CoRTX: Contrastive Learning for Real-time Explanation

Anonymous authors
Paper under double-blind review

Abstract

Recent advancements in explainable machine learning provide effective and faithful solutions for interpreting model behaviors. However, most explanation methods encounter efficiency issues, which largely limits their deployments in practical scenarios. Real-time explainer (RTX) frameworks have thus been proposed to accelerate the model explanation process by learning an one-feed-forward explainer. Existing RTX frameworks typically build the explainer under the supervised learning paradigm, which require large amounts of explanation labels as ground truth. Considering that accurate explanation labels are usually hard to obtained, due to the constrained computational resources and limited human efforts, effective explainer training is still challenging in practice. In this work, we propose a COntrastive Real-Time eXplanation (CoRTX) framework that adopts contrastive learning to relieve the intensive dependence of explainer training on explanation labels. Specifically, we design a synthetic strategy to select positive and negative samples for explanation representation learning. Theoretical analysis show that our selection strategy can benefit the contrastive learning process on explanation tasks. Experimental results on three real-world datasets further demonstrate the efficiency and effectiveness of our proposed CoRTX framework.

1 Introduction

The remarkable progress in explainable machine learning (ML) significantly improve the model transparency to human beings (Du et al., 2019). However, applying explainable ML techniques to real-time scenarios remains to be a challenging task. Real-time systems typically require explanation methods to be not only human-understandable and faithful, but also efficient (Stankovic et al., 1992). Due to the requirements from both stakeholders and social regulations (Goodman & Flaxman, 2017; Floridi, 2019), efficient explanation methods are necessary for the real-time ML systems, such as the key decisions in controlling systems (Steel & Angwin, 2010), advertisement recommendation on e-commerce systems (Yang et al., 2018), and denotations in healthcare systems (Esteva et al., 2019; Gao et al., 2017). Nevertheless, existing work on local explanation methods suffer from high explanation latency, including LIME (Ribeiro et al., 2016), KernelSHAP (Lundberg & Lee, 2017), GradCAM (Selvaraju et al., 2017), Integrated Gradient (Sundararajan et al., 2017) and RISE (Petsiuk et al., 2018). These methods rely on either multiple perturbations or backward propagation in deep neural networks (DNNs) for explanation (Covert & Lee, 2021; Liu et al., 2021), which can be time-consuming and limited in deployment on the real-time scenarios.

Real-time explainer (RTX) frameworks have thus been proposed to address such efficiency issue and provide faithful explanations for real-time systems (Dabkowski & Gal, 2017; Jethani et al., 2021b). Specifically, RTX learns a global explainer on the training set by using the ground-truth explanation labels which obtained from exact calculation or approximation. RTX then provides explanations for each testing instance via a single feed-forward process. Existing efforts on RTX can be categorized into two lines of work. The first line (Jethani et al., 2021b; Covert et al., 2022) explicitly learns an explainer to minimize the estimation error from the ground-truth explanation label. The second line (Dabkowski & Gal, 2017; Chen et al., 2018; Kanehira & Harada, 2019) trains the feature mask generators based on certain constraints on predefined label distribution for feature selection. Despite the effectiveness of existing RTX frameworks, recent advancements still rely on the large amounts of explanation label under the supervised learning paradigm. The computational cost of deriving explanation labels is extremely high (Roth, 1988; Winter, 2002), indicating that existing supervised RTXs thereby limit the wide deployment in real-world scenarios.

To tackle the aforementioned challenges, we propose a COntrastive Real-Time eXplanation (CoRTX) framework based on the contrastive learning techniques. CoRTX aims to learn the explanation representation of each data instance without any ground-truth explanation label. Contrastive learning has been widely exploited for improving the learning processes of downstream tasks by providing well pre-trained representative embeddings (Arora et al., 2019; He et al., 2020). In particular, a task-oriented selection strategy of positive and negative pairs (Chen et al., 2020a; Khosla et al., 2020) can shape the representation properties through contrastive learning. Motivated by this existing contrastive schemes, CoRTX develops an explanation-oriented contrastive learning framework to generate well pre-trained explanation representation, with the goal of further fine-tuning in the downstream explanation tasks.

CoRTX learns the explanation representation to deal with explanation tasks by minimizing the contrastive loss function (Van den Oord et al., 2018), Specifically, CoRTX framework designs a synthetic positive and negative sampling strategy to learn the explanation representation. The pre-trained explanation representation can then be transformed to feature attribution by fine-tuning an explanation head using a tiny amount of explanation label. The theoretical analysis and experimental results demonstrate that CoRTX can successfully provides effective explanation representation for different explanation tasks. Overall, the contributions can be summarized as follows:

- CoRTX provides a contrastive learning framework for generating explanation representation, which can effectively reduce the required amount of explanation labels.
- Theoretical analysis indicates our proposed CoRTX framework is able to generate effective explanation representation and bounds the explanation error.
- Empirical results demonstrate our proposed CoRTX can efficiently provide faithful explanation on both tabular and image datasets.

2 Preliminary

2.1 Notations

We consider an arbitrary DNN $f(\cdot)$ as the target model to interpret. Let input feature be $\boldsymbol{x} = [x_1, \cdots, x_M] \in \mathcal{X}$, where $x_1, \cdots, x_M$ denote the value of input feature $1, \cdots, M$, respectively. The contribution of each feature to the model output can be treated as a cooperative game on the feature set $\mathcal{X}$ (Shapley, 1953). Specifically, the preceding difference $f(\widetilde{\boldsymbol{x}}_{\mathcal{S}\cup\{i\}}) - f(\widetilde{\boldsymbol{x}}_{\mathcal{S}})$ indicates the contribution of feature i under feature subset $\mathcal{S} \subseteq \boldsymbol{\mathcal{U}} \setminus \{i\}$, where $\boldsymbol{\mathcal{U}}$ is the entire feature set. The overall contribution of feature i is formalized as the average preceding difference considering all possible feature subsets $\mathcal{S}$, which can be formally given by

$$\phi_i(\boldsymbol{x}) := \mathbb{E}_{\mathcal{S}\subseteq\boldsymbol{\mathcal{U}}\setminus\{i\}}\left[f(\widetilde{\boldsymbol{x}}_{\mathcal{S}\cup\{i\}}) - f(\widetilde{\boldsymbol{x}}_{\mathcal{S}})\right]. \tag{1}$$

where $\widetilde{\boldsymbol{x}}_{\mathcal{S}} = \mathbf{S} \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S}) \odot \boldsymbol{x}_r$ denotes the perturbed sample, $\mathbf{S} = \mathbf{1}_{\mathcal{S}} \in \{0,1\}^M$ is a masking vector of $\mathcal{S}$, and $\boldsymbol{x}_r = \mathbb{E}[\boldsymbol{x} \,|\, \boldsymbol{x} \sim P(\boldsymbol{x})]$ denotes the reference values[1] of each feature. The computational complexity of Equation 1 grows exponentially with the feature number M, which prevents its application to real-time explanation. To this end, we propose an efficient and faithful framework for the real-time explanation in this work.

2.2 Real-time Explainer

The Real-Time explainer framework (RTX) maintains a global model to provide a fast explanation for each data instance via one feed-forward process. Generally, RTX attempts to learn the global explanation distribution under two different methodologies, Shapley-sampling-based approaches (Wang et al., 2021; Jethani et al., 2021b; Covert et al., 2022) and feature-selection-based approaches (Chen et al., 2018; Dabkowski & Gal, 2017; Kanehira & Harada, 2019). The first line of approaches enforces a DNN explainer to simulate the given approximated Shapley distribution for generating explanation results. The second line of approaches assumes the specific predefined feature patterns or distributions, and formulates the explainer learning strategy based on the given hypothesis. As Shapley value is well supported with solid theoretical backbones, Shapley-sampling-based approaches are typically more faithful for training RTX frameworks.

Different from the local explanation frameworks (Lundberg & Lee, 2017; Lomeli et al., 2019) that require multiple explaining models for each data instance, RTX frameworks only require one global

[1] Other statistic probability families can also be adopted for generating the reference value.

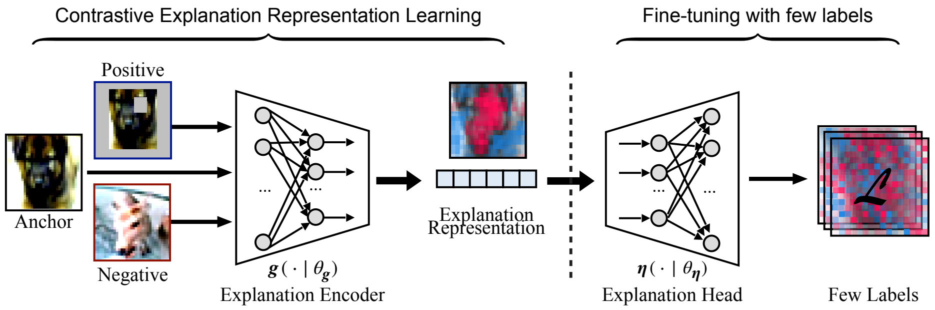

Contrastive Explanation Representation Learning
Fine-tuning with few labels
Positive
Anchor
Negative
$g(\cdot \mid \theta_g)$
Explanation Encoder
Explanation Representation
$\eta(\cdot \mid \theta_\eta)$
Explanation Head
$\mathcal{L}$
Few Labels

Figure 2: Pre-training explanation representation from the explanation encoder in CoRTX and fine-tuning on explanation head.

model for explanation. Compared to the local explanation frameworks, the advantages of RTX are as follows: (1) obtaining faster inference utility and (2) ensuring similar data instances with similar explanations. One of the work (Chen et al., 2018) provides a feature masking generator for real-time feature selection. The training process of mask generator is under the constraint of the given ground-truth label distribution. Another work, FastSHAP (Jethani et al., 2021b), proposes a state-of-the-art Shapley-sampling-based framework for learning RTX to derive feature attribution. With sufficient amounts of masking sampling, FastSHAP exploits a DNN model to imitate the Shapley distribution among the training data instances to yield an efficient RTX. In general, Shapley-sampling-based approaches are more faithful but suffered from high computational costs when examining larger feature subsets, while feature-selection-based approaches are easy-to-train but constrained under the specific distributions. In this work, we propose an unsupervised framework, CoRTX, to benefit with the advantages of both works. This makes CoRTX to be more efficient and faithful.

