International Journal of Data Science and Analytics (2024) 17:13–38
https://doi.org/10.1007/s41060-022-00365-6

REGULAR PAPER

CARE: coherent actionable recourse based on sound counterfactual explanations

Peyman Rasouli[1] · **Ingrid Chieh Yu**[1]

Received: 25 November 2021 / Accepted: 20 September 2022 / Published online: 30 September 2022

Abstract
Counterfactual explanation (CE) is a popular post hoc interpretability approach that explains how to obtain an alternative outcome from a machine learning model by specifying minimum changes in the input. In line with this context, when the model's inputs represent actual individuals, actionable recourse (AR) refers to a personalized CE that prescribes feasible changes according to an individual's preferences. Hence, the quality of ARs highly depends on the soundness of underlying CEs and the proper incorporation of user preferences. To generate sound CEs, several data-level properties, such as proximity and connectedness, should be taken into account. Meanwhile, personalizing explanations demands fulfilling important user-level requirements, like coherency and actionability. The main obstacles to inclusive consideration of the stated properties are their associated modeling and computational complexity as well as the lack of a systematic approach for making a rigorous trade-off between them based on their importance. This paper introduces CARE, an explanation framework that addresses these challenges by formulating the properties as intuitive and computationally efficient objective functions, organized in a modular hierarchy and optimized using a multi-objective optimization algorithm. The devised modular hierarchy enables the arbitration and aggregation of various properties as well as the generation of CEs and AR by choice. CARE involves individuals through a flexible language for defining preferences, facilitates their choice by recommending multiple ARs, and guides their action steps toward their desired outcome. CARE is a model-agnostic approach for explaining any multi-class classification and regression model in mixed-feature tabular settings. We demonstrate the efficacy of our framework through several validation and benchmark experiments on standard data sets and black box models.

Keywords Interpretable machine learning · Actionable recourse · Counterfactual explanations · Black box models · Multi-objective optimization

1 Introduction

Nowadays, black box machine learning (ML) models (e.g., deep neural networks) are widely used in high-stake applications that often impact human lives, for example, criminal justice [21], clinical healthcare [3], credit approval [38], to name a few. As a result, interpretability, an essential requirement for transparent decision-making, has become a crucial challenge in black box ML models. Beyond understanding *why* a model made an undesired prediction [34,35], it is beneficial to know *how* to obtain a desired decision. Counterfactual Explanations (CE) are a popular class of post hoc explanation methods that aim to answer the latter question [43]. The explanations describe what changes in the input would have resulted in an alternative decision, represented as:

> If feature(s) **X** had the value(s) of $\mathbb{V}$, then the outcome would have been $\mathbf{Y}'$ rather than **Y**.

Formally, CEs identify minimal changes in the input features that lead to the desired outcome from the model [45]. Consider a house pricing classifier as an example; a possible explanation for an instance predicted as **Low-cost** can be:

> If **area** had the value of `400 meters`, then the outcome would have been **High-cost** rather than **Low-cost**.

✉ Peyman Rasouli
peymanra@ifi.uio.no

Ingrid Chieh Yu
ingridcy@ifi.uio.no

[1] Department of Informatics, University of Oslo, Gaustadalléen 23B, 0373 Oslo, Norway

In addition to the main utility of counterfactual explanations, which is providing guidelines to change a model's predictions, they can be quite helpful in increasing the model's transparency. First, a prescribed list of modifications somehow identifies the most influencing features for current and alternative predictions of the input [27]. Second, CEs can expose potential discrimination issues in the model [37]. For example, in a human-related application, if generated explanations for a specific group of individuals mostly recommend altering sensitive features (like **race** or **gender**), it reveals the model's bias toward the studied group. Finally, we can use CEs to measure the robustness of different ML models and select the most robust one for our application [37]. Precisely, the average distance between original data points and their corresponding counterfactuals can demonstrate the resistance of a model concerning perturbations.

In line with the context of counterfactual explanations, when the model's inputs represent actual individuals, actionable recourse (AR) refers to personalized CEs that prescribe feasible changes or actions according to individuals' preferences [19]. For example, consider a person who has been denied a loan by a bank's decision system and seeks an explanation on how to proceed. A counterfactual explanation for this scenario demonstrates how to obtain the desired outcome, however, the recommended changes may not necessarily be actionable from the individual's standpoint. For instance, a possible CE may be like this:

> If **balance** had the value of `$50000`, then the outcome would have been **Granted** rather than **Rejected**.

Although this explanation seems plausible, it will not be helpful if the individual can not increase the **balance** to `$50000`. Hence, to generate an *actionable* list of changes for an individual, it is essential to effectively capture their preferences and incorporate them into the AR generation process [43]. Individuals should be able to define various constraints and prioritize them according to their circumstances. Besides, the defined constraints need to be considered within the explanation generation process, instead of a post hoc filtering step, to improve the probability of obtaining a recourse that satisfies an individual's desires [41]. Generating *diverse* recourses for every input is another requirement for enhancing the actionability of explanations, enabling individuals to choose the most suitable action list according to their situation [19].

An AR is often seen as instantaneous and atomic changes to the input features that lead to the desired outcome. But, in reality, actions are usually sequential and have a temporal ordering [43]. It would be helpful to guide the users on accomplishing the recommended actions by the recourse. For example, alongside an AR that recommends *"increase **balance** and promote **occupation**,"* providing a sequence of actions like first improve the job and then increase the balance, i.e., **occupation** → **balance** can increase the utility of explanations. Such a sequence prevents creating inconsistent or impossible states for the individual during the transition to the recourse state.

Considering that CEs constitute the basis of ARs, generating valid and statistically sound CEs is of paramount importance for obtaining feasible ARs. Here, validity refers to a counterfactual instance that provides the *desired outcome* and applies *minimum changes* to the original input [45]. Meanwhile, the soundness indicates a counterfactual data point originating from the original distribution of the data (known as *proximity*) and linking to the existing knowledge of the domain represented by the ground-truth data (known as *connectedness*) [23]. As a result, its features have plausible values and compose a fairly consistent combination, similar to the observed data. While proximity and connectedness are two different notions, the state-of-the-art explanation methods often consider proximity as the only soundness requirement. The likely reasons for neglecting connectedness are the modeling and computational complexities [24].

Although fulfilling the soundness property results in a counterfactual data point following the original data distribution, it may not fully preserve the *coherency* between feature values. This is because of the existence of feature values that have a similar distance to the original input's value and influence the model's predictions similarly. Density-based methods [6,30,42], which consider proximity and/or connectedness properties, do not differentiate between such similar values as the resultant counterfactual still lies fairly close to the overall training distribution and meets the soundness requirement. However, this is a coarse approach to modifying a feature that can lead to inconsistent explanations due to disregarding the correlation and conformity with other features. For example, in the *Adult* data set [11], the two values {`wife`, `husband`} for the **relationship** feature affect the prediction of the model similarly and their distance to the other possible values for the feature is identical as **relationship** is a categorical feature. In this case, a sound explanation will not necessarily distinguish between {`wife`, `husband`} as selecting either value fulfills the soundness property. However, from the consistency perspective, the correct recommended value for **relationship** should be chosen according to the **gender** of the individual.

It can be seen that the coherency property is a complement to the soundness property, and it is especially important when individual preferences are in play. Generally, a user specifies constraints for some features regardless of the status of other features. This is the algorithm's responsibility to ensure the generated explanation has established the consistency between changed and unchanged features. Therefore, preserving the coherency between features is crucial for creating an actionable recourse that provides an individual with consistent changes.

Generating ARs possessing the stated properties is challenging. It requires inclusive consideration of data-level (e.g., proximity and connectedness) and user-level (e.g., coherency and actionability) properties properly and in an efficient manner. In this paper, we propose a method for generating **C**oherent **A**ctionable **R**ecourse based on sound counterfactual **E**xplanations (**CARE**). Our approach creates coherent ARs on the grounds of sound CEs, a prerequisite for the feasibility of recommendations. In CARE, properties are formulated as objective functions, organized in a modular hierarchy, and optimized using non-dominated sorting genetic algorithm III (NSGA-III) [7]. The modular structure of CARE enables effective arbitration and aggregation between the desired properties. Moreover, the chosen multi-objective optimization algorithm overcomes the computational challenge of addressing a complete set of properties by making a rigorous trade-off between their corresponding objective functions.

As mentioned earlier, the quality of ARs is highly dependent on how well individuals' preferences are perceived and utilized in the explanation framework. On that account, CARE provides a flexible constraint language for the users to express and prioritize their desires. Moreover, it supplies every individual with a diverse set of ARs to choose the most suitable one concerning their situation. Further, to facilitate the user's actions toward the recourse state, CARE provides a *temporal action sequence* alongside every recourse. This sequence guides the individual on how to fulfill the recommended changes without creating inconsistent or impossible states along the way.

The chosen optimization scheme and the designed modular structure contribute to the applicability and flexibility of CARE. First, the optimization algorithm enables explaining any tabular model without requiring access to the model's internals or gradients. It treats the models as black box and only relies on their prediction function for explanation generation (i.e., model-agnostic). Second, the optimization scheme allows defining arbitrary objective functions for the *outcome* property, making our approach applicable for both multi-class classification and regression tasks. Third, it can handle any tabular data set containing numerical and categorical features without demanding feature transformation or defining customized distance functions. Finally, the modular hierarchy allows generating counterfactual explanation and actionable recourse by choice. It also makes our approach a suitable benchmark for techniques addressing similar properties.

To summarize, the main contributions of this work are as follows:

- We formulate data-level and user-level desiderata as intuitive and computationally efficient objective functions and employ the NSGA-III multi-objective optimization algorithm to generate feasible explanations by making a rigorous trade-off between all objectives.
- We propose a flexible and multi-purpose framework that can generate counterfactual explanation and actionable recourse for any multi-class classification and regression model in mixed-feature tabular settings.
- We involve individuals by providing a flexible language for defining preferences, facilitate their choice by recommending various ARs, and guide their action steps toward their desired outcome.
- We demonstrate the importance of different properties in explanation generation and the efficacy of our approach in addressing them through comprehensive validation and benchmark experiments on standard data sets and black box models.
- We provide an open-source implementation of our approach for the research community, accompanying the entire experiments for reproducing the results: https://github.com/peymanrasouli/CARE.

The rest of the paper is organized as follows: Sect. 2 provides a brief description of the actionable recourse properties with an overview of the existing approaches for addressing every property; Sect. 3 further investigates the challenges of current explanation methods and motivates our proposed solutions; Sect. 4 introduces our proposed framework, including property formulation, optimization scheme, and temporal action sequence; Sect. 5 presents the conducted experiments and discussions around the efficacy of our devised approach, including validation and benchmark experiments; Sect. 6 provides a literature review of the state-of-the-art explanation methods; finally, Sect. 7 concludes the paper and states future works.

2 Actionable recourse desiderata

To obtain a realistic and actionable recourse, an inclusive set of data- and user-level desiderata must be considered simultaneously in the explanation generation process. In the following, we discuss these requirements in conjunction with the relevant existing studies that are summarized in Table 1. However, for the readers who are already familiar with these notions can skip this section.

Outcome A counterfactual should be classified as the desired (often opposite) outcome by the ML model. This is considered the primary requirement for generating valid counterfactuals. Generally, methods that explain differentiable models consider this property as a constraint [25,31,45], while black box explanation techniques treat it as an objective function [6,37]. As a result, the former methods may fail to gener-

Table 1 A summary of characteristics and limitations of related works studied for this paper, including explanation properties, model assumptions, and application domains

Method	Explanation properties						Model assumptions		Application domains	
	Sparsity	Proximity	Connectedness	Coherency	Actionability	Diversity	Model Access	Model domain	Prediction task	Data type
CF [45]	L1	No	No	No	No	No	Gradients	Differentiable	Classification	Tabular
DACE [17]	No	Yes	No	Yes	Yes	No	Complete	Linear and tree ensemble	Classification	Tabular
REVISE [16]	No	Yes	Yes	No	No	No	Gradients	Differentiable	Classification	Image and tabular
DiCE [30]	L1 and post hoc filter	Yes	No	No	Post hoc Filter	Yes	Gradients	Differentiable	Classification	Tabular
MOC [6]	Yes	Yes	No	No	Limited	Yes	Black box	Agnostic	Classification and regression	Tabular
AR [41]	Hard constraint	No	No	No	Yes	No	Complete	Linear	Classification	Tabular
CFPrototype [42]	L1	Yes	No	No	No	No	Black box/gradients	Differentiable	Classification	Image and tabular
FACE [32]	No	Yes	Yes	No	No	No	Black box	Agnostic	Classification	Image and tabular
C-CHVAE [31]	No	Yes	Yes	No	Limited	No	Black box	Agnostic	Classification	Tabular
CERTIFAI [37]	No	No	No	No	Yes	Yes	Black box	Agnostic	Classification	Image and tabular
GRACE [25]	Yes	Yes	No	No	No	No	Gradients	Differentiable	Classification	Tabular
FOCUS [27]	L1	No	No	No	Limited	No	Gradients	Tree ensembles	Classification	Tabular
FASTAR [44]	Yes	Yes	No	No	Limited	Yes	Black box	Agnostic	Classification	Tabular

ate valid explanations for some inputs because they rely on *distance* as the optimization criterion.

Distance A counterfactual explanation should provide the desired outcome by making minimum modifications to the original input. The degree of changes is calculated by measuring the distance between original and counterfactual instances in the feature space. The typical distance metrics for numerical features are L1/L2-norm and quadratic, which are widely used by gradient-based methods [45]. However, since most tabular data sets contain both numerical and categorical attributes, such methods demand extra work handling categorical features [43]. On the contrary, techniques based on SMT solver and genetic algorithm can naturally handle mixed-feature data sets [6,18].

Sparsity A counterfactual should provide understandable and straightforward guidelines toward the desired outcome. The understandability and simplicity of explanations can be measured by counting the number of changed features. Thus, an ideal counterfactual should alter a minimum number of features, leading to a sparse change list. It should be noted that *distance* property is not equivalent to *sparsity*, as it does not determine the number of changed features. The majority of existing methods neglect *sparsity* [43], while some works aim to address it using L1-norm (instead of or alongside the L2-norm) as the distance metric [42,45].

