T(f|\text{do}(z'), C = x) = f(\text{do}(z' = 1), C = x) - f(\text{do}(z' = 0), C = x) = f(x) - f(z). 
\]

The first line is derived by using Eq. (1) in the main paper and treating \( f \) as a random variable.

Similarly, we can derive the treatment effect of \( z' \) on \( g \):

\[
T(g|\text{do}(z'), C = x') = g(\text{do}(z' = 1), C = x') - g(\text{do}(z' = 0), C = x') = g(x') - g(z'). 
\]

Then, we can see that the loss function in Eq. (9) ensures that the two treatment effects align with each other:

\[
\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in Z} \pi_x(z) (f(z) - g(z'))^2 
= \sum_{z, z' \in Z} \pi_x(z) ((f(z) - f(x)) - (g(z') - g(x')))^2 
= \sum_{z, z' \in Z} \pi_x(z) (f(z) - f(x) - g(z') + g(x')). 
\]

where \( b = f(x) - g(x') \) is a constant that does not change with different perturbations.

Let us consider the optimal situation in which the loss is perfectly minimized to 0, then according to Eq. (9) we have

\[
b = f(x) - g(x') = 0. 
\]

According to Eq. (14), \( \mathcal{L}(f, g, \pi_x) = 0 \) and \( b = 0 \) lead to

\[
T(f|\text{do}(z'), C = x) = T(g|\text{do}(z'), C = x') = \sum_{i \in \{z_i = 0\}} w_i. 
\]

The last term is derived by using Eq. (10), where \( w_i \) is the linear weight of feature \( i \) in the interpretable function \( g \). In other words, \( w_i \) is the explanation score for feature \( i \) according to LIME. The above equation shows that if LIME achieves an optimal loss, its explanation scores (\( w_i \)) for a set of features \( \{i|z'_i = 0\} \) will add up to the treatment effect of these features on model prediction \( f \) (i.e., \( T(f|\text{do}(z'), C = x) \)).

Since LIME computes the individual treatment effect \( T(f|\text{do}(z'), C = x) \), its causal graph is Fig. 1(a), according to Fig. 2.2 and Sec. 2.3.2 in [30], with the specific treatment, context, and outcome specified in Table 3. The important properties of LIME, e.g., whether it supports the causal analysis of multiple features combined, can be easily derived according to this causal graph and their learning method.

A.1.2 Shapley Values [23]. In Shapley values, the contribution of the \( i \)-th feature for input instance \( x \) is

\[
\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f\left(x_{S \cup \{i\}}\right) - f(x_S)\right],
\]

where \( F \) is the set of features in input \( x \), and \( S \) is a subset of features that does not include feature \( i \). Let us define the do-operator for \( i \) as:

\[
\text{do}(i = 0) : \text{Remove (edit) feature } i; 
\]

\[
\text{do}(i = 1) : \text{Keep feature } i \text{ unchanged.}
\]

Then the treatment effect of \( i \) on \( f \) is:

\[
T(f|\text{do}(i), C = S \cup \{i\}) = f(\text{do}(i = 1), C = S \cup \{i\}) - f(\text{do}(i = 0), C = S \cup \{i\}) = f\left(x_{S \cup \{i\}}\right) - f(x_S). 
\]

Thus, each term in Eq. (17) corresponds to a treatment effect, and we have

\[
\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f\left(x_{S \cup \{i\}}\right) - f(x_S)\right] 
= \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} T(f|\text{do}(i), C = S \cup \{i\}) 
= \sum_{S \subseteq F \setminus \{i\}} \frac{1}{|F|} \frac{1}{C_S^{[|F|-1]} - p (N_i|F| - 1) p (S|S| = N_i) T(f|\text{do}(i), C = S \cup \{i\})} 
= \frac{1}{C_{2|F|-1}} \sum_{S \subseteq F \setminus \{i\}} \left[f\left(x_{S \cup \{i\}}\right) - f(x_S)\right].
\]

In the third line, \( \frac{1}{C_{2|F|-1}} \) is a multi-nominal co-efficient. In the fourth line, \( p (N_i|F| - 1) = 1/|F| \) denotes a probability distribution in which an integer \( N_i \) sampled from this distribution is uniformed distributed from 0 to \( |F| - 1 \). \( p (S|S| = N_i) = \frac{1}{C_{2|F|-1}} \) is a probability distribution over all subsets \( S \) with \( N_i \) features. Since there are \( C_{2|F|-1} \) such subsets, sampling from \( p (S|S| = N_i) \) means that we randomly get one without any preference for one subset.

