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.

*work done during an internship at Microsoft Research Asia.
†Corresponding authors.
‡also affiliated with the State Key Laboratory of Cognitive Intelligence.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599240

ACM Reference Format:
Chenwang Wu, Xiting Wang, Defu Lian, Xing Xie, and Enhong Chen. 2023. A Causality Inspired Framework for Model Interpretation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3580305.3599240

1 INTRODUCTION

Although deep learning is widely used in fields such as medical insurance [16] and financial predictive analysis [13], deep models are mostly black boxes (complex functions that humans cannot understand). This opacity could have serious consequences, especially in high-stakes decisions [37]. For example, the pollution models claim that highly polluted air poses no threat to human health [29]. 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 [46].

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 [36], 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 [28, 42].

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 [31, 38]. 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 [31].

Due to its importance and applicability, causality has attracted increasing attention in XAI. Multiple explanation methods [24, 35, 38] 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 [35], Shapley values [24], and CXplain [38] (RQ1). Our analysis shows that their explanation scores correspond to (average) treatment effect [31] 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 **C**ausality **I**nspired **M**odel **I**nterpreter (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.

Finally, we conduct extensive experiments on four datasets. 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 [31] 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

$$\begin{aligned} T(Y|do(E)) &= \mathbb{E}_c T(Y|do(E), C = c) \\ &= \mathbb{E}_c(Y(do(E = 1), C = c) - Y(do(E = 0), C = c), \end{aligned} \tag{1}$$

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

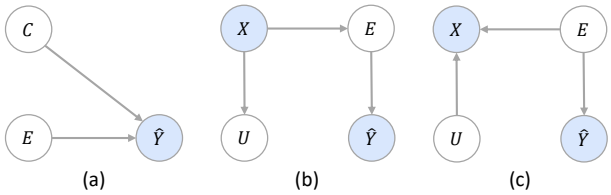

Figure 1: Comparison of causal graphs for model interpretation: (a) The causal graph for existing methods, in which explanation E is not the sore cause for model prediction $\hat{Y}$; (b) Another causal graph, in which explanations are causally sufficient for prediction, but not generalizable; (c) Our proposed causal model which explanation E is generalizable and modeled as the only cause for $\hat{Y}$. 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 [35], Shapley values [24], and CXPlain [38] actually compute or learn the treatment effect, and that their causal graph corresponds to the one shown in Fig. 1(a). Notably, here we only briefly summarize the commonalities and differences among these XAI methods by presenting the main intuition behind the mathematical analysis, and a formal theoretical analysis and more discussion is given in Appendix A.1.

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 $do(E = 1)$, these methods include the feature in the input, while $do(E = 0)$ does the opposite. Then, they obtain the model's outcome $\hat{Y}$ when E is changed and compute the treatment effect $T(\hat{Y}|do(E)) = \hat{Y}(do(E = 1), C = c) - \hat{Y}(do(E = 0), C = c)$, 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 LIME 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.

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 context, in other words, the explanation E is not causally sufficient for $\hat{Y}$. Thus, even if the users have carefully checked the explanations, the problem remains as long as the specific context is a potential cause for the model prediction, thereby the real complete reason for the model prediction cannot be seen.

Solving the causal insufficiency issue. The causal insufficiency of explanations may be addressed by removing context as a causal effect of the model prediction. Fig. 1(b) and (c) show two possible causal graphs to solve this issue. Here, X denotes the random variable for input instances. E and U are unknown random variables for explanations and non-explanations respectively, where $E = x_e$ means that the explanation for $X = x$ is x_e, and $U = x_u$ means that the non-explanation for $X = x$ is x_u. In both causal graphs, $\hat{Y}$ has the only parent (cause), which is the explanation, making the explanation sufficient to 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 [25] (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 $U = X \backslash E$ 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 $\boldsymbol{g : X \rightarrow M}$ 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 [25]. 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 [11]:*** *There exists a causal $x_i \rightarrow \hat{y}$ if the use of all available information results in a more precise prediction of $\hat{y}$ than using information excluding x_i, all causes for $\hat{y}$ are available, and x_i occurs prior to $\hat{y}$.*

Principle 2. ***Independent Causal Mechanisms Principle [33]:*** *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 $\hat{Y}$ 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

$$\mathcal{L}_{s'} = \min_{f'} \mathbb{E}_x(\ell(f(x), f'(x_e)) - \ell(f(x), f'(x_u))), \tag{2}$$

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 [47]. 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 $\mathcal{L}_{s'}$ and learn an interpreter g. In summary, the causal sufficiency loss $\mathcal{L}_s$ rewritten as follows

$$\mathcal{L}_s = \mathbb{E}_x(\ell(f(x), f(x_e)) - \ell(f(x), f(x_u))), \tag{3}$$

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 invariable 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 [25] 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:

$$x_{int} = \underbrace{g(x) \odot x}_{invariant\ explanation} + \underbrace{(1 - g(x)) \odot ((1 - \lambda) \cdot x + \lambda \cdot x')}_{intervened\ non-explanation}, \tag{4}$$

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

$$\mathcal{L}_i = \mathbb{E}_x \ell(g(x), g(x_{int})). \tag{5}$$

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 [22], 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 [38]. 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_e) as the encoder in our interpreter model g. The decoder of g is a simple neural network, denoted as ϕ. The ease of learning is supported by information bottleneck theory, which states that information in each layer decreases as we progress through the model [12]. Therefore, the input x contains the most information, while $f_e(x)$ 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 ϕ so its input concatenates the encoded embedding $f_e(x) \in \mathbb{R}^{|x| \times d}$ and the original instance embedding $v_x \in \mathbb{R}^{|x| \times d}$ along the axis 1, i.e., $[f_e(x); v_x]_1 \in \mathbb{R}^{|x| \times 2d}$, where d is the dimension of embedding, and the operator $[a; b]_i$

