Adversarial Infidelity Learning for Model Interpretation

Jian Liang1*, Bing Bai1*, Yuren Cao1, Kun Bai1, Fei Wang2
1Cloud and Smart Industries Group, Tencent, China
2Department of Population Health Sciences, Weill Cornell Medicine, USA
{joshualiang,icrbai,laurenyrcao,kunbai}@tencent.com
few2001@med.cornell.edu

ABSTRACT
Model interpretation is essential in data mining and knowledge discovery. It can help understand the intrinsic model working mechanism and check if the model has undesired characteristics. A popular way of performing model interpretation is Instance-wise Feature Selection (IFS), which provides an importance score of each feature representing the data samples to explain how the model generates the specific output. In this paper, we propose a Model-agnostic Effective Efficient Direct (MEED) IFS framework for model interpretation, mitigating concerns about sanity, combinatorial shortcuts, model identifiability, and information transmission. Also, we focus on the following setting: using selected features to directly predict the output of the given model, which serves as a primary evaluation metric for model-interpretation methods. Apart from the features, we involve the output of the given model as an additional input to learn the explainer, besides fidelity, we propose an Adversarial Infidelity Learning (AIL) mechanism to boost the explanation learning by screening relatively unimportant features. Through theoretical and experimental analysis, we show that our AIL mechanism can help learn the desired conditional distribution between selected features and targets. Moreover, we extend our framework by integrating efficient interpretation methods as proper priors to provide a warm start. Comprehensive empirical evaluation results are provided by quantitative metrics and human evaluation to demonstrate the effectiveness and superiority of our proposed method. Our code is publicly available online at https://github.com/langlrsw/MEED.

1 INTRODUCTION
The interpretation of data-driven models explains their input-output relationship, which provides information about whether the models admit some undesired characteristics, and thus can guide people to use, debug, and improve machine learning models. The model interpretation has an increasing demand in many real-applications, including medicine [36], security [8], and criminal justice [22].

Existing research on model interpretation can be categorized into model-specific methods and model-agnostic methods. Model-specific methods take advantage of the knowledge of the model itself to assist explanations, such as gradient-based methods for neural networks, whereas model-agnostic methods can explain any black-box system. Instance-wise Feature Selection (IFS) is a well known model-agnostic interpretation method. It produces an importance score of each feature for representing a data sample [12], which indicates how much each feature dominates the model’s output. For this kind of approach, desired properties for ideal explanations (feature importance scores) are as follows.

• Expressiveness: the number of features with relatively high scores should be small [27].
• Fidelity: the model output should primarily depend on high-score features [7, 9, 13, 15, 19, 27, 30, 39].
• Low sensitivity: feature scores should be robust against adversarial attacks [11, 14, 39, 40].
• Sanity: feature scores should be dependent of the model [3].

Recent research for IFS-based model explanation can be divided into (local/global) feature attribution methods [4, 39] and direct model-interpretation (DMI) methods. Local feature attribution methods provide some sensitivity scores of the model output concerning the changes of the features in the neighborhood. In contrast, global feature attribution methods directly produce the amount of change of the model output given changes of the features. Other than providing the change of the model output, DMI is a more straightforward approach to select features and use a model to approximate the output of the original black-box model [9, 35]. In this paper, we attempt to tackle the DMI problem. When given a data sample and the model to be explained, what features does the model use primarily to generate the output? A straightforward approach is to develop a feature attribution network (which we...
Figure 1: The architecture of our proposed framework. Taking a sentence as an example, we train an explainer to select important words and an approximator to predict the output of the original model. The model output is also used as an input for the explainer. The AIL module trains the explainer to render the adversarial approximator cannot predict the model output well based on the unselected words. As an extension, efficient interpretation methods, e.g., gradient-based methods, can be integrated to provide a warm start. Best view in color.

refers to as the explainer) to produce a soft/hard mask to highlight essential features, and a prediction network (which we refer to as the approximator) to approximate the output of the original model [9]. However, this straightforward approach may cause the effectiveness and efficiency related concerns in the following.

- **Sanity problem** [3]: a mask may be irrelevant to the original model, but only relate to the features of a specific sample. As a consequence, the selected features of the trained explainer may be different from the useful features. The reason is that the model may have different selected features when predicting different samples.

- **Combinatorial shortcuts problem**: the entries of the mask may not select good features, but rather act as additional features themselves for better approximation performances [17, 37], because it is a function of all the input features. For example, the explainer could choose to mask out the first half of the features for positive samples, and the second half of the features for negative samples. The approximator can utilize this pattern to predict the target while completely ignore whether good features are selected.

- **Model identifiability problem**: similar approximation performances can be achieved by different groups of feature. It is difficult to decide which group is the best.

- **Information transmission problem** [25]: it is difficult to transmit effective supervised information to the explainer, because the mask is unsupervised.

To address these issues, we propose a Model-agnostic Effective Efficient Direct (MEED) model-interpretation method for the DMI problem. The overall architecture of our proposed framework is presented in Figure 1. The major components include model output feedback, adversarial infidelity learning, and prior knowledge-based warm start, which we describe as follows.

Firstly, we propose to enrich the input information of the explainer to boost the effectiveness and the efficiency of the feature selection process. Existing research treats raw features only as the input to the neural network-based explainer [7, 9, 30]. The absence of the original model to include for the explainer’s input may render the mask out of the explainer uncorrelated with the original model and then cause the sanity problem. Nonetheless, it is not trivial to input a whole model into the neural network-based explainer. Therefore, we propose to incorporate the model output as another input signal. Apart from the sanity problem, the model output can provide rich information for the explainers to select essential features, and make the learning process more precise, especially in applications like regression or representation learning. In other words, the information transmission problem can also be mitigated.

Secondly, we propose to exploit the unselected features for mitigating the combinatorial shortcuts and model identifiability problems. Inspired by Hooker et al. [15], we attempt to achieve an auxiliary goal that the unselected features should contain the least useful information. To achieve this, we propose an Adversarial Infidelity Learning (AIL) mechanism. Specifically, we develop another approximator that learns to approximate the original model output using the unselected features. Then our explainer learns to select features to minimize such approximation accuracy. The learning processes run alternately. Intuitively, the convergence of such an adversarial learning process will render the masks uncorrelated with the model output, and then can mitigate the combinatorial shortcuts problem. On the other hand, this learning process exploits the unselected features, which are often (at least relatively) ignored, to introduce additional supervised information for a certain group of selected features, and then can improve model identifiability. These properties are demonstrated by our theoretical analysis and experimental results.

