Information Sciences 678 (2024) 120974

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

CIRF: Importance of related features for plausible counterfactual explanations

Hee-Dong Kim, Yeong-Joon Ju, Jung-Ho Hong, Seong-Whan Lee *

Department of Artificial Intelligence, Korea University, Seoul, Republic of Korea

ARTICLE INFO

Keywords:
Explainable artificial intelligence
Counterfactual explanation
Generative adversarial networks
Generative neural networks

ABSTRACT

Counterfactual explanation (CFE) provides actionable counterexamples and enhances the interpretability of the decision boundaries in deep neural networks and thereby has gained increasing interest in recent years. An ideal CFE should provide both plausible and practical examples that can alter the decision of a classifier as a plausible CFE grounded in the real world. Motivated by this issue, we propose a CFE framework for identifying related features (CIRF) to improve the plausibility of explanations. CIRF comprises the following two steps: *i*) searching for the direction vectors that contain class information; *ii*) investigating an optimal point using a projection-point, which determines the magnitude of manipulation along the direction. Our framework utilizes related features and the property of a latent space in a generative model, thereby highlighting the importance of related features. We derive points that have many related features, and show a performance gain of more than 11% on the IM1 metric compared to points that have fewer related features. We validate the versatility of CIRF by performing experiments using various domains and datasets, and the two interchangeable steps. CIRF exhibits remarkable performance in terms of plausibility across various domains, including tabular and image datasets.

1. Introduction

The black-box nature of deep neural networks has led to concerns about the lack of interpretability in critical domains such as financial [1,2], medical [3], crime [4], or autonomous driving [5,6]. To address these concerns, researchers have examined the reasons behind the decisions of these models. Explanation approaches, such as feature attribution [7–10] and CounterFactual Explanation (CFE) [11,12], enhance the reliability and applicability of deep neural networks in industries by investigating the underlying reasons of predictions. Intuitively, CFE is based on the premise that modifying certain input features alters a model's decision to align with a user's intention. CFE is a post-hoc explanation method because it analyzes the basis of a model's decision after the decision is made [10,13,14]. The perspective of CFE stems from counterfactual reasoning, which addresses the propositions of the form "If $\mathcal{A}$, then $\mathcal{B}$ [15–19]." Thus, CFEs provide an example-based explanation ($\mathcal{A}$) that alters the decision of a classifier by modifying certain features. Through examples that can alter the decision, users can implicitly understand the decision-making mechanism of a classifier. In particular, the CFE is practical for users because it provides useful examples. For example, consider a

* Corresponding author.
E-mail address: sw.lee@korea.ac.kr (S.-W. Lee).

https://doi.org/10.1016/j.ins.2024.120974
Received 16 November 2023; Received in revised form 8 March 2024; Accepted 5 June 2024

Available online 11 June 2024

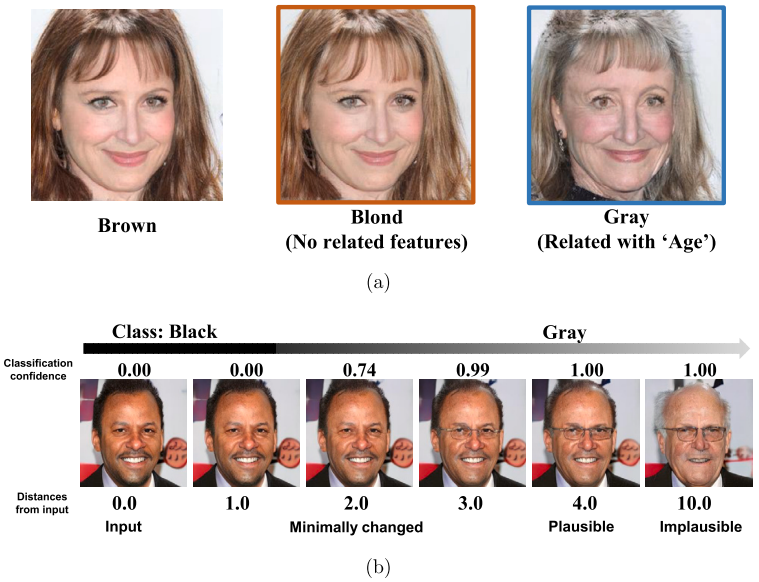

Fig. 1. (a) Comparison of non-related features and related features when altering class from 'Brown' hair to 'Blond' and 'Gray' hair. (b) Extent of manipulation required to produce a plausible CFE. The distances from the input latent codes for each image are provided at the bottom, and the classification confidence is provided at the top.

user who has been rejected for a loan using a credit rating model. The user requires insights not only into which features of their financial profile necessitate changes but also the degree to which these changes are needed. Therefore, it is advisable to provide actionable CFEs and quantify the extent of input changes so that the user adjusts their financial profile easily. The provision of plausible CFEs is thus essential. In summary, CFE modifies an input to alter the decision of a classifier, providing a sample-based explanation that enables users to understand the basis for the decision. While research aims to improve the plausibility of CFE, examining approaches to increase the plausibility remains under-explored.

Recent studies have aimed to improve the plausibility of CFE by providing examples as similar as possible to the input [11,14,20–22]. A CFE that is similar to an input is deemed actionable, thus enhancing the plausibility of CFE by being easily attainable with small modification. However, a notable drawback of previous CFE approaches is that they provide implausible examples, particularly when the relationships between features are not considered. For instance, previous methods may produce implausible explanations, such as an increase in asset with decrease in deposit, which is unlikely plausible in the financial area because of the generally positive correlation between assets and deposits. In this case, assets and deposits are "related features," which should be considered in a CFE to provide plausible examples. Such implausible examples degrade the credibility of an explanation as they are not aligned with the understanding of the relationship between features in the real world. Previous studies have examined spurious correlations, which are similar to related features [23,24]. Spurious attributes, which are the components of spurious correlations, are features associated with the class labels [25]. However, unlike spurious correlations, related features are associated with both features and the class labels. Consequently, both related features and target features collaboratively contribute to altering the class label. Although utilizing related features can enhance the plausibility of CFE by synergizing with target features, this promising approach has yet to be extensively explored.

In this study, we present a novel framework, called CFE identifying related features (CIRF), to generate plausible counterexamples that alter the decision of a classifier in a CFE manner. To provide a brief overview, we initially determine center-of-target points and direction vectors in a latent space. Subsequently, we calculate a projection-point in the latent space of a Generative Adversarial Network (GAN) [26], using a direction vector to influence the decision. The center-of-target point in the latent space is defined as a latent code that retains class features and related features identified by the classifier. As shown in Fig. 1(a), the CFE provided by our approach changes the 'Age' and 'Color' features when altering to 'Gray hair.' In contrast, the approach changes only 'Color' when altering to 'Blond hair' since 'Blond hair' lacks relationship with 'Age.' To verify whether these features are related or not, we investigated the training dataset and results at Sec. 4.5. Moreover, our framework attempts to produce the most plausible example through a projection-point, which is the point closest to the center-of-target point when the input moves along the direction vector. Due to the property that close latent codes produce similar outputs, the projection-point preserves the input features when the input is manipulated by a small magnitude. To show the plausible CFE, Fig. 1(b) illustrates three examples: the minimally changed CFE, the plausible CFE from the projection-point, and the implausible CFE due to excessive changes. As shown in Fig. 1(b), we observe that both 'Eyeglasses' and 'Age' are features associated with 'Gray hair.' Moreover, it is evident that the 'Eyeglasses' and 'Age' features influence the classification as 'Gray hair,' and the excessively changed sample loses the identity of the original input. To prevent excessive changes, CIRF specifies the magnitude of manipulation by calculating the projection-point, whereas previous approaches [27–29] investigated the latent spaces to change attributes without specific constraints on the magnitude. This procedure enables the projection-point to be located at the minimum distance from the input and the center-of-target point. Our method of calculating the

projection-point utilizes a direction vector and three reference points: the input, the center-of-target point, and the projection-point. Therefore, we address plausibility issues with the direction vector by changing the related features and specifying the magnitude of manipulation. Simultaneously, CIRF preserves the features that do not need to be changed to alter the decision, such as 'Smile' or 'Pose.' Throughout the paper, we describe the method and demonstrate the adaptability of CIRF using various datasets: image datasets (MNIST [30], CIFAR10 [31], CelebA-HQ [32], and ImageNet [33]) and a tabular dataset (HELOC [34]). Lastly, we demonstrate the practicality of CIRF by integrating CIRF with GAN inversion, mapping an image space to the latent space [35–37]. Through the integration with GAN inversion, we can deploy CIRF to real-world data by manipulating inverted latent codes. The contributions of this study are summarized below.

- We propose a novel, data-driven framework that determines the magnitude of manipulation in a latent space, thereby creating plausible examples by projection of a center-of-target point. To the best of our knowledge, this is the first study to adopt using a classifier in a data-driven approach for determining the magnitude of modification in a latent space.
- We highlight the importance of related features in generating a plausible CFE by showing related features and investigating the dataset.
- The two interchangeable steps in our framework improve the applicability of the proposed method by replacing each step with other methods. We revisit the advantage of replacing the first step with InterFaceGAN [36].

2. Related works

The CFE is a post-hoc explanation method that has attracted attention because of its practicality by providing the implications for the model decisions and ready-to-use examples [15]. Previous methods cross a decision boundary by directly perturbing an input such that a decision can be intuitively altered [27,29,38–40]. Meanwhile, the methods that calculate the gradient of a classifier and reveal feature attributions have been proposed, and they exhibit potential for applications to complex datasets [9,11,41–44]. The contrastive explanations method approach [45] provides a saliency map to divide the areas to be classified into each class. Salience maps are widely used in explanation methods because they increase the intuitiveness of the user's interpretation of the model. Counterfactual visual explanation [12] expands CFE to the vision domain by identifying the regions to be classified as counterclasses. These works report that a CFE should be distinguished from an adversarial attack [46], which is a fooling classifier. Furthermore, they have established a field of research by providing the appropriate basis for a decision of a given classifier. Studies [47,48] have provided diverse CFEs. Specifically, [47] provides nontrivial explanations to examine various decision boundaries of the model. Studies [17,49] provide model-agnostic CFEs, which can be applied to various models. The model-agnostic approach does not need an internal interpretation of the model, thereby enabling the approach to be broadly applied to various tasks. Moreover, studies have been conducted to improve the plausibility of CFE [20,39,50,51]. Gradual Construction [11] states that an explanation with a minor change is actionable, which inspires us to provide a minimally changed CFE. The approach creates an explanation by calculating the gradient of a classifier and gradually changing it from the most influential feature. Plausible exceptionality-based contrastive explanations (PIECE) [20] improves plausibility by changing exceptional features and simultaneously defining a semi-factual explanation which are feature modifications within the bounds of not changing a decision. A semi-factual explanation is practical because it guides the users to the extent that they can act within a range that does not change a decision of a model. A study [48] points out a limitation of a semi-factual explanation, which is a lack of general definition, thereby providing CFEs and semi-factual explanations for one sample. A study [50] utilized the causal relationships between the input features and an oracle to improve plausibility. Recent work [52] aims to improve the plausibility by harnessing a generative model. [52] also points out small changes that have high adversarial power, leading to degrading plausibility, which is similar to a gap between conventional CFEs and our plausible explanation. Although numerous studies have attempted to enhance the plausibility of CFEs, the utilization of related features remains unexplored.

