Generative Counterfactuals for Neural Networks via Attribute-Informed Perturbation

Fan Yang, Ninghao Liu, Mengnan Du, Xia Hu
Department of Computer Science and Engineering, Texas A&M University
{nacoyang, nhliu43, dumengnan, xiahu}@tamu.edu

ABSTRACT

With the wide use of deep neural networks (DNN), model interpretability has become a critical concern, since explainable decisions are preferred in high-stake scenarios. Current interpretation techniques mainly focus on the feature attribution perspective, which are limited in indicating *why* and *how* particular explanations are related to the prediction. To this end, an intriguing class of explanations, named *counterfactuals*, has been developed to further explore the "what-if" circumstances for interpretation, and enables the reasoning capability on black-box models. However, generating counterfactuals for raw data instances (i.e., text and image) is still in the early stage due to its challenges on high data dimensionality and unsemantic raw features. In this paper, we design a framework to generate counterfactuals specifically for raw data instances with the proposed **A**ttribute-**I**nformed **P**erturbation (**AIP**). By utilizing generative models conditioned with different attributes, counterfactuals with desired labels can be obtained effectively and efficiently. Instead of directly modifying instances in the data space, we iteratively optimize the constructed attribute-informed latent space, where features are more robust and semantic. Experimental results on real-world texts and images demonstrate the effectiveness, sample quality as well as efficiency of our designed framework, and show the superiority over other alternatives. Besides, we also introduce some practical applications based on our framework, indicating its potential beyond the model interpretability aspect.

1. INTRODUCTION

The past decade has witnessed the success of deep neural networks (DNN) in many application domains [31]. Despite the superior performance, DNN models have been increasingly criticized due to its black-box nature [6]. Interpretable machine learning techniques [7] are thus becoming significantly vital, especially in those high-stake scenarios, such as medical diagnosis. To effectively interpret black-box DNNs, most approaches investigate the feature attributions between input instances and output predictions through correlation analysis, so that humans can have a sense of which part of the instance contributes most to the model decision. A typical example is the heatmaps employed for image classification [36], where saliency scores are capable of indicating the feature importance for one prediction label.

However, existing correlation-based explanations are neither discriminative nor counterfactual [29], since they are not able to help understand why and how particular explanations are relevant to model decisions. Thus, to further explore the decision boundaries of black-box DNN, *counterfactuals* have gradually come to the attention of researchers, as an emerging technique for model interpretability. Counterfactuals are essentially some synthetic samples within data distribution, which can flip the prediction. With counterfactuals, humans can understand how input changes affect the model and conduct reasoning under "what-if" circumstances. Take a loan applicant who got rejection for instance. Correlation-based explanations may simply indicate those most contributed features (e.g., income and credit) for rejection, while counterfactuals are capable of showing how the application could be accepted with certain changes (e.g., increase the monthly income from $5,000$ to $7,000$).

Recent work have made some initial attempts on conducting counterfactual analysis. The first line of research [19; 3] employed the prototype and criticism samples in the training set as the raw ingredients for counterfactual analysis, though those selected samples are not counterfactuals in nature. Some other work [12; 1] utilized feature replacement techniques to create hypothetical instances as counterfactuals, where a query instance and a distractor instance are typically needed for counterfactual generation. Besides, contrastive intervention [5; 43] on the query instance is another way to generate counterfactuals. By reasonably perturbing input features, counterfactuals can be obtained in the form of modified data samples.

Despite the existing efforts, generating valid counterfactuals for raw data instances is still challenging due to the following reasons. First, effective counterfactuals for certain label are not guaranteed to be existed in training set, so the selected prototypes and criticisms are not always sufficient for counterfactual analysis. The related sample selection algorithms are highly possible to select some "unexpected" instances due to data constraints [19], which would largely limit the reasoning on model behaviors. Second, efficient feature replacement for raw data instances could be very hard and time-consuming [12]. Also, relevant distractor instances for replacement may not be available in particular scenarios considering privacy and security issues, such as loan applications. Third, modifying query samples with intervention can simply work on a limited types of data, such as tabular data [43] and naive image data [5]. For general raw data like real-world texts or images, intervention operation in data space can be intractable, which makes it difficult to be used in practice.

To handle the aforementioned challenges, the high-dimension data space and unsemantic raw features are the two obstacles ahead. In this paper, we design a framework to generate counterfactuals specifically for raw data instances with the proposed **A**ttribute-**I**nformed **P**erturbation (**AIP**) method. By utilizing the power of generative models, we can obtain useful hypothetical instances within the data distribution for counterfactual analysis. Essentially, our proposed AIP can guide a well-trained generative model to generate valid counterfactuals by updating representations in the attribute-informed latent space, which is a concatenated coding space for both raw features and semantic attributes. Due to the different data formats, raw features and attributes are typically encoded in different ways to construct such attribute-informed space. Compared with the input space, attribute-informed latent space has two merits for counterfactual generation: (1) raw features are encoded as low-dimension ones which are more robust and efficient for generation; (2) data attributes are modeled as joint latent features which are more semantic for conditional generation. As for the construction of attribute-informed latent space, we employ two losses to conduct the training of generative models, where the reconstruction loss is used to guarantee the quality of raw feature embedding and the discrimination loss is used to ensure the correct attribute embedding. Through the adaptive gradient-based optimization, AIP can iteratively derive valid counterfactuals which are able to flip the prediction. The main contributions of this paper are summarized as follows:

- We design a general framework to derive counterfactuals for raw data instances by employing generative models;

- We develop AIP to iteratively update the parameters of generative models in an attribute-informed latent space;

- We evaluate the designed framework with AIP on several real-world datasets including raw texts and images, and demonstrate the superiority over other alternatives.

2. PRELIMINARIES

In this section, we briefly introduce related contexts to our problem, as well as some basics of the employed techniques.

2.1 Counterfactual Explanation

Counterfactual explanation is a natural extension under the framework of example-based reasoning [34], where particular data samples are provided to promote the understandings on model behaviors. Nevertheless, counterfactuals are not common examples, since they are typically generated under the "what-if" circumstances which may not necessarily exist. According to the theory proposed by J. Pearl [30], three distinct levels of cognitive ability are needed to fully master the behaviors of a particular model, i.e., *seeing*, *doing* and *imagining* from the easiest to the hardest. In fact, counterfactual explanation is just raised to meet the imagining-level cognition for model interpretation.