Finally, we extend our framework to further mitigate the information transmission problem by integrating prior knowledge. Specifically, we integrate explanations provided by efficient interpretation methods as priors to provide a warm start. The constraints of the priors fade out when the number of training epochs grows to learn a more powerful explainer by the end-to-end framework.
We follow Chen [9] to perform a predictive evaluation to see whether the selected features contribute to sufficient approximate accuracy. Comprehensive empirical evaluation results on four real-world benchmark datasets are provided with quantitative evaluation metrics and human-evaluations to demonstrate the effectiveness and superiority of our proposed method. Moreover, we validate our method on a real-world application: teenager/adult classification based on mobile sensor data from 5 million of Tencent users who play the popular *Honor of Kings, a.k.a. Arena of Valor* game.

2 RELATED WORKS

Model interpretation methods based on IFS can be categorized into local/global methods as introduced in the introduction. Local methods includes 1) gradient-based methods, such as Gradient (Grad) [32] and Guided Back-Propagation [34], 2) sampling-based methods, i.e., perform sensitivity analysis by sampling points around the given data sample, such as LIME [27], kernel SHAP [23] and CXPlain [30], and 3) hybrid methods, such as SmoothGrad [33], Squared SmoothGrad [33], VarGrad [2], and INFD [39]. On the other hand, global methods include Gradient × Input [31], Integrated Gradients [35], DeepLIFT [31] and LRP [6], among others. These methods do not directly tackle the DMI problem.

For the DMI problem, being inherently interpretable, tree- [29] and rule-based [5] models have been proposed to approximate the output of a complex black-box model with all features. The models themselves provide explanations, including feature importance. However, they may lack the ability for accurate approximations when the original given model is complex. Recently, L2X [9] and VIBI [7] have been proposed as variational methods to learn a neural network-based approximator based on the selected features. The unselected features are masked out by imputing zeros. VIBI improves L2X to encourage the briefness of the learned explanation by adding a constraint for the feature scores to a global prior (in contrast, our priors are conditioned on each sample). Since L2X and VIBI only input features to their explainers, and they directly select features to approximate the model output, therefore, they both may suffer from the sanity, combinatorial shortcuts, model identifiability, model identifiability, and information transmission problems.

In contrast, our method tackle these problems for DMI by leveraging more comprehensive information from the model output, the proposed adversarial infidelity learning mechanism, and the proposed prior-knowledge integration. We note that Zhu et al. [41] proposed an adversarial attention network. However, their objective is to eliminate the difference in extracted features for different learning tasks, which is different from ours.

3 METHODOLOGY

In this section, we present the detailed methodology of our method. First, we define the notations and problem settings of our study.

Consider a dataset $\mathcal{D} = \{x^i\}_{i=1}^n$ consisting of $n$ independent samples. For the $i$th sample, $x^i \in \mathcal{X} \subset \mathbb{R}^d$ is a feature vector with $d$ dimensions, $y^i = \mathcal{M}(x^i) \in \mathcal{Y} \subset \mathbb{R}^c$ is the output vector of a given data-driven model $M \in \mathcal{M}$ (note that $y^i$ may be different from the true label of the sample). The conditional output distribution $p(y \mid x)$ is determined by the given model. For classification tasks, $c$ is the number of classes.

We do not assume the true label of each feature vector is available for training or inference. We develop a neural network-based IFS explainer $E$, which outputs a feature-importance-score vector $\bar{z} \in \mathcal{Z} \subset \mathbb{R}^d$ for a data sample $x$ and the model $M$. As discussed by Yeh et al. [39], the explainer should be a mapping that $E : \mathcal{X} \times \mathcal{M} \rightarrow \mathcal{Z}$. Since it is not trivial to treat an arbitrary model in $\mathcal{M}$ as an input to a neural network, we compromise by involving the model output as an alternative such that $E : \mathcal{X} \times \mathcal{M} \rightarrow \mathcal{Z}$. We select top-$k$ features according to $\bar{z}$, where $k$ is a user-defined parameter. The indices of $k$ selected features are denoted by $S \subset \{1, \ldots, d\}$. For a feature vector $x$, the selected features are denoted by $x_S$, whereas the unselected features are denoted by $x_{\bar{S}}$. Throughout the paper, we denote $[k']$ as the index set $\{1, 2, \ldots, k'\}$ for some integer $k'$.

The goal of our learning approach is to train a neural network-based explainer $E$ over the dataset $\mathcal{D}$ and then generalize it to a testing set to see whether the selected features contribute to sufficient approximate accuracy. The quantitative evaluations of the explainer are described in Section 4.0.2.

3.1 Our Framework

The architecture of our framework is illustrated in Fig. 1. We explain a given model by providing IFS for each specific data sample. The IFS is embodied as a feature attribution mask provided by a learned explainer with the features and the model output of the data sample as inputs. We train an approximator to use selected/masked features to approximate the model output. We also train an adversarial approximator to use unselected features to approximate the model output, and then train the explainer to select features to undermine the approximation, which is referred to as the AIL mechanism. As an extension, integrating efficient model-interpretation methods is also introduced to provide a warm start.

3.2 Adversarial Infidelity Learning

As discussed in the introduction, a straightforward approach to optimize the selection indices $S$ directly maximizing the mutual information $I(x_S; y)$ between selected features $x_S$, and the model output $y$ [7, 9]. To tackle the combinatorial shortcuts and model identifiability problems, we propose an auxiliary objective: minimizing the mutual information $I(x_{\bar{S}}; y)$ between unselected features $x_{\bar{S}}$ and the model output $y$. Because compared with the selected features, the unselected features should contain less useful information. Therefore, the basic optimization problem for $S$ is:

$$\max_S I(x_S; y) - I(x_{\bar{S}}; y) \text{ s.t. } S \sim E(x, y).$$

We can be guided by the Theorem 1 to optimize the explainer.

**Theorem 1.** Define

$$S^* = \arg\max_S \mathbb{E}[\log p(y \mid x_S) - \log p(y \mid x_{\bar{S}})],$$

where the expectation is over $p(y \mid x)$. Then $S^*$ is a global optimum of Problem (1). Conversely, any global optimum of Problem (1) degenerates to $S^*$ almost surely over $p(x, y)$.