3. Methods

Table 1 summarizes mathematical symbols and their descriptions. Throughout this work, calculating and manipulating vectors is the fundamental process, so vectors are indicated by arrows at the top of the symbols to aid understanding.

3.1. Preliminaries

The goal of CFE is to generate examples that can cross the decision boundaries of a given classifier. A previous study [15] formulated an objective function for CFEs by minimizing loss using the following equation:

$$\mathcal{L} = \arg\min_{x'} (F(x') - c')^2 + \tau \mathcal{D}(x, x'), \tag{1}$$

where τ is a coefficient and $\mathcal{D}$ is the Euclidean distance function. The CFE that is closer to the input is preferable because $\mathcal{D}$ calculates the distance between the input and the CFE. The quality of the CFE is determined by the evaluation metrics that comply with the objective function. We discuss these evaluation metrics in Sec. 4.2, including those based on distance ($L1$ and $L2$ [15]) and those that assess plausibility or interpretability (IM1 and IM2 [53]). CFE uses the objective function to minimize the distance between the input and the CFE, and simultaneously alters the decision.

Table 1
Notations, dimensions, functions, and descriptions of each symbol.

Notation	Dimension or Function	Description
F	$\mathbb{R}^{h\times w} \rightarrow \mathbb{R}^T$	Target classifier to interpret
T	$\mathbb{R}$	Number of total classes produced by F
x, x'	$\mathbb{R}^{h\times w}$	Input of F ($x \in \mathcal{X}$)
G	$\mathbb{R}^d \rightarrow \mathbb{R}^{h\times w}$	Generative model
$\vec{z}, \vec{z'}$	$\mathbb{R}^d$	Input latent code of G ($\vec{z} \in \mathcal{Z}$)
d	$\mathbb{R}$	Dimension of the latent code of G
$\vec{n}$	$\mathbb{R}^d$	Direction vector
$\overrightarrow{CT}$	$\mathbb{R}^d$	Center-of-target
$\overrightarrow{PP}$	$\mathbb{R}^d$	Projection-point
λ	$\mathbb{R}$	Scalar value for magnitude of manipulation
τ	$\mathbb{R}$	Coefficient of the objective fuction of CFE
α, β	$\mathbb{R}$	Coefficient of the objective fuction of Step. 1
R	$\mathbb{R}$	Filtering ratio
$D(\cdot,\cdot)$	$\mathbb{R}$	Distance function

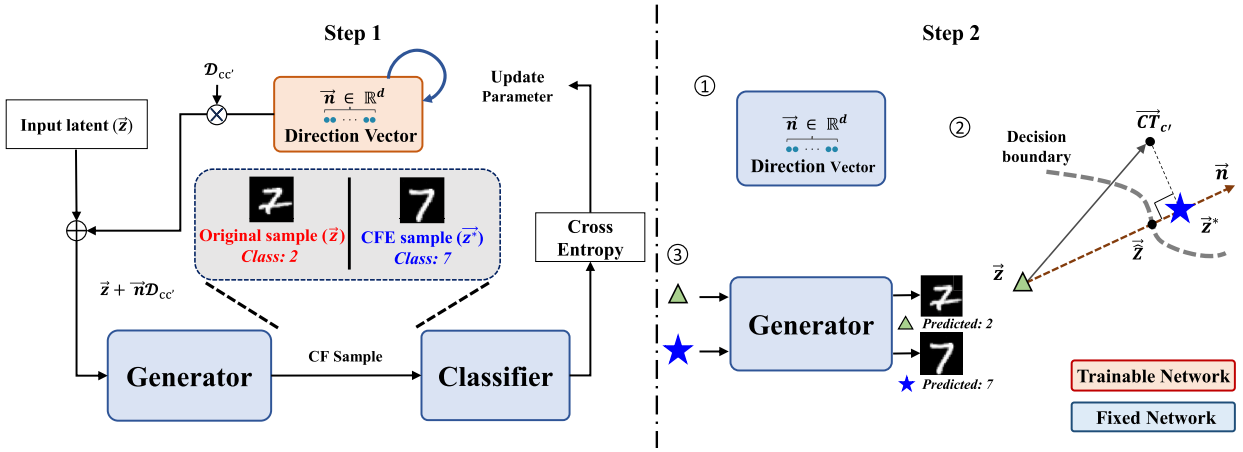

Fig. 2. Two primary steps of our framework for generating CFEs. In step 1, the direction vector is trained by pre-trained networks and class-change difficulty. The network in the blue box is a fixed network, whereas the network with the red box is a trainable network. In step 2, the projection-point is computed using the direction vector, input, and center-of-target point. Then the generator provides the CFE with the projection-point.

3.2. Problem statement

The primary objective of this study is to improve the plausibility of the generated CFE by leveraging the related features inherited in the classifier itself. To this end, CIRF calculates a direction vector, denoted as $\vec{n}$, which enables the CFE to alter a prediction by manipulating related features. Subsequently, CIRF identifies the center-of-target point, $\overrightarrow{CT_{c'}}$, which contains the related features of the target class. With $\vec{n}$, $\overrightarrow{CT_{c'}}$, and the input latent code $\vec{z}$, CIRF determines three distinct points: *i*) the minimally changed CFE; *ii*) the semi-factual explanation; and *iii*) the plausible CFE. The minimally modified CFE faithfully satisfies Eq. (1). The semi-factual explanation indicates the potential feature modifications, achieved through latent interpolation, while retaining the original class to the maximum extent possible. The plausible CFE aims to determine the magnitude by which the latent input should be manipulated with respect to the input. We solve the problem of including as many related features of the target class as possible by simplifying the problem by finding the point closest to $\overrightarrow{CT}$ of the target class within the latent space. Consequently, our goal is to identify the magnitude and direction of manipulation with consideration of related features. The following sections explain how to calculate this simplified problem with a mathematical tool that finds the shortest distance in latent space.

3.3. Latent code manipulation

We utilize the property of the latent space in which close latent codes generate similar outputs throughout the study. As given by Eq. (1), it is recommended to generate a CFE that is similar to the input. The properties that yield similar outputs using close latent codes can satisfy the second term of Eq. (1). To comply with the first term of Eq. (1), CIRF initially calculates the direction vector $\vec{n}$ that contains target class features to alter the decision of a classifier. Fig. 2 visualizes our framework, including the calculation of $\vec{n}$ (step 1) and $\overrightarrow{PP}$ (step 2). The first step involves training $\vec{n}$, and the second step entails manipulating the input latent code ($\vec{z}$) towards the direction ($\vec{n}$). In the second step, CIRF calculates the projection of $\overrightarrow{CT_{c'}}$ on $\vec{n}$ to specify the magnitude of manipulation (λ^*). Subsequently, $\vec{z}$ is directed toward the derived vector $\vec{n}$ up to λ^*. We interpolate from the input latent to the projection-point

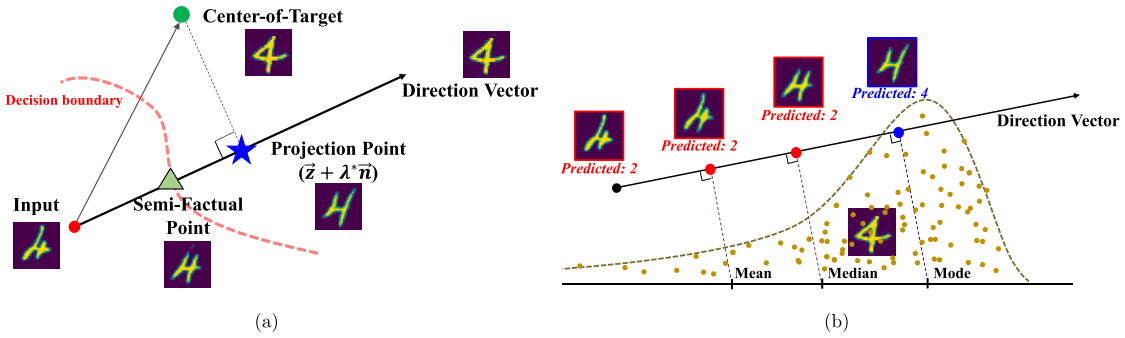

Fig. 3. Diagrams in the latent space. (a): Principle of calculating the projection-point, given the input sample and direction vector. (b): Example of results generated by asymmetric distribution of center-of-target points.

to identify the minimally changed CFE which is a CFE closest to the decision boundary of F. Finally, $\overrightarrow{PP}$ is defined as $\overrightarrow{PP} = \vec{z} + \lambda^*\vec{n}$ (see Fig. 3(a)). $\overrightarrow{PP}$ stands closest to the $\overrightarrow{CT_{c'}}$ and it maintains similarity with $\vec{z}$ in the latent space because of the close latent code property. In the following sections, we describe the calculation of $\overrightarrow{PP}$ using CIRF in two steps: *i*) Direction vector calculation and *ii*) Projection-point search.