Eq. (20) tells us that the explanation score \( \phi_i \) in Shapley values is an average treatment effect of feature \( i \) on model \( f \), in which every possible subset of features \( S \) is considered and treated in the same way, without any bias on subset size or distance.
Table 3: Comparing existing XAI methods from a unified causal view. Here, $E$, $C$, $\hat{Y}$ are the specific treatment, context, and outcome in causal graph Fig. 1(a) for each method. “MultiF” refers to whether the method can analyze the causal effect of multiple features combined, “Average treatment” means whether the corresponding treatment effect is robustly derived by averaging over varying context, “Approximation” means whether the treatment effect is directly computed in an exact way or approximated learned and can only be obtained in the optimal situation, and “Efficiency” denotes the number of interventions (or forward propagations through deep model) needed for computation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Treatment $E$</th>
<th>Context $C$</th>
<th>Outcome $\hat{Y}$</th>
<th>MultiF</th>
<th>Average treatment</th>
<th>Approximation</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIME</td>
<td>${i</td>
<td>z_i = 0}$</td>
<td>$C = x$</td>
<td>$f$</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Shapley Value</td>
<td>$i$-th feature</td>
<td>$C = S \cup {i}, S \subseteq F\setminus i$</td>
<td>$f$</td>
<td>×</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>CXPlain</td>
<td>$i$-th feature</td>
<td>$C = {x, y}$</td>
<td>$-\mathcal{L}$</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

to the original input $x$. We can then specify the treatment, context, and tested outcome in Table 3 and derive their properties similarly to LIME.

A.1.3 CXPlain [37]. In CXPlain, the contribution of each feature $i$ for instance $x$ is

$$w_i(x) = \frac{\Delta x_i}{\sum_{j=0}^{p-1} \Delta x_j},$$

where $\mathcal{L}$ is a loss function measuring the difference between model outputs $f(x)$ and the ground-truth $y$. CXplain approximates $w_i(x)$ on each input $x$ by using neural models.

Same with that in Shapley values, we define the do-operator for $i$ as

\[
\begin{align*}
&\text{do}(i = 0): \text{Remove (edit) feature } i; \\
&\text{do}(i = 1): \text{Keep feature } i \text{ unchanged.}
\end{align*}
\]

(22)

Then the treatment effect of $i$ on $-\mathcal{L}$ is:

\[
\begin{align*}
T(-\mathcal{L}\text{do}(i), C = \{x, y\}) &= -\mathcal{L}(\text{do}(i = 1), C = \{x, y\}) \\
&\quad + \mathcal{L}(\text{do}(i = 0), C = \{x, y\}) \\
&= -\mathcal{L}(y, f(x)) + \mathcal{L}(y, f(x\setminus \{i\})) \\
&= -\varepsilon_x + \varepsilon_x(\setminus \{i\}) \\
&= \Delta x_i.
\end{align*}
\]

(23)

Thus, the explanation score $w_i(x)$ that CXPlain approximates with neural networks is a normalized treatment effect that the $i$-th feature has on the negative loss function $-\mathcal{L}$. We can then specify the treatment, context, and tested outcome in Table 3 and derive their properties similarly to LIME.

A.2 Comparison with the Causal Graph in Domain Generalization

Fig. 9(a) is a typical causal graph for domain generalization (DG). For the sake of intuition, we take the image classification task as an example. The label of the image is used as the class $Y$, and the image’s pixels are denoted as the features $X$. Due to different application scenarios or human intentional design, each class will have multiple views. For example, in the image of puppy class, the puppy can be on the grass, or it can be in the room. Here each view can be considered as a different domain $U$, which should not establish a correlation with class $Y$. Images with the same class should establish a common causal variable, denoted by $S$, which

\[
\begin{align*}
X &\to S \to U \to Y \\
X &\to Y \to \hat{Y}
\end{align*}
\]

Figure 9: (a) Causal graph in domain generalization. (b) Our proposed causal model.

will uniquely determine the class of the label. To create an image, it is first possible to sample causal factors and a view (domain), such as sampling a puppy (causal factor) and grass (non-causal factor). The pixels in the image are caused by both, corresponding to the two arrows pointing to $X$. And only causal factors directly lead to class, which is reflected in $S$ pointing to $Y$.