2.3 Limitation of Supervised Framework

Supervised RTX relies on enormous quantities of ground-truth explanation annotations, which limits its application to real-world scenarios. It is experimentally proved that supervised RTX suffers from performance degradation without sufficient explanation annotations. Specifically, taking the Shapley values as the ground-truth explanation, the supervised RTX is learned to minimize the mean square error on the training dataset, and then estimates explanations on the testing dataset for evaluation. Preliminary experiments are conducted on the Census Income dataset Dua & Graff (2017) which demonstrates the necessity of supervised RTX to use enormous quantities of explanation annotations. The explanation ranking performance versus the exploitation ratio of explanation annotations for training is given in Figure 1. Implementation details are given Appendix B.

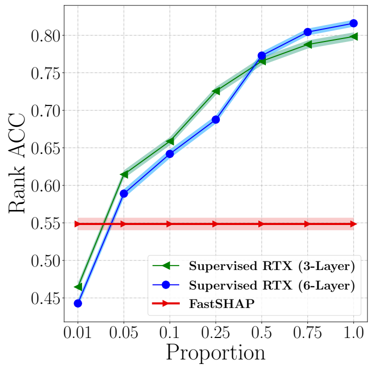


Figure 1: The explanation ranking versus the ratio of explanation annotations on supervised training.

According to Figure 1, we observe that the performance of explanation ranking significantly grows as the ratio of annotations increases. However, the complexity of estimating the ground-truth explanation is extremely high (i.e., the computational complexity of the Shapley values grows exponentially with the feature number). It is challenging to have a well-trained RTX considering the limited computational resources in real-world scenarios. To tackle this problem, we propose a CoRTX framework to pre-train the explaners without any explanation labels on the training dataset. As shown in left-hand side of Figure 2, CoRTX first propose a explanation encoder to generate explanation representation without any supervision for downstream explanation task. After the explanation encoder has been well-trained, as shown in right-hand side of Figure 2, a explanation head can converge to optimal even very limited explanation labels are available to exploit.

3 Contrastive Real-time Explanation

We systematically introduce our CoRTX framework in this section. Figure 2 demonstrates the overall pipeline of our proposed CoRTX. CoRTX is built under a constrastive learning paradigm with positive data augmentation and negative data sampling (Kim et al., 2016; Dhurandhar et al., 2018) for providing a real-time explanation.

3.1 Positive Data Augmentations toward Similar Explanation

Different from conventional data augmentation (Liu et al., 2021) for representation learning, CoRTX develops an explanation-oriented data augmenting strategy for training RTX. Consider an anchor datapoint $\boldsymbol{x} \in \mathcal{X}$, the synthetic positive data collection is randomly sampled by m times of perturbation of $\boldsymbol{x}$, which is formally defined as:

$$\mathcal{X}^{+} = \{\mathbf{S}_i \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S}_i) \odot \boldsymbol{x}_r \mid \mathbf{S}_i \sim \mathcal{B}(M, 0.5)\}, \tag{2}$$

where $\mathcal{B}(M, 0.5)$ denotes M-dimensional binomial distribution. An optimal compact positive datapoint is ideally composed of similar important features to the anchor data point, which leads to a slight variation in prediction scores between the two datapoints. Thus, CoRTX determines an compact positive datapoint $\widetilde{\boldsymbol{x}}^{+}$ by holding the minimum difference gap between the prediction scores of $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}^{+}$. Formally, we make the selection strategy as follows:

$$\widetilde{\boldsymbol{x}}^{+} = \arg \min_{\widetilde{\boldsymbol{x}}_i \in \mathcal{X}^{+}} \ |f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}}_i)|. \tag{3}$$

We further propose Theorem 1 to provide a theoretical analysis to support the proposed positive data augmentation in CoRTX. Theorem 1 basically shows that the ideal greedy selection strategy in CoRTX can degrade the error difference of explanation values between the anchor point $\boldsymbol{x}$ and the positive sample $\widetilde{\boldsymbol{x}}^{+}$. This demonstrates the compact positive sample $\widetilde{\boldsymbol{x}}^{+}$ has similar explanation to anchor point $\boldsymbol{x}$. The proof of Theorems 1 is provided in Appendix A.

Theorem 1 (Compact Alignment). *Let $f(\boldsymbol{x})$ be a K-Lipschitz continuous function for the given input sample $\boldsymbol{x}$ and $\boldsymbol{\Phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \cdots, \phi_M(\boldsymbol{x})]$ be the importance score of each feature, where $\phi_i(\boldsymbol{x}) := \mathbb{E}_{\mathcal{S} \subseteq \boldsymbol{\mathcal{U}} \setminus \{i\}} \left[f(\widetilde{\boldsymbol{x}}_{\mathcal{S} \cup \{i\}}) - f(\widetilde{\boldsymbol{x}}_{\mathcal{S}}) \right]$. Given a perturbed sample $\widetilde{\boldsymbol{x}} \in \mathcal{X}^{+}$ satisfying $\min_{1 \leq i \leq M} \phi_i(\widetilde{\boldsymbol{x}}) \geq 0$, the explanation difference $||\phi(\boldsymbol{x}) - \phi(\widetilde{\boldsymbol{x}})||_2$ is bounded by the prediction difference $|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})|$ as*

$$||\phi(\boldsymbol{x}) - \phi(\widetilde{\boldsymbol{x}})||_2 \leq (1 + \sqrt{2}\gamma_0)|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})| + \sqrt{M}\gamma_0, \tag{4}$$

where $\gamma_0 = K||\boldsymbol{x}||_2$ and $K \geq 0$ is the Lipschitz constant of function $f(\cdot)$.

According to Theorem 1, we can obtain an optimal positive sample for the least upper bound of $||\phi(\boldsymbol{x}) - \phi(\widetilde{\boldsymbol{x}})||_2$, when CoRTX is able to select an optimal compact positive sample under the selected mask conditions. The selected masking condition in Theorem 1 ensures that each non-masking feature obtains positive contributions toward the model prediction $f(\cdot)$, which prevents the ideal candidate of compact positive samples from being a set of noise sampling. However, hard positive sample selection has been proved to benefit the contrastive learning (Grill et al., 2020; Kalantidis et al., 2020). The compact positive data may be a slightly perturbed sample when selecting from the universe set of perturbed datapoints and aggravate the contrastive learning process of CoRTX. In practice, considering to generate a hard positive sample, CoRTX receives an compact positive data point $\widetilde{\boldsymbol{x}}_i^{+}$ by selecting from the subset of synthetic positive set $\mathcal{X}^{+}$.

3.2 explanation Contrastive Loss

Unlike the downstream tasks focusing by existing contrastive learning approaches (He et al., 2020; Chen et al., 2020b), CoRTX adopts the contrastive learning with positive data augmentation and negative sampling to generate the explanation representation of each data instance. Specifically, a positive pair includes an anchor data instance $\boldsymbol{x}_i$ and a compact perturbed augmentation $\widetilde{\boldsymbol{x}}_i^{+}$. A negative pair contains an anchor data instance $\boldsymbol{x}_i$ and another data point $\boldsymbol{x}_j$, where $j \neq i$.

The proposed explanation encoder $g(\cdot \mid \theta_g) : \mathbb{R}^M \to \mathbb{R}^d$ in CoRTX aims to generate the explanation representation. Let $\boldsymbol{h}_i = g(\boldsymbol{x}_i \mid \theta_g)$ be the encoded explanation representation of $\boldsymbol{x}_i$ and $\widetilde{\boldsymbol{h}}_i^{+} = g(\widetilde{\boldsymbol{x}}_i^{+} \mid \theta_g)$ be the encoded compact positive sampling $\widetilde{\boldsymbol{x}}_i^{+}$. CoRTX updates the explanation encoder $g(\cdot \mid \theta_g)$ by measuring the similarity through dot product. Given an anchor encoded representation $\boldsymbol{h}_i$, the explanation encoder $g(\cdot \mid \theta_g)$ can be optimized with one encoded positive sample pair $(\boldsymbol{h}_i, \widetilde{\boldsymbol{h}}_i^{+})$ and N encoded negative sample pairs $(\boldsymbol{h}_i, \boldsymbol{h}_j)$, where $i \neq j$. We follow the existing work (He et al., 2020; Chen et al., 2020a) to construct the contrastive loss function of CoRTX and minimize the contrastive loss function illustrated as follows:

$$\mathcal{L}_g = -\log \frac{\exp\left(\mathbf{h}_i \cdot \widetilde{\boldsymbol{h}}_i^{+} / \tau\right)}{\sum_{j=1}^{N} \exp\left(\mathbf{h}_i \cdot \mathbf{h}_j / \tau\right)}, \tag{5}$$

where τ is a temperature hyper-parameter Wu et al. (2018). Note that the form of contrastive loss function can be replaced with other implementations (Jaiswal et al., 2020).

