On the Robustness of Counterfactual Explanations to Adverse Perturbations

MARCO VIRGOLIN, Centrum Wiskunde & Informatica—the Dutch National Research Institute for Mathematics and Computer Science, the Netherlands

SAVERIO FRACAROS, Department of Mathematics and Geosciences, University of Trieste, Italy

Counterfactual explanations (CEs) are a powerful means for understanding how decisions made by algorithms can be changed. Researchers have proposed a number of desiderata that CEs should meet to be practically useful, such as requiring minimal effort to enact, or complying with causal models. We consider a further aspect to improve the usability of CEs: robustness to adverse perturbations, which may naturally happen due to unfortunate circumstances. Since CEs typically prescribe a *sparse* form of intervention (i.e., only a subset of the features should be changed), we provide two definitions of robustness, which concern, respectively, the features to change and to keep as they are. These definitions are workable in that they can be incorporated as penalty terms in the loss functions that are used for discovering CEs. To experiment with the proposed definitions of robustness, we create and release code where five data sets (commonly used in the field of fair and explainable machine learning) have been enriched with feature-specific annotations that can be used to sample meaningful perturbations. Our experiments show that CEs are often not robust and, if adverse perturbations take place (even if not worst-case), the intervention they prescribe may require a much larger cost than anticipated, or even become impossible. However, accounting for robustness in the search process, which can be done rather easily, allows discovering robust CEs systematically. Robust CEs are resilient to adverse perturbations: additional intervention to contrast perturbations is much less costly than for non-robust CEs. Our code is available at: https://github.com/marcovirgolin/robust-counterfactuals.

Additional Key Words and Phrases: counterfactual explanation, explainable machine learning, explainable artificial intelligence, robustness, algorithm development

1 INTRODUCTION

Modern Artificial Intelligence (AI) systems often rely on machine learning models such as ensembles of decision trees and deep neural networks [15, 30, 40], which are *massive* in terms of number of parameters. Massive models are appealing because, under proper training and regularization regimes, they are often unmatched by smaller models [4, 45]. However, as massive models perform myriads of computations, it can be very difficult to interpret and predict their behavior. Because of this, massive models are often called *black-box models*, and ensuring that their use in high-stakes applications (e.g., of medicine and finance) is fair and responsible can be challenging [17, 27].

The field of eXplainable AI (XAI) studies methods to dissect and analyze black-box models [1, 21] (as well as methods to generate interpretable models when possible [51]). Famous methods of XAI include feature relevance attribution [41, 50], explanation by analogy with prototypes [8, 32], and, of focus in this work, *counterfactual explanations*. Counterfactual explanations enable to reason by contrast rather than by analogy, as they show in what ways the input given to a black-box model needs to be changed for the model to make a different decision [54, 60]. A classic example of counterfactual explanation is: *"Your loan request has been rejected. If your salary was* 60 000*$ instead of* 50 000*$ and your debt was* 2500*$ instead of* 5000*$, your request would have been approved."* Remarkably, a user who obtains an unfavourable decision can attempt to overturn it by intervening according to the counterfactual explanation.

Normally, the search of counterfactual explanations is formulated as an optimization problem (see Sec. 2.1 for a formal description). Given the feature values that describe the user as starting point, we seek the minimal changes to those feature values that result in a point for which the black-box model makes a different (and oftentimes, a specific favourable) decision. We wish the changes to be minimal for two reasons: one, to learn about the behavior of the black-box model for users (points) who are similar to the current user (starting point), e.g., for matters of fairness; two, in the hope that putting the counterfactual explanation into practice with a real life intervention will require minimal effort too. For counterfactual explanations to be most useful, more desiderata than requiring minimal feature changes may need to be taken into account (see Sec. 3) [3].

In this paper, we consider a desideratum that can be very important for the usability of counterfactual explanations: *robustness to adverse perturbations*. By adverse perturbations we mean changes in feature values that happen due to unfortunate circumstances beyond the user's control, and cause intervention to be more costly than expected or even fruitless. These unfortunate circumstances can have various origins, e.g., time delays, measurement corrections, biological processes, and so on. For example, if a counterfactual explanation for improving a patient's heart condition prescribes lowering the patient's blood pressure, the chosen treatment may need to be employed for longer, or even turn out to be futile, if the patient has a genetic predisposition to resist that treatment (for more examples, see Sec. 4.1 and choices made in the coding of our experiments, in `robust_cfe/dataproc.py`).