Proximity A counterfactual should provide plausible changes to the original input that are in accordance with the observations of the model. For example, *"if* ***balance*** *had a value of* `-$10000`, *then the loan would have been* ***Granted***" is an *outlier* counterfactual because it has recommended an invalid (out of distribution) value for **balance**, which is infeasible and unrealistic. This notion is called *proximity*, which indicates a counterfactual instance should lie in the neighborhood of the ground-truth data [23]. Recent papers have proposed various ways of handling *proximity* such as using local outlier factor [2] as an objective function to prevent outlier counterfactuals [17], employing generative models (e.g., variational auto-encoders) to approximate the data distribution and then sampling realistic counterfactuals [16,28,42], and using a weighted average distance metric between the original input and its K nearest instances in the training data [6,44].

Connectedness A counterfactual should be related to the existing knowledge (represented by the training data) and not a consequence of the model's artifacts. Artifacts (i.e., blind spots) are areas on the decision surface where the model misbehaves and mispredicts the originating samples. They can be created due to a lack of training data for some regions in the feature space or the model's weaknesses. Having an explanation caused by an artifact is not associated with the domain knowledge and is undesirable in the context of interpretability and feasibility [23]. Therefore, it is necessary to create a counterfactual instance that is related to the training data continuously. This notion is called *connectedness*, which implies that a counterfactual should be connected to the same-class data points using a continuous path [23]. Along this path, features change smoothly and coherently, and each instance in the path is correlated with the preceding and succeeding instances. Therefore, this property facilitates the transition from the default state to the counterfactual state and improves the actionability of explanations.

This property is thus complementary to *proximity*, as two instances can be located closely but not necessarily linked through the same-class data points. Without considering *connectedness*, the optimization algorithm will not distinguish between the blind spots and actual decision space of the desired class and may generate a counterfactual that provides the desired label, however, based on the data distribution it is not actually a possible instance in the desired class. This issue is not handled by the *proximity* property which is broadly considered in the literature to create statistically sound explanations [6,16,17,25,30,42,44]. To the best of our knowledge, there are three works that address this property by connecting counterfactuals to original inputs via high-density paths in the data manifold [32], high-probability paths in the latent representation of the data distribution [16], and high-density regions in the feature space modeled by an autoencoder [31].

Coherency A counterfactual should preserve the correlation among features to create a consistent explanation. Density-based approaches that create sound explanations by fulfilling either the *proximity* or *connectedness* requirement, can fail to capture subtle correlations between features, especially for the scenarios in which some feature values have a similar distance to the original input's value and influence the model's predictions similarly. Consider the *Adult* data set [11] as an example in which two features **relationship** and **gender** are highly correlated with each other. The possible values for **relationship** are {`unmarried`, `wife`, `husband`} while **gender** can take values from {`male`, `female`}. For a `male` and `unmarried` individual who is predicted as $Income \leq$ **50K**, a likely counterfactual explanation may be a change in the **relationship** feature, as based on the data distribution married people tend to have $Income >$ **50K**. In this case, since two values {`wife`, `husband`} affect the prediction of the model in a similar way and since their distance to the original value (i.e., `unmarried`) is equal as they are categorical values, a sound explanation will not necessarily distinguish between {`wife`, `husband`} as selecting either value fulfills the soundness property. However, according to the gender of the individual, which is `male`, a valid recommended value for **relationship** should be `husband` which results in consistency among the features.

We call this notion *coherency*, which is a complement for the soundness property that considers features' correlations for selecting suitable values for the changed features according to the status of other features, leading to a consis-

tent state for the counterfactual explanation. Considering this property is even more essential when an individual imposes constraints over some features regardless of the status of other features. In this case, the counterfactual needs to implicitly preserve the consistency between changed and unchanged features. Although there are several works around creating statistically sound explanations [6,16,25,32,42], to the best of our knowledge, *coherency* has been only investigated in [17] in which authors emphasize the importance of feature correlation and propose a cost function based on the Mahalanobis distance and the local outlier factor model that is only applicable to white box linear and tree ensemble classifiers.

It should be noted that *coherency* is different than the *causality* concept that has been studied in some recent works [28,44]. While both concepts are being used for establishing feature consistency and seemingly overlapping, *coherency* and *causality* are two distinct notions [1]. The former indicates correlation, a statistical relationship between two variables determining how they vary together; the covariation has positive/negative and weak/strong properties and is not necessarily due to a direct or indirect causal link between the variables. Whereas the latter refers to causation, a special type of relationship between correlated variables meaning changes in one variable cause the other one to respond accordingly (cause and effect); in other words, the two variables are correlated with each other and there is also a causal link between them. Therefore, all causations are correlations, but not all correlations are causations. For example, the relationship between **education** and **income** is correlation and not causation. Although there is a strong and positive correlation between these variables, it does not necessarily imply that "*higher* **education**" causes "*higher* **income**" because there may be other causal factors like **occupation** and **working hours**.

Actionability A counterfactual should satisfy some user-defined global and local constraints to create a personalized recourse that recommends actionable changes to the user. Global constraints need to be satisfied by every counterfactual, for instance, fixing immutable features like **race** and **gender**. In contrast, local constraints are tailored to every single individual, for example, a possible range of value for feature **balance** for an individual may be [$3000, $6000] while for another one may be [$5000, $10000]. This notion is called *actionability*, implying a recourse that meets the user's preferences [41]. Some recent works allow defining preferences only in the form of immutable and mutable feature sets [6,20,30,44], while others accept a range/set of values for numerical and categorical features [37,41]. Another important criterion to categorize the existing works regarding *actionability* is how they incorporate preferences in the recourse generation. Accordingly, some techniques address this property either as a constraint in the optimization process [6,16,37] or as a post hoc filtering step after the explanation generation [10,30]. Algorithms falling in the former category are more likely to generate a recourse that fulfills an individual's preferences.

Diversity A diverse set of counterfactuals for a particular input should be generated. Often, there are several paths (i.e., recourses) that provide the desired decision of an individual. Having multiple explanations is useful in the sense of actionability as it allows the individual to select the most appropriate solution according to their circumstance. There are different ways for generating diverse explanations, like adding an optimization objective to maximize the distance between multiple counterfactuals [6,30], imposing a hard constraint [18,41], and minimizing the mutual information between all pairs of modified features [25].

3 Motivation

In reviewing state-of-the-art CE and AR generation methods, we observed several unexplored and under-explored research areas that are motivating this work:

- The *sparsity* property is either neglected or addressed partially by using L1-norm as the distance metric. Although this metric can reduce the number of changes to some extent, it is necessary to consider *sparsity* as a separate objective goal to generate simple and intuitive explanations.
- The *proximity* property is often interpreted as finding counterfactuals that are connected to the entire training data. Indeed, this can be problematic, because a sparse counterfactual instance (changes in one/two features) lies fairly close to the overall training data, however, it can be out-of-distribution with respect to a subset of data that share the same values as the changed features and belong to the same class [42]. Ideally, a counterfactual instance should be an inlier with respect to a subset of similar data points that belong to the same class as the counterfactual.
- Compared to *proximity*, the *connectedness* property has received little attention. A likely reason for neglecting connectedness as an objective goal in the explanation generation process can be its computational burden. Moreover, many works use proximity as an interchangeable property for connectedness, which is arguable, as proximity prevents generating outlier counterfactuals while connectedness results in counterfactuals that are connected to the existing domain knowledge. Disregarding connectedness results in instances being created from areas where the model has no information about (artifacts) and makes questionable improvisations (decisions) [24]. In fact, this is problematic in terms of interpretability and feasibility of explanations. Therefore, both proximity and connectedness properties are

necessary for deriving statistically sound counterfactual explanations.

- Often, there are strong correlations between some features in the data that demand coherent value allocation, like the correlation between **relationship** and **gender** features in the *Adult* data set [11]. The existing works have not explicitly studied the *coherency* between counterfactual features. Although soundness properties (like *proximity* and *connectedness*) and correlation-preserving distance functions (like Mahalanobis) can preserve feature consistency to some extent, they may fail to capture the subtle correlation between features due to the existence of feature values that have a similar distance to the original input's value and influence the model's predictions similarly.

  Using domain knowledge represented as a structural causal model (SCM), which describes the causal relationships between features in a data set, one can establish feature consistency [28]. However, this approach has several problems; first, proper domain knowledge is not available for the majority of the ML data sets, and creating one can be highly expensive; second, the domain knowledge may not include complex causal relationships between several features; and third, two correlated features may change together not necessarily because one causes the other one to change (cause and effect), but maybe there is another causal factor (confounding variable) that affects both.

  As an alternative solution for establishing feature consistency, one can formulate unary and binary constraints by knowing the relationship of the domain's features [44]. For example, "**age** *cannot decrease*" and "*increasing* **education** *increases* **age**." However, identifying complex relationships between multiple features and encoding them as constraints is challenging, especially in applications with high-dimensional feature space and high-order relationships unknown to the user.

  Therefore, it would be beneficial to establish value consistency for highly correlated features in an autonomous way by merely relying on the observational data and its contained correlation information that can be derived using statistical approaches like Spearman's Rho [5]. Such an approximate data-driven method can identify complex relationships between several features, is applicable to any data set, and does not require rarely available domain knowledge. Fulfilling *coherency* is even more important when user preferences come into play. Often, a user defines constraints for some features irrespective of the status of others. When an algorithm only satisfies the specified constraints and neglects their consistency with other features, it will create an unrealistic and impractical state for the individual. Hence, *coherency* should be considered as a prerequisite property for producing actionable recourse.
- There is a lack of a framework to capture individuals' preferences precisely and incorporate them into the AR generation process. An individual should be able to define various constraints over input features, rather than categorizing them into immutable and mutable groups. Moreover, the framework should allow prioritizing different constraints. For example, two constraints, "*fix the* ***occupation***" and "*constrain the* ***balance*** *between* [`$5000, $10000`]" have different importance, as exceeding the balance range may be acceptable, but changing the occupation can be costly. In the absence of constraint importance, the optimization algorithm makes no difference between preferences that can lead to ineffective recommendations. Besides, it is essential to utilize the specified preferences within the AR generation mechanism instead of using them for filtering the created ARs. As in the latter case, there may not exist a recourse in the solution set that satisfies the preferences, leading to a null set of results. Furthermore, an AR is often seen as instantaneous and atomic changes to the input features that lead to the desired outcome. But, in reality, actions are usually sequential and have a temporal ordering. Thus, it is important to provide a sequence of steps that guides the individual toward the recourse state without creating inconsistent or impossible states along the way.
- Above all, the state-of-the-art methods usually address a subset of the stated desiderata in Sect. 2. High computational cost, the complexity of modeling, and misleading similarity are the main barriers in handling an inclusive set of properties. As a result, the generated explanations may fail to fulfill some necessary properties. This can diminish the utility of explanations for real-world applications, especially in settings that humans are the consumers of results. Moreover, most existing works optimize a weighted sum of multiple objectives (i.e., properties) that may not be a rigorous way of specifying the importance of properties for the generated explanations. Consequently, it can lead to explanations biased toward satisfying only some specific properties. Therefore, it would be better if the optimization algorithm discovers a trade-off between the objectives based on more explicit information from the user (like a hierarchical ordering of properties for describing their relative importance).

4 The framework of CARE

This section formalizes the problem of counterfactual explanations and describes our proposed framework, called CARE,

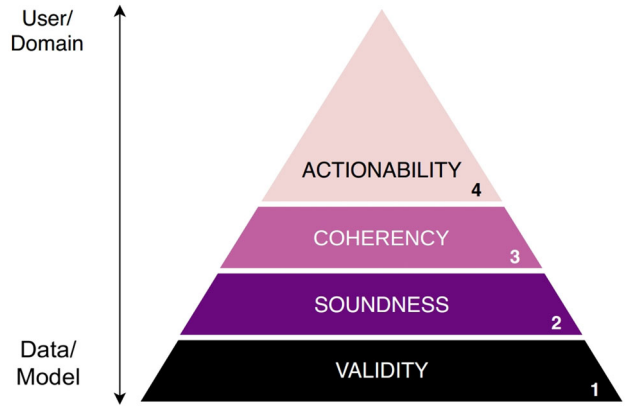


Fig. 1 The framework of CARE

for generating actionable recourse, personalized counterfactuals according to individuals' preferences. As we explained earlier, every property plays a unique and crucial role in creating sound and actionable explanations. Hence, we formulate an inclusive set of properties as objective functions and organize them in a modular hierarchy. Figure 1 illustrates the framework of CARE. It consists of four modules including VALIDITY, SOUNDNESS, COHERENCY, and ACTIONABILITY, each containing some specific properties. The low-level modules handle model- and data-related properties to create valid and statistically sound explanations, while the high-level ones personalize the explanations by establishing the consistency between individuals' attributes and incorporating their preferences in the recourse. The VALIDITY module acts as a basis for other modules, and explanations can be generated regardless of the presence of other modules. In other words, it is possible to include or exclude desired modules in the hierarchy.

The devised framework is advantageous from several perspectives. First, the hierarchical structure ranks the generated explanations with respect to the fitness of properties defined in the modules in a consecutive bottom-up order. For example, first, explanations are ranked based on the objectives in the VALIDITY module to return counterfactuals similar to the original input that provide the desired outcome, then based on the SOUNDNESS module to prioritize statistically sound explanations, and so on. This sequential consideration of properties is useful, especially in the context of AR, because it helps provide a recourse that has fulfilled fundamental properties (like soundness and coherency) before satisfying user preferences. Second, the possibility of adding arbitrary modules in the hierarchy makes our approach applicable for both counterfactual explanation and actionable recourse problems. Last but not least, the modular hierarchy enables us to scrutinize the impact of distinct properties in the quality of explanations and qualifies our framework as a benchmark for explanation methods addressing similar properties.