Within the contexts of this paper, we only discuss counterfactuals under the assumption of "closest possible world" [42], where desired outcomes can be obtained through the smallest changes to the world. To be specific and simple without loss of generality, consider a binary classification model $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \{0, 1\}$, where 0 and 1 respectively indicate the undesired and desired output. The model input $\mathbf{x} \in \mathbb{R}^d$ is further assumed to be sampled from data distribution $\mathcal{P}(\mathbf{x})$. Then, given a query instance $\mathbf{x}_0$ with the undesired model output (i.e., $f_{\boldsymbol{\theta}}(\mathbf{x}_0) = 0$), the corresponding counterfactual $\mathbf{x}^*$ can be mathematically represented as:

$$\mathbf{x}^* = \underset{\mathbf{x} | \mathcal{P}(\mathbf{x}) > \eta}{\arg \min} \, l(\mathbf{x}, \mathbf{x}_0) \quad \text{s.t.} \ f_{\boldsymbol{\theta}}(\mathbf{x}^*) = 1, \qquad (1)$$

where $l : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ is a distance measure in the input space, and $\eta > 0$ denotes the threshold which quantifies how likely the sample $\mathbf{x}$ is under the distribution $\mathcal{P}(\mathbf{x})$. The obtained counterfactual $\mathbf{x}^*$ is regarded to be valid if it can flip the target classifier $f_{\boldsymbol{\theta}}$ to the desired prediction.

Although finding counterfactuals is somewhat similar to generating adversarial examples (in terms that both tasks aim to flip the model decision by minimally perturbing the input instance), they are essentially different in nature. Following the previous settings, the adversarial sample $\mathbf{x}^{adv}$ for model $f_{\boldsymbol{\theta}}$, with query instance $\mathbf{x}_0$, can be generally indicated by:

$$\mathbf{x}^{adv} = \underset{\mathbf{x} = \mathbf{x}_0 + \boldsymbol{\delta}}{\arg \min} \, \|\boldsymbol{\delta}\|_p \quad \text{s.t.} \ f_{\boldsymbol{\theta}}(\mathbf{x}^{adv}) \neq f_{\boldsymbol{\theta}}(\mathbf{x}_0), \qquad (2)$$

where $\boldsymbol{\delta}$ denotes the adversarial perturbation on the query, $\| \cdot \|$ represents the norm operation and $p \in \{\infty, 1, 2, \cdots\}$. Comparing with Eq. 1, we note that counterfactual example has two significant differences from adversarial sample. First, counterfactual generation process is subject to the original data distribution, while adversarial samples are not constrained by the distribution. This difference brings about the fact that counterfactuals are all in-distribution samples, but adversarial examples are mostly out-of-distribution (OOD) samples. Second, counterfactual changes on the query need to be human-perceptible, while adversarial perturbations are usually inconspicuous [37]. Therefore, the key problem of counterfactual explanation actually lies in how to generate such in-distribution sample, with human-perceptible changes on the query, to flip the model decision as desired.

2.2 Generative Modeling

Generative modeling is a typical task under the paradigm of unsupervised learning. Different from discriminative ones, which involves discriminating input samples across classes, generative modeling aims to summarize the data distribution of input variables and further create new samples that plausibly fit into that distribution [28]. In practice, a well-trained generative model is capable of generating new examples that are not only reasonable, but also indistinguishable from real examples in the problem domain. Conventional examples of generative modeling include Latent Dirichlet Allocation (LDA) and Gaussian Mixture Model (GMM).

As emerging families of generative modeling, Generative Adversarial Network (GAN) [10] and Variational Auto-Encoder (VAE) [21] have been attracting lots of attentions due to their exceptional performance in a myriad of applications, especially for the task on image and text generation [41; 17]. By taking full advantage of their power on raw data with high dimensionality, we are able to better investigate how those data samples were created in the first place, which potentially benefits the generation of certain hypothetical example. To this end, we specifically employ some advanced generative models (i.e., GAN and VAE) to study the counterfactual explanation for black-box DNN on raw data instances, providing effective generative counterfactuals for better model understanding.
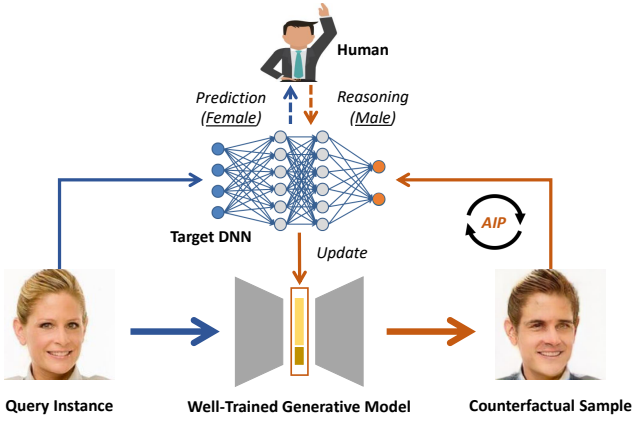
Figure 1: Designed framework for counterfactual generation.

Figure 2: General illustration of the attribute-informed latent space in generative models.

3. COUNTERFACTUAL GENERATION

In this section, we first introduce the designed generative counterfactual framework for raw data instances. Then, we present how to specifically construct the attribute-informed latent space with generative models. Finally, we show the details of our proposed AIP method.

3.1 Generative Counterfactual Framework

We design a framework to create counterfactual samples for raw data instances, as illustrated by Fig. 1. To effectively handle the high dimensionality and unsemantic features, we utilize the generative modeling techniques to aid the counterfactual generation process. Consider a target DNN $F_{\phi} : \mathbb{R}^d \rightarrow \{1, \cdots, C\}$, which is a black-box model for counterfactual analysis, where $\mathbb{R}^d$ is the input data space and $\{1, \cdots, C\}$ denotes the model prediction space with C different outputs. Given a query instance $\mathbf{x}_0$, $F_{\phi}(\mathbf{x}_0) = \mathbf{y}_0$ outputs a one-hot vector. To effectively generate a valid counterfactual sample $\mathbf{x}^* \in \mathbb{R}^d$ that can flip the F_{ϕ} decision to $\mathbf{y}^* \in \{1, \cdots, C\}$ as desired, a generative model is trained to achieve this in the framework. The applied generative modeling plays two important roles in the counterfactual generation process: (1) guarantee that all created instances are in-distribution samples, since it can be regarded as a stochastic procedure that generates samples $\mathbf{x} \in \mathbb{R}^d$ under the particular data distribution $\mathcal{P}(\mathbf{x})$; (2) assume that underlying latent variables can be mapped to the data space under certain circumstances, which ensures the sufficient feasibility for hypothetical examples. Thus, a well-trained generative model is the basis for high-quality counterfactuals within the designed framework.