3.4. Direction vector calculation

As shown in Fig. 2, $\vec{n}$ is calculated in the first step by utilizing a pre-trained generative model G and a classifier F. This algorithm can be applied to any classifier, given that a GAN is trained on the identical dataset. The training procedure begins with the identification of center-of-targets points, $\overrightarrow{CT_c}$ and $\overrightarrow{CT_{c'}}$, to capture the features of classes c and c'. Each element of $\overrightarrow{CT}$ is defined as the mode of each element of latent codes. Mode points are defined by generating random samples from G along with random noise $\mathcal{Z}$. These samples are then classified using F to extract confidently classified points along the output logits. Subsequently, the classified samples are filtered in a high logit order with a predefined filtering ratio R. Then, Gaussian Mixture Models (GMMs) learn from the filtered latent codes of each element and extract the mode points with the highest probability. We utilize GMMs to approximate unknown distributions. Candidates of $\overrightarrow{CT_{c'}}$s (mean, median, and mode) are shown in Fig. 3(b). We empirically observe that mode points are preferable in the case of asymmetric distributions, as shown in Fig. 3(b). After defining $\overrightarrow{CT}$s, we calculate the initial direction vector and normalized version of the direction vector as follows:

$$\vec{n} = \frac{\overrightarrow{n_{init}}}{||\overrightarrow{n_{init}}||_2}, \overrightarrow{n_{init}} = \overrightarrow{CT_{c'}} - \overrightarrow{CT_c}, \tag{2}$$

where $\overrightarrow{n_{init}}$ and $\vec{n}$ denote the initial and normalized direction vectors, respectively. This initialization ensures that $\vec{n}$ does not settle into undesired local optima. We simplify the optimization process by utilizing $\overrightarrow{n_{init}}$, which guides $\vec{z}$ toward the target features of c'. Before $\vec{n}$ is added to the input latent code $\vec{z}$, $\vec{n}$ is multiplied by $D_{cc'}$ where $D_{cc'} = D(\overrightarrow{CT_c}, \overrightarrow{CT_{c'}})$. $D_{cc'}$ is the reference of class difficulty, and it enables $\vec{n}$ to consider the difficulties associated with crossing the decision boundaries from the class c to c'. For instance, the distance required to alter the class from 'Dog' to 'Truck' is longer than that required to alter the class from 'Dog' to 'Cat' because the former involves changing more features. If normalization and class difficulty are neglected, then $\vec{n}$ identifies a typical direction and magnitude, thus ensuring monotonous manipulations and achieving the high class-confidence for the target class. In particular, the normalizing term prevents extremely large magnitudes, which can dominate the class-confidence. For training $\vec{n}$, we calculate the initial counterfactual example x' with the following equation:

$$x' = G(\vec{z} + D_{cc'}\vec{n}), \tag{3}$$

where x' denotes an initial CFE. Then, $\vec{n}$ is optimized by minimizing the following loss:

$$\mathcal{L} = C(F(x'), c') + \alpha\,||\vec{n}||_1 + \beta\,||\vec{n}||_2, \tag{4}$$

where C represents the cross-entropy function. The second and third terms are elastic net regularization terms, which are introduced to allow sparse manipulation of $\vec{n}$ and to prevent $\vec{n}$ from simultaneously changing a large number of features [54]. The calculation of $\vec{n}$ is shown in Algorithm 1. Note that the first step, which aims to calculate the direction vector $\vec{n}$ can be substituted with a method that identifies meaningful directions in the latent space. The alternative to the first step is discussed in Sec. 4.6.

3.5. Projection-point search

In the second step, as shown in Fig. 2, we calculate $\overrightarrow{PP}$ to determine the magnitude of latent manipulation, as shown in Fig. 3(a). $\overrightarrow{PP}$ is derived through a projection operation that fulfills three conditions: *i*) $\overrightarrow{PP}$ facilitates the class alteration from c to c', *ii*) $\overrightarrow{PP}$ remains proximate to the input $\vec{z}$, and *iii*) $\overrightarrow{PP}$ maintains closeness to $\overrightarrow{CT_{c'}}$. The first and second conditions stem from the first and second terms in the objective function of CFE (Eq. (1)). The last condition is added to capture target features and related features.

Algorithm 1 Direction Vector Calculation.

Require: Classifier F, Generator G, Input latent $\vec{z}$, Training steps S, Filtering rate R, Gaussian Mixture Model GMM, Euclidean distance function $\mathcal{D}$, Cross-Entropy function CE, Sampling function f_s samples the mode of the class from GMM, Predefined variance $\Sigma_{\mathcal{Z}}$ and $\Sigma_{\vec{n}}$

Sample $\mathcal{Z} \sim \mathcal{N}(0, \Sigma_{\mathcal{Z}})$
$N_c, N_{c'} \leftarrow \text{sort}(F(G(\mathcal{Z})))$ ▷ Sort by confidence for corresponding classes
N' is assigned the first $R \cdot \text{len}(N)$ elements from the sorted list N
Fit $GMM_c(N'_c)$ and $GMM_{c'}(N'_{c'})$ ▷ Train GMM with latent codes
$\overrightarrow{CT_{c'}} \leftarrow f_s(GMM_{c'})$
$\overrightarrow{CT_c} \leftarrow f_s(GMM_c)$
Initialize $\vec{n} = \frac{\overrightarrow{n_{init}}}{||\overrightarrow{n_{init}}||_2}$, $\overrightarrow{n_{init}} = \overrightarrow{CT_{c'}} - \overrightarrow{CT_c}$ ▷ Eq. (2)
for $s \in 1, .., S$ **do**
  $c \leftarrow F(G(\vec{z}))$
  $\mathcal{D}_{cc'} \leftarrow \mathcal{D}(\overrightarrow{CT_c}, \overrightarrow{CT_{c'}})$
  $\mathcal{L} \leftarrow CE(F(G(\vec{z} + \vec{n}\mathcal{D}_{cc'})), c') + \alpha ||\vec{n}||_1 + \beta ||\vec{n}||_2$ ▷ Eq. (3), (4)
  $\vec{n}$ is updated by $\mathcal{L}$
end for
return $\vec{n}$

In order to capture features, we need to find the closest point from $\overrightarrow{CT_{c'}}$, on the line of $\vec{n}$. We transform the problem of finding this point into the least square solution, which can be obtained using a projection operation. The projection operation is calculated by the inner product of two vectors $\overrightarrow{CT_{c'}} - \vec{z}$ and $\vec{n}$, identifying the closest point on the line of $\vec{n}$ from $\overrightarrow{CT_{c'}}$. Leveraging the properties of the generative model, $\overrightarrow{PP}$ integrates features from both $\vec{z}$ and $\overrightarrow{CT_{c'}}$, while altering class. Given that $\overrightarrow{CT_{c'}}$ contains representative information for the target class c', an ideal $\overrightarrow{CT_{c'}}$ enhances the overall performance (discussed in Table 4). We project $\overrightarrow{CT_{c'}}$ onto $\vec{n}$ to obtain $\overrightarrow{PP}$. We define the initial $\overrightarrow{PP}$, which is formulated as follows:

$$\vec{\hat{z}} = \vec{z} + \lambda\vec{n},\ \lambda = \vec{n} \cdot (\overrightarrow{CT_{c'}} - \vec{z}), \tag{5}$$

where $\vec{\hat{z}}$ represents the initial $\overrightarrow{PP}$. Thereafter, CIRF establishes the final CFE by iteratively expanding to both sides of the $\vec{n}$ vector with an initial $\overrightarrow{PP}$ as the center. This iterative search algorithm enables us to consider the imprecision of F and the entangled latent space of G. As a result, the algorithm helps prevent the classification of $\overrightarrow{PP}$ as c despite containing numerous features of c', thereby enabling that the CFE is appropriately classified as c'. Thus, CIRF calculates a new point located close to $\vec{\hat{z}}$, which is classified as c'. The formula for identifying the new point is as follows:

$$\begin{aligned} \vec{z}^* &= \vec{z} + \lambda^*\vec{n},\ \lambda^* = \lambda - \upsilon, \\ \upsilon &= \arg\min_{\upsilon}(F(G(\vec{z} + (\lambda - \upsilon)\vec{n})) - c'), \end{aligned} \tag{6}$$

where $\upsilon \in \mathbb{R}$, and $\vec{z}^*$ is the new $\overrightarrow{PP}$. This algorithm is useful when applying complex datasets that prevent the CFE from being classified as c', for example, ImageNet. The second step is represented at Algorithm 2. Moreover, our method is able to provide the following three types of explanations by interpolating the input latent $\vec{z}$ and $\vec{z} + \lambda^*\vec{n}$: *i*) $G(\vec{z} + \hat{\lambda}\vec{n})$: a minimally modified CFE that alters the class and enlightens the decision boundary, where the magnitude of the change $\hat{\lambda}$, is derived by interpolating $\vec{z}$ and $\vec{z} + \lambda^*\vec{n}$; *ii*) $G(\vec{z} + ((\hat{\lambda} - \gamma)\vec{n})$: a semi-factual explanation [20] that is close to the decision boundary but does not cross it, where γ is a small value; and *iii*) $G(\vec{z} + \lambda^*\vec{n})$: a plausible CFE derived from $\overrightarrow{PP}$.

Algorithm 2 Generating CFE.

Require: Classifier F, Generator G, Input latent $\vec{z}$ from x, Target class c', Center-of-target points $\overrightarrow{CT_{c'}}$, Direction vector $\vec{n}$ derived from Algorithm 1)

$\lambda \leftarrow \vec{n} \cdot (\overrightarrow{CT_{c'}} - \vec{z})$,
$\vec{\hat{z}} \leftarrow \vec{z} + \lambda\vec{n}$ ▷ Eq. (5)
if $F(G(\vec{\hat{z}})) = c'$ **then**:
  $\vec{z}^* \leftarrow \vec{\hat{z}}$
  return $\vec{z}^*$
else:
  $\vec{z}^* \leftarrow \vec{z} + \lambda^*\vec{n}$, where $\lambda^* = \lambda - \upsilon$,
  $\upsilon \leftarrow \text{argmin}_{\upsilon}(F(G(\vec{z} + (\lambda - \upsilon)\vec{n})) - c')$ ▷ Eq. (6)
  return $\vec{z}^*$
end if

4. Experiments

4.1. Experimental setup

We use the following four classification datasets in tabular and image domains: *a*) **HELOC** is a tabular dataset designed for loan approval/refusal classification, *b*) **MNIST** is a handwritten digit dataset, *c*) **CelebA-HQ** is a facial dataset encompassing with 40 classes, and *d*) **ImageNet** is a comprehensive image database comprising 1,000 classes. The same paired datasets ($\mathcal{Z}$ or $\mathcal{W}$ for the projected latent for StyleSwin [37] and StyleGAN-XL [55], along with $\mathcal{X}$ of G) are used to evaluate each method. Note that $\mathcal{Z}$ is replaced by the projected latent $\mathcal{W}$ in StyleSwin and StyleGAN-XL to utilize the highly disentangled latent factors of variation [56]. The settings of all the predefined hyperparameters and each model are described in the appendix.