To provide an insight on how different properties affect the generated counterfactual explanations, we explain an instance of the *Adult* data set [11], denoted as x, using incremental combinations of CARE's modules. For the sake of simplicity, we used a subset of features and reported the results in Table 2. It can be seen that the valid counterfactual (x'_{Valid}) is a minimally modified variant of the original input that only provides the desired outcome irrespective of feasibility and actionability of changes. In contrast, the sound counterfactual (x'_{Sound}) lies on the data manifold and appears more realistic, however, it has not established the correlation between features perfectly (the values of features **relationship** and **sex** do not conform with each other). The coherent counterfactual (x'_{Coherent}) resolves the consistency shortcoming of the sound counterfactual and creates a coherent state for the input features (**relationship** and **sex** features are consistent). Nevertheless, a statistically sound and coherent counterfactual may not be actionable from the user's perspective (changing the feature **sex** is impractical). Hence, we need to take into account the user's preferences containing the name of mutable/immutable features, their possible values, and their importance to create a realistic and actionable recourse ($x'_{\text{Actionable}}$).

Before diving into details, we need to clarify that CARE consists of two main phases: *fitting* and *explaining*; the former creates an *explainer* based on the training data and the ML model, while the latter generates counterfactuals for every input instance using the created *explainer*. In the following, we describe the CARE's modules and the optimization algorithm for generating explanations.

4.1 Valid counterfactual explanations

We define a valid counterfactual explanation as an instance similar to the original input with minimum changes to features that results in the desired outcome. Consider (X^m, Y) as a data set, where X^m is an m dimensional feature space and Y is a target set. Let f be a black box model trained on (X^m, Y) that maps an input to a target, i.e., $f : X^m \rightarrow Y$. Targets can be either discrete classes or continuous values depending on the prediction task (classification or regression). For a given input x with $f(x) = y$ and a desired outcome y', our goal is to find a counterfactual x' as close to x as possible such that $f(x') = y'$. The VALIDITY module addresses three fundamental properties for counterfactual explanations: *outcome*, *distance*, and *sparsity*. In the following, We define three cost functions for the stated properties that are minimized during the optimization process.

Table 2 Explanations generated using incremental combinations of CARE's modules for an instance from the *Adult* data set

EXAMPLE	Age	Capital gain	Capital loss	Hours-per-week	Education	Occupation	Relationship	Sex	Native country	*Income*
x	25	0	0	40	Bachelors	Sales	Not-in-family	Female	United States	**≤50K**
x'_{Valid}	–	7688	–	–	–	–	–	–	–	**> 50K**
x'_{Sound}	–	–	–	50	–	Exec-managerial	Husband	–	–	**> 50K**
x'_{Coherent}	–	–	–	60	–	Transport-moving	Husband	Male	–	**> 50K**
$x'_{\text{Actionable}}$	30	–	–	45	Masters	Prof-specialty	–	–	–	**> 50K**

The results show the impact of different properties on the quality of explanations

4.1.1 Outcome

We evaluate the prediction of f for a generated counterfactual x' with respect to the desired outcome. Relying on the black box prediction function allows us to define different measurements for the *outcome* property. For the classification task, we use the Hinge loss function [9]:

$$O^{C}_{\text{outcome}}(x', c, p) = \max(0, (p - f_c(x'))) \tag{1}$$

where $f_c(x')$ is the prediction probability of x' for the desired class c and p is a probability threshold that leads to a counterfactual with a desired level of confidence. This cost function considers all counterfactuals above the threshold p as valid counterfactuals. The *outcome* property for the regression task is defined as follows:

$$O^{R}_{\text{outcome}}(x', r) = \begin{cases} 0, & \text{if } f(x') \in r \\ \min\limits_{r' \in r} |f(x') - r'|, & \text{otherwise} \end{cases} \tag{2}$$

where $f(x')$ is the predicted response for the counterfactual and $r = [lb, ub]$ is a desired response range. The devised cost function considers any predicted response within the range r as valid (zero cost), otherwise the absolute distance between the prediction and the closest bound (lb or ub) is considered as the cost of x'.

4.1.2 Distance

We employ Gower's distance [14] to calculate the distance between features in a mixed-feature setting (i.e., data set contains both categorical and numerical features). For numerical features like **age**, Gower's distance applies range-normalized Manhattan distance, while for categorical features like **education**, it uses the Dice coefficient. Given an original input x and a counterfactual instance x', we measure their distance by calculating the overall difference between their feature values:

$$O_{\text{distance}}(x, x') = \frac{1}{m} \sum_{j=1}^{m} \delta(x_j, x'_j) \tag{3}$$

where m is the number of features, and δ is a metric function that returns a distance value depending on the type of the feature:

$$\delta(x_j, x'_j) = \begin{cases} \dfrac{1}{R_j} |x_j - x'_j|, & \text{if } x_j \text{is numerical} \\ \mathbb{I}_{x_j \neq x'_j}, & \text{if } x_j \text{is categorical} \end{cases} \tag{4}$$

where R_j is the value range of feature j that is extracted from the training data, and $\mathbb{I}$ is a binary indicator function, returning 1 if compared values are different, otherwise 0.

4.1.3 Sparsity

To have a simple and intuitive explanation, a counterfactual should alter a minimum number of features. Minimum feature distance is not equivalent to the minimum number of changed features. Therefore, we define the cost function O_{sparsity} for counting the number of altered features:

$$O_{\text{sparsity}}(x, x') = \sum_{j=1}^{m} \mathbb{I}_{x_j \neq x'_j} \tag{5}$$

where m is the number of features in the data set and $\mathbb{I}$ is a binary indicator function, returning 1 if compared values are different, otherwise 0. This function penalizes every change in features regardless of their type (i.e., categorical and numerical) and the magnitude of change.

4.2 Sound counterfactual explanations

As we mentioned earlier, a sound counterfactual instance should originate from the observed data (*proximity*) and connect to the existing knowledge via a continuous path (*connectedness*). Meeting these two conditions results in an inlier instance that provides interpretable and feasible explanation. Hence, the SOUNDNESS module addresses the *proximity* and *connectedness* properties via the following fitness functions that are maximized during the optimization process.

4.2.1 Proximity

Proximity indicates that the counterfactual instance lies in the neighborhood of the ground-truth samples that are predicted correctly by the model and have the same target value as the counterfactual. We utilize the proximity evaluation metric introduced in [23] as an objective function for counterfactual generation. Let x be our original input and x' be a generated counterfactual. We refer to $X^{f(x')}$ as the set of instances in the data set that are predicted correctly by f and belong to the same class (in classification task) or response range (in regression task) as x'. Consider $a_0 \in X^{f(x')}$ is the closest instance to x', i.e., $a_0 = \arg\min_{a_i \in X^{f(x')}} D(x', a_i)$, where D is a distance metric (e.g., Minkowski). The counterfactual x' fulfills the proximity criterion if it has the same distance to a_0 as a_0 has to the rest of the data ($X^{f(x')}$). The formal definition of proximity is as follows:

$$proximity(x') = \frac{D(x', a_0)}{\min_{a_i \in X^{f(x')}} D(a_0, a_i)} \tag{6}$$

A lower value of $proximity(x')$ refers to an inlier counterfactual that is located at a reasonable distance from the training data that are predicted identically. According to [23], the formal definition of *proximity* (Eq. 6) corresponds to the local outlier factor (LOF) [2] with a neighborhood size of $K = 1$, which is a well-known model for outlier detection. In the *fitting* phase of CARE, we create an LOF model for every class/response range of the samples in the training data that are predicted correctly by the model f. During the *explaining* phase and via the objective function $O_{\text{proximity}}$, we invoke the LOF model related to the class/response range of the counterfactual x' to identify its status; if x' is an inlier, the model outputs 1, otherwise, it returns 0 that refers to an outlier. Thus, the goal is to maximize $O_{\text{proximity}}$ for every counterfactual instance.

4.2.2 Connectedness

Connectedness implies the counterfactual instance is related to the existing knowledge and is not a consequence of the model's artifacts. Such a counterfactual is continuously connected to the observed data (existing knowledge) using a topological notion of path. Along this path, features change smoothly and coherently, and each instance in the path is correlated with the preceding and succeeding instances. This property facilitates the transition of changes and improves the actionability of explanations. Similarly, we benefit from the connectedness evaluation metric proposed in [23] as an objective function for counterfactual generation. The continuous path can be approximated by the notion of ϵ-*chainability* (with $\epsilon > 0$) between two instances e and a, meaning that a finite sequence $X_N = e_0, e_1, \ldots, e_N$, where $X_N \subset X$, exists such that $e_0 = e$, $e_N = a$, and $\forall i < N, D(e_i, e_{i+1}) < \epsilon$. Let x' be a counterfactual instance for an input x. We say counterfactual x' is ϵ-connected to $a \in X$ if $f(x') = f(a)$ and there exist an ϵ-chain X_N between x' and a such that $\forall e \in X_N, f(e) = f(x')$.

Although assessing ϵ-connectedness seems complex, its definition resembles the DBSCAN clustering algorithm [12]. We can acknowledge that x' is ϵ-connected to $a \in X$, if x' and a belong to the same cluster of DBSCAN algorithm with parameters `epsilon` $= \epsilon$ (maximum distance between two samples) and `min_samples` $= 2$ (number of samples in a neighborhood). Using DBSCAN clustering for every counterfactual instance is not computationally efficient. Moreover, finding an optimal `epsilon` parameter, which highly impacts the clustering results, for every class/response range is challenging. To remedy the stated issues, we employ a generalized version of DBSCAN, called HDBSCAN [4]. This algorithm adaptively selects the best `epsilon` value to produce stable clusters. We avoid computational complexity by creating HDBSCAN models on the ground-truth data in the *fitting* phase and then querying the

models for predicting the cluster membership of every potential counterfactual instance in the *explaining* stage. Since a single queried sample is not likely to impact the shape of the created clusters, we achieve a fairly accurate measurement of *connectedness*. Moreover, our approach does not require updating the clustering models, making the assessment procedure computationally efficient.

We define the objective function $O_{\text{connectedness}}$ to connect the generated counterfactuals to the existing knowledge. We categorize the ground-truth samples w.r.t every class/response range that are predicted correctly by the model f. A clustering model for every category is then constructed within the *fitting* phase of the explainer. In the *explaining* phase and using objective function $O_{\text{connectedness}}$, the clustering model corresponding to the class/response range of the counterfactual x' is queried. If x' is assigned to a cluster, the function returns 1, otherwise, it returns 0, which indicates x' is not connected to the ground-truth data. Hence, the goal is to maximize $O_{\text{connectedness}}$ for every counterfactual instance.

4.3 Coherent counterfactual explanations

Preserving the consistency between changed and unchanged features is essential for creating feasible counterfactual explanations. Fulfilling this property is even more important when the desired output is a personalized, actionable recourse. In this case, individuals specify some constraints for a handful of attributes and leave the burden of selecting consistent values for other features to the explanation method. Formally, the feature values of an original input x should be changed in a way that they remain coherent in the counterfactual instance x'. Although soundness properties can establish coherency to some extent, the consistency of features in a high-dimensional and highly correlated feature space can be overlooked. Hence, there is a need for a fine-grained approach that evaluates the coherency of every single feature w.r.t its correlated features. Precisely, the explanation method should distinguish between different values of a feature having a similar distance to the original input's value and a similar effect on the model's decisions. This prevents assigning meaningless values to a feature concerning the status of other features.

Here, we introduce an approximate data-driven method that establishes value consistency for counterfactual features in a granular and autonomous way by merely relying on observational data. We solve the problem of assigning coherent values to features using predictive models that take into account high-order and complex correlations between features. This technique is especially useful when there is unknown or limited knowledge about features' relationships.

We aim to evaluate the coherency of a potential counterfactual instance x' that has changed a list of features indicated as L. We denote the jth feature of counterfactual as x'_j, its correlated features as x'_{corr_j}, and the set/range of its possible values as R_j. Let x^{ref} be a reference counterfactual that will hold coherent values for the changed features and has initial values as $x^{\text{ref}} = x'$. We define the following function to estimate the coherent values for the changed features of x^{ref}:

$$x_j^{ref} = \underset{v \in R_j}{\arg\max}\, P_j(x'_j = v \mid x'_{\text{corr}_j}), \forall j \in L \tag{7}$$

where P_j is a conditional probability distribution function for feature j that calculates the probability of every possible value $\forall v \in R_j$ given its correlated variables corr_j. Depending on the type of j, we estimate P_j using a classification model (if j is categorical) or a regression model (if j is numerical). After constructing predictive models (also called correlation models) for the features and estimating the reference (coherent) counterfactual x^{ref}, the coherency of x' is determined via:

$$coherency(x') = 1 - \frac{\sum_{\forall j \in L} \delta(x'_j, x_j^{ref})}{|L|} \tag{8}$$

where δ is the distance function defined in Sect. 4.1.2. Equation 8 measures the differences between the potential counterfactual x' and the reference counterfactual x^{ref} w.r.t. to the changed features L. The intuition is that, for every feature $\forall j \in L$, if the values of its correlated features corr_j conform with each other, the estimated coherent value x_j^{ref} will be close to the actual value x'_j, leading to $coherency(x') \simeq 1$, implying high coherency among the counterfactual's features.

We implement this idea in the COHERENCY module, which consists of two phases: (1) constructing correlation models to approximate coherent values in the *fitting* phase (Algorithm 1), and (2) exploiting the created models in the optimization algorithm to promote coherent value assignment during the *explaining* phase (Algorithm 2).