If we change the objective law into a black-box model, we can get the causal graph of our choice, as shown in Fig. 9(b). Unlike the causal graph of DG, $\hat{Y}$ does not represent the true class, but the model’s prediction, even though it may be inconsistent with the true class. Correspondingly, the parent node $E$ of $\hat{Y}$ is not a causal factor that truly determines the class, but an explanation that is faithful to the model prediction. For example, for a picture of a puppy on grass, the model might predict it as a lamb because it uses information about the grass to make a prediction. At this time, $\hat{Y}$ is a lamb rather than a puppy, and the causal factor $E$ is the grass, not the puppy. In short, the causal graph in XAI is faithful to the model prediction rather than the objective law, which is fundamentally different from the causal graph in domain generalization.

A.3 More Details of the Experimental Setup

A.3.1 Datasets. We use four datasets from the natural language processing domain, including Clickbait [2], Hate [7], Yelp [50], and IMDB [25]. Among them, Clickbait contains 32k news headlines collected from various news websites, dedicated to predicting the headlines as clickbait or non-clickbait. Hate is a hate speech dataset that contains three categories: contains hate speech, is offensive but not hateful, and is not offensive at all. Yelp contains reviews of restaurants, hotels, and other businesses around the world, and
the task is to classify these reviews into positive or negative sentiments. IMDB contains heavily polarizing reviews from Internet Movies. To improve the efficiency of various explanation methods, similar to the preprocessing of [19, 31], we use a down-sampled subset (5000 records) for experiments, with a ratio of 8:1:1 for the training set, validation set, and test set.

A.3.2 Black-box Models. In our experiments, we consider the following classical pre-training models as target models to be explained.

BERT [8], a pre-training language model with a bidirectional transformer structure. Benefiting from the powerful expressive ability, it achieves state-of-the-art performance in multiple NLP tasks. However, the overly complex architecture allows us to know little about its prediction patterns.

RoBERTa [22], an improved version of BERT, brings better performance to downstream tasks by using a larger batch size, more training data, and dynamically adjusted mask modes.

A.3.3 Comparison Methods. The specific details of the baseline methods are shown as follows.

Gradient [41]: It uses standard backpropagation to measure the importance of individual features, i.e., the sensitivity of the input features to changes in the model’s predictions.

Attention [3]: The attention mechanism allows each feature to interact with each other and then finds the features that should be paid more attention to.

LIME [34]: It approximates the local linearity of the black-box model and measures the contribution of each feature by perturbing features.

KernelSHAP [23]: It combines Lime and SHAP, using the calculated Shapley value as a linear interpretation of the instance.

Rationale [20]: It designs a generator that adds length constraints and coherence constraints to select short and coherent pieces of text.

Probing [1]: It inserts a shallow network on sentence embeddings to probe which words are encoded by the model.

CXPlain [37]: It utilizes a causal objective function that quantifies the marginal contribution of either a single input feature towards the predictive model’s accuracy.

AXAI [33]: It calculates the importance of each feature by adversarial attacks and maps feature importance to data segments to obtain explanations with high importance density.

Smask [21]: Based on the assumption that the interpretation of the neighbor data should vary slowly, Smask constrains the generated saliency map to be consistent in the neighborhood.
A.3.4 Evaluation Metrics. In this paper, we use four metrics to measure the quality of generated explanations, where DFFOT, COMP, and SUFF focus on the causal sufficiency of explanations, while AvgSen captures the generalizability of explanations to neighboring instances. The details are as follows:

Decision Flip-Fraction of Tokens (DFFOT), which measures the minimum fraction of important tokens that need to be erased in order to change the model prediction.

\[
DFFOT = \min_{k} \left\{ \frac{1}{|B|} \right\} \text{s.t. } c(x) \neq c(x \setminus x_{k}) \quad \text{if } c(x) = c(x \setminus x_{k}) \text{ for any } k
\]

where \( c(x) \) denotes the predicted class of \( x \), and \( x : k \) is the input sequence containing only the top-\( k \) important tokens.

Comprehensiveness (COMP), which measures the faithfulness score by the change in the output probability of the original prediction class after the important tokens are removed.

\[
COMP = \frac{1}{|B|} \sum_{k \in B} (p_c(x) - p_c(x \setminus x_{k}))
\]

Sufficiency (SUFF), in contrast to COMP, it only keeps important tokens and compares the changes in output probabilities over the original predicted class.

\[
COMP = \frac{1}{|B|} \sum_{k \in B} (p_c(x) - p_c(x \setminus x_{k}))
\]

Average Sensitivity (AvgSen), while measures the average sensitivity of an explanation when the input is perturbed. Here we perturb each sample ten times and take the average result.

\[
AvgSen = \frac{1}{|\Delta|} \sum_{\delta \in \Delta} \frac{||g(x) - g(x + \delta)||}{||g(x)||}
\]

A.4 Additional Experimental Results

A.4.1 Faithfulness Comparison on RoBERTa. Table 4 shows the comparison of various methods in explaining RoBERTa with respect to the three faithfulness metrics. Although baselines perform well in some scenarios, e.g., KernelSHAP’s performance on Clickbait w.r.t. SUFF, such good performance is difficult to generalize to other datasets or metrics, which may be related to their inability to fully exploit the causality of model predictions. In contrast, the proposed CIMI consistently achieves leading or comparable performance under various settings, emphasizing that the causal principle-following explanations are more faithful to the model.