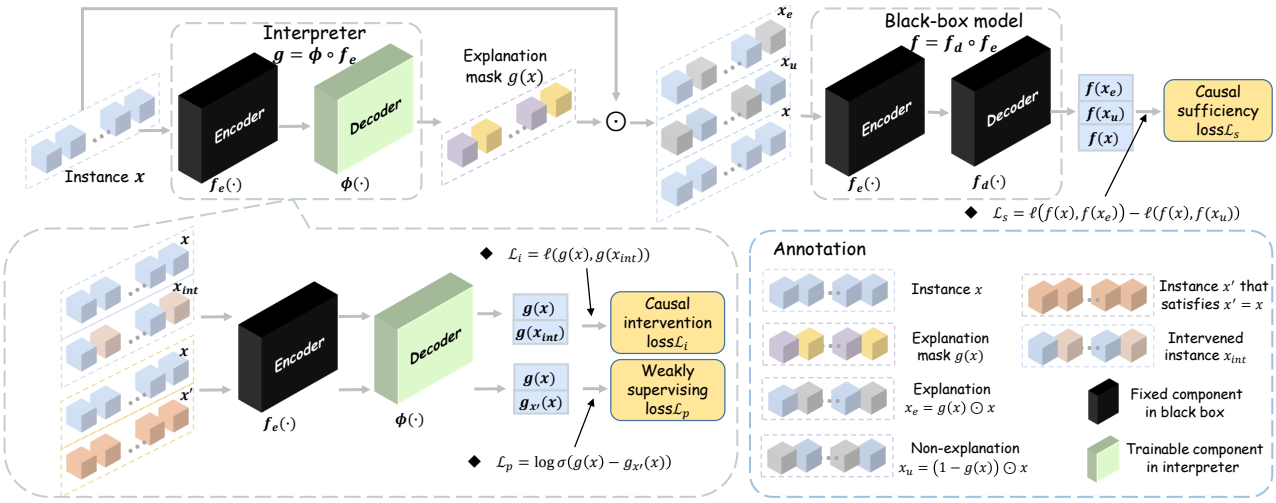

Figure 2: The framework of CIMI. The only trainable component is the decoder ϕ, which is a simple neural network that can be trained with a relatively small number of samples.

denotes the axis i along which matrix a and b will be joined. Therefore, the decoder ϕ maps input $[f_e(x); v_x] \in \mathbb{R}^{|x| \times 2d}$ 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. ϕ 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

$$g(x) = \phi([f_e(x); v_x]_1). \tag{6}$$

It is worth emphasizing that by setting the encoder in g as f_e, the architecture of g could be as complicated as f, 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 ϕ is learnable, while only requiring a few additional parameters, avoiding the issues of overfitting and large training cost.

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 [5]. 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:

$$\mathcal{L}_p = \mathbb{E}_{x,x',x \neq x'} \log \sigma(g(x) - g_{x'}(x)), \tag{7}$$

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

This weakly supervising loss prevents the interpreter from overly optimistically predicting all tokens as explanations, which helps alleviate trivial solutions.

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.

$$\min_{\phi} \mathcal{L}_s + \alpha \mathcal{L}_i + \mathcal{L}_p, \tag{8}$$

where α 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 2-layer MLP (64×16, 16×2). 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 [12]. Let us denote information in a random variable Z as $I(Z)$, then we have $I(X) \geq I(f_e(X)) \geq I(X_e) \geq I(f(X_e))$ according to the architecture in the framework, meaning that the explanations cannot have more information than $I(f_e(X))$. Then, we enforce that $I(X_e) \geq I(f(X_e)) = I(f(X))$ by using the causal sufficiency loss $\mathcal{L}_s$, thus, $I(f_e(X)) \geq I(X_e) \geq I(f(X))$. The possible noise in $I(f_e(X))$ is then removed from $I(X_e)$ by using the intervention loss $\mathcal{L}_i$ and probing loss $\mathcal{L}_p$, 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** [8], **Yelp** [52], and **IMDB** [26]. 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 [9] and RoBERTa [23] (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** [43], **Attention** [3], **LIME** [35], **KernelSHAP** [24], **Rationale** [21], **Probing** [1], **CXPlain** [38], **AXAI** [34], **Smask** [22]. 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).

Decison Flip-Fraction of Tokens (DFFOT) [39], which measures the minimum fraction of important tokens that need to be erased in order to change the model prediction.

Comprehensiveness (COMP) [10], 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) [10], 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 [6]. In addition to the above metrics, we use AvgSen to measure the explanation's generalizability.

Average Sensitivity (AvgSen) [48], 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 $1e-5$. 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 α, set it to 1, 1, 1, 0.1 on Clickbait, Hate, Yelp, and IMDB respectively. In addition, the perturbation magnitude ϵ in the causal intervention module is set to 0.2. The source code of CIMI is available at https://github.com/Daftstone/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 → IMDB). For example, the improvement over the best baselines reaches 119% on IMDB w.r.t. DFFOT 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 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 CIMI 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_e as CIMI-s, CIMI-i, CIMI-p, and CIMI-f, respectively. Their sufficient effects are shown in Fig. 5. Overall, removing any module

Table 1: Faithfulness comparison when explaining BERT. * indicate that our method's improvements over the best results of baselines results are statistically significant for $p < 0.001$.**

Method	Clickbait			Hate			Yelp			IMDB		
	DFFOT↓	COMP↑	SUFF↓	DFFOT↓	COMP↑	SUFF↓	DFFOT↓	COMP↑	SUFF↓	DFFOT↓	COMP↑	SUFF↓
Gradient	0.5139	0.3651	0.1308	0.2776	0.4880	**0.1324**	0.4245	0.1497	0.2900	0.3216	0.1390	0.3999
Attention	0.5247	0.3655	0.1213	0.5933	0.2719	0.2718	0.5890	0.0809	0.3935	0.5459	0.0453	0.4496
AXAI	0.5234	0.3641	0.1245	0.4738	0.3210	0.2299	0.5120	0.1115	0.3088	0.4449	0.0891	0.4111
Probing	0.5278	0.3606	0.1249	0.5679	0.3013	0.2462	0.7133	0.0671	0.4392	0.6535	0.0445	0.4531
LIME	0.3994	0.4374	0.0778	0.2800	0.4860	0.1441	0.3346	0.2362	0.3201	0.2777	0.1953	0.4078
KernelSHAP	0.4447	0.4183	0.0725	0.4012	0.3963	0.1897	0.5484	0.0992	0.3488	0.5189	0.0565	0.4297
Rationale	0.5250	0.3651	0.1226	0.5937	0.2719	0.2719	0.5963	0.0838	0.3892	0.5501	0.0420	0.4533
CXPlain	0.4505	0.4092	0.0952	0.3796	0.4414	0.1438	0.4544	0.2287	0.3121	0.2894	**0.3273**	0.4094
Smask	0.5268	0.3561	0.1320	0.6121	0.2722	0.2735	0.5894	0.0839	0.3863	0.5594	0.0446	0.4492
CIMI	**0.3826**	**0.4612**	**0.0416**	**0.2761**	**0.5022**	0.1497	**0.1896**	**0.3100**	**0.2500**	**0.1270**	0.3270	**0.3516**
t-test	***	***	***	***	***		***	***	***	***		***