Algorithm 1 aims to learn a computationally efficient predictive model for every feature that approximates its values based on its correlated features. It starts by extracting feature correlations from the training data X using Spearman's Rho, correlation ratio, and Cramer's V for the pairs of numerical–numerical, numerical–categorical, and categorical–categorical features, respectively [5]. It creates a symmetric correlation matrix $corr^{m \times m} \in [0, 1]$. We consider every feature j, $j \in \{1 \ldots m\}$, as a target variable that is predicted by a correlation model constructed on its correlated features denoted as $inputs$. The $inputs$ set includes features that have a correlation value above threshold $\rho \in [0, 1]$ with feature j (Line 5); the correlation threshold ρ consists of a separate threshold for every method, i.e., $\rho = \{\rho_{\text{Spearman's Rho}}, \rho_{\text{CorrelationRatio}}, \rho_{\text{Cramer's V}}\}$, to prevent bias in

Algorithm 1 Correlation Models

Require: observed data X, correlation threshold ρ, model's score threshold τ, number of features m, type of features F
Ensure: correlation models $\mathcal{M}$
1: Initialize $\mathcal{M} = \{\}$
2: $corr = \text{CalculateCorrelations}(X, F)$
3: $Xt, Xv = \text{TrainValidationSplit}(X, val\_perc = 0.2)$
4: **for** $j = 1$ **to** m **do**
5: $\quad inputs = \text{FindCorrelatedFeatures}(corr, j, \rho)$
6: $\quad Xt^j, Yt^j = Xt_{inputs}, Xt_j$
7: $\quad Xv^j, Yv^j = Xv_{inputs}, Xv_j$
8: $\quad model = \text{ConstructModel}(Xt^j, Yt^j)$
9: $\quad score = \text{ValidateModel}(model, Xv^j, Yv^j)$
10: $\quad$ **if** $score \geq \tau$ **then**
11: $\quad\quad \mathcal{M}^j_{inputs}, \mathcal{M}^j_{model}, \mathcal{M}^j_{score} = inputs, model, score$
12: $\quad$ **end if**
13: **end for**

Algorithm 2 Coherency Objective ($O_{coherency}$)

Require: input x, counterfactual x', correlation models $\mathcal{M}$, distance function δ
Ensure: coherency cost ξ
1: Initialize $\xi = 0$
2: $L = \text{ChangedFeatures}(x, x')$
3: **for all** $j \in L$ **do**
4: $\quad$ **if** $\mathcal{M}^j \neq \emptyset$ **then**
5: $\quad\quad inputs, model, score = \mathcal{M}^j_{inputs}, \mathcal{M}^j_{model}, \mathcal{M}^j_{score}$
6: $\quad\quad ref = model(x'_{inputs})$
7: $\quad\quad distance = \delta(x'_j, ref)$
8: $\quad\quad cost = score * distance$
9: $\quad\quad \xi = \xi + cost$
10: $\quad$ **end if**
11: **end for**

feature selection caused by varying measurements of the correlation methods.

Correlation-based feature selection has several benefits. First, it eliminates non-correlated features that can diminish the accuracy of correlation models, especially in high-dimensional data sets in which all features may not be informative for predicting a specific feature in the data set (the target feature that we aim to investigate its consistency). Second, considering poorly correlated features for establishing the consistency of a specific feature can cause the optimization algorithm to make changes to more features that may not be necessary, resulting in noisy and distant explanations. Last but not least, feature selection is helpful for creating computationally efficient correlation models that are frequently invoked during runtime.

We create simple classification and regression models (e.g., CART [26] and Ridge [29]) for every categorical and numerical feature, respectively (Line 8). The model construction can be accompanied by cross-validation and hyper-parameter tuning to avoid the over-fitting issue and create more accurate models. We only keep reliable models having a validation score (F1-score or R^2-score) above threshold $\tau \in [0, 1]$ (Line 10). This threshold determines the magnitude of the correlation constraints; hence, it significantly impacts the overall coherency preservation performance. At the end (Line 11), if there exists a reliable model for a feature j, $j \in \{1 \ldots m\}$, its correlation model $\mathcal{M}^j$ will include a triplet $\{inputs, model, score\}$ containing the inputs of the model, the trained model, and the score of the model.

In Algorithm 2, the created models $\mathcal{M}$ are used in the coherency objective function (i.e., $O_{\text{coherency}}$) to encourage the optimization algorithm to recommend coherent values for the features. Initially, we identify the altered features in the original input x by comparing it with the potential counterfactual x', denoted as L (Line 2). For every feature $\forall j \in L$, if a corresponding correlation model $\mathcal{M}^j = \{inputs, model, score\}$ exists, we will evaluate its consistency via the following steps. First, $model$ predicts the value of j based on the values of its correlated features specified by $inputs$ and assigns it to the reference value ref (Line 6). The correlation model considers high-order relationships between multiple features for making prediction. We calculate the coherency of feature j in the potential counterfactual, i.e., x'_j, by calculating its distance to the estimated coherent value ref (Line 7). The intuition is that if the values of the correlated features of j conform with each other, the prediction of the correlation model will be close to the recommended value by the counterfactual, i.e., $x'_j \simeq ref$, leading to a low $distance$. Although we filtered out unreliable models in Algorithm 1, here, we weigh the $distance$ according to the $score$ of the correlation model to have a truthful measurement of the coherency cost (Line 8). This procedure is repeated for every feature, and the output of $O_{\text{coherency}}$ is the total coherency cost ξ that is minimized during the optimization process.

4.4 Actionable recourse

This section describes the ACTIONABILITY module that is used for customizing the counterfactual explanations according to individuals' preferences. We introduce novel mechanisms for the users to express their circumstances and desires precisely. The captured information is then utilized in the AR generation process to enhance the likelihood of generating recommendations that match the users' preferences. To this end, we propose a constraint language (outlined in Table 3) to provide the user with a flexible set of constraints for features. The language provides diverse operators for numerical (e.g., relational operators and intervals) and categorical (e.g., a set of categories) features. We also present the notion of *constraint importance* to weigh the constraints according to their importance for the user. For example, two constraints, *"fix the* ***occupation***" and *"constrain the* ***balance*** between [`$5000, $10000`]" have different importance, as exceeding the balance range may be acceptable, but changing the occupation can be costly. In the absence of constraint importance,

Table 3 Constraint language for preference specification

Feature type	Constraint	Description
Numerical	fix	Fix the current value
	l	Greater than the current value
	g	Less than the current value
	le	Less than or equal to the current value
	ge	Greater than or equal to the current value
	$[lb, ub]$	A range of numerical values
Categorical	fix	Fix the current value
	$\{v_1, \ldots, v_n\}$	A set of categorical values

Algorithm 3 Actionability Objective ($O_{actionability}$)

Require: input x, counterfactual x', user's preference $\mathcal{P}$
Ensure: actionability cost η
1: Initialize $\eta = 0$
2: **for all** $\mathcal{C} \in \mathcal{P}$ **do**
3: $S = \text{CheckSatisfiability}(\mathcal{C}_{feature}, \mathcal{C}_{constraint}, x, x')$
4: **if** $S = \texttt{False}$ **then**
5: $\eta = \eta + \mathcal{C}_{importance}$
6: **end if**
7: **end for**

the optimization algorithm makes no difference between a set of preferences. By weighing constraints, we can prioritize them, and therefore, the optimization algorithm avoids to overstep the highly important ones.

We define a user's preference $\mathcal{P}$ as a set of constraint triplets in the form of $\mathcal{C}_j = (\text{feature}_j, \text{constraint}_j, \text{importance}_j)$ for a desired feature j, $j \in \{1 \ldots m\}$, where m is the total number of features, i.e., $\mathcal{P} = \{\mathcal{C}_x, \mathcal{C}_y, \ldots, \mathcal{C}_z\}, x, y, z \in \{1 \ldots m\} \wedge x \neq y \neq z$. An example preference can be $\hat{\mathcal{P}} = \{(\textbf{age}, ge, 4), (\textbf{race}, fix, 10)\}$. Algorithm 3 outlines our proposed objective function $O_{\text{actionability}}$ which computes the actionability cost η for a particular counterfactual x' according to the user's preference $\mathcal{P}$. For every constraint triplet $\forall \mathcal{C} \in \mathcal{P}$, Algorithm 3 checks the satisfiability of $\mathcal{C}_{\text{constraint}}$ for $\mathcal{C}_{\text{feature}}$ (Line 3); if $\mathcal{C}_{\text{constraint}}$ is not satisfied (i.e., $S = \texttt{False}$), then $\mathcal{C}_{\text{importance}}$ is added to the actionability cost η (Line 5). The $O_{\text{actionability}}$ function only evaluates the feasibility of features that are indicated in the preference set $\mathcal{P}$. Therefore, for features that are not specified by the user, the actionability cost will be 0, indicating that CARE can change them arbitrarily. The output of $O_{\text{actionability}}$ is the total incurred actionability cost η that is minimized during the optimization process.

4.5 Multi-objective optimization framework

We adopt non-dominated sorting genetic algorithm III (NSGA-III) [7] to solve our multi-objective optimization problem. Once different modules are included in the CARE's hierarchy, it is NSGA-III's responsibility to establish interaction between the objective functions existing in various modules and find a set of Pareto-optimal solutions[1] by making a trade-off between the defined objectives. Compared to other evolutionary algorithms, NSGA-III performs well with differently scaled objective values and generates diverse solutions. The first property is essential for a multi-objective counterfactual explanation method where there exists a combination of fitness and cost functions with different ranges of output and conflict goals. The second property is useful in the sense of actionability, as providing a diverse set of solutions increases the chance of obtaining a recourse complying with the user's circumstance. The objective set Obj defines the CARE's hierarchy:

$$Obj = \Big\{\{\downarrow O_{\text{outcome}}, \downarrow O_{\text{distance}}, \downarrow O_{\text{sparsity}}\}_{\mathbf{1}}, \{\uparrow O_{\text{proximity}}, \uparrow O_{\text{connectedness}}\}_{\mathbf{2}}, \{\downarrow O_{\text{coherency}}\}_{\mathbf{3}}, \{\downarrow O_{\text{actionability}}\}_{\mathbf{4}}\Big\} \tag{9}$$

Every set in Obj corresponds to a module annotated by a subscript number, i.e., **1**, **2**, **3**, and **4** (also shown in Fig. 1). Arrows indicate the type of objectives, either cost or fitness function, which should be minimized or maximized, respectively. As we mentioned earlier, the first set (VALIDITY module) is always present, while other sets can be arbitrarily included. For example, we may create the combination {**1**, **2**} which only contains the VALIDITY and SOUNDNESS modules. This modular structure allows our method to be used for both counterfactual explanation (combination {**1**, **2**}) and actionable recourse generation (combination {**1**, **2**, **3**, **4**}). Since the defined objective functions make no assumption about the nature and internal structure of the prediction model f, CARE is applicable on any ML model created for tabular classification and regression tasks. Moreover, the NSGA-III algorithm can automatically handle categorical features dispensable of auxiliary operations

[1] A solution is called Pareto-optimal, if none of the objective functions can be improved without degrading at least one of the other objectives.

Table 4 Temporal action sequences for an instance from the *Adult* data set

x	y: ≤**50K**	**age:** 22 – **education-num:** 9 – **education:** HS-grad – **marital status:** Never married – **relationship:** Not-in-family
x'	y: > **50K**	**age:** 37 – **education-num:** 14 – **education:** Masters – **marital status:** Married – **relationship:** Husband
ord_1	*cost*: 1.12	**education-num** → **marital status** → **age** → **relationship** → **education**
ord_2	*cost*: 0.34	**age** → **education-num** → **education** → **relationship** → **marital status**

(such as one-hot encoding and imposing hard-constraints), making CARE suitable for mixed-feature tabular data sets.

4.6 Temporal action sequence

An actionable recourse only identifies the features and their corresponding values for obtaining the desired outcome from the model. This information reflects the idea of instantaneous and atomic actions, but in the real world, actions are often sequential and have an ordering. It would be useful to provide discrete action steps for a recommended AR to guide and ease the individual's actions toward the AR's state. In a parallel study [44], the authors propose an explanation technique that generates sequential ARs, which every AR provides an action step guiding the individual to transfer to the recourse state.

Inspired by [44], we devised a procedure based on the coherency objective function ($O_{\text{coherency}}$) to provide temporal action sequences. However, our technique is not a part of the counterfactual generation process, because the employed optimization algorithm, i.e., NSGA-III, applies various operations (e.g., crossover and mutation) over the generations of solutions, making it impossible to trace temporal action sequences. To the best of our knowledge, it is the first post hoc approach to recommend multiple action sequences with different feasibility costs for a specific AR generated by any explanation method (i.e., method-agnostic). It is noteworthy that having a good quality data set and accurate correlation models (influential elements of $O_{\text{coherency}}$) positively impact the performance of our proposed method.

Algorithm 4 outlines our devised procedure for generating temporal action sequences denoted as T. Given an input x and its corresponding counterfactual/recourse x', we perform the following procedure: (1) identifying the list of changed features L (Line 2); (2) generating all permutations (sequences) of changed features P (Line 3); (3) for all generated sequences $\forall ord \in P$ (Line 4), applying sequential and discrete changes to the original input x according to the order of features defined in ord and measuring its coherency cost after every single change (Line 7–9). We consider every change (i.e., a new value for a feature) as an action step that transfers the individual's current state to a new state. If the new state preserves the consistency between features, it produces low cost and vice versa. Every order and its associated cost are added to T (Line 11). Eventually, we can sort the achieved orders in T based on their overall coherency cost to determine the best and worst sequences.

Algorithm 4 Temporal Action Sequence

Require: input x, counterfactual x', coherency objective $O_{coherency}$, correlation models $\mathcal{M}$
Ensure: temporal action sequences T
1: Initialize $T = \{\}$
2: $L = \text{ChangedFeatures}(x, x')$
3: $P = \text{Permutations}(L)$ ▷ all possible sequences of changed features
4: **for all** $ord \in P$ **do**
5: $\quad cost = 0$
6: $\quad x^{ord} = x$
7: $\quad$ **for all** $j \in ord$ **do**
8: $\quad\quad x^{ord}_j = x'_j$
9: $\quad\quad cost = cost + O_{coherency}(x, x^{ord}, \mathcal{M})$
10: $\quad$ **end for**
11: $\quad T = T \bigcup (ord, cost)$
12: **end for**

To illustrate the efficacy of the proposed technique, we created a **GB** classifier for the *Adult* data set [11] and generated an actionable recourse for every test input using CARE (we only imposed global constraints on features like **sex** and **race**). Then, Algorithm 4 is applied to the generated recourses to find the associated cost with every order of actions. Table 4 demonstrates the result for an instance x and its corresponding explanation x'. Due to lack of space, we only showed the changed features. In the *Adult* data set, features **education-num** and **education** are strongly correlated and they both relate to feature **age**. Likewise, there is a strong correlation between features **relationship** and **marital status**. Taking actions according to order ord_1 leads to a high coherency cost because correlated features are changed discontinuously. For example, **relationship** and **marital status** should be changed sequentially/simultaneously. Any temporal gap between these two actions creates a non-sensible state for the individual. On the other hand, ord_2 considers the relationship between features that takes the individual's state to the recourse's state without creating an unrealistic situation along the way. This example demonstrates how this method

Table 5 Summary information of the data sets

Data set	# Sample	# Num	# Cat	Target
Adult	48842	6	8	$\{\leq 50K, > 50K\}$
COMPAS	7214	4	7	{low risk, high risk}
Default of Credit Card Clients	30000	20	3	{No, Yes}
HELOC	10459	23	0	{Bad, Good}
Wine	178	13	0	{C1, C2, C3}
Iris	150	4	0	{setosa, versicolor, virginica}
Moon	500	2	0	{C1, C2}
Diabetes	442	10	0	[25.0, 346.0]
California Housing	20640	8	0	[0.15, 5.0]

Table 6 Performance of the black box models

Data set	NN	GB
Adult	0.849	0.860
COMPAS	0.788	0.808
Default of Credit Card Clients	0.794	0.797
HELOC	0.733	0.727
Wine	1.0	0.917
Iris	1.0	1.0
Moon	0.881	0.990
Diabetes	0.421	0.450
California Housing	0.778	0.776

can detect varied-cost temporal sequences for an actionable recourse. In domains where features and their relationships are known to the user, optimal sequences can be identifiable. However, temporal action sequences are more useful when it comes to the domains with high-dimensional feature space and high-order relationships unknown to the user.