To provide downstream results from pre-train representation, a explanation head $\boldsymbol{\eta} : \mathbb{R}^d \to \mathbb{R}^M$ is exploited to generate the feature explanation in the downstream tasks. Specifically, $\boldsymbol{\eta}(\cdot \mid \theta_{\boldsymbol{\eta}})$ is learned with pre-trained explanation representation made from $g(\boldsymbol{x}_i \mid \theta_g)$ and very few amount of explanation labels (e.g., the Shapley Values), which makes the computational cost be affordable in real-world scenarios. Moreover, $\boldsymbol{\eta}(\cdot \mid \theta_{\boldsymbol{\eta}})$ can be designed to adapt different scenarios of model interpretation. In this work, CoRTX is built under two common explanation scenarios, which are the feature attribution task and feature importance ranking task.

To ensure the accuracy of generated explanation from CoRTX, We here provide Theorem 2 to reveal the efficacy of the explanation error of CoRTX is under theoretical support. The proof of Theorems 2 is provided in Appendix A.

Theorem 2 (**Explanation Error Bound**). *Given a training set $\mathcal{D}$, a testing set $\mathcal{C}$, and a well-trained explainer $\hat{\boldsymbol{\phi}} = \boldsymbol{\eta} \circ g$, where g denotes the contrastive explanation encoder to generate low rank explanation embeddings and $\boldsymbol{\eta}$ represents explanation head. Assume $\hat{\boldsymbol{\phi}}(\cdot)$ is a K_h-Lipschitz continuity. For all $\boldsymbol{x}_j \in \mathcal{D}$ in training set, if there exist $\mathcal{E} > 0$, such that the training error satisfies $||\boldsymbol{\phi}(\boldsymbol{x}_j) - \hat{\boldsymbol{\phi}}(\boldsymbol{x}_j)||_2 \leq \mathcal{E}$. Then, for any testing datapoint $\boldsymbol{x}_k \in \mathcal{C}$, the testing explanation error Err($\hat{\boldsymbol{\phi}}$) $= ||\boldsymbol{\phi}(\boldsymbol{x}_k) - \hat{\boldsymbol{\phi}}(\boldsymbol{x}_k)||_2$ can be bounded as:*

$$||\boldsymbol{\phi}(\boldsymbol{x}_k) - \hat{\boldsymbol{\phi}}(\boldsymbol{x}_k)||_2 \leq (1 + \sqrt{2}\gamma_0)|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}}_k^+)| + \sqrt{M}\gamma_0 + \mathcal{E} + K_h||\boldsymbol{h}_{\widetilde{\boldsymbol{x}}_k^+} - \boldsymbol{h}_{x_k}||_2 \tag{6}$$

where $\widetilde{\boldsymbol{x}}_k^+$ is the compact positive sample, $\boldsymbol{h}_{\boldsymbol{x}_i} = g(\boldsymbol{x}_i)$, $\gamma_0 = K||\boldsymbol{x}_k||_2$, and $K_h \geq 0$ is the Lipschitz constant of prediction model $f(\cdot)$.

With theoretical analysis, we conclude two advantages brought from CoRTX in Remark 1. The first advantage lies in the explanation representation learning. The second advantage takes from the compact positive selection strategy.

Remark 1. *Within our proposed CoRTX framework, the upper bound of explanation error Err($\hat{\boldsymbol{\phi}}$) can be reduced due to the following reasons:*

- *Our proposed CoRTX minimizes the representation distance of $||\boldsymbol{h}_{\widetilde{\boldsymbol{x}}_k} - \boldsymbol{h}_{\boldsymbol{x}_k}||_2$ which contributes to minimize the last item of explanation error bound.*
- *Our proposed CoRTX framework provides a compact positive selection strategy to get smaller $|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}}_k^+)|$ than randomly selection.*

3.3 Algorithm of CoRTX

The outline of CoRTX is given in Algorithm 1. Specifically, CoRTX first follows Equation 3 to receive compact positive selection (lines 4-5), and then updates the explanation encoder $g(\cdot \mid \theta_g)$ according to Equation 5 (line 6). The training iterations stop once the explanation encoder $g(\cdot \mid \theta_g)$ converges to the optimal. After CoRTX framework converges, $g(\cdot \mid \theta_g)$ generates the explanation representation for each data instance $\boldsymbol{x} = [x_1, \cdots, x_M] \in \mathcal{X}$. A explanation head $\boldsymbol{\eta}(\cdot \mid \theta_{\boldsymbol{\eta}})$ is then fine-tuned by involving very limited explanation labels and explanation representation for generating efficient and faithful explanations.

Algorithm 1: Explanation Encoder $g(\cdot \mid \theta_g)$ in CoRTX

```
1 Input DNN model $f$, input values $\boldsymbol{x} = [x_1, \cdots, x_M]$.
2 Output Estimation contribution values of each feature $[\hat{\phi}_1, ..., \hat{\phi}_M]$.
3 while not convergence do
4     Generate the synthetic positive instances $\mathcal{X}^+ = \{\mathbf{S}_i \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S}_i) \odot \boldsymbol{x}_r \mid \mathbf{S}_i \sim \mathcal{B}(M, 0.5)\}$.
5     Select the compact positive sampling $\widetilde{\boldsymbol{x}}^+ = \arg\min_{\widetilde{\boldsymbol{x}}_i \in \mathcal{X}^+} |f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}}_i)|$ and the set of
        negative samples $\{\boldsymbol{x}_j \mid j \neq i\}$.
6     Update $g(\cdot \mid \theta_g)$ with $\widetilde{\boldsymbol{x}}^+$ and $\boldsymbol{x}_j$ to minimize loss function given by Equation 5.
7 end
```

4 Experiments

We conduct the experiments to demonstrate the effectiveness of CoRTX, aiming to answer the following research questions: **RQ1:** How does CoRTX perform compared with state-of-the-art baselines? **RQ2:** What is the impact brought from pre-trained CoRTX to reduce usage amount of ground-truth explanation labels? **RQ3:** How does the explanation representation generated by CoRTX improve the generalization to the explanation tasks?

4.1 Datasets and Baselines

Datasets. Our experiments consider two tabular datasets: Census Income (Dua & Graff, 2017) with 13 features, Bankruptcy (Liang et al., 2016) with 96 features and one image dataset: CIFAR-10 (Krizhevsky et al., 2009) with 32 × 32 pixels. The preprocessing and statistics details of three datasets are all provided in Appendix B. **Baseline Methods.** On the Census Income and Bankruptcy datasets, CoRTX is compared with two RTX models, which are Supervised RTX, FastSHAP (Jethani et al., 2021b), and two representative local explanation models, KernelSHAP (KS) (Kokhlikyan et al., 2020) and Permutation Sampling (PS) (Mitchell et al., 2021). On the CIFAR-10 dataset, CoRTX is compared with one RTX model, FastSHAP (Jethani et al., 2021b), and several local explanation models, DeepSHAP (Lundberg & Lee, 2017), Saliency (Simonyan et al., 2013), Integrated Gradients (Sundararajan et al., 2017), SmoothGrad Smilkov et al. (2017), and GradCAM (Selvaraju et al., 2017). More details about the baseline methods can be referred to Appendix B.

4.2 Experimental Settings and Evaluation Metrics

In this section, we introduce the experimental settings and metrics used for evaluating CoRTX. The task settings and implementation details are shown as follows.