4.2. Evaluation metrics

We used the following evaluation metrics to assess our method.

IM1 Metric. This metric is defined as IM1$= \frac{||I' - AE_{c'}(I')||_2^2}{||I' - AE_{c}(I')||_2^2}$, where a lower score indicates better performance [53]. I' denotes a generated CFE, and AEs are autoencoders trained by the corresponding classes. The IM1 metric evaluates both interpretability [53] and plausibility [20]. We use this metric to measure the plausibility of the target class, and the score is low when generated I' is plausible. Thus, a lower IM1 score suggests that I' closely aligns with class c' and deviates from c. Since the degree of AE training influences the IM1 score, we use multiple random seeds for the complex dataset in Sec. 4.6. The IM2 metric is given by IM2$= \frac{||AE(I') - AE_{c'}(I')||_2^2}{||I'||_1 + \epsilon}$, where AE is trained by all classes [53]. Since this metric has an inaccuracy issue [20], we will investigate its performance in the future work.

MC-mean and NN-dist Metrics. These two metrics were introduced in a recent study [20]. MC-mean assesses the robustness of a CFE. It activates the dropout of the fully-connected layers of F and checks the classification of the CFE into c' across multiple inference trials. We performed 1,000 inferences to ensure consistency with the baseline [20]. NN-dist assesses the distance between the input(x) and the CFE(x') using the feature space rather than the input space. Hence, it is particularly suitable for image domain evaluations. It becomes feasible to differentiate CFEs from adversarial attacks [46,57] by comparing similarities in the feature space, particularly because adversarial examples exhibit high similarity in the input space. Conventional distance metrics in the input space, such as $L1$ and $L2$ metrics [15], frequently fail to differentiate between adversarial attacks and CFEs.

Validity score. The validity score quantifies the fraction of CFEs correctly classified as c' by a given classifier.

Note that we did not evaluate the Fréchet Inception Distance (FID) or other metrics for generative models for several reasons. Firstly, as CIRF employs a pre-trained generative model, the FID of CFEs inherently depends on the pre-trained generator. Furthermore, the primary purpose of CFEs is to probe the decision boundary of a classifier, which does not necessitate an evaluation of the image structure. The reason is that CFE approaches are designed to alter the prediction of the classifier efficiently, so the generated CFEs are not required to closely resemble the training dataset in the feature space. Consequently, our evaluation mainly concentrated on the IM1 metric, assessing the plausibility and interpretability.

4.3. Shifting latents towards the center-of-target point

In this section, we describe how shifting a latent closer to the $\overrightarrow{CT}$s enables the latent to provide more plausible examples. This plausibility is attributed to the $\overrightarrow{CT}$s containing the critical features that are considered by the classifier. We present an experiment to demonstrate the capability of $\overrightarrow{CT}$s in generating more plausible examples. This example shows that the points close to $\overrightarrow{CT}$s provide more plausible examples for the classifier than those far from $\overrightarrow{CT}$s. First, we trained a GAN with a three-dimensional latent space using the MNIST dataset in order to visualize the latent space. Fig. 4 represents the three-dimensional latent space of the trained GAN. We created grid sample points in the latent space and classified the samples generated from these points using a classifier. As shown in Fig. 4(a), the latent codes that generate samples to be classified into the same class were grouped. Subsequently, only the points classified with high confidence were filtered out, leading to clearer groupings as depicted in the right part of Fig. 4(a). The outline was drawn using the Convex Hull construction, where the dotted line represents a pseudo-decision boundary. Furthermore, we noticed that the centroids of the grouped latent codes generated examples that appeared more plausible to humans as evident in the centers corresponding to classes 1 and 0 in Fig. 4(b). Through this experiment, we empirically demonstrate that the closer a latent code is to the centroid of the grouped latent codes, the more likely it is to be classified into the corresponding class. As a result, our goal is narrowed down to defining the center of the grouped latent codes and determining the magnitude of manipulation of the input latent code. In the following section, we present a performance comparison of $\overrightarrow{PP}$s using multiple $\overrightarrow{CT}$s to determine the manipulation.

4.4. Comparison of center-of-targets

As discussed in Sec. 3.5, $\overrightarrow{CT_{c'}}$ strongly affects the plausibility performance. We empirically know the mode is better in case of non-symmetric distribution, like Fig. 3 (b). We thus use the mode point of each component from the filtered samples. Before deriving the $\overrightarrow{CT_{c'}}$, we first analyze the distributions of the latent codes in the MNIST training dataset. As shown in Table 2, the distributions

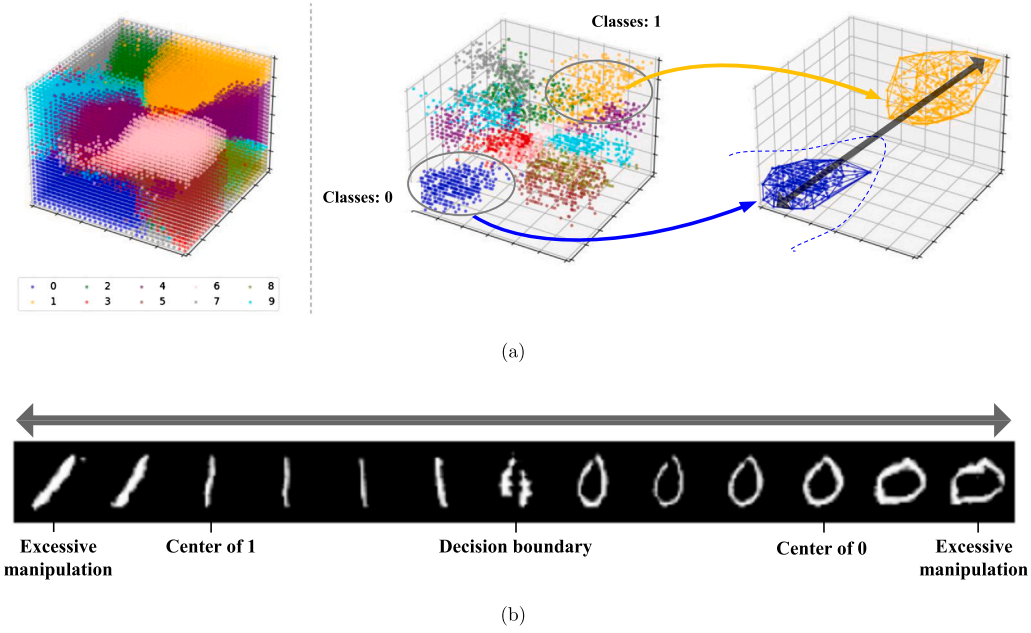

Fig. 4. The latent space of a GAN is trained with a three-dimensional latent space. Each color corresponds to a specific class. (a): Latents and their corresponding classes as classified by the classifier. The left side presents the classification results for grid sample points in latent space, while the right side presents the outcomes after filtering confidently classified points. (b): Gradual changes along the black arrow shown in (a). The lined up samples are results generated by gradually manipulating latent codes directed by the black arrow in (a).

Table 2
Distributions of each latent element of training datasets tested using the Kolmogorov-Smirnov test. The 'Gaussian' and 'Else' distributions are described in Fig. 5.

Dataset	Class	Gaussian	Else	Total
HELOC	Entire	0	128	128
	Each	6	250	256
MNIST	Entire	28	72	100
	Each	824	166	1000

had characteristics of either a Gaussian distribution or a gamma distribution. Even in the case of an 'else' distribution, it takes on a specific form that closely resembles a Gaussian distribution, as shown in Fig. 5. A Kolmogorov-Smirnov test is conducted with a significance level of 0.05 to determine whether the distributions fit the Gaussian or gamma distribution. In addition, we further analyze the percentile probability for each value of the latent element. We sample 10,000 values and calculate the probability of each value using trained GMMs. Subsequently, we extract the 0.25, 0.5, 0.75, and 1.0 percentile points in ascending order of probability. As shown in Fig. 6, the overall performances get better as the probability of an element's value increases, particularly reflected in the IM1 score. This is because $\overrightarrow{CT_{c'}}$s contain the features of c', thereby enabling $\overrightarrow{PP}$ to have the features of c'. Therefore, the target features are effectively captured using the mode as $\overrightarrow{CT_{c'}}$, thus improving the overall performance.

4.5. Investigation of related features

This section describes the identification of related features by classifier F for the CelebA-HQ dataset, which is used for training F. In addition, we demonstrate how these features manifest in plausible CFEs. The CelebA-HQ dataset comprises 30,000 facial images and 40 classes. Each class corresponds to each attribute. These rich attribute labels enable us to effectively explore related features. First, we examine the relationship between the classes 'Gray hair' and 'Young' classes. Since the 'Young' class is negatively related features to 'Gray hair' 1,242 samples are labeled as having 'Gray hair,' out of which 43 are labeled as 'Young' and 1,199 are labeled as 'not Young.' We analyzed odds ratio (OR) to investigate relationships between features. An OR greater than 1 indicates a positive association, whereas an OR less than 1 suggests a negative association. The odds of having 'Gray hair' given being 'Young' are 0.0018 and for not being 'Young' are 0.22, resulting in an OR of 0.0083 (see Table 3). As the OR value for the 'Young' class was extremely lower than 1, the 'Young' and 'Gray hair' features can be represented as negatively correlated. When the classifier is trained using this dataset, it naturally associates 'Gray hair' with being 'not Young.' As a result, if an example contains both related features, the classifier considers it to be more plausible. The results of setting c' to 'Gray hair' are shown in Fig. 7. In addition to the

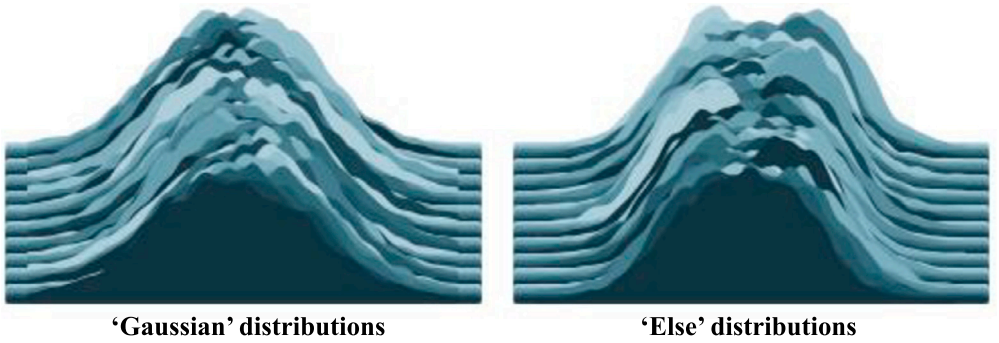

Fig. 5. 'Gaussian' and 'Else' distributions of each latent element on the training dataset.