5 Experiments and discussion

This section first describes the evaluation setup, including data sets, models, and hyper-parameters. Second, we reveal the impact of the stated desiderata in Sect. 2 and the efficacy of CARE's modules in addressing them using multiple validation experiments. Finally, we demonstrate the overall performance of CARE by benchmarking against state-of-the-art explanation methods concerning various evaluation metrics.

5.1 Experimental setup

For the experiments, we utilized commonly used classification data sets in counterfactual explanation literature, including *Adult* [11], *COMPAS* [33], *Default of Credit Card Clients* [11], and *HELOC* [13]. To demonstrate the efficacy of our approach in handling multi-class scenarios, we also employed *Wine* [11] and *Iris* [11] data sets. We evaluated the performance of CARE with respect to regression data sets named *Diabetes* [40] and *California Housing* [39]. Summary information of the data sets is reported in Table 5. For every data set, the numerical features were standardized by removing the mean and scaled to unit variance. We converted original categorical features to ordinal encoding and used their corresponding one-hot encoding for creating black box models.

We split the data sets into 80% *train set* and 20% *test set*. The *train set* was used for creating black box and the explainer models while the *test set* was used for evaluating the model performance and generating explanations. We created a multilayer perceptron neural networks (**NN**) consisted of two hidden layers each with 50 neurons and a gradient boosting machines (**GB**) comprised of 100 estimators as black box classifiers and regressors. Table 6 reports the performance of the created models for every data set in terms of F1-score (classification task) and R^2-score (regression task).

CARE is mainly consisted of two phases: *fitting* and *explaining*. Given an ML model f and its corresponding *train set*, an explainer during the *fitting* phase is created. We will then use the explainer to explain every desired instance in the *explaining* phase. Within the *fitting* phase, correctly predicted samples by black box f for every class (in classification task) or response range (in regression task) is identified, and unique LOF (used in $O_{\text{proximity}}$) and HDBSCAN (used in $O_{\text{connectedness}}$) models for every class/response range are constructed. Correctly predicted samples in classification task are simply detected by comparing ground-truth labels and predicted labels. In the regression task, we assume a sample x is correctly predicted if it meets the following condition: $|r_x - \hat{r}_x| \leq MAE$, where r_x is the ground-truth response value, $\hat{r}_x$ is the predicted response value, and MAE is the mean absolute error of all predictions of the training data. We can create several response ranges for a regression task by dividing the original response range into intervals. In the experiment, we used 4-quantiles as cutting points for inter-

vals. In this case, a counterfactual can be a sample from a desired interval. We selected the neighbor interval as the default response target. Similarly, for the classification task we can select any desired class $c \in Y$ for counterfactual generation. The probability threshold for the classification task was set to $p = 0.5$.

Creating correlation models (Algorithm 1) is another task that is handled in the *fitting* phase. We employed CART decision tree [26] and Ridge regression model [29] for predicting categorical and numerical features, respectively. We set the minimum correlation threshold ρ for every correlation metric in $\Omega = \{Spearman's Rho, Correlation Ratio, Cramer's V\}$ as the average correlation values of their covered features, i.e., $\rho = \{\overline{corr_\omega} : \forall \omega \in \Omega\}$, where $corr_\omega \subseteq corr$ is a subset of the correlation matrix that is achieved by the correlation metric ω. We used F1-score and used R^2-score to measure the performance of CART and Ridge models, respectively. We filtered out unreliable correlation models by setting the model's score threshold as $\tau = 0.7$.

For NSGA-III optimization algorithm, we used the two-point crossover operator with percentage $pc = 60\%$ and the polynomial mutation operator with percentage $pm = 30\%$ that are applied on the population in every generation. The number of generations was set to $n_{\text{generation}} = 10$. Although it is possible to define arbitrary population size $n_{\text{population}}$, we determined it adaptively concerning the number of objectives using a standard formula described in the original paper of NSGA-III [7]. This approach is more suitable for our modular framework in which the number of objectives would vary with respect to different combinations. For every input x, we initialize the individuals of the population by randomly sampling from three sets: instance x, samples in the neighborhoods of x belonging to the counterfactual class/response range, and random instances, with sampling probabilities $P_x = 0.3$, $P_{\text{neighbor}} = 0.6$, and $P_{\text{random}} = 1$. We scaled both numerical and ordinal features between [0, 1] for the optimization algorithm while user preferences are given in the data set's original format.

Regarding user preferences for actionable recourse, we only set global constraints according to our basic knowledge about the domains for all inputs. For example, constraint *fix* (i.e., fix the current value) was set for features like **race** and **sex** while constraint *ge* (i.e., greater than or equal to the current value) was set for feature **age**. Thus, actionable recourses are not locally personalized with respect to every specific input. To have a fair comparison regarding the actionability objective function (i.e., $O_{\text{actionability}}$) among baseline methods, we set equal importance values for all constraints, i.e., $C_{\text{importance}} = 1$.

CARE has been developed using Python programming language, and experiments were run on a system with Intel Core i7-8650U processor and 32GB of memory. A complete implementation of the method, including validation and benchmark experiments, is available at: https://github.com/peymanrasouli/CARE.

5.2 Illustration of soundness properties

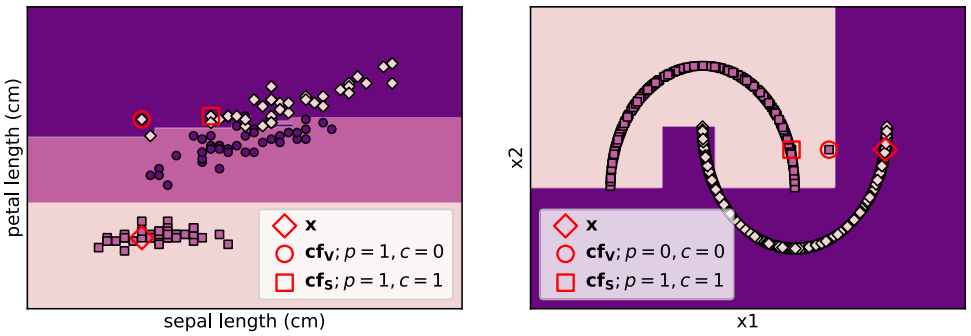
We visualize the impact of SOUNDNESS module in generating statistically sound counterfactuals on the *Iris* and *Moon* data sets. We created Gradient Boosting (**GB**) classifiers on these data sets and generated $\mathbf{cf_V}$ and $\mathbf{cf_S}$ counterfactuals using combinations $\{\mathbf{1}\}$ and $\{\mathbf{1}, \mathbf{2}\}$, respectively. Figure 2a demonstrates counterfactuals from the furthest class for a sample from the *Iris* data set, and Figure 2b depicts counterfactuals from the opposite class for an instance from the *Moon* data set. The counterfactuals are annotated with the values of $O_{\text{proximity}}$ and $O_{\text{connectedness}}$ denoted as p and c, respectively.

By observing the location of the generated counterfactuals in Fig. 2a, b, the importance of SOUNDNESS module is revealed. It can be seen that minimum distance is the only important criterion for $\mathbf{cf_V}$ counterfactuals while $\mathbf{cf_S}$ counterfactuals are associated with the same-class training data ($p = 1$ and $c = 1$), therefore they are connected to the previous knowledge and achieve a high prediction probability from the classifier. We stated that *proximity* is different than *connectedness* and they are complementary criteria for a sound counterfactual explanation. Figure 2a demonstrates this distinction clearly as $\mathbf{cf_V}$ is located at the proximity of the training data ($p = 1$), but it is not connected to a high-density region ($c = 0$) which declines its interpretability and actionability.

5.3 Validation of coherency module

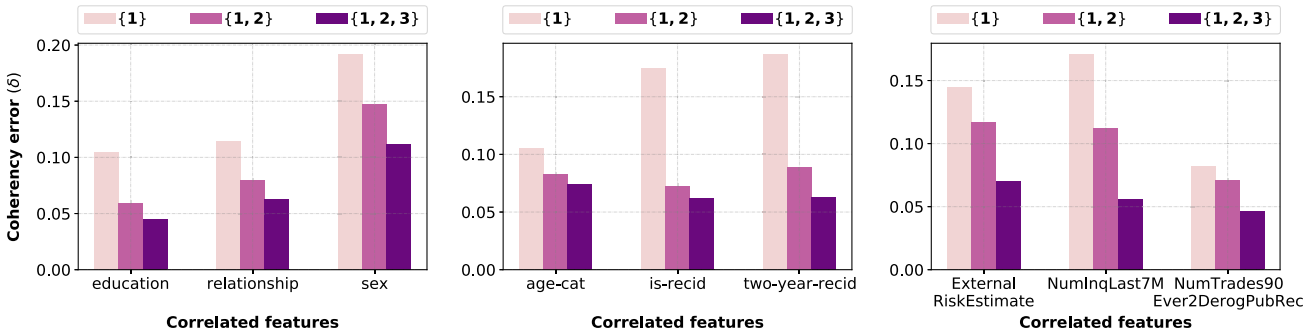
Counterfactuals are, in fact, a set of new instances in desired classes sharing similar properties as the set of original data points. Therefore, it is possible to measure the correlation discrepancy between these sets (denoted as δ) to reveal how much the explanation method ignored the consistency between features while generating counterfactual instances. If δ is large, it indicates that the explanation method has changed featuress regardless of their relationships with other features. Therefore, we prefer a method that generates counterfactual data points leading to minimum correlation difference with the original inputs, i.e., small δ.

The COHERENCY module preserves the consistency between changed and unchanged features that results in feasible explanations. To validate the impact of this module in the CARE's hierarchy, we designed an experiment to measure the correlation discrepancy in counterfactuals generated by different module combinations of CARE. We have used the *Adult*, *COMPAS*, and *HELOC* data sets for the experiment. To have a fair comparison between different combinations, we have selected three highly correlated features in each data



(a) Iris data set. (b) Moon data set.

Fig. 2 The impact of soundness properties (i.e., proximity and connectedness) on the generated counterfactuals



(a) Adult data set. (b) COMPAS data set. (c) HELOC data set.

Fig. 3 Coherency validation results. It shows the performance of different combinations of CARE in preserving the consistency of highly correlated features in the data sets

set because their relationships with other features are likely to be preserved by different combinations as dictated by data distribution.

The steps of the experiment are as follows: (1) create a correlation matrix for every data set using their *train set* to identify three features that are highly correlated with others denoted as $F = \{j_1, j_2, j_3\}$; (2) randomly select n samples from the *test set* of each data set and name as *explain set* (for this experiment, we set $n = 500$; the larger n, the more reliable results); (3) extract the correlation matrix for the *explain set* indicated as $corr$; (4) generate different sets of counterfactuals for samples of *explain set* using three combinations of CARE including $\{\mathbf{1}\}$, $\{\mathbf{1}, \mathbf{2}\}$, and $\{\mathbf{1}, \mathbf{2}, \mathbf{3}\}$ with respect to every correlated feature $\forall j \in F$, given the constraint that the value of j *must* change in every explanation; this helps to establish a fair comparison between distinct combinations as each tends to alter a different set of features; (5) calculate the correlation matrix for every set of counterfactuals denoted as $corr^{\{\mathbf{1}\}}$, $corr^{\{\mathbf{1},\mathbf{2}\}}$, and $corr^{\{\mathbf{1},\mathbf{2},\mathbf{3}\}}$; (6) measure the coherency error δ for feature j that is incurred by every combination by calculating mean absolute error (MAE) between the correlation matrix of the original and counterfactual sets, i.e., $\delta_j^{\text{comb}} = \dfrac{\sum_{k=1}^{m} |corr_{j,k} - corr_{j,k}^{\text{comb}}|}{m}$, $\forall \text{comb} \in \big\{\{\mathbf{1}\}, \{\mathbf{1}, \mathbf{2}\}, \{\mathbf{1}, \mathbf{2}, \mathbf{3}\}\big\}$, where m is the dimension of features. Accordingly, a coherency preservation method alters feature j and its correlated features in a way to create a consistent state for the counterfactual, leading to a small value for δ.