Feature Attribution Task. The task-specific CoRTX under such training is denoted as CoRTX-MSE. CoRTX-MSE first provides pre-trained explanation representation $\boldsymbol{h}_i$ of input instance $\boldsymbol{x}_i$ from $g(\boldsymbol{x}_i \mid \theta_g)$. After this, feature attribution score is estimated based on a regression-oriented explanation head $\boldsymbol{\eta}(\boldsymbol{h}_i \mid \theta_{\boldsymbol{\eta}})$ with tiny amounts of Shapley-based explanation labels. The explanations are generated by $[\hat{\phi}_1, \cdots, \hat{\phi}_M] = \boldsymbol{\eta}(\boldsymbol{h}_i \mid \theta_{\boldsymbol{\eta}})$, where $\hat{\phi}_i$ indicates the contribution of feature i. The explanation head $\boldsymbol{\eta}(\boldsymbol{h}_i \mid \theta_{\boldsymbol{\eta}})$ learns to minimize the mean-square loss $\mathcal{L}_{\text{MSE}} = \frac{1}{M}\sum_{i=1}^{M}(\hat{\phi}_i - \phi_i)^2$ between estimated scores $\boldsymbol{\eta}(\boldsymbol{h}_i \mid \theta_{\boldsymbol{\eta}}) = \hat{\phi}_i$ and explanation labels ϕ_i. Since CoRTX-MSE is evaluated on Shapley-based explanation values, we adjust the predicted explanation scores to fit the additive property of Shapley values on evaluation stage, following the common additive efficient normalization (Jethani et al., 2021b; Ruiz et al., 1998). To evaluate the performance of CoRTX, the estimated feature attribution scores are measured by the $\boldsymbol{\ell}_2$-error (Jethani et al., 2021b). The mean $\boldsymbol{\ell}_2$-error indicates the Euclidean distance error from the explanation labels given by

$$\boldsymbol{\ell}_2\text{-error} = \frac{1}{N}\sum_{n=1}^{N}\left[\sqrt{\sum_{i=1}^{M}(\phi_i - \hat{\phi}_i)^2}\,\right]_n, \tag{7}$$

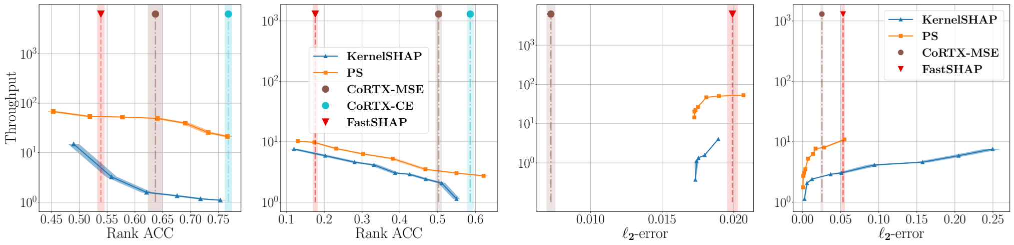
where N denotes the cardinality of testing set and M is feature numbers of each data instance.

Feature Importance Ranking Task. In this set of experiments, our explanation head $\boldsymbol{\eta}(\boldsymbol{h}_i \mid \theta_{\boldsymbol{\eta}})$ aims to estimate the feature importance ranking index $[\hat{\mathbf{r}}_1, \cdots, \hat{\mathbf{r}}_M]$. The task-specific CoRTX in this scenario is denoted as CoRTX-CE. Here, the ground-truth ranking annotations are assumed to be the ranking index from explanation scores. Given the ranking annotations $[\mathrm{r}_1, \cdots, \mathrm{r}_M]$ of each training sample, the explanation head $\boldsymbol{\eta}(\boldsymbol{h}_i \mid \theta_{\boldsymbol{\eta}})$ learns the feature importance ranking with pre-trained explanation representation. CoRTX-CE minimize the loss function given by $\mathcal{L}_{\text{CE}} = \sum_{i=1}^{M} l(\hat{\mathbf{r}}_i, \mathrm{r}_i; \theta_h)$, where $l(\hat{\mathbf{r}}_i, \mathrm{r}_i; \theta)$ denotes the cross entropy loss. The feature importance ranking is evaluated by the ranking accuracy (Wang et al., 2022). Specifically, it indicates the ratio of the correct-ranked features to all features . By considering more significance for the important features than the trivial features, the accuracy of descending feature ranking is given by:

$$\text{Rank ACC} = \frac{\sum_{m=1}^{M}\frac{\mathbf{1}_{\hat{r}_m = r_m}}{m}}{\sum_{m=1}^{M}\frac{1}{m}}, \tag{8}$$

where the factor $\frac{1}{m}$ at the numerator highlights the contributions of important features to the accuracy, and the factor $\sum_{m=1}^{M}\frac{1}{m}$ at the denominator normalizes the accuracy such that $0 \leq \text{Rank ACC} \leq 1$.

Evaluation of Efficiency. To evaluate speed performance of the above explanation scenarios, we adopt algorithmic throughput to evaluate the efficiency (Wang et al., 2022). The measurement is calculated by $\text{Throughput} = \frac{N_{\text{test}}}{t_{\text{total}}}$, where N_{test} and t_{total} denote the testing instance number and the overall time consumption of generating the explanations, respectively. For the three datasets, the testing instance number N_{test} is given in Appendix B, and t_{total} is measured based on the physical computing infrastructure given in Appendix B.



(a) Census Income. (b) Bankruptcy. (c) Census Income. (d) Bankruptcy.

Figure 3: Explanation throughput versus ranking accuracy on Census Income (a) and Bankruptcy dataset (b). Explanation throughput versus ℓ_2-error on Census Income (c) and Bankruptcy dataset (d).

Implementation Details. Three different prediction models $f(\cdot)$ are applied for verifying the agnostic property of proposed CoRTX. AutoInt (Song et al., 2019) is adopted for Census Income dataset, MLP model is used for Bankruptcy dataset, and ResNet-18 is considered for CIFAR-10. To generate ground truth explanation labels, we calculate the exact Shapley values for Census Income dataset. However, due to the high computational resource requirements, we utilize the approximated methods to generate explanation labels until the convergence Covert & Lee (2021); Jethani et al. (2021b). Since estimated values from Antithetical Permutation Sampling (APS) (Mitchell et al., 2021; Lomeli et al., 2019) and KernelSHAP Lundberg & Lee (2017) are proved to approach to exact Shapley values when it obtains high samples Covert & Lee (2021). Thus, APS is set to generate the explanation labels for Bankruptcy dataset and KernelSHAP yields the explanation labels for CIFAR-10. More implementation details are introduced at Appendix B.

4.3 Tabular Experiments

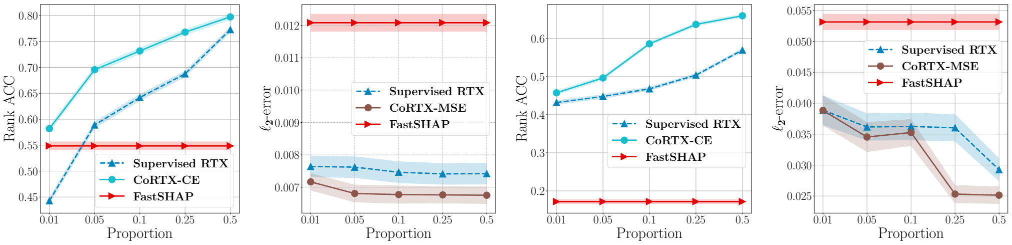
4.3.1 Explanation and Efficiency Performance (RQ1)

We compare CoRTX with the baseline methods on efficacy and generating speed of the predicted explanations. The results are shown by ℓ_2-error versus throughput on feature attribution task and $\mathrm{Rank\ ACC}$ versus throughput on feature importance ranking task. RTX frameworks, such as CoRTX and FastSHAP, build on the training set and evaluate on the testing dataset. For other local methods, such as KernelSHAP and Permutation Sampling, the explanations are generated based on 2^n times of model evaluation, where $3 \leq n \leq 11$. Larger times of model evaluation given in local methods indicate a smaller throughput. The reported results of CoRTX adopting 25% of the explanation labels on fine-tuning stage. The outcomes of explanation throughput versus explanation performance on the Census Income dataset is illustrated in Figures 3 (a) and (c) and demonstrate Figures 3 (b) and (d) on the Bankruptcy dataset under feature attribution task and ranking task.

- **CoRTX vs. Local Methods**: For KernelSHAP and PS, a sharp decrease of ranking accuracy and an increase of the ℓ_2-error can be observed as the growth of throughput. Compared with local explanation methods obtaining large times of model evaluations, CoRTX achieves competitive explanation performance. For example, CoRTX is competitive to PS with 2^8 times of model evaluation and KernelSHAP with 2^{10} times on Census Income dataset. This indicates KernelSHAP and PS suffer from an undesirable tradeoff between explanation speed and performance. In contrast, our proposed CoRTX provides both fast and accurate explanation.
- **CoRTX vs. FastSHAP**: CoRTX outperforms FastSHAP on Rank ACC and ℓ_2-error under the same level of throughput. This shows that our proposed CoRTX provides accurate and faithful explanations in the scenario of RTX.
- **CoRTX-MSE vs. CoRTX-CE**: CoRTX successfully provides accurate solutions for two explanation tasks. CoRTX-CE outperforms CoRTX-MSE in the feature ranking task, while CoRTX-MSE still remains competitive performance on feature attribution task. The results indicate the necessity of selecting appropriate downstream loss on fine-tuning stage. To sum up, CoRTX has the potential capability and flexibility to be applied to different scenarios of model explanation.

4.3.2 Impact from Pre-trained Explanation Representation (RQ2)

In this set of experiments, we investigate the effects of pre-trained explanation representation brought to the usage amount of explanation labels. Given the fixed pre-trained explanation representation, Figure 4 demonstrates the explanation performance of the RTX frameworks with different annotation usage ratios. We compare CoRTX with two baseline RTX frameworks (i.e., FastSHAP and supervised RTX). For the fair comparison, CoRTX and supervised RTX adopt the same loss function under the same task. We summarize the key results as follows:



(a) Census Income. (b) Census Income. (c) Bankruptcy. (d) Bankruptcy.

Figure 4: Explanation performance with different proportion of explanation label usage on two tabular dataset. CoRTX outperforms SOTA baselines by using only 5% of labels.