The proof is deferred to Appendix A.1. Problem (1) and Theorem 1 show that the auxiliary objective exploits $x_{\bar{S}}$ to involve additional supervised information, and then improves model identifiability.
According to Theorem 1, we develop an approximator $A_y : X \rightarrow Y$ to learn the conditional distribution $p(y | x_S)$. We achieve this by optimizing a variational mapping: $x_S \rightarrow q_y(y | x_S)$ to let $q_y(y | x_S)$ approximate $p(y | x_S)$. We define $q_y(y | x_S) \propto \exp(-\ell_y(y, A_y(x_S)))$, where $\ell_y$ denotes the loss function corresponding to the conditional distribution $p(y | x_S)$ (e.g., mean square error for Gaussian distribution, and categorical cross entropy for categorical distribution), and $x_S \in X$ which is defined as: $(x_S)_j = x_j$ if $j \in S$ and $(x_S)_j = 0$ otherwise. We let $q_m(y | x_S)$ denote the output distribution of $M(x_S)$. We approximate $y$ by $A_y$ instead of $M$, because as discussed by Hooker et al. [15], $p(y | x_S) \neq q_m(y | x_S)$, then $A_y(x_S)$ may approximate more accurately than $M(x_S)$ does.

Similarly, we develop another approximator $A_u : X \rightarrow Y$ to learn $q_u(y | x_S) \propto \exp(-\ell_u(y, A_u(x_S)))$ to approximate $p(y | x_S)$. Then we can show that Problem (1) can be relaxed by maximizing variational lower bounds and alternately optimizing:

$$\max_{A_y, A_u} \mathbb{E}[\log q_y(y | x_S) + \log q_u(y | x_S)] \text{ s.t. } S \sim E(x, y),$$

$$\max_{A_y} \mathbb{E}[\log q_y(y | x_S) - \log q_u(y | x_S)] \text{ s.t. } S \sim E(x, y).$$

First, Problem (3) is optimized to learn $q_y(y | x_S)$ and $q_u(y | x_S)$ to approximate $p(y | x_S)$ and $p(y | x_S)$, respectively. Then Problem (4) is optimized to learn the explainer $E$ to find good explanations according to Theorem 1. Since 1) $q_y(y | x_S)$ is maximized by optimizing $A_y$ and then minimized by optimizing $E$, which is an adversarial learning process, and 2) minimizing $q_u(y | x_S)$ represents indelibility, i.e., undermining performance to approximate $y$ (by excluding selected features), the alternate optimization process can be regarded as an adversarial indelibility learning mechanism.

Since optimizing Problems (3) and (4) for all possible $S$ requires $D^2$ times of computation for the objectives, we follow L2X [9] to apply the Gumbel-softmax trick to approximately sample a $k$-hot vector. Specifically, let $\xi = E(x, y)$ for a pair of inputs $(x, y)$, where for $j \in [d]$, $z_j \geq 0$ and $\sum_j z_j = 1$. Then we define the sampled vector $v \in \mathcal{V} \subset \mathbb{R}^d$, where for a predefined $\tau > 0$,

$$u_j = \max_{l \in [k]} \frac{\exp((\log z_j + \xi_{jl}) / \tau)}{\sum_{j'=1}^m \exp((\log z_{j'} + \xi_{j'l}) / \tau)}, \quad j \in [d],$$

$$\xi_{jl} = -\log(-\log u_{jl})$, $u_{jl} \sim \text{Uniform}(0, 1)$, $j \in [d], l \in [k]$.

Denote the above random mapping by $G : \mathbb{Z} \rightarrow \mathcal{V}$, approximate $x_S \approx x \oplus G(E(x, y))$ and $x_S \approx x \oplus (1^d - G(E(x, y)))$, where $1^d \in \mathbb{R}^d$ with all elements being 1, and $\oplus$ denotes element-wise product. Define the losses for selected and unselected features, respectively,

$$L_S = \frac{1}{n} \sum_{i=1}^n \ell_y(y, A_y(x^i \odot G(E(x^i, y^i))))$$

$$L_u = \frac{1}{n} \sum_{i=1}^n \ell_u(y, A_u(x^i \odot (1^d - G(E(x^i, y^i)))).$$

Then we can relax Problems (3) and (4) as

$$\min_{A_y, A_u} L_S + L_u,$$

$$\min_{E} L_S - L_u.$$

For inference, one can select the top-$k$ features of $E(x^i, y^i)$ for a testing sample $x^i$, where $y^i = M(x^i)$.

### 3.3 Theoretical Analysis

In this section, we first derive variational lower bounds to show the connection between the goal in Eq. (1) and the realization in Eq. (3) and (4). The derivation also shows that the approximator is possible to be superior to the given model to predict the model output using masked features. We also show that our ALL mechanism can mitigate the combinatorial shortcuts problem.

The variational lower bounds are as follows and derived in Appendix A.2. First, for selected features, we have for any $q_y(y | x_S)$:

$$I(x_S; y) = \mathbb{E}_x \mathbb{E}_y \mathbb{E}_{\xi | x_S} \mathbb{E}_{\xi | x_S} [\log p(y | x_S)] + \text{Const},$$

$$= \mathbb{E}_x \mathbb{E}_{\xi | x_S} [\log q_y(y | x_S)] + \text{Const},$$

where the equality holds if and only if $q_y(y | x_S) = p(y | x_S)$. Therefore, if $M(x_S)$’s output distribution $q_m(y | x_S) \neq p(y | x_S)$, it is possible that $\mathbb{E}_y [\log q_y(y | x_S)] > \mathbb{E}_y q_m(y | x_S)$, i.e., $A_y(x_S)$ can be more accurate than $M(x_S)$ to estimate $y$.

Similarly, for unselected features, we have for any $q_u(y | x_S)$:

$$I(x_S; y) = \mathbb{E}_x \mathbb{E}_y \mathbb{E}_{\xi | x_S} \mathbb{E}_{\xi | x_S} [\log p(y | x_S)] + \text{Const},$$

$$= \mathbb{E}_x \mathbb{E}_{\xi | x_S} [\log q_u(y | x_S)] + \text{Const},$$

On the other hand, since $A_y$ actually receives the selected features $x_S$ and the feature-attribution mask $v$ as inputs, where $v = G(E(x, y))$, what $A_y$ actually learns is the conditional distribution $p(y | x_S, v)$. Through the straightforward learning mentioned in the introduction, i.e., removing $L_u$ in Eq. (7) and (8), it could cause the combinatorial shortcuts problem for $A_y$ to learn $p(y | v)$ only, resulting in the feature selection process $x_S$ meaningless. Fortunately, Theorem 2 shows that our ALL can help to avoid this problem by encouraging the independence between $v$ and $y$, then it will be hard for $A_y$ to approximate $y$ solely by $v$. Thus, $A_y$ will have to select useful features from $x$. The proof can be found in Appendix A.3.