Figure 3 depicts the results of the above-mentioned experiment. As expected, the valid counterfactuals generated by combination $\{\mathbf{1}\}$ fail to establish a great extent of consistency between the features, leading to high δ for every correlated feature of all data set. It is because the minimum distance, as the main property for this approach, often leads to outlier counterfactuals that may not comply with the ground-truth distribution. This issue has been considerably resolved by the combination $\{\mathbf{1}, \mathbf{2}\}$ that generates sound counterfactuals linked to the observed data. However, as we described earlier, the soundness property does not ensure consistent explanations for every input due to the existence of feature values that have a similar distance to the original input's value and influence the model's predictions similarly. For example, in the *Adult* data set [11], the two values {`wife`, `husband`} for the **relationship** feature have an identical distance to each other (as they are categorical values) and affect the model's predictions in a similar way. In this case, a sound explanation will not necessarily distinguish between {`wife`, `husband`} as selecting either value fulfills the soundness property. However, from the consistency perspective, the cor-

Table 7 Efficacy of CARE's modules

Data set	Combination	$\downarrow O_{\text{outcome}}$	$\downarrow O_{\text{distance}}$	$\downarrow O_{\text{sparsity}}$	$\uparrow O_{\text{proximity}}$	$\uparrow O_{\text{connectedness}}$	$\downarrow O_{\text{coherency}}$	$\downarrow O_{\text{actionability}}$
Adult	{**1**}	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.02 \pm 0.0}$	$\mathbf{1.41 \pm 0.7}$	0.58 ± 0.5	0.20 ± 0.4	0.05 ± 0.2	0.08 ± 0.3
	{**1**, **2**}	0.00 ± 0.0	0.12 ± 0.1	2.84 ± 1.7	$\mathbf{1.00 \pm 0.0}$	$\mathbf{1.00 \pm 0.0}$	0.10 ± 0.3	0.36 ± 0.6
	{**1**, **2**, **3**}	0.00 ± 0.0	0.12 ± 0.1	3.05 ± 1.7	1.00 ± 0.0	1.00 ± 0.0	$\mathbf{0.00 \pm 0.0}$	0.35 ± 0.6
	{**1**, **2**, **3**, **4**}	0.00 ± 0.0	0.11 ± 0.1	2.76 ± 1.8	1.00 ± 0.0	0.91 ± 0.3	0.00 ± 0.0	$\mathbf{0.00 \pm 0.0}$
COMPAS	{**1**}	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.03 \pm 0.0}$	$\mathbf{1.94 \pm 0.9}$	0.58 ± 0.5	0.34 ± 0.5	0.01 ± 0.1	0.49 ± 0.5
	{**1**, **2**}	0.00 ± 0.0	0.08 ± 0.1	2.63 ± 1.2	$\mathbf{1.00 \pm 0.0}$	$\mathbf{1.00 \pm 0.0}$	0.01 ± 0.1	0.84 ± 0.8
	{**1**, **2**, **3**}	0.00 ± 0.0	0.08 ± 0.1	2.79 ± 1.4	1.00 ± 0.0	1.00 ± 0.0	$\mathbf{0.00 \pm 0.0}$	0.89 ± 0.8
	{**1**, **2**, **3**, **4**}	0.00 ± 0.0	0.07 ± 0.1	2.34 ± 1.3	0.99 ± 0.1	0.67 ± 0.5	0.00 ± 0.0	$\mathbf{0.00 \pm 0.0}$
Wine	{**1**}	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.04 \pm 0.0}$	$\mathbf{2.33 \pm 2.3}$	0.30 ± 0.5	0.03 ± 0.2	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.00 \pm 0.0}$
	{**1**, **2**}	0.00 ± 0.0	0.09 ± 0.1	5.70 ± 3.6	$\mathbf{1.00 \pm 0.0}$	$\mathbf{0.53 \pm 0.5}$	0.00 ± 0.0	0.00 ± 0.0
	{**1**, **2**, **3**}	0.00 ± 0.0	0.10 ± 0.1	7.17 ± 4.7	1.00 ± 0.0	0.53 ± 0.5	0.00 ± 0.0	0.00 ± 0.0
	{**1**, **2**, **3**, **4**}	0.00 ± 0.0	0.09 ± 0.1	6.63 ± 4.5	1.00 ± 0.0	0.53 ± 0.5	0.00 ± 0.0	0.00 ± 0.0
Diabetes	{**1**}	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.02 \pm 0.0}$	$\mathbf{1.85 \pm 1.3}$	0.84 ± 0.4	0.36 ± 0.5	0.03 ± 0.1	0.10 ± 0.3
	{**1**, **2**}	0.00 ± 0.0	0.05 ± 0.0	3.06 ± 2.0	$\mathbf{1.00 \pm 0.0}$	$\mathbf{1.00 \pm 0.0}$	0.05 ± 0.1	0.08 ± 0.3
	{**1**, **2**, **3**}	0.00 ± 0.0	0.04 ± 0.0	2.20 ± 1.2	1.00 ± 0.0	0.81 ± 0.4	$\mathbf{0.00 \pm 0.0}$	0.08 ± 0.3
	{**1**, **2**, **3**, **4**}	0.00 ± 0.0	0.04 ± 0.0	2.00 ± 1.2	1.00 ± 0.0	0.79 ± 0.4	0.00 ± 0.0	$\mathbf{0.00 \pm 0.0}$
California Housing	{**1**}	$\mathbf{0.00 \pm 0.0}$	$\mathbf{0.00 \pm 0.0}$	$\mathbf{1.67 \pm 1.0}$	0.51 ± 0.5	0.83 ± 0.4	0.01 ± 0.0	$\mathbf{0.00 \pm 0.0}$
	{**1**, **2**}	0.00 ± 0.0	0.01 ± 0.0	2.05 ± 1.2	$\mathbf{1.00 \pm 0.0}$	$\mathbf{1.00 \pm 0.0}$	0.02 ± 0.0	0.00 ± 0.0
	{**1**, **2**, **3**}	0.00 ± 0.0	0.01 ± 0.0	1.81 ± 0.9	0.97 ± 0.2	0.93 ± 0.3	$\mathbf{0.00 \pm 0.0}$	0.00 ± 0.0
	{**1**, **2**, **3**, **4**}	0.00 ± 0.0	0.01 ± 0.0	1.71 ± 1.0	0.91 ± 0.3	0.91 ± 0.3	0.00 ± 0.0	0.00 ± 0.0

The results reveal the role of every module in fulfilling actionable recourse desiderata when it is included (or absent) in the explanation framework

rect recommended value for **relationship** should be chosen according to the **gender** of the individual. This statement has been reflected in the results. In contrast, the combination {**1**, **2**, **3**} equipped with the COHERENCY module can distinguish such subtle discrepancies by considering feature correlations in explanation generation. This property has substantially preserved the consistency among counterfactuals' features, leading to the lowest δ for all scenarios.

5.4 Efficacy of CARE's modules

Modules in the CARE's hierarchy are rigorously designed to handle important desiderata for actionable recourse. Earlier, we demonstrated the role of CARE's modules in approaching different properties via an illustrative example (Table 2). In this section, we quantitatively evaluate the impact of each module by conducting the following experiment. We consider four combinations {**1**}, {**1**, **2**}, {**1**, **2**, **3**}, and {**1**, **2**, **3**, **4**}. We are interested to know the behavior of CARE when some modules are absent. For example, how much a counterfactual generated by combination {**1**} will satisfy objectives defined in combination {**1**, **2**}. Using this information, we can determine the importance of the devised modules. To this end, we created **GB** models for *Adult*, *COMPAS*, *Wine*, *Diabetes*, and *California Housing* data sets and generated counterfactual explanations using the specified module combinations for 500, 500, 30, 80, and 400 samples of their *test set*, respectively. Eventually, their results regarding the objectives defined in the last combination (i.e., {**1**, **2**, **3**, **4**}), which contains all CARE's modules, were measured. By observing the results demonstrated in Table 7, we can conclude several points about the performance of different modules:

- CARE generates valid counterfactuals for all combinations and data sets since VALIDITY module is the basis of our methodology.
- The counterfactuals generated by combination {**1**} are best at fulfilling the O_{distance} and O_{sparsity} objectives because closeness and sparsity are the essential goals in this setting. Moreover, their cost for the $O_{\text{actionability}}$ objective is usually low since changing immutable features leads to a counterfactual with a dramatic distance to the original input, therefore, the optimization algorithm usually avoids manipulating such features.
- The small values for $O_{\text{proximity}}$ and $O_{\text{connectedness}}$ objectives in combination {**1**} confirm that valid counterfactuals do not necessarily originate from or connect to the ground-truth data. Furthermore, the significant difference between the values of these objectives clarifies the distinction between *proximity* and *connectedness* properties.
- The SOUNDNESS module has significantly improved the $O_{\text{proximity}}$ and $O_{\text{connectedness}}$ objectives in combi-

nations {**1**, **2**}, {**1**, **2**, **3**}, and {**1**, **2**, **3**, **4**}. Meanwhile, since sound counterfactuals originate from high-density regions, their O_{distance} and O_{sparsity} costs are expectedly increased.

- The devised COHERENCY module has effectively preserved the consistency among counterfactuals' features, resulting in $O_{\text{coherency}} = 0$ for combinations equipped with the module (i.e., {**1**, **2**, **3**} and {**1**, **2**, **3**, **4**}). Interestingly, the coherency preservation rate has not been compromised even when user preferences are in play (combination {**1**, **2**, **3**, **4**}).
- According to the results, combination {**1**, **2**, **3**, **4**} has effectively incorporated user preferences ($O_{\text{actionability}} = 0$), creating actionable recourse for all instances. An interesting point in this setting is a reduction in the $O_{\text{connectedness}}$ objective function, which aims to generate actionable explanations by relying on the observed data. This is due to the conflict between user-specified constraints and the existing relationship between data features, which in this case, user preferences are prioritized.

5.5 Benchmark results

To evaluate the efficacy of CARE, we compared its performance with state-of-the-art explanation methods CFPrototype [42], DiCE [30], and CERTIFAI [37]. We selected these techniques because they address similar properties as CARE. Precisely, CFPrototype uses a loss function called *prototype loss* to generate sound counterfactuals located in the proximity of the same-class training data. DiCE generates diverse counterfactuals and allows imposing actionability constraints on the input features. CERTIFAI is a genetic-based and model-agnostic approach that can generate diverse and actionable explanations for any tabular classifier. To balance between diversity and proximity of the generated counterfactuals in DiCE, we set the corresponding hyper-parameters as equal $\lambda_1, \lambda_2 = 1$. We used CFPrototype and CERTIFAI with the default hyper-parameters stated in their papers. The methods were applied on four binary classification data sets including *Adult*, *COMPAS*, *Default of Credit Card Clients* (*DCCC* for short), and *HELOC*. We created an **NN** black box model for every data set using their *train set* and explained 500 samples from their *test set*. The number of generated counterfactuals in CARE, DiCE, and CERTIFAI was set to $N = 10$.

5.5.1 Property fulfillment

CARE's objective functions evaluate the competence of explanations regarding various properties in an intuitive and straightforward manner. Some of the objectives are even directly derived from existing metrics like *proximity* and *connectedness* [23]. Hence, they can act as reliable metrics for evaluating the generated explanations by baseline methods regarding property fulfillment.

Table 8 reports the evaluation of counterfactuals with respect to the CARE's objective functions. Compared to the baselines, CARE has successfully generated valid counterfactuals (i.e., instances from the opposite class) for every data point of all data sets. It is because the baseline approaches follow a bottom-up approach as they iteratively increase changes until a sample with the desired label is found, or a perturbation threshold is met. In contrast, our algorithm follows a top-down procedure, as it first creates valid counterfactuals and then refines the solutions during determined iterations. Therefore, CARE is less dependent on hyper-parameters to generate valid explanations.

Given the fact that properties like soundness, coherency, and actionability impose more changes to the original input to create an inlier data point, CARE has performed effectively regarding the O_{distance} and O_{sparsity} objectives. The generated counterfactuals are located fairly close to the explained instances, and the number of alterations is considerably low, resulting in more interpretable explanations.

CARE's counterfactuals originate from the distribution of the same-class ground-truth data (high value for $O_{\text{proximity}}$) and connect to the existing knowledge by a continuous path (high value for $O_{\text{connectedness}}$). The former results in inlier instances with plausible feature values, while the latter facilitates transitions from the original state to the counterfactual state.

According to $O_{\text{coherency}}$, CARE has effectively established the coherency relationships between counterfactual features in all scenarios. It generates consistent explanations while satisfying user-defined actionability constraints ($O_{\text{actionability}} = 0$). It is noteworthy that the coherency preservation rate is directly related to the number of correlation models that we had generated; the more reliable correlation models exist, the more complex correlations are preserved.

CARE produced actionable recourse for every input of the model ($O_{\text{actionability}} = 0$), meanwhile, it had satisfied other properties effectively. This comprehensive performance originates from the selected optimization scheme, which formulates every property as an objective function and provides a universally effective solution. In contrast, baseline methods either consider user preferences as an optimization constraint (e.g., CERTIFAI) or apply a post hoc filtering step to remove infeasible explanations (e.g., DiCE). As demonstrated by the results, these approaches may fail to create actionable recourse for some inputs ($O_{\text{actionability}} > 0$) or find sound and coherent actionable recourses at the same time ($O_{\text{proximity}}$, $O_{\text{soundness}} \lesssim 0.5$ and $O_{\text{coherency}} > 0$).