- **Effectiveness of Pre-training**: By providing pre-trained explanation representation, CoRTX-MSE and CoRTX-CE consistently outperform the supervised RTX on different label proportions. The results on two explanation tasks shows that CoRTX provides effective explanation representation $\boldsymbol{h}_i$ on fine-tuning the explanation head $\boldsymbol{\eta}(\boldsymbol{h}_i \mid \theta_{\boldsymbol{\eta}})$.
- **Sparsity of Annotations**: CoRTX-MSE and CoRTX-CE can both provide faithful explanation results when explanations labels are very limited. This indicates CoRTX can be potentially applied to real-world large scale datasets because the computational complexity of generating the explanation labels is extremely high in practice.
- **Effectiveness of Fine-tuning**: Both supervised RTX and CoRTX outperform FastSHAP on ranking accuracy and ℓ_2-error under 5% of explanation label usage. CoRTX can significantly reduce the usage of good quality labels by providing pre-trained explanation representation. For the fast explanation yielding, FastSHAP synchronously generates the approximated explanation labels during the explainer training. However, comparing FastSHAP to supervised RTX, the results indicates that exact explanation labels usage (e.g., exact Shapley values) benefits the explanation performance than approximated labels. This reveals that CoRTX obtains the advantages from limited usage of exact labels and leads to fast estimation and accurate explanation.

4.3.3 Ablation Studies on Synthetic Positive Augmentations

In this section, we conduct an ablation test on the greedy positive augmenting strategy in Section 3.1 to evaluate the efficacy on explanation. We perform the ablation experiments under the Census dataset on the feature ranking and feature attribution task. First, the greedy positive augmenting strategy is replaced with a random selection of candidate synthetic positive data, which is denoted as *CoRTX w/o Compact Alignment (CA)*. Second, we replace the proposed greedy positive augmenting strategy with a maximum positive augmenting strategy in our ablation studies, which is denoted as *CoRTX w/ Maximum Alignment (MA)*. The other settings of *CoRTX w/o CA* and *CoRTX w/ MA* follow the traditional contrastive learning, which uniformly samples the positive data. Figure 5 demonstrates the results of *CoRTX w/o CA* and *CoRTX w/ MA* comparing to our proposed CoRTX. As shown in the figure, we can observe that CoRTX outperforms *CoRTX w/o CA* and *CoRTX w/ MA* under different proportions of explanation label usage on two explanation tasks. The results reveal that the proposed explanation-oriented sampling strategy in CoRTX can significantly benefit the efficacy of explanation results. The results in this ablation study demonstrate the necessity of adopting Compact Alignment in the CoRTX framework instead of positive sampling from conventional contrastive learning.

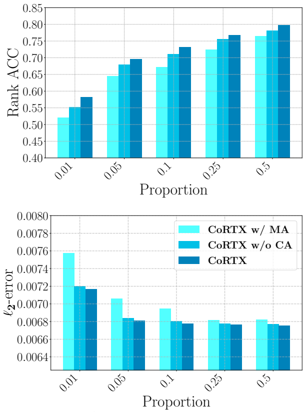


Figure 5: Ablation Study on Positive Augmentations.

4.4 Image Experiments (RQ3)

Unlike tabular data that obtains relatively few features, image data is composed of high-dimension pixel features. In this section, we evaluate the performance of explanation representation generated by CoRTX on the CIFAR-10 dataset. We compare CoRTX with FastSHAP utilizing all explanation labels and six other local image explaining methods. CoRTX generates the efficient image explanation through one-feed-forward model prediction process. The explanation results of CoRTX are fine-tuned using 5% ground-truth explanation labels. CoRTX and FastSHAP output 2×2 superpixel attribution (Jethani et al., 2021b) for explaining the outcome of image classification.

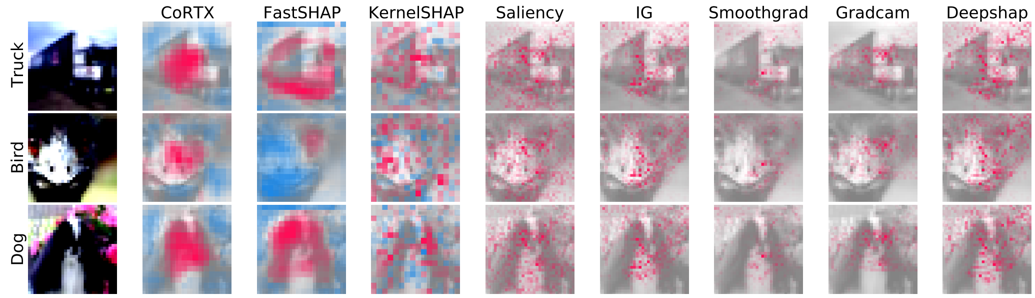


Figure 6: Explanations generated on CIFAR-10 Dataset.

	Top-1 Accuracy		ΔLog-odds	
	Exclusion	Inclusion	Exclusion	Inclusion
CoRTX	**0.373** ± 0.004	**0.764** ± 0.022	**2.790** ± 0.060	**1.615** ± 0.026
FastSHAP	0.420 ± 0.005	**0.782** ± 0.005	**2.896** ± 0.045	1.642 ± 0.033
KernelSHAP	0.395 ± 0.005	0.764 ± 0.004	2.449 ± 0.055	1.687 ± 0.031
Saliency	0.497 ± 0.006	0.730 ± 0.004	2.075 ± 0.052	2.055 ± 0.036
IG	0.560 ± 0.005	0.726 ± 0.004	1.818 ± 0.053	2.040 ± 0.039
Smoothgrad	0.472 ± 0.007	0.731 ± 0.005	2.287 ± 0.054	2.111 ± 0.036
Gradcam	0.563 ± 0.006	0.734 ± 0.004	1.824 ± 0.049	2.062 ± 0.037
Deepshap	0.555 ± 0.006	0.730 ± 0.005	1.870 ± 0.054	2.040 ± 0.035

Table 1: Exclusion and Inclusion AUCs and ΔLog-odds. The evaluation scores of each methods are calculated from the average scores of five times repetitions. The model performs better when obtaining lower Exclusion AUC and Inclusion ΔLog-odds; and encountering higher Inclusion AUC and ΔLog-odds.

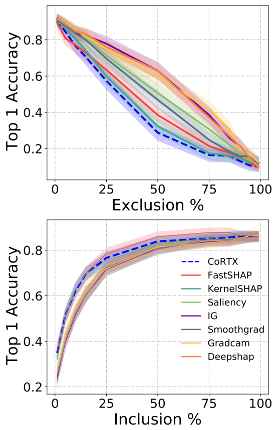


Figure 7: AUC on CIFAR-10

4.4.1 A Case Study of CoRTX

We visualize the predicted explanation results in Figure 6. It shows that CoRTX, FastSHAP, and GradCAM are able to highlight the relevant objects corresponding to the image labels. Specifically, CoRTX provides a more faithful explanation result and highlights a precise region which is important for model prediction. In contrast, we observe that the importance regions localized by FastSHAP are larger than relevant objects, which reveals to be less accurate to the explanations. Moreover, GradCAM provides importance regions only on partial relevant objects, which makes the explanations less faithful. The other baselines are less competitive due to the wrong regions highlighted, or noisy saliency maps provided. These observations validate that the explanations from CoRTX are more faithfulness. More case studies are available in Appendix D.

4.4.2 Quantitative Evaluation

To investigate the quality of the estimated explanation results on CIFAR-10, we compare CoRTX with two Shapley-based models and several gradient-based local explainers on exclusion and inclusion tasks. Following the experimental setting from existing work (Petsiuk et al., 2018; Jethani et al., 2021b), we utilize Top-1 accuracy and ΔLog-odds from image classification task as the metric. The evaluated images are orderly masked according to the estimation of feature importance scores. Once the important pixels are removed from the input images on exclusion, we expect the Top-1 accuracy to drop drastically and obtain a lower exclusion AUC. On the contrary, we expect to gain higher inclusion AUC for better performance on the inclusion task. ΔLog-odds reveals the opposite instructions where higher exclusion AUC and lower inclusion task denotes better explanation performance. Figure 6 and Table 1 reveal the results of AUC curves for the exclusion and inclusion tasks under a set of 1000 images. We observed that CoRTX outperforms all the other baselines on the two tasks, which obtains the lowest exclusion AUC with Top-1 Accuracy and lowest inclusion AUC with ΔLog-odds. CoRTX is also competitive with the state-of-the-art baseline, FastSHAP, on two other remaining tasks and performs significantly better than other gradient-based explainers since gradient-based methods are less faithful than perturbation-based methods.

5 Conclusion

In this work, we propose a contrastive RTX framework, CoRTX, which significantly reduces the usage amount of explanation labels. Specifically, CoRTX introduces a synthetic positive samples selection on contrastive learning for generating pre-trained explanation representation and fine-tunes on the downstream explanation tasks. We also provide theoretical analysis to support the effectiveness of CoRTX in learning explanation representation. The experimental results on three datasets demonstrate that CoRTX works more effectively and faithfully compared with other RTX frameworks and local explanation methods. As for the future exploration, we will explore more on explanation representation and reduce the limited labels requirement into zero label exploitation.

REFERENCES

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pp. 883–892. PMLR, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Ian Covert and Su-In Lee. Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics*, pp. 3457–3465. PMLR, 2021.

Ian Covert, Chanwoo Kim, and Su-In Lee. Learning to estimate shapley values with vision transformers. *arXiv preprint arXiv:2206.05282*, 2022.

Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *Advances in neural information processing systems*, 30, 2017.

Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Advances in neural information processing systems*, 31, 2018.

Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019.

Dheeru Dua and Casey Graff. UCI machine learning repository. (2007), 2017. URL `http://archive.ics.uci.edu/ml`.

Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

Luciano Floridi. Establishing the rules for building trustworthy ai. *Nature Machine Intelligence*, 1(6): 261–262, 2019.

Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. *arXiv preprint arXiv:2006.01272*, 2020.

Jun Gao, Ninghao Liu, Mark Lawley, and Xia Hu. An interpretable classification framework for information extraction from online healthcare forums. *Journal of healthcare engineering*, 2017, 2017.

Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In *International Conference on Artificial Intelligence and Statistics*, pp. 1459–1467. PMLR, 2021a.

Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. Fastshap: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2021b.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33: 21798–21809, 2020.

Atsushi Kanehira and Tatsuya Harada. Learning to explain with complemental examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8603–8611, 2019.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Deron Liang, Chia-Chi Lu, Chih-Fong Tsai, and Guan-An Shih. Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. *European Journal of Operational Research*, 252(2):561–572, 2016.

Yang Liu, Sujay Khandagale, Colin White, and Willie Neiswanger. Synthetic benchmarks for scientific research in explainable machine learning. 2021.

Maria Lomeli, Mark Rowland, Arthur Gretton, and Zoubin Ghahramani. Antithetic and monte carlo kernel estimators for partial rankings. *Statistics and Computing*, 29(5):1127–1147, 2019.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777, 2017.

Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. Sampling permutations for shapley value estimation. *arXiv preprint arXiv:2104.12199*, 2021.

Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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*, pp. 1135–1144, 2016.

Alvin E Roth. Introduction to the shapley value. *The Shapley value*, pp. 1–27, 1988.

Luis M. Ruiz, Federico Valenciano, and Jose M. Zarzuelo. The family of least square values for transferable utility games. *Games and Economic Behavior*, 24(1):109–130, 1998. ISSN 0899-8256. doi: https://doi.org/10.1006/game.1997.0622. URL https://www.sciencedirect.com/science/article/pii/S0899825697906229.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Lloyd S Shapley. *A Value for N-Person Games*. 1953.

Weichen Shen. Deepctr: Easy-to-use,modular and extendible package of deep-learning based ctr models. https://github.com/shenweichen/deepctr, 2017.

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

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

Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1161–1170, 2019.

John A Stankovic et al. Real-time computing. *Byte, pág*, pp. 155–162, 1992.

Emily Steel and Julia Angwin. On the web's cutting edge, anonymity in name only. *The Wall Street Journal*, 4, 2010.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pp. arXiv–1807, 2018.

Guanchu Wang, Yu-Neng Chuang, Mengnan Du, Fan Yang, Quan Zhou, Pushkar Tripathi, Xuanting Cai, and Xia Hu. Accelerating shapley explanation via contributive cooperator selection. *arXiv preprint arXiv:2206.08529*, 2022.

Rui Wang, Xiaoqian Wang, and David I Inouye. Shapley explanation networks. *arXiv preprint arXiv:2104.02297*, 2021.

Eyal Winter. The shapley value. *Handbook of game theory with economic applications*, 3:2025–2054, 2002.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Fan Yang, Ninghao Liu, Suhang Wang, and Xia Hu. Towards interpretation of recommender systems with sorted explanation paths. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 667–676. IEEE, 2018.

Appendix

A Proof of Theorem

Theorem 1 (Compact Alignment). *Let $f(\boldsymbol{x})$ be a K-Lipschitz continuous function for given input sample $\boldsymbol{x}$ and $\boldsymbol{\Phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \cdots, \phi_M(\boldsymbol{x})]$ be the importance score of each feature, where $\phi_i(\boldsymbol{x}) := \mathbb{E}_{\mathcal{S} \subseteq \boldsymbol{\mathcal{U}} \setminus \{i\}} \big[f(\widetilde{\boldsymbol{x}}_{\mathcal{S} \cup \{i\}}) - f(\widetilde{\boldsymbol{x}}_{\mathcal{S}}) \big]$. Given a perturbed sample $\widetilde{\boldsymbol{x}} \in \mathcal{X}^+$ satisfying $\min_{1 \le i \le M} \phi_i(\widetilde{\boldsymbol{x}}) \ge 0$, the explanation difference $||\phi(\boldsymbol{x}) - \phi(\widetilde{\boldsymbol{x}})||_2$ is bounded by the prediction difference $|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})|$ as*

$$||\phi(\boldsymbol{x}) - \phi(\widetilde{\boldsymbol{x}})||_2 \le (1 + \sqrt{2}\gamma_0)|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})| + \sqrt{M}\gamma_0,$$

where $\gamma_0 = K||\boldsymbol{x}||_2$ and $K \ge 0$ is the Lipschitz constant of function $f(\cdot)$.

Proof. In order to calculate the importance score $\phi_i(\boldsymbol{x}) := \mathbb{E}_{\mathcal{S} \sim \boldsymbol{\mathcal{U}} \setminus \{i\}} \big[f(\widetilde{\boldsymbol{x}}_{\mathcal{S} \cup \{i\}}) - f(\widetilde{\boldsymbol{x}}_{\mathcal{S}}) \big]$ of the feature subset $\mathcal{S}$, we recast the formulation under the expression of $\mathbf{S} = \mathbf{1}_{\mathcal{S}} \in \{0,1\}^M$, which is given as follows:

$$\phi_i(\boldsymbol{x}) = \mathbb{E}_{\mathbf{S} \in \{0,1\}^{M-1}} \big[f(\mathbf{S} \cup [1]_i \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r) - f(\mathbf{S} \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S}) \odot \boldsymbol{x}_r) \big]$$

Following Equation 9, we can now discuss explanation difference by each feature. Without lost of generality, we first discuss the score difference of feature i,

$$|\phi_i(\boldsymbol{x}) - \phi_i(\widetilde{\boldsymbol{x}})| \tag{9}$$

$$= \Big| \mathbb{E}_{\mathbf{S}} \big[f(\mathbf{S} \cup [1]_i \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r) - f(\mathbf{S} \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S}) \odot \boldsymbol{x}_r) - \big(f(\mathbf{S} \cup [1]_i \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r) - f(\mathbf{S} \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S}) \odot \boldsymbol{x}_r) \big) \big] \Big| \tag{10}$$

$$\le \mathbb{E}_{\mathbf{S}} \Big[\Big| f(\mathbf{S} \cup [1]_i \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r) - f(\mathbf{S} \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S}) \odot \boldsymbol{x}_r) - \big(f(\mathbf{S} \cup [1]_i \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r) - f(\mathbf{S} \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S}) \odot \boldsymbol{x}_r) \big) \Big| \Big] \tag{11}$$

$$= \frac{1}{2^{M-1}} \Big[\sum_{\mathbf{S}} |f(\mathbf{S} \cup [1]_i \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r) - f(\mathbf{S} \cup [1]_i \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r)| + \sum_{\mathbf{S}} |f(\mathbf{S} \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S}) \odot \boldsymbol{x}_r) - f(\mathbf{S} \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S}) \odot \boldsymbol{x}_r)| \Big] \tag{12}$$

We then discuss the case from Equation 16. We have

$$\frac{1}{2^{M-1}} \Big[\sum_{\mathbf{S}} |f(\mathbf{S} \cup [1]_i \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r) - f(\mathbf{S} \cup [1]_i \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r)| + \sum_{\mathbf{S}} |f(\mathbf{S} \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S}) \odot \boldsymbol{x}_r) - f(\mathbf{S} \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S}) \odot \boldsymbol{x}_r)| \Big] \tag{13}$$

$$\le \frac{1}{2^{M-1}} \Big[\sum_{\mathbf{S}} \big(|f(\mathbf{S} \cup [1]_i \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r) - f(\mathbf{S} \cup [1]_i \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r)| \big) + \sum_{\mathbf{S}} K||\mathbf{S} \odot \widetilde{\boldsymbol{x}} - \mathbf{S} \odot \boldsymbol{x}||_2 \Big] \tag{14}$$

$$= \frac{1}{2^{M-1}} \Big[\sum_{\mathbf{S}} |f(\mathbf{S} \cup [1]_i \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r) - f(\mathbf{S} \cup [1]_i \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r)| + \sum_{\mathbf{S}} K||\mathbf{S} \odot (\widetilde{\mathbf{S}} - \mathbf{1}) \odot \boldsymbol{x}||_2 \Big] \tag{15}$$

$$\le \underbrace{\frac{1}{2^{M-1}} \sum_{\mathbf{S}} \Big(|f(\mathbf{S} \cup [1]_i \odot \boldsymbol{x} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r) - f(\mathbf{S} \cup [1]_i \odot \widetilde{\boldsymbol{x}} + (\mathbf{1} - \mathbf{S} \cup [1]_i) \odot \boldsymbol{x}_r)| \Big)}_{\delta_i} + K||\boldsymbol{x}||_2 \tag{16}$$