**Theorem 2.** For the optimized problem in Eq. (7) and (8), the independence between $v$ and $y$ is encouraged.

### 3.4 Extension Considering Prior Knowledge

As described in Section 3.2, the feature attribution layer (the output layer of the explainer) is in the middle of networks. Since it is optimized by an end-to-end learning process, the information transmission is inefficient. Therefore, we propose to involve efficient interpretations as good priors of feature attribution.

Let $r \in \mathbb{R}^d$ be a feature-importance-score vector generated by another interpretation method $H$ for a sample $x$ and the model $M$. Assume for $j \in [d]$, $r_j \geq 0$ and $\sum_j r_j = 1$, which can be easily achieved through a softmax operation.

Given $z = E(x, M(x))$ for a sample $x$, we can regard $z_j$ as $p(\delta = j \mid x, M, E)$ for $j \in [d]$, where $\delta \in [d]$ denotes whether the $j$th feature should be selected. Similarly, we can regard $r_j$ as $p(\delta = j \mid x, M, H)$.

Therefore, conditional independence between the interpretation models $E$ and $H$ given $\delta, x$ and $M$, we can obtain

$$\rho(\delta = j \mid x, M, E, H) = \frac{\rho(\delta = j \mid x, M, E)p(\delta = j \mid x, M, H)}{\sum_{j'=1}^d \rho(\delta = j' \mid x, M, E)p(\delta = j' \mid x, M, H)}$$

The derivation details can be found in Appendix A.4.
We conduct comprehensive evaluations on five datasets: MNIST, Fashion-MNIST, IMDB sentiment analysis, Zebra, and Google’s Honor of Kings game for teenager recognition, which we refer to as Tencent Gaming Dataset (TGD) in this paper. The detailed re-organization process for each data will be introduced in the following sections.

4 EXPERIMENTS

We conduct comprehensive evaluations on five datasets:

- IMDB sentiment analysis dataset [24],
- MNIST dataset [20] to classify 3 and 8,
- Fashion-MNIST dataset [38] to classify Pullover and Coat,
- ImageNet dataset [10] to classify Gorilla and Zebra,
- our established mobile sensor dataset from Tencent Honor of Kings game for teenager recognition.

We also report the average explanation Time (by second) Per Sample (TPS) on a single NVidia Tesla m40 GPU.

4.0.3 Implementation Details. In Eq. (6), \( \ell_p \) adopts cross-entropy, \( \ell_z \) adopts cross-entropy for IMDB, MNIST, and TGD, and adopts Wasserstein distance [21] for Fashion-MNIST and ImageNet. The weights for \( \ell_{m} \) in Eq. (7) and (8) are 1e-3 for MNIST and \( 1 \) for all other datasets. We adopt the GI method to provide the prior explanation. \( \ell_{c} \) in Eq. (13) is mean absolute error with the weight to be 1e-3 for ImageNet and 0 for others. \( r = 0.5 \) for all the datasets. We constrain the model-capacity of each method to be the same to acquire fair comparisons. For each dataset, we split half test data for validation. For each method on each dataset, we repeat 20 times and report the averaged results. Our implementation uses Keras with Tensorflow [1] backends. We list all other details in Appendix B.

4.1 IMDB

The IMDB [24] is a sentiment analysis dataset which consists of 50,000 movie reviews labeled by positive/negative sentiments. Half reviews are for training, and the other half is for testing. We split half of the testing reviews for validation. For the given model, we follow Chen et al. [9] to train a 1D convolutional neural network (CNN) for binary sentiment classification, and achieve the test accuracy of 88.60%. We develop our approximator with the same architecture as the given model. And we develop our explainer with the 1D CNN used by L2X [9] with a global and a local component. For a fair comparison, each method selects top-10 important words as an explanation for a review.

As shown in Table 1, our method significantly outperforms state-of-the-art baseline methods. Especially, our FS-M score shows nearly optimal fidelity, which is objectively validated by the original given model. Given that our FU-A score is similar to those of baselines, which shows that our selected features are indeed important, we demonstrate that the effectiveness and superiority of our method are significant. We present some examples of selected words of our method and the state-of-the-art baseline VIBI in Fig. 2. As shown in Fig. 2, our method not only selects more accurate keywords, but also provides more interpretable word combinations, such as “I enjoyed it”, “fabulous dancer”, “extremely bad movie”, and “excellent thriller”. Even though “I”, “it”, “dancer”, “movie”, and “thriller” are not apparent whether they are positive or negative for human evaluation, we also follow Chen [9] to evaluate Fidelities of Selected features denoted by FS-H(\%), i.e., whether the predictions made by a human using selected features are consistent with those made by the given model using all the features. We adopt this metric to evaluate whether human can understand how the given model makes decisions. Note that sometimes human may not understand or be satisfied with features selected by the given model. After all, we are explaining the given model, not human.
I have to say that this film was excellently produced and tops the ratings as a typical sci-fi film! I enjoyed it. Its effect film, if you want a thriller watch another channel. This is what the sci-fi lovers want. Excellently produced by one of Sci-Fi's best producers: Robert Vandiver! OK the special effects weren't excellent, but what a great cast! Some more money could have been used for effects but then again what sci-fi is that like these type of films. Films like Mission Impossible and Braveheart are great but those are entirely different types of films. Sci-fi produces excellent films like Sabretooth , Aliens, Hunter etc... Well done... I never realized what a gorgeous Lara Turner was until I saw this movie. She was only 19 years old and gorgeous. What a pleasure to watch her dance with George Murphy. The story line was typical for its day but the dancing was really special. I never tire of watching Fred and Ginger but Lara Turner in this movie was just as terrific. I always thought of Lana as a soap actress who tended to over act. She should have done more dancing and less of the Madison X and Peyton Place roles. I had a new appreciation for her after seeing this movie and her "Oscar" dancing. Too bad the "Academy" doesn't give an Oscar for dancing.

(1) Given Model: Positive; VIBI: Positive; Ours: Positive
(2) Given Model: Positive; VIBI: Positive; Ours: Positive
(3) Given Model: Negative; VIBI: Positive; Ours: Negative
(4) Given Model: Positive; VIBI: Negative; Ours: Positive

Legend: Words selected by VIBI; Words selected by Ours; Words selected by both models

Figure 2: Examples of explanations on the IMDB dataset. The labels predicted by the original given model using all the words, the words selected by the state-of-the-art baseline VIBI, and the words selected by our method are shown, respectively, at the bottom of each panel. Keywords picked by VIBI, our method, and both methods are highlighted in blue, green, and yellow, respectively. In (1) and (2), the given model output consistent predictions using the selected words by both VIBI and ours, whereas (3) and (4), the prediction using full words is inconsistent with that using VIBI’s selected words but is still consistent with that using our selected words. Best view in colors.