The employed generative model specifically serves two sub-tasks for counterfactual generation, i.e., data *encoding* and *decoding*. For raw data instances like images, the input space $\mathbb{R}^d$ could be extremely large, which makes it difficult and inefficient to directly create counterfactuals for the query. In our designed framework, data encoding is conducted to map the input data space to a low-dimension attribute-informed latent space, which is formulated as a joint embedding space for both raw features and data attributes. In this way, each data sample $\mathbf{x}$ can be effectively encoded through the function $G_{\psi}^{enc} : \mathbb{R}^d \rightarrow \mathbb{R}^k \oplus \mathbb{R}^t$, where $\mathbb{R}^k$ is the latent space for raw feature embeddings, $\mathbb{R}^t$ indicates the data attribute space, and $\oplus$ represents a con-
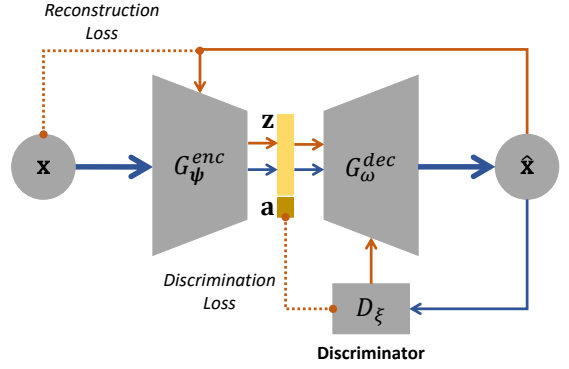
catenation operator. Reversely, the mapping for decoding is from the attribute-informed latent space to the original data space. The decoder function can be similarly indicated by $G_{\omega}^{dec} : \mathbb{R}^k \times \mathbb{R}^t \rightarrow \mathbb{R}^d$. Although G_{ψ}^{enc} and G_{ω}^{dec} typically have two different focuses, they are jointly trained as a whole generative model in an end-to-end manner. The issues about how to derive G_{ψ}^{enc} and G_{ω}^{dec} will be particularly discussed in Sec. 3.2.

To finally obtain the counterfactual $\mathbf{x}^*$ for model F_{ϕ} with query $\mathbf{x}_0$, we further need to modify the attribute-informed latent space of the deployed generative model. Specifically, we use the proposed AIP to update the attribute-informed latent vector of $\mathbf{x}_0$, according to the counterfactual loss calculated. Assuming $G_{\psi}^{enc}(\mathbf{x}_0) = \mathbf{z}_0 \oplus \mathbf{a}_0$ ($\mathbf{z}_0 \in \mathbb{R}^k, \mathbf{a}_0 \in \mathbb{R}^t$), AIP method can jointly update $\mathbf{z}_0$ and $\mathbf{a}_0$, so as to minimize the corresponding loss counter-factually. The overall counterfactual loss consists of two parts, i.e., *prediction loss* and *perturbation loss*. Prediction loss is set to ensure the flip of model decisions, and perturbation loss is involved to guarantee the "closest possible" changes on the query, which are both indispensable for counterfactual generation. For the prediction loss, we simply follow the common cross-entropy term, expressed as $L_d(F_{\phi}(\mathbf{x}), \mathbf{y}^*) = -\mathbf{y}^* \log(F_{\phi}(\mathbf{x})) - (1 - \mathbf{y}^*) \log(1 - F_{\phi}(\mathbf{x}))$. For the perturbation loss, we employ two l_2 norms respectively on $\mathbb{R}^k$ and $\mathbb{R}^t$, indicated by $L_b(\mathbf{z}, \mathbf{a}, \mathbf{z}_0, \mathbf{a}_0) = \|\mathbf{z} - \mathbf{z}_0\|_2 + \|\mathbf{a} - \mathbf{a}_0\|_2$ ($\mathbf{z} \in \mathbb{R}^k, \mathbf{a} \in \mathbb{R}^t$), to restrain the query changes, which can also be regarded as a regularization term. Further, the overall counterfactual loss can thus be represented as follows:

$$L_c(\mathbf{z}, \mathbf{a}, \mathbf{z}_0, \mathbf{a}_0, \mathbf{y}^*) = L_d\left(F_{\phi}(G_{\omega}^{dec}(\mathbf{z}, \mathbf{a})), \mathbf{y}^*\right) + \alpha L_b(\mathbf{z}, \mathbf{a}, \mathbf{z}_0, \mathbf{a}_0), \quad (3)$$

where α is a balance coefficient between the two loss terms. With the proposed AIP method, the designed framework can generate the valid counterfactual example $\mathbf{x}^*$ with the aid of optimized $\mathbf{z}^*, \mathbf{a}^*$ through the decoder function (i.e., $\mathbf{x}^* = G_{\omega}^{dec}(\mathbf{z}^*, \mathbf{a}^*)$). The details of the proposed AIP method will be introduced in Sec. 3.3.

3.2 Attribute-Informed Latent Space

Constructing an appropriate attribute-informed latent space is the key part for generative modeling in our designed framework, which has direct influences on the quality of generated counterfactuals. To achieve this, we need to well train a gen-

erative model, better capturing the raw data features as well as relevant data attributes, where embedded features can bring about more robust bases for counterfactual analysis, and incorporated attributes are able to provide more semantics for conditional generation. Here, the data attributes mainly indicate those extra information from humans along with raw instances, such as annotations or labels, which can usually be represented as one-hot vectors.

In practice, it is common that different generative models are employed for different tasks or data. Since different models typically involve disparate architectures, their training schemes can totally differ from each other. Take the GAN and VAE for example, where GAN is usually trained to obtain an equilibrium between a generator and a discriminator function, while VAE is typically trained to maximize a variational lower bound of the data log-likelihood. Therefore, to better introduce how to specifically construct the attribute-informed latent space with generative models, we present a general illustration shown by Fig. 2, although it may not be fully representative for all kinds of models.

We generally introduce the data modeling process with an encoder-decoder structure, which corresponds to the data encoding and decoding in our designed framework. Essentially, the attribute-informed latent space can be regarded as an extended code space of auto-encoders, where attribute information is properly incorporated into the representation beyond the raw feature inputs. By concatenating attribute vector $\mathbf{a}$ to raw feature embedding $\mathbf{z}$, the decoder function aims to achieve the conditional generation based on $\mathbf{a}$. To ensure the attribute consistency between original sample $\mathbf{x}$ and generated sample $\hat{\mathbf{x}}$, discriminator D_ξ is particularly employed, which is trained separately and used to classify the attributes of $\hat{\mathbf{x}}$. To effectively train such generative model, two basic loss terms are required, which are the discrimination loss and reconstruction loss. The overall training can be indicated by:

$$\min_{\psi, \omega} \mathbb{E}_{\substack{\mathbf{x} \sim \mathcal{P}(\mathbf{x}) \\ \mathbf{a} \sim \mathcal{P}(\mathbf{a})}} \sum_{i=1}^{t} -a_i \log D_\xi^i(\hat{\mathbf{x}}) - (1-a_i) \log \left(1 - D_\xi^i(\hat{\mathbf{x}})\right)$$
$$+ \mathbb{E}_{\mathbf{x} \sim \mathcal{P}(\mathbf{x})} \|\mathbf{x} - \hat{\mathbf{x}}\|_2, \tag{4}$$

where a_i denotes the i-th attribute in $\mathbf{a}$, and D_ξ^i indicates the prediction of D_ξ on the i-th attribute. After sufficient training, G_ψ^{enc} and G_ω^{dec} can be effectively obtained, and the attribute-informed latent space can be constructed with the aid of G_ψ^{enc}. For specific tasks and architectures, the generative modeling process could be further enhanced with other loss terms, such as the adversarial losses employed in [16] for better visual quality.