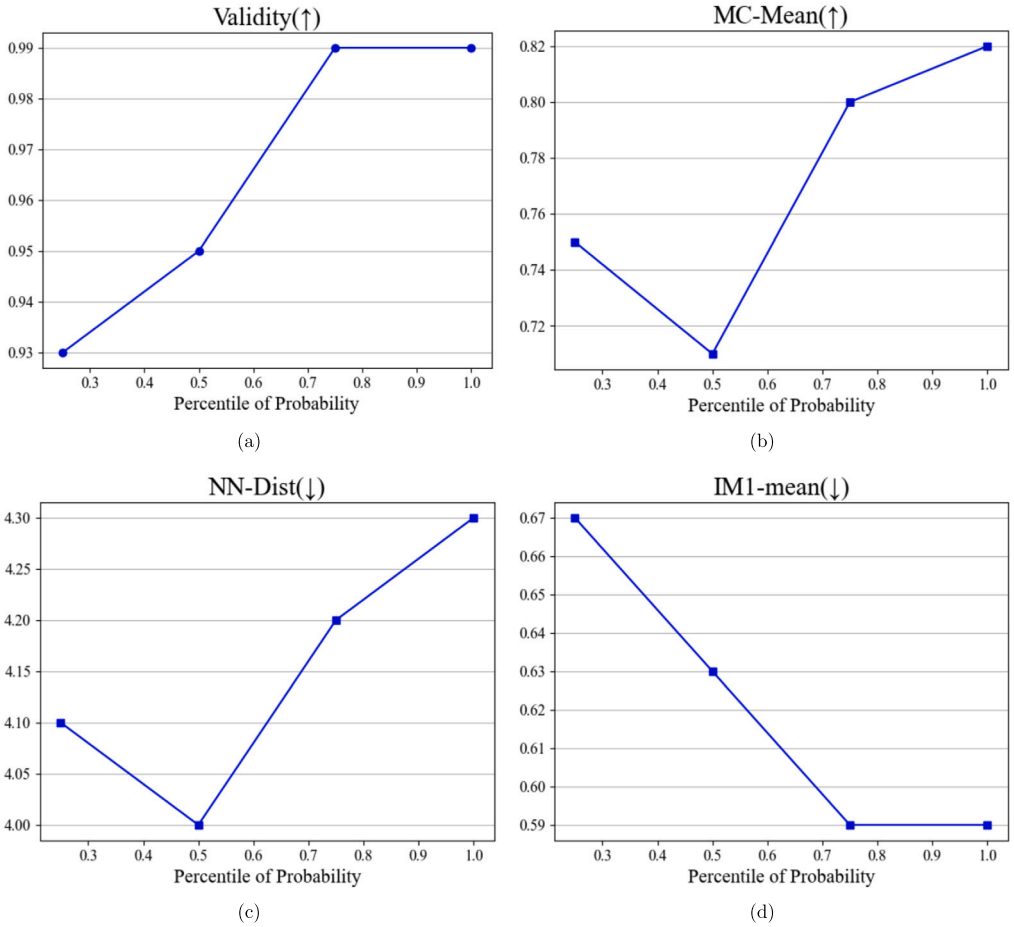

Fig. 6. Brief comparison of $\overline{CT}$s. The overall performance improves as the percentile increases on the MNIST dataset.

'Young' class, we also examine other related features 'Male' and 'Eyeglasses,' as shown in Table 3 and Fig. 7. A significant number of examples change the 'Young,' 'Male,' and 'Eyeglasses' classes together when altering class to 'Gray hair.' This alignment enhances the plausibility of decisions within the scope of F. As the CIRF algorithm calculates $\overrightarrow{PP}$ by projecting $\overrightarrow{CT_{c'}}$ onto $\vec{n}$, the related features considered by F are reflected in the CFEs. The reason for this is that F influences $\overrightarrow{CT_{c'}}$ and $\vec{n}$. Conversely, Fig. 8 represents cases of low relationships between features. When the target class has low relationships with other features, CIRF only changes the target features. Details on the OR for 'Brown hair' and 'Blond hair' are presented in the appendix.

Table 3
Related features in the CelebA-HQ dataset, which consists of 30,000 face images and 40 labels.

Odds of having 'Gray hair' given features ($Odds(Gray\|features)$)		Odds ratio	Variations
Male (0.10)	not Male (0.012)	8.3	not Male → Male
Young (0.0018)	not Young (0.22)	0.0083	Young → not Young
Eyeglasses (0.27)	no Eyeglasses (0.034)	7.9	no Eyeglasses → Eyeglasses

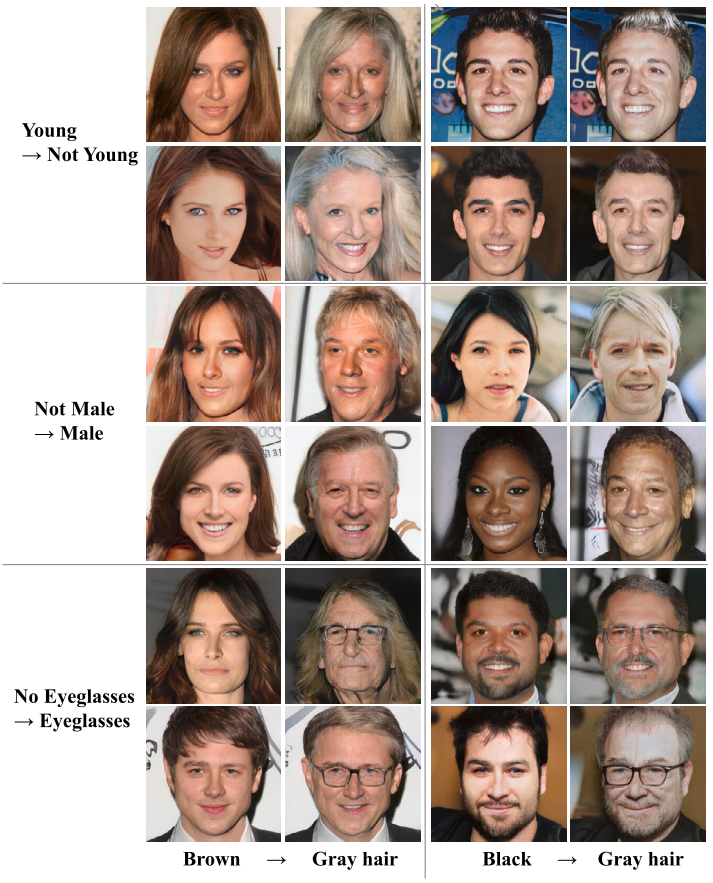

Fig. 7. Altering class to 'Gray hair' to examine related features. 'Gray hair' is associated with certain related features, such as 'Age,' 'Gender,' and 'Eyeglasses.'.

4.6. Evaluation of plausibility

This section describes the evaluation results and the assessment settings and highlights how CIRF achieves high performance in terms of plausibility through the IM1 metric. Table 4 shows the performance of the CFEs. The terms 'Ours w/ SVM' and 'Ours w/ St.1' denote the usage of the Support Vector Machine (SVM) in InterFaceGAN [36] and step 1 in our method to compute $\vec{n}$, respectively. As shown in the table, 'Ours w/ SVM' and 'Ours w/ St.1' exhibit the highest performance in terms of the IM1 metric. This is because CIRF simultaneously changes the critical and related features for the target class. Direction vector $\vec{n}$ guides the manipulation of related features, and $\overrightarrow{PP}$ represents the point close to $\overrightarrow{CT_{c'}}$. In contrast, the baseline only changes critical features to alter the decision. To ensure a fair comparison, Table 4 is obtained using publicly available code and uses the same datasets, classifier, and generator as the baseline. Note that the baseline only evaluates 163 samples for MNIST and 60 samples for CIFAR10, and this was conducted for only one target class. In contrast, our evaluation encompasses 108 samples for MNIST (split evenly between 54 misclassified and 54 correctly classified samples) and the first 300 samples of the CIFAR10 test dataset, covering all classes. Consequently, we assess over 18 times more samples than those used in the baseline for evaluation. Additionally, no attempts are made to simultaneously and correctly evaluate the IM1 and MC-Mean metrics on financial datasets. Therefore, we carefully compare the performance of

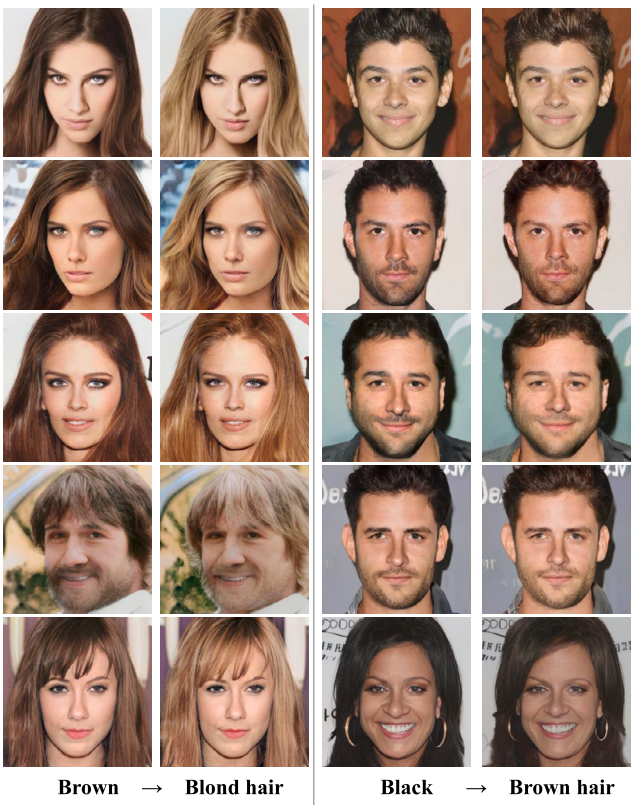

Fig. 8. Samples altering the class to 'Blond' and 'Brown hair' exhibit little correlation with other features. CIRF primarily changes the target feature due to the low relationships between features.