To support the conducted quantitative evaluation, we provided a few example explanations in Table 9 to show the

Table 8 Property fulfillment

Data set	Method	$\downarrow O_{\text{outcome}}$	$\downarrow O_{distance}$	$\downarrow O_{\text{sparsity}}$	$\uparrow O_{\text{proximity}}$	$\uparrow O_{\text{connectedness}}$	$\downarrow O_{\text{coherency}}$	$\downarrow O_{\text{actionability}}$
Adult	CFPrototype	0.005 ± 0.0	$\mathbf{0.080 \pm 0.1}$	2.930 ± 1.5	0.516 ± 0.5	0.062 ± 0.2	0.208 ± 0.4	0.267 ± 0.5
	DiCE	0.123 ± 0.2	0.204 ± 0.1	3.787 ± 1.6	0.203 ± 0.4	0.384 ± 0.5	0.548 ± 0.7	0.000 ± 0.0
	CERTIFAI	0.000 ± 0.0	0.215 ± 0.2	6.719 ± 2.1	0.350 ± 0.5	0.041 ± 0.2	0.886 ± 0.9	0.177 ± 0.4
	CARE	$\mathbf{0.000 \pm 0.0}$	0.106 ± 0.1	$\mathbf{2.684 \pm 1.7}$	$\mathbf{0.996 \pm 0.1}$	$\mathbf{0.896 \pm 0.3}$	$\mathbf{0.000 \pm 0.0}$	$\mathbf{0.000 \pm 0.0}$
COMPAS	CFPrototype	0.006 ± 0.0	0.087 ± 0.1	3.162 ± 1.2	0.578 ± 0.5	0.412 ± 0.5	0.036 ± 0.2	0.782 ± 0.5
	DiCE	0.090 ± 0.2	0.351 ± 0.1	5.682 ± 1.3	0.380 ± 0.5	0.070 ± 0.3	1.656 ± 0.5	0.000 ± 0.0
	CERTIFAI	0.000 ± 0.0	$\mathbf{0.054 \pm 0.1}$	3.588 ± 0.9	0.554 ± 0.5	0.204 ± 0.4	0.193 ± 0.4	0.004 ± 0.1
	CARE	$\mathbf{0.000 \pm 0.0}$	0.074 ± 0.1	$\mathbf{2.384 \pm 1.5}$	$\mathbf{0.986 \pm 0.1}$	$\mathbf{0.666 \pm 0.5}$	$\mathbf{0.002 \pm 0.0}$	$\mathbf{0.000 \pm 0.0}$
Default of Credit Card Clients	CFPrototype	0.000 ± 0.0	0.107 ± 0.0	10.506 ± 5.2	0.278 ± 0.4	0.014 ± 0.1	0.968 ± 0.6	0.466 ± 0.5
	DiCE	0.145 ± 0.1	0.216 ± 0.1	16.764 ± 2.4	0.116 ± 0.3	0.000 ± 0.0	2.419 ± 1.2	0.000 ± 0.0
	CERTIFAI	0.350 ± 0.2	0.575 ± 0.1	21.486 ± 1.0	0.186 ± 0.4	0.000 ± 0.0	5.291 ± 1.0	0.796 ± 0.6
	CARE	$\mathbf{0.000 \pm 0.0}$	$\mathbf{0.079 \pm 0.1}$	$\mathbf{3.084 \pm 2.3}$	$\mathbf{0.994 \pm 0.1}$	$\mathbf{0.690 \pm 0.5}$	$\mathbf{0.000 \pm 0.0}$	$\mathbf{0.000 \pm 0.0}$
HELOC	CFPrototype	0.003 ± 0.0	$\mathbf{0.026 \pm 0.0}$	7.280 ± 2.7	0.902 ± 0.3	0.594 ± 0.5	0.003 ± 0.0	0.000 ± 0.0
	DiCE	0.264 ± 0.2	0.172 ± 0.0	13.710 ± 1.7	0.150 ± 0.4	0.000 ± 0.0	0.439 ± 0.3	0.000 ± 0.0
	CERTIFAI	0.074 ± 0.2	0.147 ± 0.1	19.604 ± 2.5	0.406 ± 0.5	0.448 ± 0.5	0.266 ± 0.5	0.000 ± 0.0
	CARE	$\mathbf{0.000 \pm 0.0}$	0.045 ± 0.0	$\mathbf{7.082 \pm 5.1}$	$\mathbf{0.976 \pm 0.2}$	$\mathbf{0.848 \pm 0.4}$	$\mathbf{0.000 \pm 0.0}$	$\mathbf{0.000 \pm 0.0}$

The results present the performance of CARE and baselines in satisfying various actionable recourse properties, measured by CARE's objective functions

performance of baseline methods qualitatively. We used the *Adult* data set for this experiment because its features and corresponding correlations are more perceptible by average users compared to other studied data sets. The demonstrated examples reveal several interesting points linked to the quantitative evaluation as: (1) disregarding the *actionability* property leads to infeasible recourses; for example, the CFPrototype method changed **sex** and **native country** features in EXAMPLE 1 and EXAMPLE 3 respectively, confirming its high average cost for $O_{actionability}$ objective reported in Table 8; (2) sound counterfactual explanations complement user preferences and lead to reasonable values for unconstrained features; although DiCE and CERTIFAI techniques have satisfied general user-defined constraints for all examples, they made anomalous changes to other features like **capital gain** and **education**; (3) the *coherency* property plays a crucial role in creating a consistent state of changes for an individual; consequently, CARE recommended conformable values for features **age** and **education** in EXAMPLE 3 and EXAMPLE 4 as opposed to CERTIFAI in EXAMPLE 1; (4) considering *sparsity* as a distinct objective function improves the simplicity and intuitiveness of explanations; as a result, CARE changed a small number of features that are decisive for achieving the desired outcome, rather than altering many features slightly (e.g., CERTIFAI explanations).

5.5.2 Diversity of explanations

We benchmarked CARE versus DiCE and CERTIFAI regarding the diversity of the generated counterfactuals. A diverse set of explanations for a particular input provides the user with more alternatives to obtain the desired outcome. We define two diversity metrics called feature-based diversity (d_F) and value-based diversity (d_V). The former measures the variation of the changed attributes between the generated counterfactuals, while the latter calculates the diversity of prescribed values for the counterfactuals' features.

The feature-based diversity metric (d_F) is defined as follows:

$$d_F = 1 - \Big(\frac{1}{C_N^2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} Jaccard(\mathcal{S}_i, \mathcal{S}_j)\Big) \tag{10}$$

where C is combination operator, N is the total number of counterfactuals, and $\mathcal{S}$ is the set of explanations. We calculate the Jaccard index between the feature names in every pair of explanations for calculating d_F. We believe this is a representative metric for diversity because a user is generally interested in a set of counterfactuals providing various combinations of changes instead of recommending multiple values for one or a few specific features. However, for some inputs, only a few features determine the model's decision, which in this case, suggesting different values for those features is inevitable.

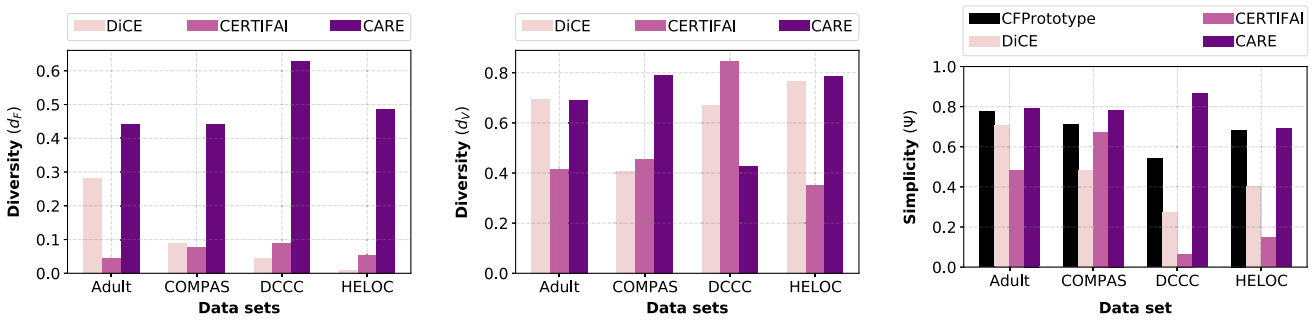
Table 9 Qualitative evaluation. The table reports explanations generated by CARE and baseline methods for a few instances from the *Adult* data set

EXAMPLE 1	age	capital gain	capital loss	hours-per-week	education	occupation	marital status	sex	native country	*Income*
x	19	0	0	35	Some college	Other service	Never married	Male	United States	≤**50K**
$x'_{\text{CFPrototype}}$	26	13059	43	42	_	_	_	Female	_	>**50K**
x'_{DiCE}	_	98968	_	_	_	_	_	_	_	>**50K**
$x'_{CERTIFAI}$	20	8538	109	_	Doctorate	_	Married-AF-spouse	_	_	>**50K**
x'_{CARE}	38	_	_	40	_	Tech-support	Married-civ-spouse	_	_	>**50K**

EXAMPLE 2	age	capital gain	capital loss	hours-per-week	education	occupation	marital status	sex	native country	*Income*
x	57	0	0	70	10th	Adm-clerical	Divorced	Female	Germany	≤**50K**
$x'_{CFPrototype}$	_	12501	_	_	_	_	_	_	_	>**50K**
x'_{DiCE}	_	99965	_	71	5th-6th	_	_	_	_	>**50K**
$x'_{CERTIFAI}$	65	4964	53	71	Doctorate	Priv-house-serv	Separated	_	_	>**50K**
x'_{CARE}	_	18692	_	64	_	_	_	_	_	>**50K**

EXAMPLE 3	age	capital gain	capital loss	hours-per-week	education	occupation	marital status	sex	native country	*Income*
x	18	0	0	16	11th	Prof-specialty	Never married	Female	United States	≤**50K**
$x'_{CFPrototype}$	31	17166	_	17	_	_	_	_	Scotland	>**50K**
x'_{DiCE}	_	99894	_	_	5th-6th	_	Married-civ-spouse	_	_	>**50K**
$x'_{CERTIFAI}$	20	19914	61	19	9th	Exec-managerial	_	_	_	>**50K**
x'_{CARE}	28	_	_	45	Bachelors	_	Married-civ-spouse	_	_	>**50K**

EXAMPLE 4	age	capital gain	capital loss	hours-per-week	education	occupation	marital status	sex	native country	*Income*
x	24	0	0	40	HS-grad	Machine-op-inspct	Married-civ-spouse	Female	United States	≤**50K**
$x'_{CFPrototype}$	25	7045	_	_	_	_	_	_	_	>**50K**
x'_{DiCE}	_	91270	_	_	_	_	_	_	_	>**50K**
$x'_{CERTIFAI}$	28	4761	2	41	9th	Tech-support	_	_	_	>**50K**
x'_{CARE}	29	_	_	_	Masters	Prof-specialty	_	_	_	>**50K**



(a) Feature-based diversity. (b) Value-based diversity. (c) Simplicity score.

Fig. 4 Comparison of explanation methods regarding the diversity and simplicity of explanations

We define the value-based diversity metric (d_V) as follows:

$$d_V = 1 - \left(\frac{1}{C_N^2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(\frac{\sum_{\forall k \in (\mathcal{S}_i \cap \mathcal{S}_j)} \mathbb{I}_{\mathcal{S}_i^k = \mathcal{S}_j^k}}{|\mathcal{S}_i \cap \mathcal{S}_j|} \right) \right) \quad (11)$$

where $\mathbb{I}$ is a binary indicator function, returning 1 if compared values are identical, otherwise 0. A commonly used method for measuring value-based diversity is computing the difference between feature values of every pair of counterfactuals in $\mathcal{S}$, which can be a biased measurement as every method tends to alter a varied number of features (due to using different mechanisms for the sparsity of explanations). So instead, we measure the difference between values of jointly changed features in every pair of explanations. This approach standardizes d_V values, enabling a fair comparison between the baselines methods.

Figure 4 illustrates the results of diversity evaluation. Although we have not explicitly formulated diversity as an objective goal, the choice of the optimization algorithm (i.e., NSGA-III) has resulted in highly diverse explanations. According to the feature-based diversity results depicted in Fig. 4a, CARE mostly tends to generate counterfactuals that are varied concerning the feature names (high d_F compared to the baselines). Besides, the result of the value-based diversity (Fig. reffig:metricsb) shows the solid performance of our approach when changing multiple features is not applicable ($d_V \gtrsim 0.7$ for the majority of data sets).

5.5.3 Simplicity of explanations

The *simplicity* of explanations is an important aspect from the end-user's perspective. Users often prefer simple and intuitive explanations that suggest straightforward guidelines toward the desired outcome. To this end, we devise the following metric to measure the simplicity of explanations based on the number of changed features:

$$\Psi = 1 - \frac{\sum_{j=1}^{m} \mathbb{I}_{x_j \neq x'_j}}{m} \quad (12)$$

where x is the original input, x' is the counterfactual instance, m is the data set's feature dimension, and $\mathbb{I}$ is a binary indicator function. Simply put, the fewer altered features, the more intuitive and straightforward explanation (i.e., higher Ψ).

Figure 4c illustrates the performance of different methods regarding the defined simplicity metric. It shows that CARE generates simple explanations for every data set ($\Psi \approx 0.8$). Compared to the baselines that either neglect the sparsity of changes or address it interchangeably using the L1-norm distance metric, CARE formulates the sparsity as a concrete objective function (i.e., $O_{sparsity}$), leading to straightforward explanations. Specifically, this function penalizes every feature alteration regardless of its type (numerical or categorical) and the magnitude of change that encourages the optimization algorithm to generate simple counterfactuals recommending few changes in the original inputs.

5.5.4 Computational complexity

Computational complexity is an essential matter of a counterfactual explanation method. Our proposed approach uses intuitive and computationally efficient objective functions to address the counterfactual desiderata. Furthermore, the modular structure enables including arbitrary modules depending on the application (CF or AR generation) and the desired properties for explanations, easing the computational burden. Importantly, CARE generates multiple explanations for a given input via a single optimization process with a reasonable execution time regardless of the ML model's complexity (i.e., a model-agnostic approach).

We have adopted the NSGA-III algorithm to generate explanations by optimizing the defined objective functions for the properties. The overall complexity of the NSGA-III algorithm is $O(MN^2)$ where M is the number of objectives,

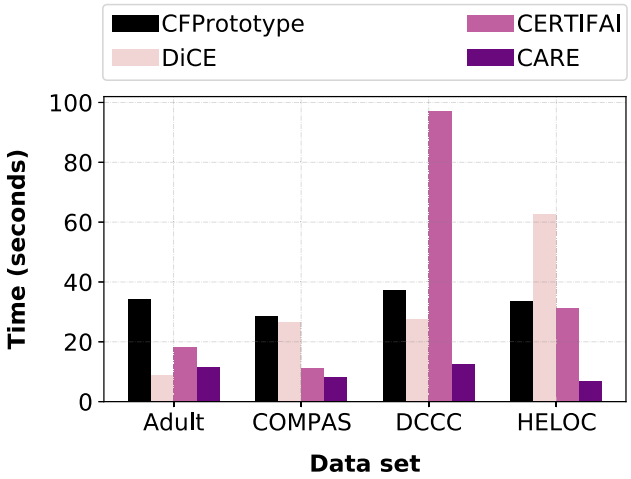


Fig. 5 Computational complexity: average required time for explaining a single instance

and N is the population size. CARE does not require creating a new model for explaining every instance; once a CARE's explainer for a black box model and its corresponding data set is built, it takes $O(MN^2)$ to explain every data point. As stated earlier, we used an adaptive mechanism to set the population size N with respect to the number of objectives M. This approach is suitable for our modular framework in which modules are arbitrarily included.

We benchmarked the computational complexity of CARE against the baseline methods. Figure 5 shows the average time spent by each algorithm for explaining a single instance. Although the employed data sets have different feature dimensions and proportions of numerical and categorical features, CARE generates explanations in a reasonable and similar amount of time compared to the baseline methods. Several factors influence the effectiveness of our approach. First, the size of the solution space in our algorithm is determined by the number of objectives, rather than the dimension of features. Therefore, CARE is less dependent on the feature dimensionality. Accordingly, the complexity of CARE is substantially reduced as fewer modules are used, for example, in the counterfactual generation setting. Second, the selected optimization algorithm does not demand extra work to convert categorical features to numerical features and treats them equally. Last but not least, the objective functions are designed in a way to perform lightweight operations in the *explaining* phase, like querying models and simple arithmetical calculations.