Table 1: Results on the IMDB dataset. \(^1\) denotes the method uses additional information.

| Method | FS-M | FU-M | FS-A | FS-A | FS-H | TPS |
|--------|------|------|------|------|------|-----|
| Grad   | 85.58 | 87.58 | 86.18 | 85.79 | 73.58 | 5e-5 |
| GI     | 87.31 | 86.25 | 87.88 | 83.86 | 78.23 | 5e-5 |
| LME    | 89.75 | 82.13 | 88.53 | 82.96 | 83.98 | 3e-2 |
| SHAP   | 50.17 | 99.16 | 50.24 | 99.50 | 53.22 | 4e-2 |
| †CXP   | 90.60 | 80.01 | 90.70 | 83.04 | 84.97 | 1e-4 |
| INFN   | 40.50 | 99.80 | 64.50 | 96.70 | 46.27 | 3e-0 |
| L2X    | 89.23 | 82.90 | 89.05 | 83.81 | 83.49 | 1e-4 |
| VIBI   | 90.79 | 80.36 | 90.12 | 82.57 | 84.33 | 1e-4 |
| Ours   | 98.48 | 59.05 | 98.70 | 81.83 | 92.98 | 1e-4 |

4.1.1 Human Evaluation. We also evaluate with the help of humans to quantify how interpretable are those selected words. We randomly select 500 reviews in the testing set for this human evaluation. We invite 20 Tencent employees to infer the sentiment of a review given only the selected words. The explanations provided by different interpretation methods are randomly mixed before sent to these employees. The final prediction of each review is averaged over multiple human annotations. For the explanations that are difficult to infer the sentiments, we ask the employees to provide random guesses. Finally, as shown by the FS-H scores in Table 1, our method significantly outperforms baseline methods as well.

Table 2: Results of the ablation study.

| Method | FS-M | FU-M | FS-A | FS-A | FU-A |
|--------|------|------|------|------|------|
| Ours   | 98.48 | 59.05 | 98.70 | 81.83 | 81.83 |
| w/o Output | 92.47 | 54.28 | 91.99 | 79.82 | 81.83 |
| w/o AIL | 78.31 | 97.62 | 99.33 | 94.42 | 81.83 |
| w/o Prior | 98.38 | 60.22 | 98.97 | 81.07 | 81.07 |

4.1.2 Ablation Study. We evaluate three variants of our method by ablating our three components, i.e., the model output feedback (Output), AIL, and prior knowledge-based warm start (Prior). In Table 2, we show the effectiveness of both the model output feedback and AIL. It is worthy of mentioning that, although the warm start strategy does not improve the final scores, it boosts the convergence rate at the start of optimization, as shown in Fig. 3. 4.1.3 Sanity Check. We perform the model and data randomization tests suggested by Adebayo et al. [3]. We evaluate the sanity by the cosine correlation between binary masks, i.e., the original mask, and the other one resulted from randomization. The sanity scores for model and data randomization tests are 9.39% and 10.25%, respectively, which shows that our explanations are dependent on models and then are valid.

4.2 MNIST

We select the data of 3 and 8 of the MNIST dataset [20] for binary classification with 11,982 training and 1984 testing images. We train a 2D CNN with two convolutional layers for the classification and achieve the test accuracy of 99.89%. We develop our approximators are by the same architecture as the given model, whereas we develop...
Figure 3: The FS-M scores of our method with and without the prior knowledge-based warm start.

Figure 4: Examples of explanations on the MNIST dataset. In each panel, an original image and the masked images by VIBI and our method are presented from left to right. The labels inferred by the original model using the full images are 3 and 8 for panels (1) and (2), respectively.

Table 3: Results on the MNIST dataset.

| Method | FS-M | FU-M | FS-A | FU-A | SEN | FS-H | TPS |
|--------|------|------|------|------|-----|------|-----|
| Grad   | 98.19 | 68.75 | 99.55 | 99.55 | 139 | 94.37 | 5e-5 |
| GI     | 99.45 | 67.64 | 99.55 | 99.24 | 120 | 94.58 | 5e-5 |
| LIME   | 80.37 | 99.75 | 82.46 | 99.80 | 62.6 | 70.29 | 3e-2 |
| SHAP   | 63.29 | 55.78 | 93.97 | 95.58 | 84.59 | 4e-2 |
| †CXP   | 59.65 | 16.50 | 94.85 | 95.10 | 107 | 1e-4 |
| INFD   | 59.65 | 16.50 | 94.85 | 95.10 | 107 | 1e-4 |
| L2X    | 91.38 | 91.18 | 98.54 | 99.65 | 6.90 | 86.58 | 1e-4 |
| VIBI   | 98.29 | 86.29 | 99.29 | 99.65 | 6.35 | 92.17 | 1e-4 |
| Ours   | 99.04 | 74.70 | 99.80 | 99.70 | 6.11 | 94.46 | 1e-4 |

Table 4: Results on the Fashion-MNIST dataset.

| Method | FS-M | FU-M | FS-A | FU-A | SEN | TPS |
|--------|------|------|------|------|-----|-----|
| Grad   | 58.60 | 76.20 | 93.70 | 95.45 | 2528 | 5e-5 |
| GI     | 62.15 | 66.25 | 93.35 | 94.45 | 2662 | 5e-5 |
| LIME   | 75.63 | 94.30 | 73.60 | 97.35 | 61.03 | 3e-2 |
| SHAP   | 63.29 | 55.78 | 93.97 | 95.58 | 84.59 | 4e-2 |
| †CXP   | 59.65 | 16.50 | 94.85 | 95.10 | 107 | 1e-4 |
| INFD   | 100.0 | 45.80 | 100.0 | 100.0 | 87.37 | 5e-1 |
| L2X    | 77.30 | 87.30 | 89.85 | 96.00 | 1.76 | 1e-4 |
| VIBI   | 84.10 | 70.85 | 91.90 | 94.40 | 17.36 | 1e-4 |
| Ours   | 97.80 | 66.65 | 99.80 | 99.65 | 0.70 | 0.70 | 1e-4 |

As shown in Fig. 4 (2), though the VIBI masked image is similar to 8, but it is also close to a 3. In contrast, our masked image can never be 3. Since we are interpreting the recognition logic of model rather than that of humans, it is important to select features in favor of the machine logic, e.g., considering both possibility and impossibility.