We show that, if adverse perturbations might happen, one can and *should* seek counterfactual explanations that are robust to such perturbations. A particular novelty of our work is that we distinguish between whether perturbations impact the features that counterfactual explanations prescribe to *change* or *keep as they are*. This is because counterfactual explanations are normally required to be *sparse* in terms of the intervention they prescribe (i.e., only a subset of the features should be changed), for better usability (see Sec. 2.1). As it will be shown, making this discrimination allows to improve the effectiveness and efficiency with which robustness can be accounted for.

In summary, this paper makes the following contributions: (1) We propose two workable definitions of robustness of counterfactual explanations that concern, respectively, the features prescribed to be changed and those to be kept as they are; (2) We release code to support further investigations, where five existing data sets are annotated with perturbations and plausibility constraints that are tailored to the features and type of user seeking recourse; (3) We provide experimental evidence that accounting for robustness is important to prevent adverse perturbations from making it very hard or impossible to achieve recourse through counterfactual explanations, when adverse perturbations are sampled from a distribution (i.e., they are not necessarily worst-case ones).

2 ROBUST COUNTERFACTUAL EXPLANATIONS

In the following sections, we present the two new notions of robustness for counterfactual explanations. We begin with some preliminary assumptions and definitions that are used for the remainder of the paper. Next, we provide the definitions of robustness, one for the features that a counterfactual explanation prescribes to change, and one for the features those that are prescribed to be kept as they are.

2.1 Problem statement

A counterfactual *example* for a point $\mathbf{x} = (x_1, \dots, x_d)^\top \in \Omega^d = \mathbb{R}^{d_1} \times \mathbb{N}^{d_2}$ (i.e., a point with d_1 numerical features and d_2 categorical ones, here treated as integers) is a point $\mathbf{z} = (z_1, \dots, z_d)^\top \in \Omega^d$ such that, given a classification (black-box) machine learning model $f : \Omega^d \rightarrow \{c_1, c_2, \dots\}$ (c_i is a decision or *class*), $f(\mathbf{z}) \neq f(\mathbf{x})$. We wish $\mathbf{z}$ to be *close* to $\mathbf{x}$ under a meaningful distance function $\delta : \Omega^d \times \Omega^d \rightarrow \mathbb{R}_0^+$ that is problem-specific and meets several desiderata (see Sec. 3). For example, a commonly-used distance (capable of handling both numerical and categorical features) is Gower's (see Eq. (6)) [18], or variations thereof (see, e.g., the distance used in [20]). Often, when dealing with more than two classes, we also impose $f(\mathbf{z}) = c^\star$, i.e., of what specific class we desire $\mathbf{z}$ to be. Other times, we wish to find a *set* of counterfactual examples $\{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$, possibly of different classes, to obtain multiple means of recourse or simply gain information on the decision boundary of f nearby $\mathbf{x}$ (e.g., to explain f's local behavior) [43, 52, 60].

For the sake of readability, we provide formal definitions only for the case $\Omega^d = \mathbb{R}^d$, i.e., assuming that all features are numerical. For completeness, we include explanations of how to deal with categorical features in the running text. Furthermore, we assume the features to be independent. Even though in real-world practice this assumption rarely holds entirely, this assumption does not impact the validity of counterfactual explanations (but can make finding them less efficient), it is commonly made in literature (see Sec. 3), and allows us to greatly simplify the introduction of the concepts hereby presented. We discuss potential limitations linked to this assumption in Sec. 6.

A counterfactual *explanation* is represented by a description of how $\mathbf{x}$ needs to be changed to obtain $\mathbf{z}$. In other words, a counterfactual explanation is a prescription on what interventions should be made to *reach* the respective counterfactual example. For example, under the assumption of independence and all-numerical features, the difference $\mathbf{z} - \mathbf{x}$ is typically considered to be the counterfactual explanation for how to reach $\mathbf{z}$ from $\mathbf{x}$. What particular form counterfactual explanations take is not crucial to our discourse, and we will use $\mathbf{z} - \mathbf{x}$ for simplicity.