6 Related work

This section briefly reviews the CE and AR generation methods related to our work.

In [45], the authors propose the initial concept of counterfactual explanations and assess the extent to which they fulfill the GDPR's requirements. They pose the counterfactual generation as an optimization problem that aims to find the closest instance to an original data point that is classified as the desired class by the machine learning model. The closeness of data points is the primary objective of this technique that is measured via a relevant distance metric to the data set (e.g., L1/L2, quadratic, or customized distance functions). The stated objective is defined in a differentiable form that is optimized using a gradient-based algorithm. This results in a computationally efficient approach with limited applicability to non-differential models.

In [17], the authors overcome the challenge of generating non-actionable recourse that is not executable by the user. They propose DACE, a mixed-integer linear optimization approach equipped with a novel cost function to establish feature correlation and minimize the risk of generating outlier counterfactual instances. DACE is applicable to linear and tree ensemble models in the tabular classification setting, given that it has complete access to the model's internals.

In [16], an algorithm for generating individual actionable recourse, called REVISE, is introduced. The novelty of this approach is modeling the underlying data distribution to generate a minimal and actionable set of changes for an individual's features. The actionability of changes is ensured via traversing the latent representation of the data manifold that allows sampling high probability paths of changes close to the original input features. The approach is applicable to differential models in the image and tabular settings.

In [30], the authors propose DiCE, a counterfactual explanation method focusing on two properties: feasibility and diversity of explanations. In this approach, the feasibility is quantified using a distance metric between the original input and the potential counterfactual instances, while the diversity is captured by a novel metric based on determinantal point processes (DPP) model [22]. The metrics are combined in a loss function that enables a trade-off between feasibility and diversity, and can be optimized using the gradient decent algorithm [36]. The generated explanations undergo a post hoc filtering step to omit non-feasible explanations based on user-defined constraints. DiCE is designed for differential models in the tabular setting.

In [6], the authors present a multi-objective counterfactual explanation method, called MOC, which concentrates on the proximity and actionability properties. MOC contains four objectives to generate counterfactuals that are close to the original input and originate from the data distribution. The optimization problem is solved using NSGA-II algorithm [8] that results in a diverse set of explanations for a particular input. It is presumed that an individual is likely to find an actionable recourse among multiple explanations that comply with their preferences. This approach is applicable to any

ML model created for tabular classification and regression data sets.

In [41], an approach for generating actionable recourse for the prediction of linear classifiers (e.g., logistic regression models and linear support vector machines) is proposed. The authors formulate the problem as an Integer Program (IP) and solve it using an IP solver like CPLEX [15]. The set of mutable and immutable features related to an individual are encoded as constraints in the cost function. In addition to a list of actionable changes for an individual, the proposed method provides an evaluation of the feasibility and cost of the changes for the individual.

In [42], the authors present a model-agnostic counterfactual explanation method based on class prototypes to speed up the search for counterfactuals and enhance the interpretability of explanations. They consider a counterfactual interpretable if it lies close to the distribution of the same-class training data. The prototype for every class in the data set is obtained using either an encoder or a nearest-neighbor model. Consequently, the prototypes are incorporated in the objective function to efficiently guide the changes to the original input toward an interpretable counterfactual. The presented method is applicable to any classifier in the image and tabular settings.

In [32], a counterfactual explanation method, called FACE, is introduced to mitigate the feasibility shortcoming of explanations in real-world contexts. FACE generates counterfactuals that respect the underlying data distribution and are connected to the original inputs via high-density paths. A high-density path crosses the densely populated regions in the feature space and is identified through a neighborhood graph (e.g., KNN or ϵ-graph) constructed over the training data points. A resultant counterfactual is likely to prescribe a feasible list of changes for input features, leading to practical actions and facilitating the transition from the default state to the counterfactual state. FACE is a model-agnostic approach that explains the outcome of any image and tabular classifier.

In [31], the authors propose C-CHVAE approach that concentrates on *proximity* and *connectedness* properties to generate faithful counterfactual explanations. For every input, C-CHVAE finds multiple counterfactuals with high occurrence probability that are proximate and connected to high-density regions in the feature space modeled by an Auto-Encoder (AE). The framework is compatible with AEs that allow modeling of heterogeneous data and approximating the conditional log-likelihood of mutable and immutable features, which are considered as necessary requirements for generating attainable counterfactuals. C-CHVAE is a model-agnostic method that is applicable to tabular classifiers.

In [37], a unified model-agnostic counterfactual explanation approach, called CERTIFAI, for addressing the robustness, transparency, interpretability, and fairness of machine learning models is introduced. This approach uses a custom genetic algorithm to generate counterfactual instances. The algorithm considers the closeness of original input to the potential counterfactuals as the objective function and allows imposing arbitrary constraints for creating actionable explanations. Furthermore, by relying on the distance information obtained from counterfactuals, two metrics for evaluating the robustness and fairness of the black box model are proposed. This approach is applicable to any classifier in the image and tabular settings.

In [25], the authors propose GRACE to generate informative and understandable explanations for neural network classifiers in high-dimensional tabular data settings. This approach considers fidelity, conciseness, and informativeness constraints for generating counterfactual instances. Specifically, it uses an entropy-based forward features ranking procedure to find instance-dependent features satisfying the stated constraints. Further, it creates contrastive samples using the selected features. Finally, it produces user-friendly explanation texts based on the created samples illuminating *"why outcome X rather than Y"*.

In [27], the authors propose FOCUS, a counterfactual explanation approach built upon the optimization framework of [45] to extend its applicability for an important class of non-differentiable models, i.e., tree ensembles. FOCUS uses a probabilistic approximation approach to construct a differentiable version of a tree-based model. The approximated model can be utilized by the gradient descent algorithm to generate counterfactuals for tabular classifiers via minimal perturbation of input features.

In [44], a stochastic-control-based approach, called FASTAR, for generating sequential ARs is proposed. A sequential AR contains a sequence of action steps that lead an individual to the recourse state. The presentation of ARs as a set of discrete and sequential steps is closer to real-world actions, which helps individuals focus their effort on changing a small number of features at a time. To provide realistic ARs, the authors have considered several desiderata such as *sparsity*, *proximity*, and *actionability* in the explanation generation. Moreover, they have used hand-crafted causal constraints like *"Age and Education cannot decrease, increasing Education increases Age"* to generate feasible ARs. FASTAR is a model-agnostic and amortized approach that can be applied to any tabular classification model and generate ARs for several data points simultaneously.

7 Conclusion and future work

In this paper, we proposed CARE, a modular explanation framework for generating actionable recourse. We demonstrated the crucial role of different data-level and user-level properties and the necessity of their simultaneous consideration for creating feasible explanations. CARE overcomes

the modeling and computational complexity of the properties through intuitive and computationally efficient objective functions. It organizes the properties in a modular hierarchy that enables the arbitration and aggregation of different properties and increases the framework's flexibility in generating actionable recourse and counterfactual explanation. Finally, CARE generates explanations by making a rigorous trade-off between properties via a multi-objective optimization algorithm. Given the important role of individuals during and after recourse generation, CARE provides several approaches to incorporate their preferences and facilitate their action steps toward their desired outcome. We demonstrated the ability of our framework in explaining different predictive models and data sets originating from the employed optimization scheme. Several validation and benchmark experiments on standard data sets and black box models confirmed the efficacy of our devised approach and its superior performance compared to the baselines. In future work, we will create a human-in-the-loop mechanism for preference specification that learns from individuals' feedback to suggest practical constraints. Further, we aim to investigate the benefit of domain knowledge, formulated as knowledge graphs and taxonomies, to create more realistic and user-friendly actionable recourses.

Funding Open access funding provided by University of Oslo (incl Oslo University Hospital).

Declarations

Conflicts of interest The authors declare that there is no conflict of interest.

References

1. Barrowman, N.: Correlation, causation, and confusion. New Atlantis **43**, 23–44 (2014)
2. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104 (2000)
3. Callahan, A., Shah, N.H.: Machine learning in healthcare. In: Sheikh, A., Wright, A., Cresswell, K.M., Bates, D.W. (eds.) Key Advances in Clinical Informatics, pp. 279–291. Elsevier, Amsterdam (2017)
4. Campello, R.J., Moulavi, D., Zimek, A., Sander, J.: Hierarchical density estimates for data clustering, visualization, and outlier detection. ACM Trans. Knowl. Discov. Data (TKDD) **10**(1), 1–51 (2015)
5. Chen, P.Y., Smithson, M., Popovich, P.M.: Correlation: Parametric and Nonparametric Measures (No. 139). SAGE, Thousands Oaks (2002)
6. Dandl, S., Molnar, C., Binder, M., Bischl, B.: Multi-objective counterfactual explanations. In: International Conference on Parallel Problem Solving from Nature, pp. 448–469. Springer, Berlin (2020)
7. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2013)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
9. Dogan, Ü., Glasmachers, T., Igel, C.: A unified view on multiclass support vector classification. J. Mach. Learn. Res. **17**(45), 1–32 (2016)
10. Downs, M., Chu, J.L., Yacoby, Y., Doshi-Velez, F., Pan, W.: Cruds: counterfactual recourse using disentangled subspaces. In: ICML Workshop on Human Interpretability in Machine Learning (2020)
11. Dua, D., Graff, C.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2017). http://archive.ics.uci.edu/ml
12. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 226–231. AAAI Press (1996)
13. FICO-Community: HELOC Data Set (2018). https://community.fico.com/s/explainable-machine-learning-challenge
14. Gower, J.C.: A general coefficient of similarity and some of its properties. Biometrics **27**, 857–871 (1971)
15. Atamtürk, A., Savelsbergh, M.W.: Integer-programming software systems. Ann. Oper. Res. **140**(1), 67–124 (2005)
16. Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., Ghosh, J.: Towards realistic individual recourse and actionable explanations in black-box decision making systems. arXiv preprint arXiv:1907.09615 (2019)
17. Kanamori, K., Takagi, T., Kobayashi, K., Arimura, H.: Dace: distribution-aware counterfactual explanation by mixed-integer linear optimization. In: Bessiere, C (Ed) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. International Joint Conferences on Artificial Intelligence Organization, pp. 2855–2862 (2020)
18. Karimi, A.H., Barthe, G., Balle, B., Valera, I.: Model-agnostic counterfactual explanations for consequential decisions. In: International Conference on Artificial Intelligence and Statistics, pp. 895–905. PMLR (2020)
19. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: contrastive explanations and consequential recommendations. ACM Comput. Surv. (2022). https://doi.org/10.1145/3527848
20. Karimi, A.H., Schölkopf, B., Valera, I.: Algorithmic recourse: from counterfactual explanations to interventions (2020). arXiv preprint arXiv:2002.06278
21. Kehl, D.L., Kessler, S.A.: Algorithms in the criminal justice system: assessing the use of risk assessments in sentencing. Berkman Klein Center for Internet & Society (2017)

22. Kulesza, A., Taskar, B., et al.: Determinantal point processes for machine learning. Found. Trends Mach. Learn. **5**(2–3), 123–286 (2012)
23. Laugel, T., Lesot, M.J., Marsala, C., Detyniecki, M.: Issues with post-hoc counterfactual explanations: a discussion (2019). arXiv preprint arXiv:1906.04774
24. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: The dangers of post-hoc interpretability: unjustified counterfactual explanations (2019). arXiv preprint arXiv:1907.09294
25. Le, T., Wang, S., Lee, D.: Grace: Generating concise and informative contrastive sample to explain neural network model's prediction. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 238–248 (2020)
26. Loh, W.Y.: Classification and regression trees. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **1**(1), 14–23 (2011)
27. Lucic, A., Oosterhuis, H., Haned, H., de Rijke, M.: Focus: flexible optimizable counterfactual explanations for tree ensembles (2019). arXiv preprint arXiv:1911.12199
28. Mahajan, D., Tan, C., Sharma, A.: Preserving causal constraints in counterfactual explanations for machine learning classifiers. In: CausalML: Machine Learning and Causal Inference for Improved Decision Making Workshop, NeurIPS 2019, Dec 2019
29. McDonald, G.C.: Ridge regression. Wiley Interdiscip. Rev. Comput. Stat. **1**(1), 93–100 (2009)
30. Mothilal, R.K., Sharma, A., Tan, C.: 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 607–617 (2020)
31. Pawelczyk, M., Broelemann, K., Kasneci, G.: Learning model-agnostic counterfactual explanations for tabular data. In: Proceedings of The Web Conference, pp. 3126–3132 (2020)
32. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: Face: feasible and actionable counterfactual explanations. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 344–350 (2020)
33. ProPublica: Compas data set (2017)
34. Rasouli, P., Yu, I.C.: Explan: explaining black-box classifiers using adaptive neighborhood generation. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–9. IEEE (2020)
35. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)
36. Ruder, S.: An overview of gradient descent optimization algorithms (2016). arXiv preprint arXiv:1609.04747
37. Sharma, S., Henderson, J., Ghosh, J.: Certifai: a common framework to provide explanations and analyse the fairness and robustness of black-box models. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, AIES'20, New York, pp. 166–172. Association for Computing Machinery (2020)
38. Siddiqi, N.: Credit risk scorecards: developing and implementing intelligent credit scoring, vol. 3. Wiley, Hoboken (2012)
39. StatLib-Repository: California housing data set (1997)
40. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. Ann. Stat. **32**(2), 407–499 (2004)
41. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 10–19 (2019)
42. Van Looveren, A., Klaise, J.: Interpretable counterfactual explanations guided by prototypes. In: Machine Learning and Knowledge Discovery in Databases. Research Track, pp. 650–665. Springer: Cham (2021)
43. Verma, S., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: a review (2020). arXiv preprint arXiv:2010.10596
44. Verma, S., Hines, K., Dickerson, J.P.: Amortized generation of sequential counterfactual explanations for black-box models (2021). arXiv preprint arXiv:2106.03962
45. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harv. J. Law Technol. **31**, 841 (2017)