4.2.1 Human Evaluation. We randomly select 500 images in the testing set for this human evaluation. We invite 15 Tencent employees who are experts in machine learning to perform the same binary classification given only the masked images. Specifically, we ask the subjects to provide two scores in the range of [0, 1] for each image. Each score is for the possibility of each class (3 or 8). We perform $\ell_1$ normalization for the scores as the final prediction probabilities. Other settings and procedures are similar to Section 4.1.1. Finally, as shown by the FS-H scores in Table 3, our method significantly outperforms baseline methods as well.

4.3 Fashion-MNIST

The Fashion-MNIST dataset [38] is a dataset of Zalando’s article images, which consists of 28 x 28 images of 10 classes. We select the data of Pullover and Shirt for binary classification with 12,000 training and 2000 testing images. We train a 2D CNN with the same architecture as that for MNIST for the classification and achieve the test accuracy of 92.20%. The architectures of our approximators and explainer are also the same as those for MNIST. Each method selects top-64 pixels as an explanation for an image.

As shown in Table 4, our method outperforms state-of-the-art baseline methods except for the INFD method. Since INFD adds...
Intense Operation with Both Hands

Figure 6: Examples of explanations on the ImageNet dataset. In each panel, an original image and the masked images by VIBI and our method are presented from left to right. Best view in colors.

Table 5: Results on the ImageNet dataset.

| Method     | FS-M | FU-M | FS-A | FU-A | SEN  | TPS  |
|------------|------|------|------|------|------|------|
| Grad       | 55   | 91   | 66   | 99   | 1e+6 | 2e-3 |
| GI         | 56   | 95   | 74   | 89   | 1e+6 | 2e-3 |
| LIME       | 77   | 98   | 72   | 96   | 76.1 | 1e-0 |
| SHAP       | 66   | 96   | 75   | 90   | 89.4 | 2e-0 |
| CXP        | 77   | 96   | 61   | 95   | 41.38| 5e-3 |
| INFD       | -    | -    | -    | -    | -    | 4e+3 |
| L2X        | 78   | 99   | 75   | 96   | 41.32| 5e-3 |
| VIBI       | 78   | 98   | 78   | 96   | 40.10| 5e-3 |
| Ours       | 83   | 98   | 90   | 93   | 39.91| 5e-3 |

4.5 TGD

Last, we apply our method to the Tencent Gaming Dataset (TGD), which consists of 100 million samples from 5 million gamers. Each sample is a $3 \times 643$ time-series data with the time dimension and feature dimension of 3 and 643, respectively. We extract the features from inertia sensors and touch information of mobile phones, in both time and frequency domains, and categorize in 41 groups. Each feature vector of a sample corresponds to a 2-second operation during the game. Three vectors are ordered by time, but not necessarily continuous in time. The learning task is the teenage gamer (age $\leq 17$) recognition. The original model is a stacked LSTM with an accuracy of 90.16%. The approximator uses the same structure, and the explainer is also a stacked LSTM. Our method achieves the FS-M, FU-M, FS-A, FU-A, and SEN scores of 95.68%, 82.24%, 95.33%, 82.37%, and 0.18%, respectively, selecting only 10% of features. We show examples of selected features in Fig. 7. In Fig. 7 (1), the teenage gamer performs a complex operation excitedly at the start but performs a monotonous/regular operation at the end. Whereas, in Fig. 7 (2), the adult gamer starts with casual flipping of the mobile phone, and ends with a complex/skilled operation.

5 CONCLUSION

In this paper, we investigate the model interpretation problem in the favored direction of Instance-wise Feature Selection (IFS). We propose a Model-agnostic Effective Efficient Direct (MEED) IFS framework for model interpretation. Specifically, we consider the model output feedback as an additional input to learn an explainer to mitigate the sanity and information transmission problems. Furthermore, we propose an adversarial infidelity learning (AIL) mechanism to screen relative unimportant features for mitigating the combinatorial shortcuts and the model identifiability problems. Our theoretical analyses show that AIL can mitigate the model identifiability and combinatorial shortcuts problems. Our experimental results reveal that AIL can mitigate the model identifiability problem and learn more necessary features. Moreover, our extension...
to integrate efficient interpretation methods as proper priors has been shown to provide a warm start and mitigate the information transmission problem. Comprehensive empirical evaluation results provided by quantitative metrics and human evaluation demonstrate the effectiveness, superiority, and robustness of our method.

ACKNOWLEDGMENTS

Jian Liang, Bing Bai, Yuren Cao, and Kun Bai would like to acknowledge the support from AWS Machine Learning for Research Award and Google Faculty Research Award.

REFERENCES

[1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning. In OSDI. Vol. 16.

[2] Julius Adiaboy, Justin Gilmer, Ian Goodfellow, and Been Kim. 2018. Local explanation methods for deep neural networks lack sensitivity to parameter values. arXiv preprint arXiv:1810.03307 (2018).

[3] Julius Adiaboy, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In Advances in Neural Information Processing Systems. 9595–9515.

[4] Marco Ancona, Enesa Celzini, Cengiz Oztireli, and Markus Gross. 2017. A unified view of gradient-based attribution methods for deep neural networks. In NIPS 2017-Workshop on Interpreting, Explaining and Visualizing Deep Learning. ETH Zurich.

[5] Robert Andrews, Joachim Diederich, and Alan B Tickle. 1995. Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-based systems 8, 6 (1995), 373–389.

[6] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one 10, 7 (2015), e0131462.

[7] Seojin Bang, Pengtao Xie, Wei Wu, and Eric Xing. 2019. Explaining a black-box using Deep Variational Information Bottleneck Approach. arXiv preprint arXiv:1902.06918 (2019).

[8] Tathagata Chakraborti, Anagha Kulkarni, Sarah Sreedharan, David E Smith, and Subbarao Kambhampati. 2019. Explainability? Legibility? Predictability? Transparency? Privacy? Security? the emerging landscape of interpretable agent behavior. In Proceedings of the International Conference on Automated Planning and Scheduling. Vol. 29. 86–96.

[9] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In International Conference on Machine Learning. 883–892.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: a large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition. Ieee, 248–255.