Table 4
Quantitative evaluation of plausibility. 'Ours w/ SVM' and 'Ours w/ St. 1' achieved the highest or performance on IM1 and validity.

Dataset	Method	Validity(↑)	MC-Mean(↑)	MC-STD(↓)	NN-Dist(↓)	IM1-mean(↓)
HELOC	PIECE	0.60	**0.60**	0.49	4.5	**0.99**
	Ours w/ SVM	0.65	0.48	0.66	**4.4**	**0.99**
	Ours w/ St. 1	**0.91**	0.28	**0.27**	**4.4**	**0.99**
MNIST	PIECE	0.84	0.78	0.30	**3.3**	1.05
	Ours w/ SVM	0.95	**0.82**	**0.15**	4.4	**0.54**
	Ours w/ St. 1	**0.99**	**0.82**	0.19	4.3	0.59
CIFAR10	PIECE	0.25	0.27	0.33	**1.34**	1.16
	Ours w/ SVM	0.94	**0.50**	0.37	1.41	**1.0**
	Ours w/ St. 1	**1.0**	0.33	**0.34**	1.4	1.1

our method with that of the baseline, while considering that the baseline study does not focus on tabular datasets. Moreover, the results obtained using the SVM not only demonstrate the CIRF's versatility but also suggest its potential for further enhancement when integrated with other advanced methods. Table 5 presents a comparison of plausibility across selected classes in the image dataset: 'Brown,' 'Black,' 'Blond,' and 'Gray hair.' The classes and datasets are selected on the basis of the baseline study. To mitigate performance variability due to different AE training levels, particularly in complex datasets, we employ three random seeds across 300 samples. As the reconstruction errors of AEs in the IM1 metric are influenced by the training level of AEs, we use three random seeds and calculate the mean. CIRF achieves competitive performance in terms of the IM1 metric when modifying related features.

4.7. Visualization of plausibility

To qualitatively assess our method, plausible CFEs are visualized using widely recognized handwritten digits (Fig. 9), images (Fig. 10), and face datasets (Fig. 11). We collect the samples that the classifier misidentifies to present intuitive examples, as shown in Fig. 9. The F and G use identical architectures and parameters for all methods to ensure fair comparison. 'Ours w/ SVM' and 'Ours w/ St. 1' successfully generate plausible examples that resemble handwritten words. CIRF accurately promotes the positive

Table 5
Plausibility evaluation on image datasets. The average of three evaluations is calculated to account for performance variations due to the extent of AE training.

Dataset	Method	IM1-Mean(↓)
CelebA-HQ	DiVE	1.10
	Ours w/ St. 1	**1.09**

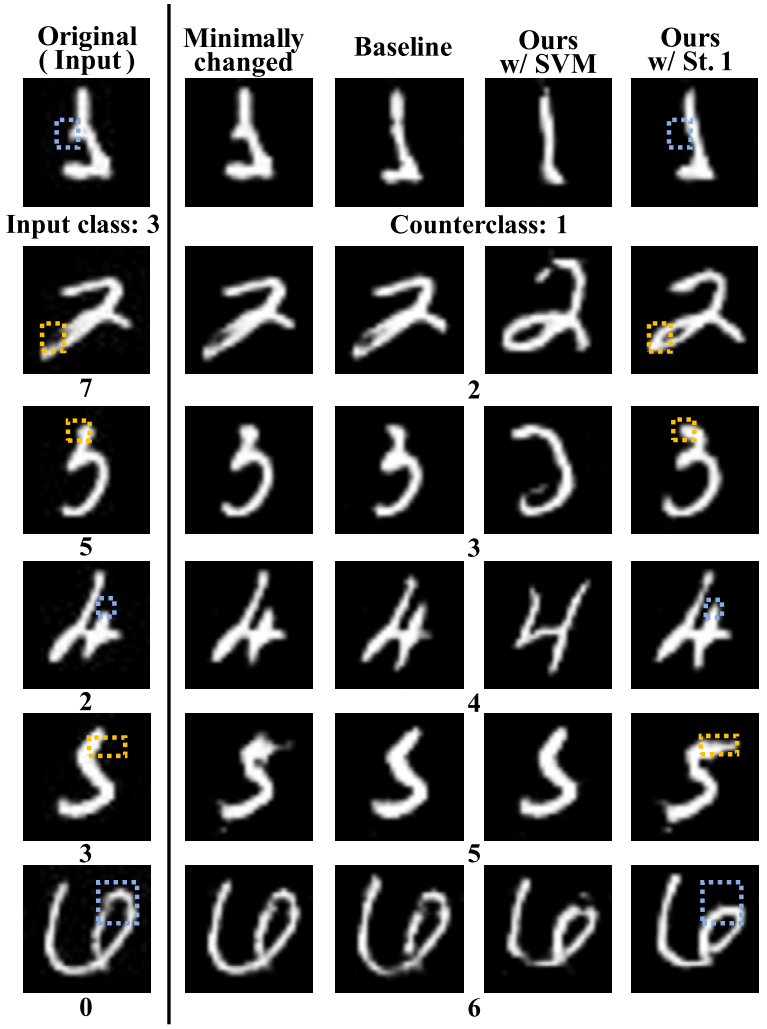

Fig. 9. Comparison of our method with the existing methods for the MNIST dataset. The highlighted yellow (prominent) and blue (suppressed) dotted boxes illustrate critical areas for altering the decision.

area (yellow dotted boxes) and suppresses the negative area (blue dotted boxes) in Fig. 9. $\overline{PP}$ prevents the generation of implausible examples by limiting excessive manipulation. The 'Minimally changed' explanations in the second column provide only slight changes in the input to alter the decision. The 'Minimally changed' explanation is designed to investigate the decision boundary of F by perturbing a marginal amount of the latent code. In addition, the most critical parts are identified through both CFEs: minimally changed explanations and plausible examples ('Ours w/ SVM' and 'Ours w/ St. 1').

Fig. 10 intuitively shows the effect that modifying related features has on improving plausibility and interpreting classifiers in the CIFAR-10 dataset. The leftmost 'Original' column is the original data, and the second 'Inversion' column is data restored through the inversion of the generative model. The more similar these two columns are, the better the inversion is. The third column 'Ours with St. 1' column is the plausible example made by CIRF. As shown in the figure, when changing the 'Airplane' class to 'Horse,' not only the object but also the background changes from air to grass. If generated CFE modifies only the object 'Airplane' to 'Horse,' an

Fig. 10. Examples of related features showing images more plausible.

implausible example ('Horse' floating in the sky) will be created. However, in the real world, images of a 'Horse' floating in the sky are not plausible, and the dataset on which the classifier trained also rarely contains a 'Horse' floating in the sky. Similarly, when changing from 'Ship' to 'Horse,' the sea in the background gradually transforms into grassland.

Fig. 11 shows a comparison with DiVE [47], which utilized autoencoders to create CFEs on a facial dataset. Given that the extent of manipulation is not apparent in a complex dataset, we measure the distances in the latent space below the samples. The CFEs obtained using our method modify related features, such as race when changing from 'Blond' to 'Black' hair. This is attributed to the training dataset for F, which contains a relatively large proportion of races with 'Black' hair. The CFEs with the related features enable users to figure out that race affects the decision of F. Thus, CIRF modifies the related features to alter the class, thereby providing natural and high-quality CFEs compared to the blurrier CFEs produced by DiVE.

Table 6 shows the qualitative results obtained for the HELOC dataset. Several highlighted features are simultaneously modified, whereas unrelated features are preserved. For example, the 'MaxDelqEver' and 'NetFractionInstallBurden' features refer to delayed payments and total assets, respectively. Intuitively, people with numerous assets infrequently make late payments. We consider the 'MaxDelqEver' and 'NetFractionInstallBurden' features to be negatively related. As assets increase, the number of late payments decreases, and vice versa, thus naturally providing plausible CFEs. We exclude two types of outliers from the HELOC dataset. The first outlier data contains a value of -9, and the second is an externally referenced feature, namely 'ExternalRiskEstimate.'

In conclusion, as discussed in Sec. 4.5, the incorporation of related features enhances plausibility and provides a clearer depiction of the decision boundary of the classifier itself, which constitutes our primary focus.

4.8. Real-world samples

To verify the applicability of our method to real-world samples, we apply CIRF to the ImageNet dataset, incorporating the inversion process. We successfully generate CFEs for real-world data by integrating CIRF with the GAN inversion process. We employ

Fig. 11. Comparison of our method with DiVE for the CelebA-HQ dataset. The distances ($D(\vec{z}, \overrightarrow{PP})$ / $D(\vec{z^*}, \overrightarrow{CT_{c'}})$) from the input to $\overrightarrow{CT_{c'}}$ and $\overrightarrow{PP}$ are shown below the images.

the StyleSwin inversion process for ImageNet and use instance-level optimization for the MNIST, CIFAR10, and HELOC datasets [20,35,37]. Although the inverted images are slightly blurrier than the original images, features such as ingredients and texture are well-preserved, as shown in Fig. 12. It should be noted that when an image of 'Carbonara' is changed to that of 'Mashed potatoes,' the texture of carbonara noodles is pasted onto the texture of mashed potatoes, and ingredients are suppressed, which are the related features of 'Carbonara.' As a result, it captures more related features compared to the minimally changed images.

5. Conclusion

This study introduced the CIRF framework to enhance the plausibility of counterfactual explanation (CFE) by defining center-of-target points, direction vectors, and projection-points. Within CIRF, projection-points served as CFEs, resembling the input while altering the classifier's decision. Leveraging the inherent property of generative models, wherein close latent codes generated similar outputs, the framework incorporated both input and class features. Furthermore, the framework's utilization of related features further enhanced CFE plausibility, underscoring the significance of these related features. The given classifiers were successfully utilized to determine the magnitude of manipulation of the latent code by the projection-point, preventing the generation of an implausible example. While the results indicated that a new approach incorporating related features can improve the plausibility of CFE, we acknowledge a disadvantage of the heavy computational costs when estimating data distributions for center-of-target points. Future approaches are anticipated to efficiently capture the related features, with advancements in computational hardware expected to significantly expedite processing speeds. This study employs GANs as the generative model, which introduces a potential limitation since the effectiveness of CFEs is closely tied to the performance of the chosen generative model. Given the rapid advancements in diffusion models, which possess different latent spaces, we designed the framework to enable the integration of alternative generative

Table 6
Additional results for the HELOC dataset. The CFEs are generated using 'Ours w/ St. 1.'