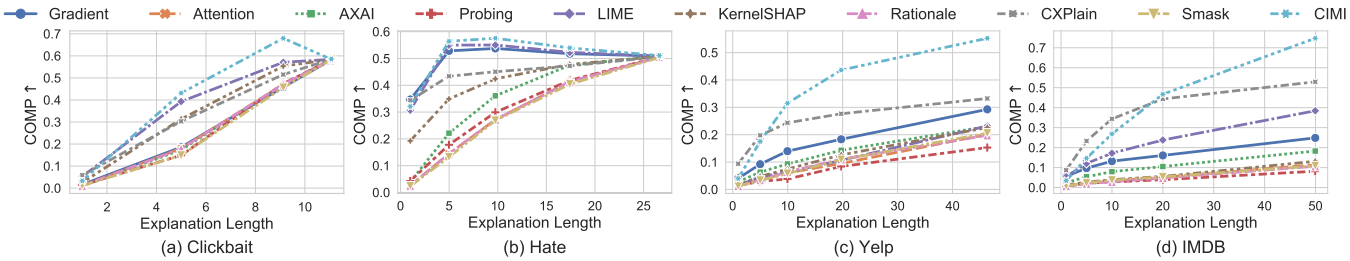

Figure 3: Performance comparison concerning COMP under different length explanations.

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

Method	Clickbait		Hate		Yelp		IMDB		AVG_IMP(%)
	AvgSen↓	IMP(%)	AvgSen↓	IMP(%)	AvgSen↓	IMP(%)	AvgSen↓	IMP(%)	
Gradient	0.2182	43.18	0.4530	123.68	0.7934	561.19	0.8088	612.60	335.16
Attention	0.2155	41.42	0.5413	167.29	0.8642	620.18	0.9482	735.44	391.09
Lime	0.2036	33.59	0.4689	131.56	0.7880	556.68	0.8555	653.74	343.89
KernelSHAP	0.2022	32.66	0.4989	146.38	0.8280	589.96	0.9180	708.82	369.45
Rationale	0.2163	41.93	0.5440	168.64	0.8650	620.87	0.9497	736.73	392.04
Probing	**0.1460**	-4.23	0.2093	3.34	0.2953	146.08	0.2873	153.14	74.58
CXPlain	0.2101	37.86	0.5066	150.19	0.8135	577.95	0.8315	632.58	349.65
AXAI	0.2146	40.79	0.5127	153.17	0.8221	585.08	0.9103	702.05	370.27
Smask	0.2179	42.98	0.5532	173.20	0.8662	621.80	0.9468	734.16	393.03
Our	0.1524		**0.2025**		**0.1200**		**0.1135**		

leads to performance degradation, which justifies the design of three modules. Specifically, first, we found that the removal of the causal sufficient module (CIMI-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 (CIMI-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.

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

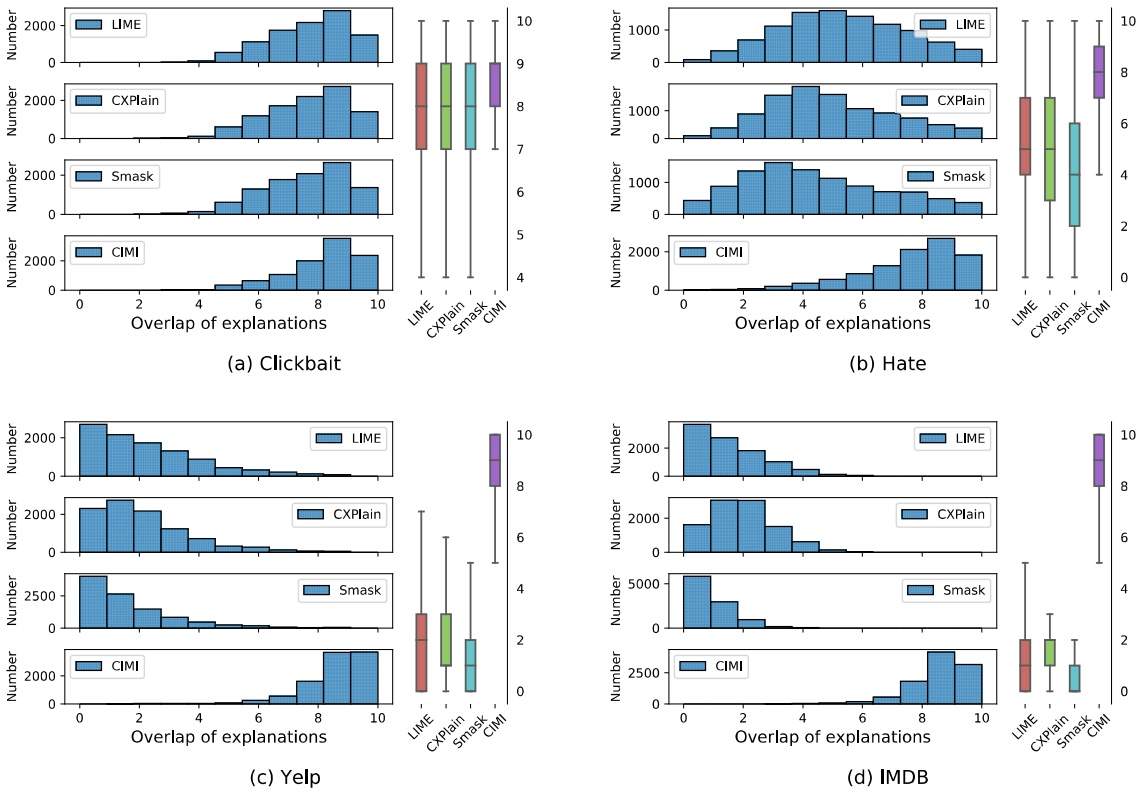

Figure 4: The distribution of explanation overlaps (top-10 tokens) before and after the input change of five words.