[11] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. 2019. Explanations can be manipulated-and-geometry-is-to-blame.pdf

[12] Mengnan Du, Ninghao Liu, and Xia Hu. 2019. Techniques for interpretable machine learning. Comm. ACM 63, 1 (2019), 68–77.

[13] Satoshi Hara, Kouchi Bere, Tasuku Soma, and Takanori Maehara. 2019. Feature Attribution As Feature Selection. https://openreview.net/forum?id=H1IS6oa5SYQ

[14] Juyeon Heo, Sungwhan Joo, and Taesup Moon. 2019. Fooling Neural Network Interpretations via Adversarial Model Manipulation. In Advances in Neural Information Processing Systems 32: H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (Eds.) Curran Associates, Inc., 13567–13578. http://papers.nips.cc/paper/9511-explanations-can-be-manipulated-and-geometry-is-to-blame.pdf

[15] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A benchmark for interpretability methods in deep neural networks. In Advances in Neural Information Processing Systems. 9734–9745.

[16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04681 (2017).

[17] Sarthak Jain and Byron C Wallace. 2019. Attention is Not Explanation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 3543–3556.

[18] Alexa Jolicoeur-Martineau. 2018. The relativistic discriminator: a key element missing from standard GAN. arXiv preprint arXiv:1807.00734 (2018).

[19] Ashkan Khakzar, Soroosh Baselzadeh, Saurabh Khanduja, Seong Tae Kim, and Nasir Navab. 2019. Explaining Neural Networks via Perturbing Important Learned Features. arXiv preprint arXiv:1911.11081 (2019).

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86, 11 (1998), 2278–2294.

[21] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. 2019. Sliced Wasserstein discrepancy for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 10285–10295.

[22] Zachary C Lipton. 2018. The mythos of model interpretability. Queue 16, 3 (2018), 31–57.

[23] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems. 4765–4774.

[24] Andrew L. 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-volume 1. Association for Computational Linguistics, 142–150.

[25] Seongsik Park, Sejoon Kim, Hyeokjun Choe, and Sungroh Yoon. 2019. Fast and efficient information transmission with burst spikes in deep spiking neural networks. In 2019 56th ACM/IEEE Design Automation Conference (DAC). IEEE, 1–6.

[26] Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. 2018. Model agnostic supervised local explanations. In Advances in Neural Information Processing Systems. 2515–2524.

[27] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 1135–1144.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention. Springer, 234–241.

[29] Patrick Schwab and Helmut Hlavacs. 2015. Capturing the essence: Towards the automated generation of transparent behavior models. In Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference.

[30] Patrick Schwab and Walter Karlen. 2019. CXPlain: Causal explanations for model interpretation under uncertainty. In Advances in Neural Information Processing Systems. 10220–10230.

[31] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 3545–3553.

[32] 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).

[33] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viegas, and Martin Wattenberg. 2017. Smalldrop: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017).

[34] J. Springenberg, Alexey Dosovitskiy, Thomas Brox, and M. Biedermier. 2015. Striving for Simplicity: The All Convolutional Net. In ICLR (workshop track).

[35] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 3319–3328.

[36] Fei Wang, Raimu Kaulash, and Dhiruv Khullar. 2019. Should Health Care Demand Interpretable Artificial Intelligence or Accept “AI Black Box”? Medicine? Annals of Internal Medicine (2019).

[37] Sarah Wiegreffe and Yuval Pianer. 2019. Attention is not not Explanation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 11–20.

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

[39] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. 2019. On the (In) Fidelity and Sensitivity of Explanations. In Advances in Neural Information Processing Systems. 10965–10976.

[40] XinYang Zhang, Ningfei Wang, Shouling Ji, Hua Shen, and Ting Wang. 2018. Interpretable Deep Learning under Fire. arXiv preprint arXiv:1812.00891 (2018).

[41] XinYang Zhang, Shoushan Li, and Guodong Zhou. 2019. Adversarial Attention Modeling for Multi-dimensional Emotion Regression. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 471–480.
A PROOFS AND DERIVATIONS

A.1 Proof of Theorem 1

Proof. The proof follows that of Theorem 1 of Chen et al. [9].
(1) Forward direction: Given the definition of $S^*$, we have for any pair $(x, y)$, and any explainer $E : S \mid x, y$,
\[
\mathbb{E}[(\log p(y \mid x) - \log p(y \mid S^*)) - \log p(y | \Delta^*_{y})]
\]
(14)
In the case when $S^*$ is a set instead of a singleton, we identify $S^*$ with any distribution that assigns arbitrary probability to each element in $S^*$, and with zero probability outside $S^*$. With abuse of notation, $S^*(x, y)$ indicates both the set function that maps every pair $(x, y)$ to a set $S^*$ and any real-valued function that maps $(x, y)$ to an element in $S^*$. Taking expectation over the distribution of $(x, y)$, and adding $\mathbb{E}[(\log p(y))]$ at both sides, we have
\[
I(x^*; y) - I(x^*; y) \leq I(x^*; y) - I(x^*; y)
\]
(15)
for any explainer $E : S \mid x, y$.
(2) Reverse direction: The reverse direction is proved by contradiction. Since the optimal explanation $S^*$ for any element in $S^*$ is a deterministic function in the set of distributions $S$, and with zero probability outside $S^*$. Outside $S$, we always have
\[
\mathbb{E}[(\log p(y \mid x) - \log p(y \mid S^*)) - \log p(y | \Delta^*_{y})] = \mathbb{E}[(\log p(y \mid x) - \log p(y \mid S^*)) - \log p(y | \Delta^*_{y})]
\]
(18)
which is a contradiction to Eq. (16).

A.2 Derivations For The Variational Lower Bounds

Proof. First, for selected features, we have:
\[
I(x^*; y) = \mathbb{E}[(\log p(y \mid x^*) - \log p(y \mid x^*)) - \log p(y)]
\]
(21)
For any $q_s(y \mid x)$, we obtain the lower bound by applying the Jensen's inequality:
\[
\mathbb{E}_{y|x_s}[\log p(y \mid x_s)] \geq \int [\log q_s(y \mid x_s)]dp(y \mid x_s)
\]
(22)
It is similar for unselected features.