Features	Input_1	CFE_1	Input_2	CFE_2	Input_3	CFE_3
MSinceOldestTradeOpen	143	81	46	273	175	193
MSinceMostRecentTradeOpen	3	4	2	5	6	11
AverageMInFile	54	27	17	119	99	106
NumSatisfactoryTrades	15	0	13	28	24	26
NumTrades60Ever2DerogPubRec	0	2	2	0	0	0
NumTrades90Ever2DerogPubRec	0	1	2	0	0	0
PercentTradesNeverDelq	100	71	92	100	100	100
MSinceMostRecentDelq	-7	52	-7	-7	-7	-7
MaxDelq2PublicRecLast12M	7	0	5	7	7	7
MaxDelqEver	**8**	**2**	**7**	**8**	**8**	**8**
NumTotalTrades	15	4	16	47	25	26
NumTradesOpeninLast12M	1	1	4	4	2	1
PercentInstallTrades	26	86	43	40	56	62
MSinceMostRecentInqexcl7days	5	0	0	2	1	1
NumInqLast6M	0	1	1	1	2	1
NumInqLast6Mexcl7days	0	1	1	1	2	1
NetFractionRevolvingBurden	12	11	53	3	85	13
NetFractionInstallBurden	**65**	**92**	**84**	**64**	**68**	**68**
NumRevolvingTradesWBalance	1	1	6	3	5	6
NumInstallTradesWBalance	3	-8	4	4	3	3
NumBank2NatlTradesWHighUtilization	0	-8	1	0	3	3
PercentTradesWBalance	36	100	89	89	80	79
Labels	1	0	0	1	0	1

models. Future work will extend this verification to contemporary generative models, underscoring the need for continued research into the diverse properties of latent spaces among various generative models.

CRediT authorship contribution statement

Hee-Dong Kim: Conceptualization, Data curation, Formal analysis, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Yeong-Joon Ju:** Conceptualization, Investigation, Supervision, Writing – original draft, Writing – review & editing. **Jung-Ho Hong:** Investigation, Methodology, Project administration, Resources, Software, Writing – review & editing. **Seong-Whan Lee:** Conceptualization, Formal analysis, Supervision, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II190079, Artificial Intelligence Graduate School Program, Korea University and No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation).

Appendix A. Hyperparameters and model architectures

We outline the predefined hyperparameters and model architectures used in the experiments. The filtering rates are shown in Table A.7, and they indicate the extent to which the data used to calculate $\overline{CT}$s, are filtered based on confidence. A low filtering rate implies that the data with a high classification confidence are used to calculate $\overline{CT}$s. The perturbation counts indicate the number of times that the perturbing algorithm is used. As the perturbation magnitude is 0.02, we apply the same number of perturbations to all data. α and β in Eq. (4) are set 0.001 and 0.01, respectively [54].

Fig. 12. Adapting real image data using inversion. The inversion images are generated by the inverted latent code optimized by original images. Minimally changed and 'Ours w/ St. 1' suppress the texture of noodles and alter it to that of mashed potatoes.

Table A.7
Hyperparameters for each dataset.

Hyperparameters	HELOC	MNIST	CIFAR10	CelebA-HQ	ImageNet
Filtering rate	0.2	0.2	0.2	0.2	0.05
Latent dimension	128	100	100	512	512
Perturbation count / magnitude	15 / 0.02	15 / 0.02	15 / 0.02	15 / 0.02	15 / 0.02
α, β	0.1, 0.01	0.1, 0.01	0.1, 0.01	0.1, 0.01	0.1, 0.01
The number of GMM components	5	5	5	5	5
Sampling variances $\Sigma_{\mathcal{Z}}, \Sigma_{\vec{n}}$	1	1	1	1	1

Table A.8 presents the classifiers and generators that were used in the experiment. Each pair of a classifier and generator is trained by the same dataset. The MLP model consists of three linear layers, and the CNN model consists of five convolution blocks. We use the default hyperparameters of CTGAN, which is used to generate tabular datasets. We also use pre-trained generators for complex datasets (CelebA-HQ and ImageNet). Details of experiments can be found at https://github.com/poongi/CIRF_CFE.

Appendix B. Additional qualitative results on each dataset

HELOC We conduct a further qualitative analysis of the HELOC dataset using a probability distribution approach. As shown in Fig. A.13, the actual distribution of highly modified features in the training data is presented in terms of labels. The comparison of

Table A.8
Model setup for each dataset.

Datasets	Classifier	Generator
HELOC [34]	MLP	CTGAN [58]
MNIST [30]	CNN	DCAN [59]
CIFER10 [31]	ResNet-18 [60]	DCGAN
CelebA-HQ [32]	ResNet-18	StyleSwin [37]
ImageNet [33]	ResNet-50	StyleGAN-XL [55]

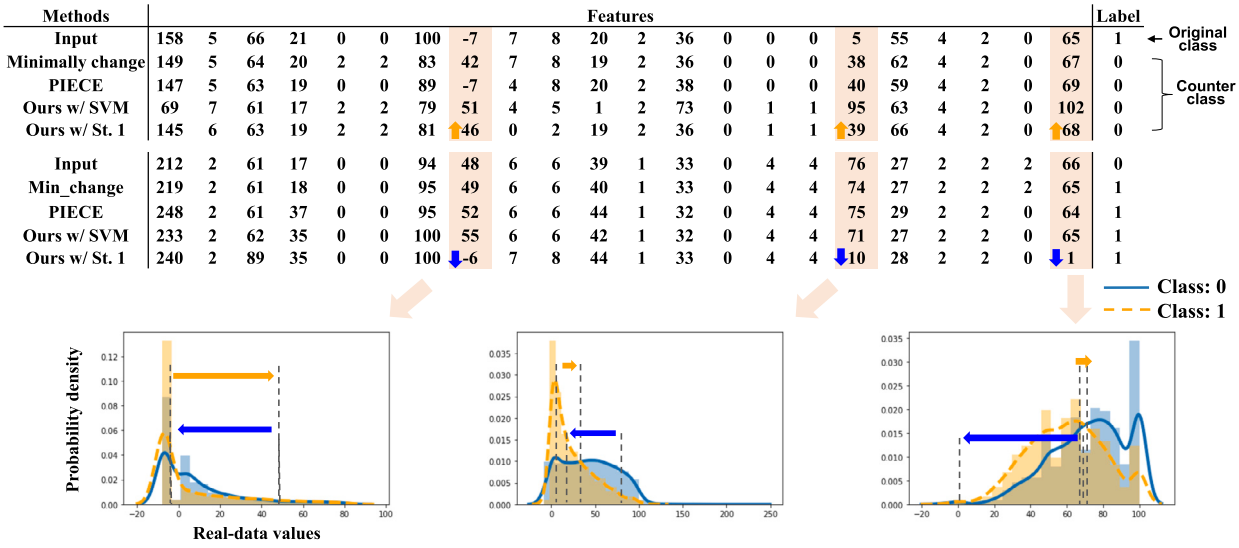

Fig. A.13. Visualization of the distribution of real data. Orange and blue colors represent different classes, respectively, and they are visualized as distributions.

Table C.9
Odds ratio for brown hair class.

Odds of having brown hair given features ($Odds(Brown\|features)$)		Odds ratio
'Male' (0.21)	not 'Male' (0.36)	0.60
'Young' (0.34)	not 'Young' (0.17)	1.94
'Eyeglasses' (0.086)	no 'Eyeglasses' (0.31)	0.27

the input data and CFE shows that the distribution of highly modified features exhibits significant variation among different classes, even in the actual dataset. This demonstrates that the provided CFE reflects the distribution of actual data and generates plausible examples.

Appendix C. Odds ratio for other features

The odds ratio (OR) is a statistical measure used to assess the strength and direction of the association between two binary variables. It quantifies how the odds of one outcome in the presence of a particular condition compare to the odds of the same outcome in the absence of that condition. An OR greater than 1 indicates a positive association, while an OR less than 1 suggests a negative association. An OR equal to 1 implies no association between the variables. Table C.9 and Table C.10 represent the relationships of both 'Brown hair' and 'Blond hair' class with 'Young' and 'Eyeglasses.' As seen in the tables, the OR is relatively not too high or lower than 1, compared to the 'Gray hair' class in Table 3. Consequently, CIRF made relatively minor changes in the features 'Young' and 'Eyeglasses' when altering the class to 'Blond hair' or 'Brown hair.'

Table C.10
Odds ratio for 'Blond hair' class.

Odds of having blond hair given features ($Odds(blond|features)$)		Odds ratio
'Male' (0.02)	not 'Male' (0.35)	0.057
'Young' (0.23)	not 'Young' (0.14)	1.6
'Eyeglasses' (0.047)	no 'Eyeglasses' (0.22)	0.22

Appendix D. Visualizations of the tabular dataset

The distribution of the tabular dataset is visualized to verify whether a manipulation to change the class resides within the distribution of real data. The values that deviate from the distribution of real data are unrealistic and require verification, because this may decrease the plausibility. Fig. A.13 illustrates the distribution of real values for each class, highlighting the features that require substantial modifications to change the class and those that require only minor adjustments. The three highlighted distributions represent the features that have been actively modified in contrast to PIECE. All features remain within the distribution, thus preventing the generation of implausible CFEs. These highlighted features are 'MSinceMostRecentDelq' (last period since delinquency), 'NetFractionRevolvingBurden' (revolving asset), and 'PercentTradesWBalance' (balance transaction), respectively. The differences between classes are also noticeable in the real data distribution for each class. In other words, the highlighted features are important for predicting an individual's credit score. We determined the features that are the primary contributors to changing a classifier's decision and the magnitude of change in these features.