3.3 Attribute-Informed Perturbation

With the obtained G_ψ^{enc} and G_ω^{dec} for generative modeling, we then introduce the proposed AIP method to finally derive the counterfactual for target DNN F_ϕ with the query $\mathbf{x}_0$. To guarantee the quality of the generated counterfactuals, AIP needs to find the sample that can minimize the counterfactual loss indicated by Eq. 3. Under the "closest possible world" assumption, the corresponding counterfactual sample can be denoted as:

$$\mathbf{x}^* = G_\omega^{dec} \left(\underset{\mathbf{z} \in \mathbb{R}^k, \ \mathbf{a} \in \mathbb{R}^t}{\arg\min} L_c(\mathbf{z}, \mathbf{a}, \mathbf{z}_0, \mathbf{a}_0, \mathbf{y}^*) \right). \tag{5}$$

Algorithm 1: Attribute-Informed Perturbation (AIP)

Input: F_ϕ, G_ψ^{enc}, G_ω^{dec}, $\mathbf{x}_0$, $\mathbf{y}^*$, μ, γ, α, β, $n_{\max}$
Output: Counterfactual sample $\mathbf{x}^*$

1 Initialize μ, γ, α, β ;
2 Initialize $n = 0$, $\mathbf{x} = \mathbf{x}_0$;
3 Construct the latent space with $\mathbf{z} \oplus \mathbf{a} \leftarrow G_\psi^{enc}(\mathbf{x})$;
4 **while** $F_\phi(G_\omega^{dec}(\mathbf{z}, \mathbf{a})) \neq \mathbf{y}^*$ or $n \leq n_{\max}$ **do**
5   Update $\mathbf{z}$ and $\mathbf{a}$ according to Eq. 6 ;
6   Update step sizes with $\mu \leftarrow \beta\mu$ and $\gamma \leftarrow \beta\gamma$;
7      $n \leftarrow n + 1$;
8 Reconstruct the sample with $\mathbf{x}^* \leftarrow G_\omega^{dec}(\mathbf{z}, \mathbf{a})$;
9 **if** $F_\phi(\mathbf{x}^*) == \mathbf{y}^*$ **then**
10   **Return** $\mathbf{x}^*$ as the counterfactual for F_ϕ with $\mathbf{x}_0$;
11 **else**
12   **Return** None – No valid counterfactual exists;

To effectively solve Eq. 5, the proposed AIP method utilizes an iterative gradient-based optimization algorithm with dynamic step sizes (controlled by a decaying factor β), which helps the iteration process converge faster. In each iteration, the updated $\mathbf{z}$ and $\mathbf{a}$ can be derived as follows:

$$\begin{cases} \mathbf{z}^{(n+1)} = \mathbf{z}^{(n)} - \mu^{(n)} \nabla_\mathbf{z} L_c \left(\mathbf{z}^{(n)}, \mathbf{a}^{(n)}, \ \mathbf{z}_0, \ \mathbf{a}_0, \ \mathbf{y}^* \right) \\ \mathbf{a}^{(n+1)} = \mathbf{a}^{(n)} - \gamma^{(n)} \nabla_\mathbf{a} L_c \left(\mathbf{z}^{(n)}, \mathbf{a}^{(n)}, \ \mathbf{z}_0, \ \mathbf{a}_0, \ \mathbf{y}^* \right) \end{cases}, \tag{6}$$

where n indicates the iteration index, μ and γ respectively denotes the step sizes of updates on $\mathbf{z}$ and $\mathbf{a}$. Specifically, the proposed AIP method can be summarized in Algorithm 1. It is important to note that AIP only works on the optimization of $\mathbf{z}, \mathbf{a}$, and does not involve the parameter update on F_ϕ, G_ψ^{enc}, G_ω^{dec}. Thus, the proposed AIP method should be less time-consuming and easily deployed for counterfactual generation task, compared with those generative frameworks which need extra model training [35; 38].

4. EXPERIMENTS

In this section, we evaluate the designed counterfactual generation framework with the proposed AIP on several real-world datasets, both quantitatively and qualitatively. Overall, we conduct two sets of experiments respectively on *text* and *image* counterfactual generation, by utilizing different data modeling techniques. With conducted experiments, we aim to answer the following four research questions:

- How *effective* is the designed framework in generating counterfactuals with AIP?

- How is the *quality* of created counterfactuals from our designed framework aided by AIP?

- How *efficient* is the counterfactual generation under the designed framework with AIP?

- Can we *benefit* other practical tasks with the counterfactuals generated from our design framework with AIP?

4.1 Experimental Settings

4.1.1 Real-World Datasets.

Throughout the whole experiments, we employ three real-world datasets to evaluate the performance of the designed

Table 1: Dataset statistics in experiments.

Datasets	#Instance	#Attribute	Type	Domain
Yelp	455,000	1	Texts	Sentiment
Amazon	558,000	1	Texts	Sentiment
CelebA	202,599	13	Images	Human Face

framework with AIP, including both raw texts and images. The relevant data attributes depend on the particular tasks, which are collected either from labels or annotations. The statistics of the involved datasets are shown in Table 1.

- **Yelp User Review Dataset**[1] [2]: This dataset consists of user reviews from the Yelp associated with relevant rating scores. We involve a tailored and modified version of this data for our experiments on text counterfactuals. Specifically, we consider the reviews with ratings higher than three as *positive* samples and regard the others as *negative* ones, and we further use these sentiment labels as the relevant attribute for data modeling. The vocabulary of our involved Yelp data contains more than 9,000 words, and the average review length is around 9 words.

- **Amazon Product Review Dataset**[2] [15]: This dataset is also involved as a raw textual dataset for our experiments. Similar to the Yelp data, we modify the original rating information of reviews into the sentiment categories (i.e., *positive* and *negative*), and further model these labels as an sentiment attribute of the raw textual reviews. Amazon dataset has more than 50,000 words in vocabulary, and the average length is around 15.