Note that the difference of prediction scores $|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})|$ is equal to the summation of contribution score among all features, where $|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})| = |\sum_{i=1}^{M} \delta_i| = \sum_{i=1}^{M} |\delta_i|$. In this manner, we have the upper bound of the explanation difference $||\phi(\boldsymbol{x}) - \phi(\widetilde{\boldsymbol{x}})||_2$ given by

$$\begin{aligned}
||\phi(\boldsymbol{x}) - \phi(\widetilde{\boldsymbol{x}})||_2 &= \sqrt{\sum_{i=1}^{M} |\phi_i(\boldsymbol{x}) - \phi_i(\widetilde{\boldsymbol{x}})|^2} \\
&\leq \sqrt{\sum_{i=1}^{M} (\delta_i + K||\boldsymbol{x}||_2)^2} \\
&\leq \sqrt{\sum_{i=1}^{M} (\delta_i)^2} + \sqrt{2} \cdot (K||\boldsymbol{x}||_2) \cdot \sqrt{\sum_{i=1}^{M} \delta_i} + \sqrt{\sum_{i=1}^{M} (K||\boldsymbol{x}||_2)^2} \\
&= \left\{||[\delta_1, \delta_2, \cdots, \delta_M]||_2 + \sqrt{2}\gamma_0 |f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})| + \sqrt{M}\gamma_0\right\} \\
&\leq \left\{||[\delta_1, \delta_2, \cdots, \delta_M]||_1\right\} + \sqrt{2}\gamma_0 \cdot \left\{|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})|\right\} + \sqrt{M}\gamma_0 \\
&= (1 + \sqrt{2}\gamma_0)|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}})| + \sqrt{M}\gamma_0
\end{aligned}$$

where $\gamma_0 = K||x||_2$ □

Theorem 2 (Explanation Error Bound). *Given a training set $\mathcal{D}$, a testing set $\mathcal{C}$, and a well-trained explainer $\hat{\boldsymbol{\phi}} = \boldsymbol{\eta} \circ g$, where g denotes the contrastive explanation encoder to generate d-dimensional explanation embeddings and $\boldsymbol{\eta}$ represents explanation head. Assume $\hat{\boldsymbol{\phi}}(\cdot)$ is a K_h-Lipschitz continuity. For all $\boldsymbol{x}_j \in \mathcal{D}$ in training set, if there exist $\mathcal{E} > 0$, such that the training error satisfies $||\boldsymbol{\phi}(\boldsymbol{x}_j) - \hat{\boldsymbol{\phi}}(\boldsymbol{x}_j)||_2 \leq \mathcal{E}$. Then, for any testing datapoint $\boldsymbol{x}_k \in \mathcal{C}$, the testing explanation error Err($\hat{\boldsymbol{\phi}}$) $= ||\boldsymbol{\phi}(\boldsymbol{x}_k) - \hat{\boldsymbol{\phi}}(\boldsymbol{x}_k)||_2$ can be bounded as:*

$$||\boldsymbol{\phi}(\boldsymbol{x}_k) - \hat{\boldsymbol{\phi}}(\boldsymbol{x}_k)||_2 \leq (1 + \sqrt{2}\gamma_0)|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}}_k^+)| + \sqrt{M}\gamma_0 + \mathcal{E} + K_h||\boldsymbol{h}_{\widetilde{\boldsymbol{x}}_k^+} - \boldsymbol{h}_{x_k}||_2$$

where $\widetilde{\boldsymbol{x}}_k^+$ is the compact positive sample, $\boldsymbol{h}_{\boldsymbol{x}_i} = \mathrm{g}(\boldsymbol{x}_i)$, $\gamma_0 = K||\boldsymbol{x}_k||_2$, and $K_h \geq 0$ is the Lipschitz constant of prediction model $f(\cdot)$.

Proof. Without loss of generality, we consider $\boldsymbol{\ell}_2$ norm to evaluate the distance between predicted explanation scores and ground truth explanation scores. For any testing datapoint $x_k \in \mathcal{C}$, we have

$$\begin{aligned}
||\boldsymbol{\phi}_{x_k} - \hat{\boldsymbol{\phi}}_{x_k}||_2 &= ||\boldsymbol{\phi}_{x_k} - \boldsymbol{\phi}_{\widetilde{\boldsymbol{x}}_k} + \boldsymbol{\phi}_{\widetilde{\boldsymbol{x}}_k} - \hat{\boldsymbol{\phi}}_{\widetilde{\boldsymbol{x}}_k} + \hat{\boldsymbol{\phi}}_{\widetilde{\boldsymbol{x}}_k} - \hat{\boldsymbol{\phi}}_{x_k}||_2 \\
&\leq ||\boldsymbol{\phi}_{x_k} - \boldsymbol{\phi}_{\widetilde{\boldsymbol{x}}_k}||_2 + ||\boldsymbol{\phi}_{\widetilde{\boldsymbol{x}}_k} - \hat{\boldsymbol{\phi}}_{\widetilde{\boldsymbol{x}}_k}||_2 + ||\hat{\boldsymbol{\phi}}_{\widetilde{\boldsymbol{x}}_k} - \hat{\boldsymbol{\phi}}_{x_k}||_2 \\
&\leq (1 + \sqrt{2}\gamma_0)|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}}_k^+)| + \sqrt{M}\gamma_0 + \mathcal{E} + K_{\mathrm{h}}||\mathrm{g}(\widetilde{\boldsymbol{x}}_k) - \mathrm{g}(x_k)||_2 \\
&= (1 + \sqrt{2}\gamma_0)|f(\boldsymbol{x}) - f(\widetilde{\boldsymbol{x}}_k^+)| + \sqrt{M}\gamma_0 + \mathcal{E} + K_{\mathrm{h}}||\boldsymbol{h}_{\widetilde{\boldsymbol{x}}_k} - \boldsymbol{h}_{x_k}||_2
\end{aligned}$$

□

B Datasets and Implementation Details about Experiments

Our experiments are conducted with the following details on two tabular datasets and one image dataset. The details of the datasets are provided as follows. **Census Income:** A collection of human social information with 26048 samples for training and validating; and 6513 samples for testing. Each sample has five continuous features and eight one-hot encoded categorical features. **Bankruptcy:** A dataset contains 5455 samples of various companies training and validating and 1364 instances for testing. Each sample has 96 features characterizing each company and whether it went bankrupt or not. **CIFAR-10:** An image dataset with 60000 images in 10 different classes, where each image has 32×32 pixels. We follow the original dataset division for training, validating, and testing. For the tabular datasets, we consider two common explanation tasks: feature attribution estimation and feature importance ranking. For the image datasets, the explanation results are presented by using heatmap for the case study and evaluated by including and excluding the importance pixels for perturbed image classification.

All baseline algorithms are implemented under the open-source package [2] and the hyper-parameters are all decided with the optimal model convergence. We here provide some detailed information about the baselines on tabular dataset. **Supervised RTX:** A supervised RTX-based MLP model trains with raw features of data instances and ground-truth explanation labels from scratch. **FastSHAP:** A state-of-the-art RTX method which adopts an DNN model to universally learn the approximated Shapley values (Jethani et al., 2021b). **KernelSHAP (KS):** The model proposes the weighted linear regressions to estimate the Shapley additive explanations from the prediction model scores (Kokhlikyan et al., 2020). **Permutation Sampling (PS):** PS estimates the feature attribution based on calculating the sensitivity gap of the scores from prediction model while inputting randomly masking data features (Mitchell et al., 2021). To evaluate proposed explanation representation, we follow the common protocol setting (He et al., 2020). First, we freeze the input features (explanation representation). Second, we train the explanation head $\boldsymbol{\eta}(\cdot \mid \theta_{\boldsymbol{\eta}})$ with limited amount of explanation labels. The input of $\boldsymbol{\eta}(\cdot \mid \theta_{\boldsymbol{\eta}})$ is the pre-trained explanation representation generated from explanation encoder $g(\cdot \mid \theta_g)$, and the output is the corresponding explanation scores among each feature. In this work, we adopt a 3-layer MLP model as $g(\cdot \mid \theta_g)$ in two tabular datasets (Census Income and Bankruptcy) and utilize a ResNet-18 model replacing the last layer with 1-layer MLP in image dataset (CIFAR-10). Considering to the two different common explanation task settings, we exploit a 3-layer MLP model as $\boldsymbol{\eta}(\cdot \mid \theta_{\boldsymbol{\eta}})$ with cross-entropy (CE) loss on feature attribution task and with mean-square-error (MSE) loss on feature importance ranking task.

Tabular Dataset: The experiments details on each tabular dataset follow the pipeline below. We verify the feature ranking task by using CoRTX-CE and the feature attribution task by exploiting CoRTX-MSE. As for the ***Prediction Model***, we exploit different prediction models $f(\cdot)$ among three different datasets to evaluate the model-agnostic property of CoRTX. AutoInt (Song et al., 2019) is adopted for Census Income dataset and MLP model is for Bankruptcy dataset. The two prediction models are trained until the convergence and the implementation are based on the DeepCTR [3] package (Shen, 2017). Due to the task given in two datasets, a binary cross entropy loss is given as the loss function of AutoInt and MLP model for Census Income dataset and Bankruptcy dataset, respectively. All hyper-parameters are decided by grid search throughout the classification results, including model layers, hidden units, and etc. Considering to the ***Explanation Benchmarks***, we use brute-force algorithm to calculate the exact Shapley value for Census Income dataset as the explanation annotations. However, Bankruptcy dataset contains large amount of features (96 features), it is hard to gain the exact Shapley value for Bankruptcy dataset due to the extremely high computational cost. We hereby adopt the proximity of Shapley values as the explanation annotations by using Antithetical Permutation Sampling(APS) (Mitchell et al., 2021; Lomeli et al., 2019) to convergence. This is because APS has shown to converge to exact Shapley values when countering high permutation times on tabular datasets. As for other detailed hyper-parameter setting, the cardinality of synthetic positive data collection is set to 30 and 300 for Census Income and Bankruptcy dataset, respectively. The temperature hyper-parameter τ is given as 0.02 for both tabular datasets. While conducting the feature attribution task, the explanation heads $\boldsymbol{\eta}(\cdot \mid \theta_{\boldsymbol{\eta}})$ are set under the Adam optimizer with weight decay rate from 10^{-3} to 10^{-6}. Every training processes of CoRTX stop until the convergence.