We proceed by considering a traditional setting, where we seek the (explanation relative to the) *optimal* $\mathbf{z}^\star$ with:

$$\begin{aligned} \mathbf{z}^\star &= \text{argmin}_{\mathbf{z}} \delta(\mathbf{z}, \mathbf{x}) \\ \textit{with} \;\; \delta(\mathbf{z}, \mathbf{x}) &:= ||\mathbf{z} - \mathbf{x}||_1 + \lambda ||\mathbf{z} - \mathbf{x}||_0 \\ \textit{and subject to} \;\; f(\mathbf{z}) &= c^\star \;\; \text{and} \;\; \mathbf{z} - \mathbf{x} \in \mathcal{P}. \end{aligned} \tag{1}$$

In other words, δ is a linear combination, weighed by λ, of the sum of absolute distances between the feature values of $\mathbf{x}$ and $\mathbf{z}$, and the count of feature values that are different between $\mathbf{x}$ and $\mathbf{z}$. Moreover, the difference $\mathbf{z}$ - $\mathbf{x}$ must abide to some plausibility constraints specified in a collection $\mathcal{P}$. For example for a private individual who wishes to be granted a loan, one of such constraints may specify that the he or she cannot reasonably intervene to change the inflation level of a currency (such a feature is called *mutable but not actionable*).

We particularly consider the $L1$-norm (i.e., the term $|| \cdot ||_1$ of δ in Eq. (1)) because it is reasonable to think that, for independent features, the total cost of intervention (i.e., the effort the user must put) is the sum of the costs of intervention for each feature separately, and that these costs grow linearly. Some works (e.g., [20, 37]) choose the $L2$-norm ($|| \cdot ||_2$, also known as Euclidean norm) instead of the $L1$-norm; the definitions of robustness given in this paper can be easily adapted for the $L2$-norm. Regarding the $L0$-norm (i.e., the term $|| \cdot ||_0$ of δ in Eq. (1)), this term explicitly promotes a form of sparsity, as it seeks to minimize how many features have a different value between $\mathbf{z}$ and $\mathbf{x}$. This is desirable because, oftentimes, the user can only reasonably intervene upon a limited number of features, as opposed to all of them (even if this amounts to a larger total cost in terms of L1).

We use the form of sparsity just mentioned to partition the features into two sets. We call the set that contains the (indices of the) features which values should *change* $C = \{i \in \{1, \ldots, d\} \mid z_i \neq x_i\}$, and its complement, i.e., the set of the (indices of the) features which values should be *kept as they are*, $\mathcal{K} = \{i \in \{1, \ldots, d\} \mid z_i = x_i\}$. Typically, because a sufficiently large λ is used, or because of $\mathcal{P}$, $\mathcal{K} \neq \emptyset$. Now, for both features in C or $\mathcal{K}$, the values of those features may be subject to perturbations caused by events that are beyond one's control. For example, consider the case of a patient who is at high risk of a certain cardiac condition, and a medical doctor who intends to treat the condition with a drug to lower the patient's blood pressure. The doctor can reasonably expect that blood pressure levels may naturally vary by some extent, possibly requiring to prolong the treatment to ensure its success. Such a situation calls for an assessment of robustness with respect to C, because possible perturbations concern a feature (blood pressure) that the doctor intends to change (with the drug for blood pressure). Furthermore, other features of the patient than those that are explicitly being treated may vary, such as vitamin deficiency levels, also causing complications. This calls for an assessment of robustness with respect to $\mathcal{K}$, because possible perturbations concern a feature (vitamin deficiency) that the doctor does not plan to change (the drug treats blood pressure and not vitamin deficiency). In the next section, we present our first notion of robustness, which concerns C.

2.2 C-robustness

We begin by focusing on the features that should be changed according to a given counterfactual explanation, i.e., the features (whose indices are) in C. Let us hypothesize that we can identify, thanks to expert knowledge, what extent of maximal perturbations can happen to each feature[1]. Depending on the problem and the feature under consideration, these perturbations can be imagined to be *relative* (e.g., due to market fluctuations, a return on investment may be smaller than anticipated by 5% of the expected value), or, more simply, *absolute* (e.g., due to complex physiological processes of a patient, a certain vitamin level may decrease by up to $5ng/mL$, irrespective of its current level). For simplicity, we will proceed by assuming that perturbations can only be absolute, and explain how to handle relative perturbations later on. We define a $2 \times d$-dimensional vector $\mathbf{p} = \left(p_1^{\{-\}}, p_1^{\{+\}} \dots, p_d^{\{-\}}, p_d^{\{+\}}\right)^\top$, where $p_i^{\{-\}} \leq 0$, and $p_i^{\{+\}} \geq 0$ represent, respectively, the smallest negative and largest positive (absolute) perturbations that can reasonably happen to the i^{th} feature. For example, if the i^{th} feature represents the blood pressure of a patient, then $p_i^{\{-\}}$ tells by how much the blood pressure might lower at most (e.g., as a consequence of dehydration) and $p_1^{\{+\}}$ tells by how much the blood pressure might raise at most (e.g., as a consequence of anti-inflammatory drug intake). In general, the magnitudes of $p_i^{\{-\}}, p_i^{\{+\}}$ need not be the same, i.e., $|p_i^{\{-\}}| \neq |p_i^{\{+\}}|$. Using $\mathbf{p}$, we can provide the following:

Definition 2.1. (Worst-case C-setback of a counterfactual example) Given a model f, a point $\mathbf{x}$, a respective counterfactual example $\mathbf{z}$, and a vector of possible perturbations $\mathbf{p}$, we call the *worst-case C-setback of* $\mathbf{z}$ the vector:

$$\mathbf{w} = (w_1, \dots, w_d)^\top \;\; where \;\; w_i := \begin{cases} p_i^{\{-\text{sign}(z_i - x_i)\}}, & \text{if } |p_i^{\{-\text{sign}(z_i - x_i)\}}| \leq |z_i - x_i| \wedge i \in C \\ -(z_i - x_i), & \text{if } |p_i^{\{-\text{sign}(z_i - x_i)\}}| > |z_i - x_i| \wedge i \in C \\ 0, & \text{otherwise, i.e., } i \notin C \end{cases} \tag{2}$$

This definition tells us how to build a $(\mathbf{z}, \mathbf{x})$-specific vector $\mathbf{w}$, with elements that are taken from $\mathbf{p}$ (with magnitude capped at $|z_i - x_i|$) or set to 0 (if the feature is not among those to change), that maximally increases the cost needed to reach $\mathbf{z}$ from $\mathbf{x}$. In fact, we can interpret the meaning of $\mathbf{w}$ by considering that the point obtained by $\mathbf{z} + \mathbf{w}$ is the point to which $\mathbf{w}$ maximally *pushes* us *away* from $\mathbf{z}$ and *back* towards $\mathbf{x}$ (hence the term *worst-case*). Let us analyze the cases put by Def. 2.1 for the values that w_i can have. We begin with the third case. We set $w_i = 0$ for $i \notin C$ because robustness in terms of features in $\mathcal{K}$ is treated separately Sec. 2.3 (without loss of generality under the assumption that features are independent of one another). The other two cases are more interesting as w_i is not necessarily null. The first case is $w_i = p_i^{\{-\text{sign}(z_i - x_i)\}}$. Here, sign choices play a key role. The perturbations of the i^{th} feature that are specified by $p_i^{\{-\}}, p_i^{\{+\}} \in \mathbf{p}$ are, depending on whether they have the same or the opposite sign of $z_i - x_i$, *fortunate* or *unfortunate*, respectively. Back to the example of treating a patient to lower his or her blood pressure, accidental events that cause the blood pressure to decrease are actually welcome, as reaching the desired level will then require a smaller intervention. Conversely, and of interest in this paper, events that cause the blood pressure to raise will result in extra

[1] Robustness is commonly set to account for worst-case scenarios, see, e.g., https://en.wikipedia.org/wiki/Robust_optimization.

intervention to be needed. This means that we must consider the element of $\mathbf{p}$ that is of opposite sign of $z_i - x_i$; by exploiting the choice of notation we made for the elements of $\mathbf{p}$, that element is $p_i^{\{-\text{sign}(z_i - x_i)\}}$. The last case to consider is the second one, which aims at limiting the extent of the setback. Note that the first and second cases, together with their conditions, can be re-written as $w_i = -\text{sign}(z_i - x_i) \max\left(|p_i^{\{-\text{sign}(z_i - x_i)\}}|, |z_i - x_i|\right)$. The reason why we cap the magnitude of the perturbation w_i is to prevent $z_i + w_i$ to be further away from z_i than the starting value x_i. In fact, from a practical perspective, it can be unreasonable to make an intervention for which perturbations can have a larger effect than the intervention itself. For example, if the effect of natural fluctuations of the blood pressure can prevail upon the effect a certain blood pressure treatment, then a doctor would most probably dismiss the idea of using that treatment. Moreover, from a theoretical perspective, degenerate situations can happen if we allow $|w_i| > |z_i - x_i|$. For example, w_i can even become advantageous. Consider, for the sake of the argument, a one-dimensional space (here, $\mathbf{x} = x_1 = x$), where $x = 4, z = 6, f(x) = 0, f(z) = c^\star = 1$, and the decision boundary of f is such that the desired class $c^\star = 1$ is given for points ≤ 1 and ≥ 6 (e.g., some governments provide financial benefits only for incomes that are sufficiently low or high). We intend to act to increase x by $z - x = +2$ to obtain a point ≥ 6. If, however, $w = -4$ (i.e., $|w| > |z - x|$), then $z + w = 6 - 4 = 2$. Thus, it can be more advantageous to intervene with -1 to obtain a point ≤ 1, rather than attempting to obtain a point ≥ 6.