- **CelebFaces Attributes (CelebA) Dataset**[3] [24]: This is a large-scale face attributes dataset, containing tons of raw face images with human annotations. We employ this dataset for our experiments on image counterfactual generation, and select 13 representative face attributes (out of 40) for data modeling along with raw face images. The involved thirteen attributes include: *Male*, *Young*, *Blond_Hair*, *Pale_Skin*, *Bangs*, *Mustache* and etc.

4.1.2 Target Model for Interpretation.

Since we mainly discuss the counterfactuals of raw data instances, DNN is a better choice as our target model. For the target DNN F_ϕ in our experiments, we employ some regular structures for the corresponding tasks. Particularly, for the text sentiment classification, we use a common convolutional architecture in [20] to pre-train a DNN classifier for further counterfactual analysis. For the image attribute classification task, similarly, we utilize a simple convolutional network [13] to prepare a target classifier, where the model is trained with one of those attributes as the label. During the evaluations on counterfactual generation, target DNNs are fixed without further training.

4.1.3 Employed Generative Modeling Techniques.

In our experiments, different data modeling techniques are employed for different types of data. In particular, we use

different generative models to construct the corresponding attribute-informed latent space, regarding to text and image. For textual reviews (Yelp, Amazon), we utilize the modeling techniques introduced in [17], and build a transformer-based VAE to effectively formulate the relevant attribute-informed latent space. For face images (CelebA), we mainly follow the modeling method of AttGAN [16], where more complicated training schemes are employed, compared with the general one shown in Sec. 3.2, for better visual quality of the generated images. Both of the employed generative models should be well-trained on the corresponding datasets before the counterfactual generation process, so as to guarantee the high quality of generated counterfactuals.

4.1.4 Alternative Methods and Baselines.

To effectively evaluate the performance of the designed framework with AIP, we incorporate following alternative methods and baselines for comparison.

- **TextBugger** [22]: This is a general method for adversarial text generation, which is built based on the word attribution and bug selection. The created text samples can effectively flip the prediction of the target classifier. We employ this method as a baseline to specifically compare with our generated text counterfactuals.

- **DeepWordBug** [9]: This is another method focusing on the adversarial text generation, where a token scoring strategy is utilized to guide the character-level adversarial perturbation. This method is employed as a baseline for text counterfactuals as well.

- **FGSM** [11]: Fast gradient sign method is a common way to generate image adversarial samples, by using the gradients of the loss with respect to the input. The sample created by this method can effectively maximize the loss, so as to flip the original prediction. We employ this method as a baseline specifically for our generated image counterfactuals.

- **Counter_Vis** [12]: This is a recent method in generating image counterfactuals, where particular image regions are replaced to flip the model decision. We employ this method as an alternative method for image counterfactual generation.

- **CADEX** [27]: This is a state-of-the-art method for counterfactual generation, where the gradient-based method is directly applied to modify the input space of query. This method is originally proposed for tabular data, and we modify it simply as an alternative for image counterfactuals, due to the particularity of texts.

- **xGEMs** [18]: This is a state-of-the-art method for generating counterfactuals, which also employs the generative modeling technique for sample generation. This method only involves the latent space modeling and cannot achieve the conditional generation with semantic attributes. We employ this method as an important alternative for both text and image counterfactuals.

- **AIP_R**: This is the random version of our proposed AIP method, which updates all parameters in a random way.
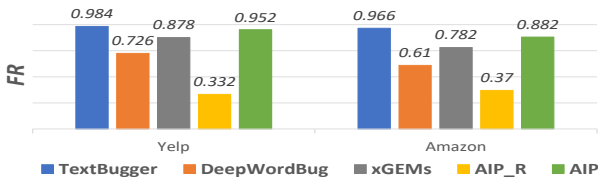
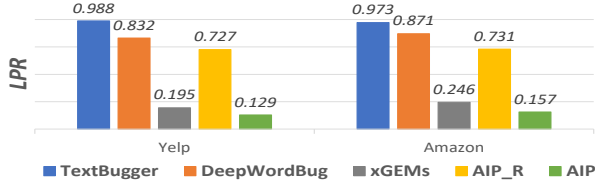Figure 3: Effectiveness evaluation for text counterfactuals.

Figure 4: Quality evaluation for text counterfactuals.

4.2 Text Counterfactual Evaluations

In this part, we evaluate the designed framework with AIP in generating text counterfactuals, regrading to a convolutional neural net (CNN) built for sentiment classification. The involved raw texts for target DNN come from the user/product reviews in Yelp and Amazon datasets, where 90% are used for training, 5% for development and 5% for testing.

4.2.1 Effectiveness Evaluation.

In order to evaluate the effectiveness for text counterfactuals, we employ the metric *Flipping Ratio* (FR) to measure the relevant performance, which reflects how likely the generated text samples would flip the model decision to $\mathbf{y}^*$. Specifically, FR can be calculated as follows:

$$\text{FR} = |\mathcal{X}_f| \big/ |\mathcal{X}_q| \quad (\mathbf{x}_0 \in \mathcal{X}_f \text{ if } F_\phi(\mathbf{x}_0) = \mathbf{y}^*), \quad (7)$$

where $\mathcal{X}_f$ indicates the set of query samples with which new flipping instances can be generated by particular methods, and $\mathcal{X}_q$ denotes the set of all testing queries. In our experiments, there are 500 testing queries in total (i.e., $|\mathcal{X}_q| = 500$), which are randomly selected from the test set. Fig. 3 illustrates our experimental results on both Yelp and Amazon datasets. According to the numerical results, we note that our designed framework with AIP can work well on both datasets, and has competitive performance among all other alternatives as well as baselines, although TextBugger achieves the highest FR score with better robustness (i.e., the performance variance across different datasets). Besides, we also observe that AIP_R does not effectively work for generating flipping samples, which indicates that random optimization in attribute-informed latent space cannot help for counterfactual sample generation.

4.2.2 Quality Evaluation.

As for the quality assessment of counterfactual samples, we employ the *Latent Perturbation Ratio* (LPR) metric to measure the latent closeness between the generated sample $\mathbf{x}^*$ and original query instance $\mathbf{x}_0$. Since high-quality counterfactual samples typically need to ensure sparse changes in the robust feature space, thus the smaller the LPR is, the better the counterfactual we have. To be specific, the LPR
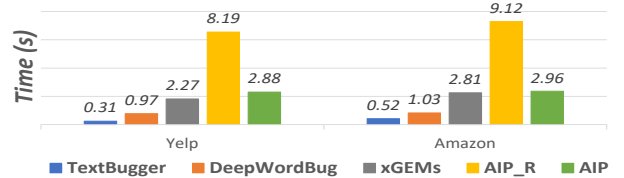

Figure 5: Efficiency evaluation for text counterfactuals.