Image Dataset: Our experiments on the image dataset are conducted under CoRTX-MSE. The experiments follow the pipeline as follows. ***Prediction Model***: We utilize ResNet-18 as the prediction models $f(\cdot)$ and train it from scratch until the model convergence. ***Explanation Benchmarks***: We calculate the approximation to Shapley values as the training annotations by adopting KernelSHAP until the convergence (Covert & Lee, 2021; Jethani et al., 2021b). The estimated values from KernelSHAP can infinitely approach to exact Shapley values when encountering the optimal convergence (Lundberg & Lee, 2017). We observe that the important regions on images are typically consecutive. FastSAHP is better to meet the property since it adopts pre-trained weight of ResNet for training a CNN-based explainer, which provides average pooling for smoothing effects. Thus, for the fair comparison in case studies, the outcomes of CoRTX are processed through the moving average for smoothing visualization. The adjustment maintains a similar trend of performance in visualization showcase from original of CoRTX. In other words, the adjusted explanation is not efficacy once the

[2] https://captum.ai

[3] https://github.com/shenweichen/deepctr

original explanation are definitely inaccurate. For quantitative experiments on exclusion and inclusion tasks, we exploit the original outputs from CoRTX and evaluate the performance on Top-1 Accuracy and ΔLog-odds. Following the same experiment setting from (Jethani et al., 2021b;a; Frye et al., 2020), we adopt the supervised surrogate model while performing the explanation tasks on CoRTX and FastSHAP. For more hyper-parameter settings, the temperature hyper-parameter τ is given as 0.02 and an Adam optimizer is given to train with both explanation encoder $g(\cdot \mid \theta_g)$ and explanation head $\boldsymbol{\eta}(\cdot \mid \theta_{\boldsymbol{\eta}})$. Every single training processes of CoRTX are guarantee to stop until converge.

Computation Infrastructure All experiments are conducted under the following physical computing infrastructure. The details of memory footprint and throughput are given in Table 2.

Device Attribute	Value
Computing infrastructure	GPU
GPU model	Nvidia-A40
GPU number	1
GPU Memory	46068 MB

Table 2: Computing infrastructure for the experiments.

C Related Work

Local Methods. The local methods separately generate individual explanations for each data sample. Existing works of local methods can be categorized into three groups. The first group of methods adopts linear regressions to fit the local explanation, such as LIME (Ribeiro et al., 2016) and KernelSHAP (Lundberg & Lee, 2017). Another group of methods adopts the preceding difference of the value function for the explanation, such as RISE (Petsiuk et al., 2018), Permutation Sampling (Mitchell et al., 2021) and SHEAR (Wang et al., 2022). The last group estimates the gradient towards the input data for the explanation, such as the GradCAM (Selvaraju et al., 2017), Integrated Gradient (Sundararajan et al., 2017) and SmoothGrad (Smilkov et al., 2017). Even though the local methods can provide faithful explanation for DNN models, these group of methods suffer from high computational complexity since each data sample requires one local explainer to yield the explanation.

Real-time Explainer (RTX). We here review the existing RTX framework based on DNN approaches. Unlike the local methods, RTX framework maintains an unified explainer to generate the explanation among each data sample. The explanation can be generated via single feed-forward process, which is much faster than the local methods. One learning strategy of RTX framework formulates the explainer learning by given strong assumption on prior feature distribution (Chen et al., 2018; Dabkowski & Gal, 2017; Kanehira & Harada, 2019). One of work (Chen et al., 2018) utilizes the instance-wise feature selection by maintaining a feature masking generator via maximizing the mutual information between selected features and corresponding labels. A well-trained feature masking generator is able to provide real-time explanation under single feed-forward process. Another framework of RTX is to adopt the exact or approximated Shapley values as the ground-truth annotations to learn the explainers (Wang et al., 2021; Jethani et al., 2021b; Covert et al., 2022). However, the exploitation of exact Shapley values suffers from extremely high computational complexity. To address this problem, FastSHAP (Jethani et al., 2021b) proposes a Monte-Carlo-based method to learn the explainer under RTX framework. Specifically, FastSHAP generates the approximated Shapley values by randomly samples batches of feature masks during the training process. Meanwhile, it updates the explainer to minimize the mean-square error between the overall contribution scores of masked features and outputs from DNN explainer. FastSHAP enforces the explanation performance without utilizing ground-truth Shapley value which can be demonstrated by the experiment results.

D Additional Results on Image Dataset

We demonstrate more explanation results on CIFAR-10 generated by CoRTX compared to other baselines. The results show that CoRTX can identify more accurate regions as the explanation results toward the significant object related to the image classes.

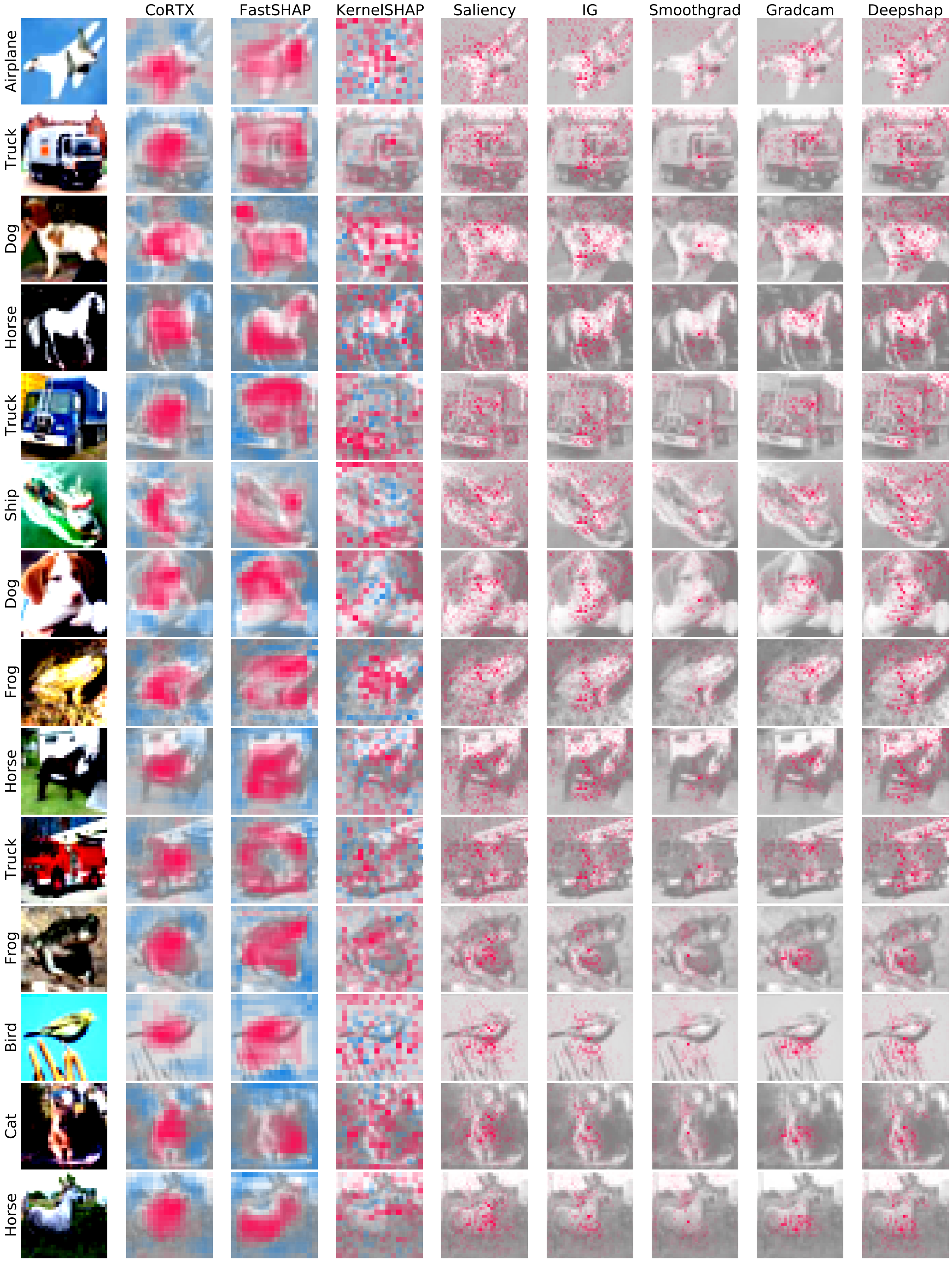


Figure 8: Explanations generated on CIFAR-10 Dataset.