A.4.2 Generalizability Comparison on RoBERTa. Table 5 shows the generalizability comparison of various methods in explaining RoBERTa in the face of neighboring instances. We can get similar findings to Section 4.3, the proposed method achieves optimality in most cases, and this superiority becomes more obvious as the dataset becomes more complex. This further emphasizes the powerful generalizable ability of CIMI to capture common causes of different instances.

A.4.3 Performance comparison concerning SUFF under different length explanations. Fig. 10 illustrates the performance comparison concerning SUFF under different explanation lengths. On Clickbait, Hate, and IMDB, the proposed method significantly outperforms the baselines. On Yelp, our method is inferior to some comparable methods when the explanation is long. However, it is worth mentioning that the quality of the explanations on Yelp is satisfactory when the explanations are short. Given that explanations are often concise in order to be user-friendly, this ability to capture the most important tokens is very important.

A.4.4 Additional Results Regarding Sampling Efficiency. Fig. 11 shows the performance comparison w.r.t. SUFF with different sampling numbers. Similar to the findings in Section 4.5, CIMI requires only a very small number of sampling to achieve good performance. In addition, from IMDB, we found that for the proposed CIMI, the explanation generated by more sampling numbers is not better. This is a good property that may allow our training to produce high-quality explanations with only a small amount of sampling.

A.4.5 Performance Comparison of Different Sampling Strategies in Causal Sufficiency Module. To optimize the causal sufficiency loss \( \mathcal{L}_c \) in Section 3.3.1, we tested three different methods for sampling \( x \):

- **Vanilla**: The default way in CIMI, where each \( x \) is one instance in the inputs \( X \).
- **Intervention-based**: Similar to the intervention in Causal Intervention Module, given on input instance \( x \), we randomly select another instance \( x' \) from \( X \), and take the sample \( x_{\text{int}} \) after intervention as:
  \[
x := x_{\text{int}} = g(x) \odot x + (1 - g(x)) \odot ((1 - \lambda) \cdot x + \lambda \cdot x').
\]
- **Mixed**: Combining the samples from the two above.

We define the versions using the latter two sampling strategies as CIMI-si and CIMI-sm, respectively. Fig. 12 presents the performance comparison of these sampling strategies in terms of causal sufficiency (COMP and SUFF) and generalizability (AvgSen). In terms of causal sufficiency, CIMI (vanilla sampling) is the best on all datasets, while also the most stable in the comparison of generalizability. This shows that our interpreter only needs to learn from observed data to be able to generalize. The possible reason is that our interpreter design follows the information bottleneck theory, so that the information utilized by the interpreter is already noise-filtered, which allows us to use fewer samples to learn the interpreter faster and easier.

A.5 The Detail of Shortcuts Discovery

Principle. If the model is trained on a noisy dataset containing shortcut features, it is possible to learn wrong correlations. If the model uses these shortcut features, and the interpretable method is faithful to the model, the generated explanation should mix these shortcuts. Although we do not know which features in the explanations are shortcuts, the test set is clean, which allows us to compare whether the tokens of the generated explanations appear in the test set. In other words, a token is suspect if it is in the explanation of the training set but not in the test set.

Experiment Details. In order to highlight the potential of explanation in improving the model, we only use 200 training set data for training, among which there are many shortcut tokens that are pseudo-related to categories such as “posts”. Here we choose BERT as the classification model. After the training is completed, we use various explanation methods to explain each training sample and
To evaluate the accuracy of finding shortcuts, we remove these words from the training set and retrain the classification model. Feature engineering inspires us that the more the performance of the retrained model is improved, the more accurate the searched shortcut will be. Therefore, we use the classification performance of the retrained model as a measure to explain the debugged model.

Figure 10: Performance comparison concerning SUFF under different length explanations

Figure 11: Performance comparison with the different number of sampling (perturbation) w.r.t. SUFF.

Figure 12: Performance comparison under different sampling strategies.