Table 2: Case studies on generated text samples.

Counterfactual on *Negative* sentiment (Yelp)
Query: this is the worst walmart neighborhood market out of any of them
TextBugger: this is the worst walmart neighborho0d market out of a ny of them
DeepWordBug: this id the wosrt walmart neighobrhood market out of any of htem
xGEMs: that is good walmart market out of any neighborhood
AIP: this is the best walmart neighborhood market for all of them
Counterfactual on *Positive* sentiment (Amazon)
Query: this item works just as i thought it would
TextBugger: this item w0rks just as i tho ught it wou1d
DeepWordBug: this item wroks just ae i thought it wolud
xGEMs: this item works out poorly just as i thought disappointed
AIP: this item works bad just as i thought it would not play

can be calculated by:

$$\text{LPR} = \left\| \mathbf{z}^* - \mathbf{z}_0 \right\|_0 \Big/ k, \quad (8)$$

where $\| \cdot \|_0$ indicates the l_0 norm operation, $\mathbf{z}^*$ and $\mathbf{z}_0$ are the raw feature embeddings respectively for $\mathbf{x}^*$ and $\mathbf{x}_0$. To make a fair comparison, we use the same encoder function G_ψ^{enc} for all generated samples to obtain the corresponding latent representation vectors. In this set of experiments, the latent dimension is 256 (i.e., $k = 256$), and the final LPR value for a particular method is recorded with the average over 500 testing queries. The relevant numerical results are presented in Fig. 4. From the experiments, it is noted that xGEMs and the proposed AIP method significantly outperform other baselines, indicating that the corresponding generated samples actually maintain more robust features regarding to the query. Furthermore, the proposed AIP is noted to be slightly better than xGEMs, which may partially result from the conditional generation brought by attribute vector $\mathbf{a}$. This set of results also validate a fact that adversarial samples typically utilize some artifacts to flip the model decisions, instead of using some robust features.

4.2.3 Efficiency Evaluation.

To compare the efficiency, we record the time consumption for each method over 500 testing queries in the generation phase on the same machine. Specifically, we calculate the average time cost for one query, and further employ this as the metric to access the efficiency for particular methods. Fig. 5 shows the relevant experimental results. Based on the statistics, it is observed that adversarial related methods (i.e., TextBugger and DeepWordBug) consume less time per query on average, compared with the counterfactual generation methods, which is mainly due to the fact that adversarial methods do not need to conduct encoding computations before sample generation. As for our proposed AIP method, the time efficiency is roughly the same as the alternative xGEMs, but it is significantly better than its random version AIP_R which needs more iterations to converge.

4.2.4 Qualitative Case Studies.

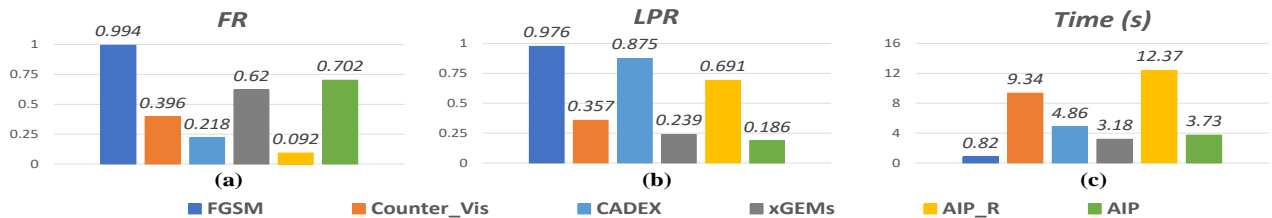Here, we present several representative case studies from

Figure 6: Evaluations for image counterfactual generation.

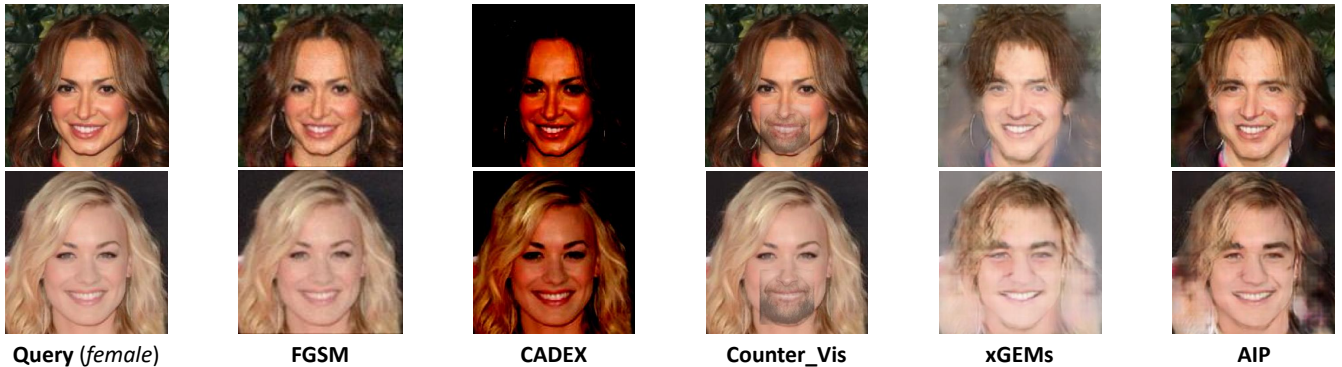| **Query** (*female*) | **FGSM** | **CADEX** | **Counter_Vis** | **xGEMs** | **AIP** |

Figure 7: Qualitative case studies on generated image samples.

different methods, shown in Tab. 2, aiming to provide a qualitative comparison for generated text samples. Based on the Tab. 2, we can see that adversarial texts typically provide limited insights for humans on counterfactual analysis, since they mainly make use of the model artifacts to flip the prediction. Nevertheless, with the samples generated by xGEMs and AIP, we can easily observe some sentiment variation regarding to the query instance, which sheds light on model behaviors and facilitates further human reasoning on black-box models. Besides, compared with xGEMs, the proposed AIP method usually can generate more sensible counterfactuals with the aid of attribute conditions.

4.3 Image Counterfactual Evaluations

In this part, we specifically evaluate the designed framework with AIP on image counterfactual generation. Instead of simply considering one attribute for conditional generation in texts, we take multiple attributes into account for image counterfactuals. In this set of experiments, our target DNN follows the common CNN architecture and is trained as a gender classifier, which can classify an input image as *Male* or *Female*. All involved raw images for target DNN come from the CelebA dataset, and we use 90% data for training, 5% for development, 5% for testing. The relevant quantitative results are all illustrated by Fig. 6.