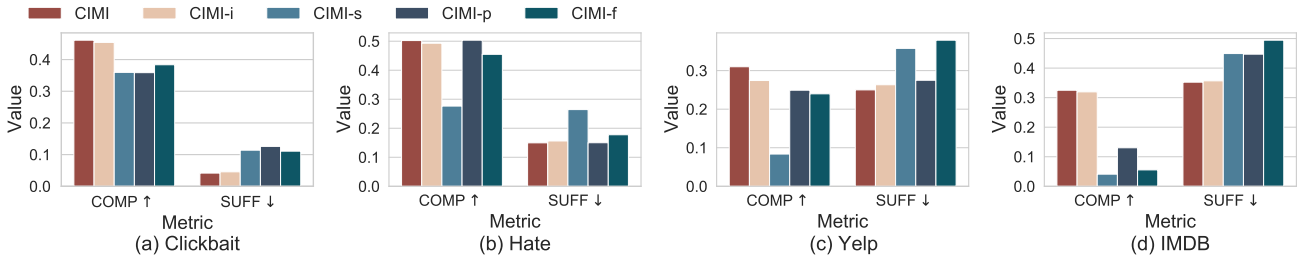

Figure 5: The faithful effect of the causal modules concerning COMP and SUFF.

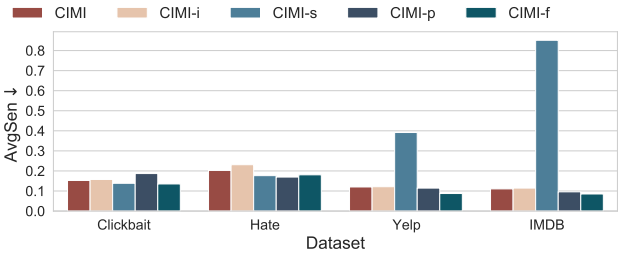

Figure 6: The generalizable effect of the causal modules concerning AvgSen.

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

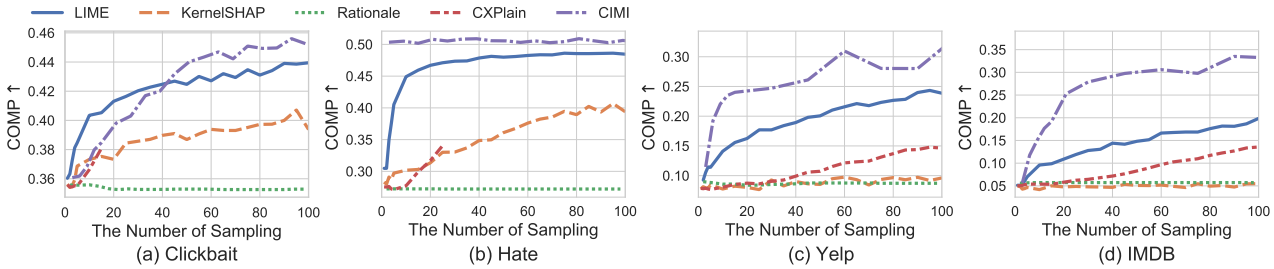

Figure 7: Performance comparison with the different number of sampling (perturbation) w.r.t. SUFF.

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

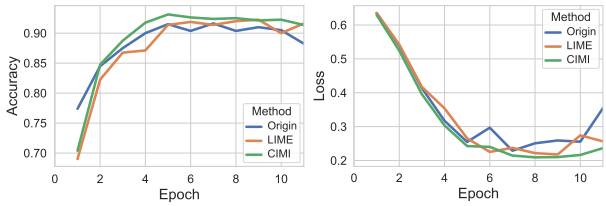

Figure 8: Usefulness evaluation. Classification performance comparison before and after deleting shortcuts.

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.

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 [35], 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 [27, 30].

The self-explaining method focuses on building model architectures that are self-explainable and transparent [46], such as decision tree [19, 51], rule-based models [45], self-attention mechanisms, [3, 50]. In order to provide rules that are easy for humans to understand, they are often too simple to enjoy both interpretability and predictive performance [20]. Recently, methods of integrating add-on modules have received increasing attention [7, 14, 17, 20]. 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 [46]. Gradient-based methods [24, 40, 43, 44, 49] 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 [18] also has been introduced to understand models, which efficiently approximates the impact of perturbations of training data through a second-order optimization strategy. Recently, casual 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 [24], LIME [35], and Smask [22] implicitly use causal inference, and their explanatory scores correspond exactly to the (average) treatment effect [31]. 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 [38] explicitly considered the non-informative features should have no effect on model predictions.

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.

ACKNOWLEDGMENTS

The work was supported by grants from the National Key R&D Program of China (No. 2021ZD0111801) and the National Natural Science Foundation of China (No. 62022077).

REFERENCES

[1] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*. International Conference on Learning Representations, ICLR.

[2] Aman Anand. 2020. Clickbait Dataset. (2020). https://www.kaggle.com/datasets/amananandrai/clickbait-dataset.

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of CVPR'17*. 6541–6549.

[4] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* (2013).

[5] Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. 2020. Differentiable causal discovery from interventional data. *Advances in Neural Information Processing Systems* 33 (2020), 21865–21877.

[6] Chun Sik Chan, Huanqi Kong, and Liang Guanqing. 2022. A Comparative Study of Faithfulness Metrics for Model Interpretability Methods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5029–5038.

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems* 29 (2016).

[8] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, Vol. 11. 512–515.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. *Transactions of the Association for Computational Linguistics* (2020).

[11] Clive WJ Granger. 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society* (1969), 424–438.

[12] Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. 2019. Towards a deep and unified understanding of deep neural models in nlp. In *International conference on machine learning*. PMLR, 2454–2463.

[13] JB Heaton, Nicholas G Polson, and Jan Hendrik Witte. 2016. Deep learning in finance. *arXiv preprint arXiv:1602.06561* (2016).

[14] Beta-vae Higgins. [n. d.]. Learning basic visual concepts with a constrained variational framework. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*.

[15] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[16] Yahui Jiang, Meng Yang, Shuhao Wang, Xiangchun Li, and Yan Sun. 2020. Emerging role of deep learning-based artificial intelligence in tumor pathology. *Cancer communications* 40, 4 (2020), 154–166.

[17] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[18] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*. PMLR, 1885–1894.

[19] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1675–1684.