A.3 Proof of Theorem 2

Proof. For the problem in Eq. (7) and (8), $A_h$ learns $p(y \mid x, 1)$, where 1 is a vector with all the elements being 1. Therefore, our AIL mechanism learns $y$ to minimize $p(y \mid x, 1 - v)$. Since
\[
I(x^*_{s}, 1 - v; y)
\]
(23)
the mutual information $I(x^*_{s}, 1 - v; y)$ is minimized. By the property of mutual information, we assume that the minimization of $I(x^*_{s}, 1 - v; y)$ encourages the independence between $(x^*_{s}, 1 - v)$ and $y$, which leads to
\[
p(x^*_{s}, 1 - v, y) = p(x^*_{s}, 1 - v)p(y)
\]
(24)
Thus, by marginalizing $x^*_{s}$ at both sides, we have
\[
\int_{x^*_{s}} p(x^*_{s}, 1 - v, y) = \int_{x^*_{s}} p(x^*_{s}, 1 - v)p(y)
\]
(25)
\[
\Rightarrow p(1 - v, y) = p(1 - v)p(y).
\]
Therefore, the independence between $1 - v$ and $y$ is encouraged as well. Since $1 - v$ and $y$ are deterministic between each other, then for any set $S_1$ such that $v \in S_1$, there exists a fixed set $S_2$ such that $1 - v \in S_2$, and vice versa. Thus we have for any set $S_1$,
\[
p(y \in S_1, y) = p(1 - v \in S_2, y)
\]
(26)
Thus, the independence between $v$ and $y$ is also encouraged.

A.4 Derivation for Eq. (11)

Proof.
\[
p(\delta = j \mid x, M, E, H)
\]
(27)
where
\[ C(E, H | x, M) = \frac{p(E | x, M)p(H | x, M)}{p(E, H | x, M)}. \] (28)

The third equation is from the assumption of conditional independence between the interpretation models \( E \) and \( H \) given \( x \) and \( M \). Assuming \( p(\delta = j | x, M) = \frac{1}{d} \) for all \( j \in [d] \), because we have no knowledge of the explainer, then we have
\[
p(\delta = j | x, M, E, H) = \frac{p(\delta = j | x, M, E, H)}{\sum_{j' = 1}^{d} p(\delta = j' | x, M, E, H)} = \frac{p(\delta = j | x, M) p(\delta = j | x, M, H)}{\sum_{j' = 1}^{d} p(\delta = j' | x, M, H)}. \] (29)

\[ \square \]

\section*{B IMPLEMENTATION DETAILS}

\subsection*{B.1 Details for Our Method}

For \( \ell_u \) with cross entropy loss, in Eq. (8) we still minimize \( L_u \), and just replace the target \( y \) for \( \ell_u \) by \( 1 - y \), following the suggestion of the Relativistic GAN et al. [18]. For \( \ell_d \) with the Sliced Wasserstein distance [21], the number of random vectors is 128 for Fashion-MNIST and 256 for ImageNet.

For optimizers, we use RMSprop for IMDB and Adadelta for image datasets, with the default hyperparameters. The learning rates are fixed. For TGD, we use Adam with learning rate of \( 2e - 4 \), \( \beta_1 = 0.5, \beta_2 = 0.999 \), with the learning-rate decay of 30% for every 50,000 steps. The batch size is 32 for IMDB and ImageNet, 128 for MNIST and Fashion-MNIST, and 1024 for TGD. The hyperparameter tuning set for both \( L_u \) and \( L_c \) is \( \{0, 10^{-3}, 10^{-2}, 10^{-1}, 1\} \).

For IMDB, we adopt the structure of the original model of L2X [9] for the same structure for our original model and approximators. We adopt the structure of the explainer of L2X [9] for the structure for our explainer, with a global and a local component. For MNIST and Fashion-MNIST, the structure of our original model and approximators is shown in Table 6. Whereas the structure for our explainer is shown in Table 7. For ImageNet, we adopt the MobileNet module in the package of keras.applications.mobilenet without the top layer as our explainer. The model parameters pre-trained on ImageNet are fixed. We only stack a global max-pooling layer and learn a full-connected top layer. We adopt the pre-process_input function in the keras.applications.mobilenet package for image pre-processing. We perform a max-pooling of kernel size and stride of 4, before generate the feature important scores, and perform an up-sampling with kernel size and stride of 4 when masking pixels. We adopt the adopt the U-Net [28] for our explainer, whose structure is complex and then omitted due to space limitations. Similarly, the structures of our modules for TGD are also omitted. Readers may refer to the publicly-available code for more implementation details.

For IMDB, the model output \( y \) is input to a MLP with three hidden layers with 100 neurons and the ReLu activation, before being concatenated to the global component of the explainer. For image datasets, the model output \( y \) is linearly mapped to the same shape with the first channel of an image and is concatenated to the raw image as an additional channel. For TGD, the model output \( y \) is linearly mapped to the same shape as an raw data sample and concatenated to the data sample in the feature dimension.

\begin{table}[h]
\centering
\caption{The CNN structure for our original model and approximators for MNIST and Fashion-MNIST.}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Layer & \# Filters & Kernel Size & Stride & \# Padding & Activation \\
\hline
Convolution & 32 & 3 & 3 & default & ReLu \\
\hline
Convolution & 64 & 3 & 3 & default & ReLu \\
\hline
Max-pooling & - & 2 & 2 & default & - \\
\hline
Dropout(0.25) & - & - & - & - & - \\
\hline
Flatten & - & - & - & - & - \\
\hline
Fully-Connected & 128 & - & - & - & ReLu \\
\hline
Dropout(0.5) & - & - & - & - & - \\
\hline
Fully-Connected & 2 & - & - & - & Softmax \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{The CNN structure for our explainer for MNIST and Fashion-MNIST.}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Layer & \# Filters & Kernel Size & Stride & \# Padding & Activation \\
\hline
Convolution & 32 & 3 & 3 & same & ReLu \\
\hline
Convolution & 64 & 3 & 3 & same & ReLu \\
\hline
Convolution & 1 & 3 & 3 & same & Linear \\
\hline
\end{tabular}
\end{table}

\subsection*{B.2 Details for Baseline Methods}

For Grad, we compute the gradient of the selected class with respect to the input feature and uses the absolute values as importance scores. We perform summation operations to form the importance scores with proper shapes. For GI, the gradient is multiplied by the input feature before calculate the absolute value. For INF-D, we select the Noisy Baseline method for consistent comparisons, since its another method Square is only suitable for image datasets. The structures of explainers are the same for CXP, L2X, VIBI, and Ours. The structures of original models and approximators are the same for L2X, VIBI, and Ours.

The hyper-parameters of each method are tuned according the strategy mentioned in their respective papers.

On ImageNet, for all the baseline methods, we perform a max-pooling of kernel size and stride of 4 for the feature important scores, and perform an up-sampling with kernel size and stride of 4 when masking pixels.