4.3.1 Effectiveness Evaluation.

For the effectiveness assessment, we still use the FR metric indicated by Eq. 7. In the experiments, we set $|\mathcal{X}_q| = 500$, and aim to test how many of them can be effectively flipped with particular methods. Fig. 6(a) illustrates the relevant numerical results, where adversarial method FGSM performs the best on FR and can flip nearly every testing query. We note that the proposed AIP method ranks the second, and outperforms other counterfactual genera-

tion methods. Besides, it is also observed that CADEX and AIP_R performs relatively bad for the image counterfactual task within certain iterations, even though CADEX is proved to work well for tabular instances [27].

4.3.2 Quality Evaluation.

Similar to text counterfactuals, we employ the LPR metric, shown as Eq. 8, to measure the quality of the generated image counterfactuals. In experiments, the latent dimension k constructed by G_{ψ}^{enc} is $1,024$ (i.e., $k = 1024$), and the LPR for particular method is recorded by calculating the average over 500 testing queries. Relevant experimental results are shown by Fig. 6(b). Based on the LPR comparison, we note that the samples generated by FGSM and CADEX change a lot in the latent feature space, because both methods directly rely on the input perturbation for sample generation. As for the proposed AIP, it achieves the lowest LPR among all the alternatives and baselines, and it is significantly better than its random version AIP_R.

4.3.3 Efficiency Evaluation.

We similarly employ the average time consumption per query to evaluate the efficiency aspect for image counterfactual generation. Specifically, the average time is obtained over the 500 testing queries randomly selected from the test set. Fig. 6(c) shows the relevant experimental results. According to the statistics and comparison, we note that FGSM is the most efficient one, and xGEMs consumes the least time on average among all other counterfactual-based methods. As for the proposed AIP, a competitive efficiency performance is observed, and is remarkably superior compared with that of Counter_Vis, CADEX and AIP_R.

4.3.4 Qualitative Case Studies.

To facilitate a qualitative comparison among different meth-

ods, we specifically show some case studies, illustrated by Fig. 7. We select several query instances whose model predictions are female, and then employ different methods to generate the corresponding image samples which flip the model decisions for counterfactual purpose. According to the results, we note that the samples generated by FGSM and CADEX do not have salient visual changes regarding to the query instances, which largely limits the human reasoning on model behaviors. Among other alternative methods, it is observed that the proposed AIP is capable of generating counterfactuals with better visual quality, which present much smoother transitions from female to male.

4.4 Influence of Hyper-parameter α

In this part, we show some additional results on the influence of hyper-parameter α in Eq. 3. Other experimental settings keep unchanged. The relevant results are shown by Fig. 8.
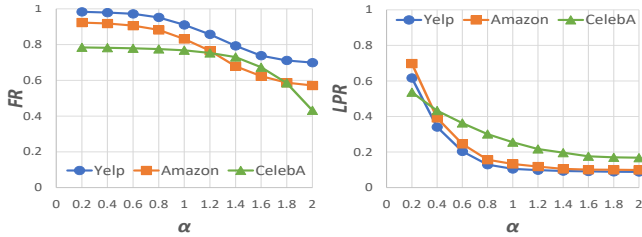

Figure 8: Influence of α on FR and LPR metrics.

Based on the results, we observe that α serves as a knob to control the effectiveness and sample quality of the designed framework. To select an appropriate α, we actually need to strike a balance between FR and LPR, where the larger the α is, the lower the effectiveness is and the higher the sample quality is. Different data types may also have different trade-off curves.

4.5 Applications

In this part, we focus on some practical scenarios which may benefit from the counterfactual samples generated by our designed framework. In particular, we show the applications of the framework respectively on *feature interaction* and *data augmentation*.

4.5.1 Feature Interaction

Understanding the feature interaction could be very important in lots of real-world domains. A typical example is the bias detection task, where humans aim to find out a related set of features which can significantly influence the correctness or fairness of model decision. Utilizing our designed framework for counterfactual analysis can partially help this practical task. By observing the perturbation scale on attribute vector **a** of the generated counterfactual, humans can have a sense on which semantic features contribute significantly to the flipping of model decision. To illustrate the point, we show another case result from the designed framework with AIP in Fig. 9. Here, we train an age classifier on the CelebA dataset as our target DNN, and aim to analyze the feature interaction of a query prediction as "Old". Based on the attribute perturbations of the generated sample, we note that the top semantic attributes are "Male", "Bushy_Eyebrows", "Black_Hair" and "Bangs", besides the target attribute. This result directly demonstrates
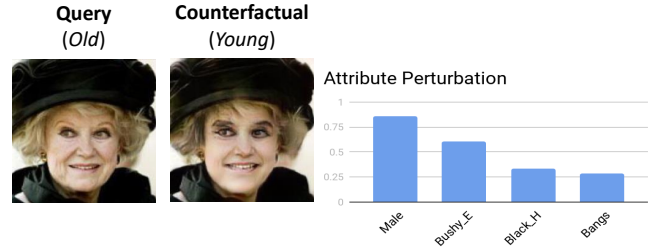

Figure 9: Feature interactions for the decision change.

Table 3: Model performance with data augmentation.

Dataset		CNN [20]	VDCNN [4]
Yelp	Initial	82.33% ($\pm$ 0.61%)	88.79% ($\pm$ 0.53%)
	Augmented	83.16% ($\pm$ 0.57%)	89.95% ($\pm$ 0.46%)
Amazon	Initial	81.96% ($\pm$ 0.52%)	88.55% ($\pm$ 0.63%)
	Augmented	82.41% ($\pm$ 0.49%)	88.76% ($\pm$ 0.55%)
Dataset		CNN [13]	ResNet [14]
CelebA	Initial	87.32% ($\pm$ 0.22%)	90.96% ($\pm$ 0.27%)
	Augmented	88.85% ($\pm$ 0.21%)	91.35% ($\pm$ 0.25%)

the fact that the "Male" attribute has a strong interaction with the predicted attribute for this particular query, and the target DNN exists potential gender bias for its age predictions. Please note that such feature interaction indicates the attribute sensitivity for flipping model decisions, instead of the attribution for predictions, which is different from the existing local interpretation methods (e.g., LIME [32]).

4.5.2 Data Augmentation

Another application of the designed framework is the data augmentation for model training. By taking full advantage of the generated counterfactual samples as new training instances, we aim to obtain better DNN models with higher performance and robustness. Specifically, to test the improvement, we train several DNN models on relatively smaller training sets, which are essentially the subsets of original data. For the sentiment classifiers on Yelp and Amazon, our initial training size is $20,000$, containing $10,000$ positive and $10,000$ negative reviews. The extra counterfactual training size is $2,000$ whose queries are randomly selected from the initial training set. For the binary age classifier on CelebA, we employ a similar setting for training, where each class includes $10,000$ initial samples, and $2,000$ generated counterfactual samples are further incorporated for augmentation. Relevant experimental results are shown in Tab. 3. Based on the statistics, we note that the augmented training with counterfactual samples typically achieves higher classification accuracies with smaller variances, which can also be observed under some advanced DNN structures.