[20] Seungeon Lee, Xiting Wang, Sungwon Han, Xiaoyuan Yi, Xing Xie, and Meeyoung Cha. 2022. Self-explaining deep models with logic rule reasoning. In *Advances in Neural Information Processing Systems*.

[21] Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 107–117.

[22] Dohun Lim, Hyeonseok Lee, and Sungchan Kim. 2021. Building reliable explanations of unreliable neural networks: locally smoothing perspective of model interpretation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6468–6477.

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[24] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[25] Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. 2022. Causality Inspired Representation Learning for Domain Generalization. In *Proceedings of CVPR'22*. 8046–8056.

[26] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 142–150.

[27] Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *Comput. Surveys* 55, 8 (2022), 1–42.

[28] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. 2020. Explainable reinforcement learning through a causal lens. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 2493–2500.

[29] Michael McGough. 2018. How bad is Sacramento's air, exactly? Google results appear at odds with reality, some say. *Sacramento Bee* 7 (2018).

[30] Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. 2020. Causal interpretability for machine learning-problems, methods and evaluation. *ACM SIGKDD Explorations Newsletter* 22, 1 (2020), 18–33.

[31] Brady Neal. 2020. Introduction to causal inference from a machine learning perspective. *Course Lecture Notes (draft)* (2020).

[32] Yuzuru Okajima and Kunihiko Sadamasa. 2019. Deep neural networks constrained by decision rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 2496–2505.

[33] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

[34] Arash Rahnama and Andrew Tseng. 2021. An adversarial approach for explaining the predictions of deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3253–3262.

[35] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of KDD'16*. 1135–1144.

[36] Jonathan G Richens, Ciarán M Lee, and Saurabh Johri. 2020. Improving the accuracy of medical diagnosis with causal machine learning. *Nature communications* 11, 1 (2020), 3923.

[37] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215.

[38] Patrick Schwab and Walter Karlen. 2019. Cxplain: Causal explanations for model interpretation under uncertainty. *Advances in Neural Information Processing Systems* 32 (2019).

[39] Sofia Serrano and Noah A Smith. 2019. Is Attention Interpretable?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2931–2951.

[40] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).

[41] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 180–186.

[42] Steven Sloman. 2005. *Causal models: How people think about the world and its alternatives*. Oxford University Press.

[43] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017).

[44] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.

[45] P.-N. Tan. 2018. *Introduction to data mining*. India.

[46] G Xu, TD Duong, Q Li, S Liu, and X Wang. 2020. Causality Learning: A New Perspective for Interpretable Machine Learning. *IEEE Intelligent Informatics Bulletin* (2020).

[47] Ruichao Yang, Xiting Wang, Yiqiao Jin, Chaozhuo Li, Jianxun Lian, and Xing Xie. 2022. Reinforcement subgraph reasoning for fake news detection. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2253–2262.

[48] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. 2019. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems* 32 (2019).

[49] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*. Springer, 818–833.

[50] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. 2018. Interpretable convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8827–8836.

[51] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. 2019. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6261–6270.

[52] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems* 28 (2015).

A APPENDIX

A.1 Revisiting XAI Through Causal Lens

In this section, we show that perturbation-based methods such as LIME [35], Shapley values [24], and CXPlain [38] 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 [35]. LIME learns an interpretable function g to approximate the original black box f by minimizing the following objective function:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z,z' \in Z} \pi_x(z)(f(z) - g(z'))^2, \tag{9}$$

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':

$$g(z') = \sum w_i z'_i. \tag{10}$$

We can then define the do-operator based on z':

$$\begin{aligned} &do(z' = 0) : \text{Remove (edit) the features whose values in } z' \text{ are } 0 \\ &do(z' = 1) : \text{Keep the features whose values in } z' \text{ are } 0 \end{aligned} \tag{11}$$

And the treatment effect of z' on f is:

$$\begin{aligned} T(f|do(z'), C = x) &= f(do(z' = 1), C = x) - f(do(z' = 0), C = x) \\ &= f(x) - f(z). \end{aligned} \tag{12}$$

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

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

$$\begin{aligned} T(g|do(z'), C = x') &= g(do(z' = 1), C = x') - g(do(z' = 0), C = x') \\ &= g(x') - g(z'). \end{aligned} \tag{13}$$

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

$$\begin{aligned} \mathcal{L}(f, g, \pi_x) &= \sum_{z,z' \in Z} \pi_x(z)(f(z) - g(z'))^2 \\ &= \sum_{z,z' \in Z} \pi_x(z)((f(z) - f(x)) - (g(z') - g(x')) + b)^2 \\ &= \sum_{z,z' \in Z} \pi_x(z)(-T(f|do(z'), C = x) + T(g|do(z'), C = x') + b)^2, \end{aligned} \tag{14}$$

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

$$b = f(x) - g(x') = 0. \tag{15}$$

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

$$T(f|do(z'), C = x) = T(g|do(z'), C = x') = \sum_{i \in \{z'_i = 0\}} w_i. \tag{16}$$

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., $T(f|do(z'), C = x)$).

Since LIME computes the individual treatment effect $T(f|do(z'), C = x)$, its causal graph is Fig. 1(a), according to Fig. 2.2 and Sec. 2.3.2 in [31], 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 [24]. In Shapley values, the contribution of the i-th feature for input instance x is

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f\left(x_{S \cup \{i\}}\right) - f\left(x_S\right) \right], \tag{17}$$

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:

$$\begin{aligned} &do(i = 0) : \text{Remove (edit) feature } i; \\ &do(i = 1) : \text{Keep feature } i \text{ unchanged.} \end{aligned} \tag{18}$$

Then the treatment effect of i on f is:

$$\begin{aligned} T(f|do(i), C = S \cup \{i\}) &= f(do(i = 1), C = S \cup \{i\}) \\ &\quad - f(do(i = 0), C = S \cup \{i\}) \\ &= f\left(x_{S \cup \{i\}}\right) - f\left(x_S\right). \end{aligned} \tag{19}$$

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

$$\begin{aligned} \phi_i &= \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f\left(x_{S \cup \{i\}}\right) - f\left(x_S\right) \right] \\ &= \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} T(f|do(i), C = S \cup \{i\}) \\ &= \sum_{S \subseteq F \setminus \{i\}} \frac{1}{|F|} \frac{1}{C^{|S|}_{|F|-1}} T(f|do(i), C = S \cup \{i\}) \\ &= \sum_{S \subseteq F \setminus \{i\}} p\left(N_s||F| - 1\right) p\left(S||S| = N_s\right) T(f|do(i), C = S \cup \{i\}) \\ &= \mathbb{E}_{\substack{N_s \sim U\{0,|F|-1\} \\ S \sim P(S||S|=N_S)}} T(f|do(i), C = S \cup \{i\}). \end{aligned} \tag{20}$$

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

Eq. (20) tells us that **the explanation score ϕ_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 treament" 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.