References

[1] T.-C. Hsu, S.-T. Liou, Y.-P. Wang, Y.-S. Huang, et al., Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction, in: IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2019, pp. 1572–1576.
[2] P. Danenas, G. Garsva, Selection of support vector machines based classifiers for credit risk domain, Expert Syst. Appl. (2015) 3194–3204.
[3] T. Davenport, R. Kalakota, The potential for artificial intelligence in healthcare, Future Healthcare J. (2019) 94.
[4] C. Rigano, Using artificial intelligence to address criminal justice needs, Nat. Inst. Just. J. (2019) 17.
[5] R. Garg, V.K. Bg, G. Carneiro, I. Reid, Unsupervised cnn for single view depth estimation: geometry to the rescue, in: European Conference on Computer Vision, Springer, 2016, pp. 740–756.
[6] H. Xu, Y. Gao, F. Yu, T. Darrell, End-to-end learning of driving models from large-scale video datasets, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 2174–2182.
[7] P. Dabkowski, Y. Gal, Real time image saliency for black box classifiers, in: Advances in Neural Information Processing Systems, 2017, pp. 6970–6979.
[8] R.C. Fong, A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2017, pp. 3429–3437.
[9] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS ONE (2015) e0130140.
[10] J.-H. Hong, W.-J. Nam, K.-S. Jeon, S.-W. Lee, Towards better visualizing the decision basis of networks via unfold and conquer attribution guidance, in: Proceedings of the AAAI Conference on Artificial Intelligence, no. 7, 2023, pp. 7884–7892.
[11] H.-G. Jung, S.-H. Kang, H.-D. Kim, D.-O. Won, S.-W. Lee, Counterfactual explanation based on gradual construction for deep networks, Pattern Recognit. (2022) 108958.
[12] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, S. Lee, Counterfactual visual explanations, in: International Conference on Machine Learning, PMLR, 2019, pp. 2376–2384.
[13] Y.-J. Ju, J.-H. Park, S.-W. Lee, Neuroinspect: interpretable neuron-based debugging framework through class-conditional visualizations, preprint, arXiv:2310.07184, 2023.
[14] S.-H. Na, W.-J. Nam, S.-W. Lee, Toward practical and plausible counterfactual explanation through latent adjustment in disentangled space, Expert Syst. Appl. (2023) 120982.
[15] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: automated decisions and the gdpr, Harv. J. Law Technol. (2017) 841.
[16] S. Verma, V. Boonsanong, M. Hoang, K.E. Hines, J.P. Dickerson, C. Shah, Counterfactual explanations and algorithmic recourses for machine learning: a review, preprint, arXiv:2010.10596, 2020.
[17] G. Nápoles, F. Hoitsma, A. Knoben, A. Jastrzebska, M.L. Espinosa, Prolog-based agnostic explanation module for structured pattern classification, Inf. Sci. (2023) 1196–1227.
[18] X. Shao, H. Wang, X. Zhu, F. Xiong, T. Mu, Y. Zhang, Effect: explainable framework for meta-learning in automatic classification algorithm selection, Inf. Sci. (2023) 211–234.
[19] I. Stepin, J.M. Alonso-Moral, A. Catala, M. Pereira-Fariña, An empirical study on how humans appreciate automated counterfactual explanations which embrace imprecise information, Inf. Sci. (2022) 379–399.
[20] E.M. Kenny, M.T. Keane, On generating plausible counterfactual and semi-factual explanations for deep learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 11575–11585.
[21] D. You, S. Niu, S. Dong, H. Yan, Z. Chen, D. Wu, L. Shen, X. Wu, Counterfactual explanation generation with minimal feature boundary, Inf. Sci. (2023) 342–366.
[22] W. Ding, M. Abdel-Basset, H. Hawash, A.M. Ali, Explainability of artificial intelligence methods, applications and challenges: a comprehensive survey, Inf. Sci. (2022).
[23] K. Ahuja, E. Caballero, D. Zhang, J.-C. Gagnon-Audet, Y. Bengio, I. Mitliagkas, I. Rish, Invariance principle meets information bottleneck for out-of-distribution generalization, in: Advances in Neural Information Processing Systems, 2021, pp. 3438–3450.

[24] S. Sagawa, P.W. Koh, T.B. Hashimoto, P. Liang, Distributionally robust neural networks for group shifts: on the importance of regularization for worst-case generalization, in: International Conference on Learning Representations, 2020.
[25] S. Wu, M. Yuksekgonul, L. Zhang, J. Zou, Discover and cure: concept-aware mitigation of spurious correlation, in: Proceedings of the 40th International Conference on Machine Learning, 2023, pp. 37765–37786.
[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM (2020) 139–144.
[27] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, F. Turini, Factual and counterfactual explanations for black box decision making, IEEE Intell. Syst. (2019) 14–23.
[28] R. Guidotti, A. Monreale, S. Matwin, D. Pedreschi, Black box explanation by learning image exemplars in the latent feature space, in: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2020, pp. 189–205.
[29] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, M. etyniecki, Comparison-based inverse classification for interpretability in machine learning, in: Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations, Springer International Publishing, 2018, pp. 100–111.
[30] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE (1998) 2278–2324.
[31] A. Krizhevsky, Learning multiple layers of features from tiny images, https://api.semanticscholar.org/CorpusID:18268744, 2009.
[32] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2015.
[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[34] FICO, Explainable machine learning challenge, FICO Commun. (2017).
[35] A. Creswell, A.A. Bharath, Inverting the generator of a generative adversarial network, IEEE Trans. Neural Netw. Learn. Syst. (2019) 1967–1974.
[36] Y. Shen, J. Gu, X. Tang, B. Zhou, Interpreting the latent space of gans for semantic face editing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9243–9252.
[37] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, B. Guo, Styleswin: transformer-based gan for high-resolution image generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11304–11314.
[38] A.C. Bueff, M. Cytryński, R. Calabrese, M. Jones, J. Roberts, J. Moore, I. Brown, Machine learning interpretability for a stress scenario generation in credit scoring based on counterfactuals, Expert Syst. Appl. (2022) 117271.
[39] A.-H. Karimi, G. Barthe, B. Balle, I. Valera, Model-agnostic counterfactual explanations for consequential decisions, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 895–905.
[40] A. Lucic, H. Oosterhuis, H. Haned, M. Rijke, Focus: flexible optimizable counterfactual explanations for tree ensembles, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 5313–5322.
[41] W.-J. Nam, S. Gur, J. Choi, L. Wolf, S.-W. Lee, Relative attributing propagation: interpreting the comparative contributions of individual units in deep neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 2501–2508.
[42] H. Maeng, S. Liao, D. Kang, S.-W. Lee, A.K. Jain, Nighttime face recognition at long distance: cross-distance and cross-spectral matching, in: Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part II 11, Springer, 2013, pp. 708–721.
[43] M.-C. Roh, T.-Y. Kim, J. Park, S.-W. Lee, Accurate object contour tracking based on boundary edge selection, Pattern Recognit. 40 (3) (2007) 931–943.
[44] M. Augustin, V. Boreiko, F. Croce, M. Hein, Diffusion visual counterfactual explanations, in: Advances in Neural Information Processing Systems, 2022.
[45] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, P. Das, Explanations based on the missing: towards contrastive explanations with pertinent negatives, in: Advances in Neural Information Processing Systems, 2018.
[46] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations, 2015.
[47] P. Rodriguez, M. Caccia, A. Lacoste, L. Zamparo, I. Laradji, L. Charlin, D. Vazquez, Beyond trivial counterfactual explanations with diverse valuable explanations, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1056–1065.
[48] R.R. Fernández, I.M. de Diego, J.M. Moguerza, F. Herrera, Explanation sets: a general framework for machine learning explainability, Inf. Sci. (2022) 464–481.
[49] H. Meng, C. Wagner, I. Triguero, Explaining time series classifiers through meaningful perturbation and optimisation, Inf. Sci. (2023) 119334.
[50] D. Mahajan, C. Tan, A. Sharma, Preserving causal constraints in counterfactual explanations for machine learning classifiers, preprint, arXiv:1912.03277, 2019.
[51] K. Kanamori, T. Takagi, K. Kobayashi, Y. Ike, Counterfactual explanation trees: transparent and consistent actionable recourse with decision trees, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 1846–1870.
[52] J. Del Ser, A. Barredo-Arrieta, N. Díaz-Rodríguez, F. Herrera, A. Saranti, A. Holzinger, On generating trustworthy counterfactual explanations, Inf. Sci. 655 (2024) 119898.
[53] A. Van Looveren, J. Klaise, Interpretable counterfactual explanations guided by prototypes, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 650–665.
[54] A. Abid, M. Yuksekgonul, J. Zou, Meaningfully debugging model mistakes using conceptual counterfactual explanations, in: International Conference on Machine Learning, PMLR, 2022, pp. 66–88.
[55] A. Sauer, K. Schwarz, A. Geiger, Stylegan-xl: scaling stylegan to large diverse datasets, https://arxiv.org/abs/2201.00273, 2022.
[56] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4401–4410.
[57] M. Pawelczyk, C. Agarwal, S. Joshi, S. Upadhyay, H. Lakkaraju, Exploring counterfactual explanations through the lens of adversarial examples: a theoretical and empirical analysis, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 4574–4594.
[58] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional gan, in: Advances in Neural Information Processing Systems, 2019.
[59] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, in: International Conference on Learning Representations, 2016.
[60] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

Hee-Dong Kim received the B.S. degree in electronic engineering from Soongsil University, Seoul, South Korea, in 2012. He is currently pursuing the Ph.D. degree with the Department of Artificial Intelligence, Korea University, Seoul, South Korea. His current research interests include computer vision and explainable artificial intelligence.

Yeong-Joon Ju received the B.S. degree in computer engineering from Sejong University, Seoul, South Korea, in 2020. He is currently pursuing the Ph.D. degree with the Department of Artificial Intelligence, Korea University, Seoul, South Korea.

Jung-Ho Hong received a B.S. degree in aero-software engineering from Hanseo University, Taean, South Korea, in 2019, and is a student of Dept. of Artificial Intelligence in Korea University, Seoul. His current research of interests includes machine learning, explainable artificial intelligence, and computer vision, and studying explainable artificial intelligence in a vision domain.

Seong-Whan Lee (Fellow, IEEE) received the B.S. degree in Computer Science and Statistics from Seoul National University, South Korea, in 1984, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, South Korea, in 1986 and 1989, respectively. He is currently the Head of the Department of Artificial Intelligence, Korea University, Seoul. His current research interests include artificial intelligence, pattern recognition, and brain engineering. He is a Fellow of the International Association of Pattern Recognition and the Korea Academy of Science and Technology.