5. RELATED WORK

Generating counterfactuals is just one of interpretation methods for black-box models, which generally belongs to the family of interpretable machine learning. According to the particular problems, interpretation methods can be divided into the following three categories in general.

The first category of methods aims to answer the "*What*"-type questions, i.e., what part of the input mostly contribute to the model prediction. A representative work in this cat-

egory is LIME [32], where authors employ linear models to approximate the local decision boundary and further formulate it as a sub-modular optimization problem for model interpretation. The feature importance in LIME is obtained by observing the prediction changes after perturbing input samples. Related methods can also be found in Anchors [33] and SHAP [25]. Another common methodology under this category is to utilize the model gradient information, where gradients are regarded as an indicator for perturbation sensitivity. Related methods can be found in GradCAM [36], Integrated Gradients [40], and SmoothGrad [39].

The second category aims to answer the "*Why*"-type questions, i.e., why the input is predicted as label *A* instead of *B*. The methods under this category can be quite different from the previous ones, since these methods need to consider two labels simultaneously. There are several different methods proposed for this problem. For example, the authors in [5] design a contrastive perturbation method to derive related positive and negative features of inputs regarding the concerned label. Besides, a general method based on structural causal models is proposed in [26] to tackle the problem in classification and planning scenarios. Also, a generative framework CDeepEx is designed in [8] to particularly investigate this problem for images by utilizing GAN.

The third category lies in the "*How*"-type questions, i.e., how to particularly modify the input so as to flip the model prediction to the preferred label. This problem is a natural extension of the "Why"-type, and it can somewhat to be handled by the second category of methods under some simple scenarios. However, for problems with high-dimension space, previous categories of methods typically fail due to the intractable computation for sample modification. Several particular methods are raised to solve this issue. For example, authors in [12] propose a straightforward solution with image region replacement, which is essentially a feature replacement process for input with the aid of a distractor. In work [1], authors novelly use the input itself as the distractor for feature replacement by utilizing GAN for inpainting. Besides, generative modeling is another potential way for this problem, and related methods can be found in [38; 18; 23]. Our work belongs to this branch of methodology.

6. CONCLUSION AND FUTURE WORK

In this paper, we design a framework to generate counterfactual explanation for black-box DNN models specifically with raw data instances. By taking advantage of the generative modeling, we effectively construct an attribute-informed latent space for particular data, and further utilize this space for counterfactual generation. To guarantee the validity of the generated samples, we propose the AIP method to iteratively optimize the specific attribute-informed latent vectors according to the counterfactual loss term, from which the counterfactuals can be finally obtained through data reconstruction. We evaluate the designed framework with AIP on several real-world datasets, including both texts and images, and demonstrate its effectiveness, sample quality as well as efficiency. Future extension of this work may possibly include the investigation under the "close possible worlds" assumption, where the goal is to find an optimal set of counterfactuals for a query instead of a single sample. Besides, employing causal models for counterfactual generation is another promising direction to explore.

7. REFERENCES

[1] C. Agarwal, D. Schonfeld, and A. Nguyen. Removing input features via a generative model to explain their attributions to classifier's decisions. *arXiv preprint arXiv:1910.04256*, 2019.

[2] N. Asghar. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*, 2016.

[3] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, pages 8928–8939, 2019.

[4] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116, 2017.

[5] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in neural information processing systems*, pages 592–603, 2018.

[6] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[7] M. Du, N. Liu, and X. Hu. Techniques for interpretable machine learning. *Commun. ACM*, 63(1):68–77, 2020.

[8] A. Feghahati, C. R. Shelton, M. J. Pazzani, and K. Tang. Cdeepex: Contrastive deep explanations. 2018.

[9] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi. Blackbox generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE, 2018.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations*, 2015.

[12] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee. Counterfactual visual explanations. In *ICML*, pages 2376–2384, 2019.

[13] T. Guo, J. Dong, H. Li, and Y. Gao. Simple convolutional neural network on image classification. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pages 721–724. IEEE, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] R. He and J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of WebConf*, 2016.

[16] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, 2019.

[17] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1587–1596, 2017.

[18] S. Joshi, O. Koyejo, B. Kim, and J. Ghosh. xgems: Generating examplars to explain black-box models. *arXiv preprint arXiv:1806.08867*, 2018.

[19] B. Kim, R. Khanna, and O. O. Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in neural information processing systems*, pages 2280–2288, 2016.

[20] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on EMNLP*, pages 1746–1751, 2014.

[21] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[22] J. Li, S. Ji, T. Du, B. Li, and T. Wang. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018.

[23] S. Liu, B. Kailkhura, D. Loveland, and H. Yong. Generative counterfactual introspection for explainable deep learning. Technical report, Lawrence Livermore National Lab.(LLNL), 2019.

[24] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

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

[26] T. Miller. Contrastive explanation: A structural-model approach. *arXiv preprint arXiv:1811.03163*, 2018.

[27] J. Moore, N. Hammerla, and C. Watkins. Explaining deep learning models with constrained adversarial examples. In *Pacific Rim International Conference on Artificial Intelligence*, pages 43–56. Springer, 2019.

[28] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[29] J. Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[30] J. Pearl and D. Mackenzie. *The book of why: the new science of cause and effect*. Basic Books, 2018.

[31] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, 51(5):1–36, 2018.

[32] M. T. Ribeiro, S. Singh, and C. 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*, pages 1135–1144, 2016.

[33] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *32nd AAAI Conference on Artificial Intelligence*, 2018.

[34] E. L. Rissland. Example-based reasoning. *Informal reasoning in education*, pages 187–208, 1991.

[35] P. Samangouei, A. Saeedi, L. Nakagawa, and N. Silberman. Explaingan: Model explanation via decision boundary crossing transformations. In *Proceedings of ECCV*, pages 666–681, 2018.

[36] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[37] A. Sen, X. Zhu, L. Marshall, and R. Nowak. Should adversarial attacks use pixel p-norm? *arXiv preprint arXiv:1906.02439*, 2019.

[38] S. Singla, B. Pollack, J. Chen, and K. Batmanghelich. Explanation by progressive exaggeration. In *8th International Conference on Learning Representations*, 2020.

[39] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[40] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.

[41] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

[42] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[43] A. White and A. d. Garcez. Measurable counterfactual local explanations for any classifier. *arXiv preprint arXiv:1908.03020*, 2019.