Method	Treatment E	Context C	Outcome $\hat{Y}$	MultiF	Average treatment	Approximation	Efficiency
LIME	Features $\{i\|z'_i = 0\}$	$C = x$	f	✓	×	✓	×
Shapley Value	i-th feature	$C = S \cup \{i\}, S \subseteq F \backslash i$	f	×	✓	×	×
CXPlain	i-th feature	$C = \{x, y\}$	$-\mathcal{L}$	×	×	✓	✓

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 [38]. In CXPlain, the contribution of each feature i for instance x is

$$\begin{aligned} w_i(x) &= \frac{\Delta\varepsilon_{x,i}}{\sum_{j=0}^{p-1} \Delta\varepsilon_{x,j}}, \\ \Delta\varepsilon_{x,i} &= \varepsilon_{x\backslash\{i\}} - \varepsilon_x, \\ \varepsilon_x &= \mathcal{L}(y, f(x)), \end{aligned} \tag{21}$$

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

$$\begin{aligned} &do(i = 0) : \text{Remove (edit) feature } i; \\ &do(i = 1) : \text{Keep feature } i \text{ unchanged.} \end{aligned} \tag{22}$$

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

$$\begin{aligned} T(-\mathcal{L}|do(i), C = \{x, y\}) &= -\mathcal{L}(do(i = 1), C = \{x, y\}) \\ &\quad + \mathcal{L}(do(i = 0), C = \{x, y\}) \\ &= -\mathcal{L}(y, f(x)) + \mathcal{L}(y, f(x\backslash\{i\})) \\ &= -\varepsilon_x + \varepsilon_{x\backslash\{i\}} \\ &= \Delta\varepsilon_{x,i}, \end{aligned} \tag{23}$$

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

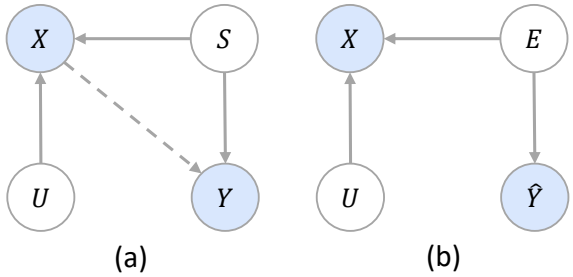

Figure 9: (a) Causal graph in domain generalization. (b) Our proposed causal model.

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** [8], **Yelp** [52], and **IMDB** [26]. 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

Table 4: Faithfulness comparison when explaining RoBERTa. * indicate that our method's improvements over the best results of baselines results are statistically significant for $p < 0.001$.**

Method	Clickbait			Hate			Yelp			IMDB		
	DFFOT↓	COMP↑	SUFF↓	DFFOT↓	COMP↑	SUFF↓	DFFOT↓	COMP↑	SUFF↓	DFFOT↓	COMP↑	SUFF↓
Gradient	0.6203	0.2615	0.0999	**0.3211**	0.5659	0.0700	0.5590	0.1167	0.3263	0.4476	0.1001	0.3266
Attention	0.6330	0.2562	0.0959	0.6535	0.3169	0.2930	0.7155	0.0727	0.3775	0.6438	0.0344	0.4314
Probing	0.6486	0.2472	0.1117	0.5928	0.3556	0.2494	0.7987	0.0622	0.4000	0.8530	0.0172	0.5496
AXAI	0.6326	0.2573	0.0967	0.6512	0.3177	0.2915	0.7107	0.0727	0.3770	0.6471	0.0358	0.4282
LIME	0.6385	0.2546	0.1122	0.6587	0.3278	0.3121	0.7095	0.0755	0.3771	0.6287	0.0361	0.4340
KernelSHAP	0.5778	0.2838	**0.0724**	0.4443	0.4554	0.1475	0.6821	0.0773	0.3533	0.6006	0.0410	0.3620
Rationale	0.6335	0.2561	0.0964	0.6533	0.3172	0.2929	0.7172	0.0699	0.3817	0.6434	0.0313	0.4292
CXPlain	0.5934	0.2786	0.0759	0.4581	0.4817	0.0978	0.5070	0.2217	0.3153	0.4149	**0.2367**	0.2830
Smask	0.6399	0.2525	0.0933	0.6688	0.3087	0.3067	0.7196	0.0759	0.3670	0.6398	0.0347	0.4164
CIMI	**0.5612**	**0.2899**	0.0770	0.3275	**0.5700**	**0.0662**	**0.2324**	**0.2487**	**0.2810**	**0.3140**	0.1232	**0.2678**
t-test	***	***		***		***	***	***	***	***		***

Table 5: Generalizability Comparison under RoBERTa. IMP indicates the improvement of our method compared to various baselines.

Method	Clickbait		Hate		Yelp		IMDB		AVG_IMP(%)
	AvgSen↓	IMP(%)	AvgSen↓	IMP(%)	AvgSen↓	IMP(%)	AvgSen↓	IMP(%)	
Gradient	0.2151	50.99	0.4126	170.10	0.7597	422.21	0.9101	535.36	294.66
Attention	0.2204	54.76	0.5408	254.00	0.8593	490.68	0.9483	562.06	340.37
Lime	0.2274	59.67	0.5263	244.51	0.8524	485.96	0.9502	563.37	338.38
KernelSHAP	0.2021	41.87	0.4780	212.89	0.8258	467.70	0.9208	542.87	316.33
Rationale	0.2208	55.02	0.5455	257.07	0.8646	494.35	0.9440	559.03	341.37
Probing	**0.1202**	-15.64	0.2520	64.95	0.2260	55.33	0.2423	69.17	43.45
CXPlain	0.2169	52.32	0.4968	225.21	0.7810	436.89	0.9153	538.99	313.35
AXAI	0.2204	54.71	0.5398	253.36	0.8573	489.32	0.9491	562.56	339.99
Smask	0.2199	54.38	0.5473	258.23	0.8544	487.37	0.9496	562.96	340.73
Our	0.1424		**0.1528**		**0.1455**		**0.1432**		

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 [20, 32], 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 [9], 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 [23], 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 [43]: 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 [35]: It approximates the local linearity of the black-box model and measures the contribution of each feature by perturbing features.

KernelSHAP [24]: It combines Lime and SHAP, using the calculated Shapley value as a linear interpretation of the instance.

Rationale [21]: 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 [38]: It utilizes a causal objective function that quantifies the marginal contribution of either a single input feature towards the predictive model's accuracy.

AXAI [34]: 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 [22]: 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:

Decison 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.

$$DFFOT = \begin{cases} \min \frac{k}{|x|} & \text{s.t. } c(x) \neq c\left(x \backslash x_{:k}\right) \\ 1 & \text{if } c(x) = c\left(x \backslash x_{:k}\right) \text{ for any k} \end{cases},$$

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.

$$COMP = \frac{1}{|B|} \sum_{k \in B} \left(p_{c(x)}(x) - p_{c(x)}(x \backslash x_{:k})\right)$$

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

$$COMP = \frac{1}{|B|} \sum_{k \in B} \left(p_{c(x)}(x) - p_{c(x)}(x_{:k})\right)$$

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 $\mathcal{L}_s$ 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_{int} after intervention as x:

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

- **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.4.6 Performance Comparison of Different Discretization Strategies. In CIMI, we use Softmax function for differentiable training, in this section, we added two variants of our method that discretize U and E by using Gumbel-Softmax [15] and Deep Hash Learning [4], denoted as CIMI-Gumbel and CIMI-DHL, respectively. As shown in Fig. 13 below, discrete masks help improve the explanation generalizability at the expense of a significant performance decline w.r.t. faithfulness. Specifically, in most cases, Deep Hash Learning contributes to explanations' generalizability (AvgSen), because the change in the mask value domain ($[0, 1] \rightarrow \{0, 1\}$) enables the explanations to be insensitive to noise. However, this discrete mask cannot distinguish the relative importance between features (e.g., when the probabilities of two words being explanations are 0.9 and 0.6, respectively, they are considered indistinguishable explanations after discretization), leading to a significant decline in performance during faithfulness tests that require a correct ordering of features according to their relative importance. We will add these analyses

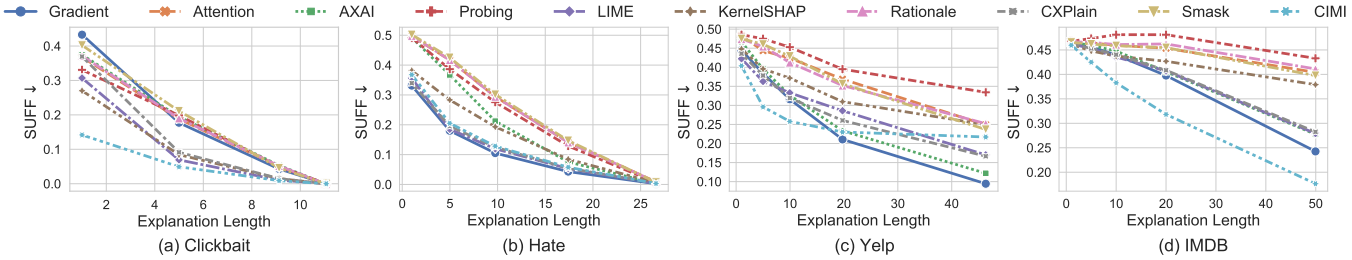

Figure 10: Performance comparison concerning SUFF under different length explanations

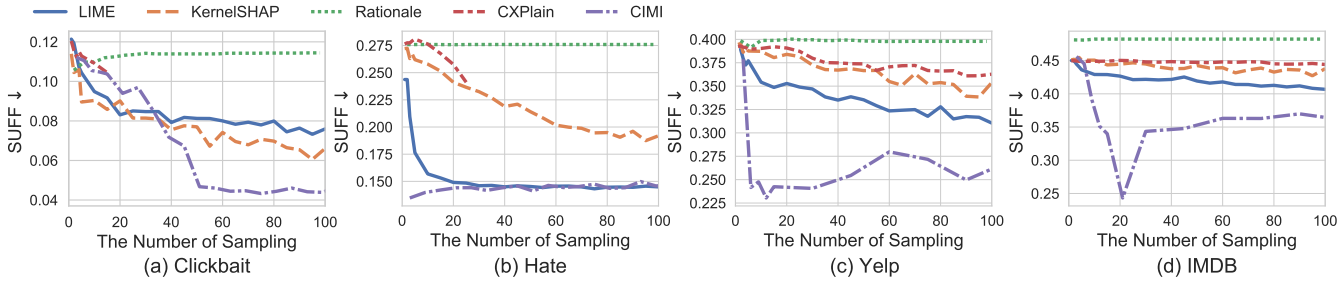

Figure 11: Performance comparison with the different number of sampling (perturbation) w.r.t. SUFF.

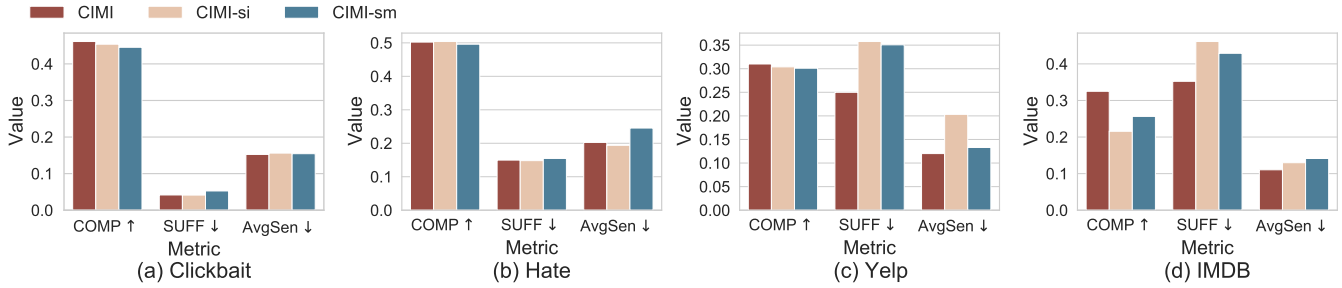

Figure 12: Performance comparison under different sampling strategies.

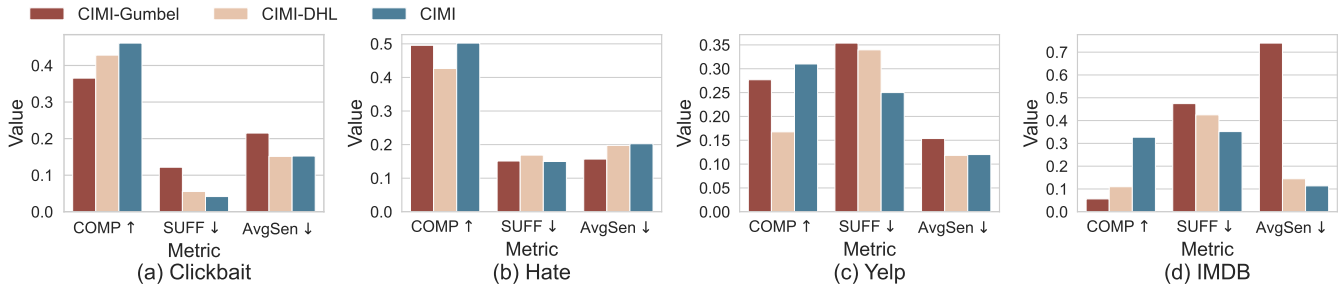

Figure 13: Performance comparison under different discretization strategies.

to the paper, which we believe will help readers better understand the motivation for differentiable training.

A.4.7 Robustness Evaluation. We conduct a robustness comparison of different interpretable methods under the attack [41], and the results verify the impressive robustness of the proposed CIMI (as shown in Table 6 below). Specifically, since the attack aims to conceal biased explanations in any biased classifier, then a robust interpreter should be able to recognize biased explanations even

when the classifier is attacked. We perform the evaluation based on the Clickbait dataset. To obtain a biased classifier, we construct a biased dataset by randomly adding the word "post" at the beginning of the input, labeling the instance as clickbait if "post" is present; otherwise, it is non-clickbait. Table 1 shows the proportion of "post" in the clickbait instances considered top-1/top-3 explanations. From columns 1-2, it can be found that most explanation methods are adept at identifying the biased token "post". However, after being attacked, the proportion of biased explanations in the baselines significantly decreases (columns 3-4), indicating that the attack effectively conceals the biased explanation. In contrast, the proposed CIMI consistently demonstrates high robustness in all settings.

Table 6: Comparison of robustness under the attack in [1]. % of instances for which a biased token (i.e., "post") shows up in top-1/top-3 (according to various interpreters' ranking of token importance) for the biased classifier/attack classifier. Best results are highlighted in bold.

Method	Biased classifier		Attack classifier	
	Top-1	Top-3	Top-1	Top-3
Gradient	0.9649	0.9917	0.6639	0.8804
LIME	0.9918	0.9999	0.0144	0.0206
KernelSHAP	0.0455	0.3864	0.0000	0.1590
CIMI	**0.9991**	**1.0000**	**0.9979**	**0.9999**

A.4.8 Case Study. To help better understand the effects of interpretation, we present examples of explanations and non-explanations on model output, as shown in Table 7 and 8. The cases intuitively show that explanations generated by CIMI are more faithful to the model prediction. Specifically, The last three columns display the model's prediction scores when the original sentences, explanations, and non-explanations are input into the model. Here, the explanation is the three words underlined in the sample. Ideally, the model's prediction with the original sentence should be close to that with the explanation as input (i.e., SUFF metric requirements), and deviate from the prediction when the non-explanation is input (i.e., COMP metric requirements). We can find that the CIMI's explanations adhere more closely to the ideal condition. Taking the first example as an instance, if CIMI only uses the three keywords (drunk, this, bitch) it discovered to replace the original sentence as input, the model output will only fluctuate slightly (0.9695→0.9139). Instead, if we remove these three keywords, the output changes significantly (0.9695→0.1336).

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 get the corresponding explanation (we choose the top-10 important tokens). If a token in the explanation does not appear in the test set, we treat it as a potential short-cut. Finally, the 30 most frequently occurring short-cut tokens among these candidate tokens are selected as shortcuts discovered by each explanation method. 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.

Table 7: Cases of generated explanations on Hate

Method	Sample (explanation is highlighted)	Label	Hate Score of Original Input	Hate Score of Explan.	Hate Score of Non-Explan.
LIME	Mother Nature is drunk again, this bitch knows how to party.	Hate	0.9695	0.8808	0.1813
KernelSHAP	Mother Nature is drunk again, this bitch knows how to party.	Hate	0.9695	0.2323	0.967
CIMI	Mother Nature is drunk again, this bitch knows how to party.	Hate	0.9695	0.9139	0.1336
LIME	RT @SimplyPerfectt_: Girls, don't let a guy treat you like a yellow starburst. You are a pink starburst.	Non-hate	0.1636	0.297	0.1982
KernelSHAP	RT @SimplyPerfectt_: Girls, don't let a guy treat you like a yellow starburst. You are a pink starburst.	Non-hate	0.1636	0.2996	0.1669
CIMI	RT @SimplyPerfectt_: Girls, don't let a guy treat you like a yellow starburst. You are a pink starburst.	Non-hate	0.1636	0.2227	0.2759

Table 8: Cases of generated explanations on Clickbait

Method	Sample (explanation is highlighted)	Label	Clickbait Score of Original Input	Clickbait Score of Explan.	Clickbait Score of Non-Explan.
LIME	Drop Everything, Emo Kids, Because Jack's Mannequin Is Reuniting	Clickbait	0.9991	0.0024	0.9819
KernelSHAP	Drop Everything, Emo Kids, Because Jack's Mannequin Is Reuniting	Clickbait	0.9991	0.2337	0.9990
CIMI	Drop Everything, Emo Kids, Because Jack's Mannequin Is Reuniting	Clickbait	0.9991	0.9972	0.7694
LIME	Rodriguez Takes the Field for Yanks, and Few Seem to Notice	Non-clickbait	0.0027	0.0013	0.0016
KernelSHAP	Rodriguez Takes the Field for Yanks, and Few Seem to Notice	Non-clickbait	0.0027	0.2348	0.0011
CIMI	Rodriguez Takes the Field for Yanks, and Few Seem to Notice	Non-clickbait	0.0027	0.0027	0.0179