For intermediate situations, i.e., not worst-case ones, we define a C-setback of $\mathbf{z}$ to be the vector $\tilde{\mathbf{w}}$ that is built similarly to $\mathbf{w}$, except for allowing that $|\tilde{w}_i| \leq |w_i|$, as long as $\exists i : \tilde{w}_i \neq 0$ (else $\tilde{\mathbf{w}}$ would be the zero-vector and have no effect). Regarding plausibility constraints, it is reasonable to assume that if $\mathbf{z} - \mathbf{x} \in \mathcal{P}$, then $\mathbf{z} + \tilde{\mathbf{w}} - \mathbf{x} \in \mathcal{P}$ holds as well, since the point $\mathbf{z} + \tilde{\mathbf{w}}$ "preceeds" $\mathbf{z}$ when moving from $\mathbf{x}$ to $\mathbf{z}$. However, this is not true in general: in principle, $\mathcal{P}$ may impose implausible regions between $\mathbf{z}$ and $\mathbf{x}$. In that case, plausibility constraints should be taken into account when building $\tilde{\mathbf{w}}$. The discourse made so far holds for absolute perturbations. For an i^{th} feature for which perturbations should be relative, e.g., to z_i, then it suffices to change Eq. (2) of Def. 2.1 to consider $z_i \times p_i^{\{-\text{sign}(z_i - x_i)\}}$ instead of $p_i^{\{-\text{sign}(z_i - x_i)\}}$. Moreover, we only described the case for numerical features. For an i^{th} feature that is categorical, decreases or increases $p_i^{\{-\}}, p_i^{\{+\}}$ are not meaningful. Rather, $\mathbf{p}$ should contain elements that represent what categorical perturbations are possible for that feature (in absolute terms or, possibly, relative to the categories of x_i or z_i).

Perhaps the most interesting scenario for considering (worst-case) C-setbacks is when dealing with $\mathbf{z}^\star$: the optimal counterfactual example, i.e., the one that is (ideally) provided to the user. The following simple result holds for $\mathbf{z}^\star$:

Proposition 2.2. *For any C-setback $\tilde{\mathbf{w}}$ of $\mathbf{z}^\star$, $f(\mathbf{z}^\star + \tilde{\mathbf{w}}) \neq c^\star$.*

Proof. We use *reduction ad absurdum*. Let us assume the opposite of what said in Proposition 2.2, i.e., there exists $\tilde{\mathbf{w}}$ such that $f(\mathbf{z}^\star + \tilde{\mathbf{w}}) = c^\star$. Let $\mathbf{z}' := \mathbf{z}^\star + \tilde{\mathbf{w}}$, and so $f(\mathbf{z}') = c^\star$. By construction of $\tilde{\mathbf{w}}$, $\delta(\mathbf{z}', \mathbf{x}) = \delta(\mathbf{z}^\star + \tilde{\mathbf{w}}, \mathbf{x}) < \delta(\mathbf{z}^\star, \mathbf{x})$. In other words, $\mathbf{z}'$ is of the desired class and is closer to $\mathbf{x}$ than $\mathbf{z}^\star$ is. This contradicts the fact that $\mathbf{z}^\star$ is optimal. □

Now, because of Proposition 2.2, we are *guaranteed* that if a C-setback $\tilde{\mathbf{w}}$ happens to $\mathbf{z}^\star$, the resulting point will no longer be classified as $c^\star$. However, it is important to note that, with additional intervention, the user may still be able to reach $\mathbf{z}^\star$. Under the $L1$-norm (as per the choice of δ in Eq. (1)), the cost associated with such additional intervention is simply $||\tilde{\mathbf{w}}||_1$. Now, recall that a $\tilde{\mathbf{w}}$ depends the counterfactual example $\mathbf{z}$ for which it is generated, since $\tilde{w}_i \neq 0$ if $z_i \neq x_i$, i.e., $i \in C$. Thus, there may exist a different counterfactual example $\mathbf{z}'$ for $\mathbf{x}$ that is also classified as $c^\star$, for which the features to change are different from the ones of $\mathbf{z}^\star$, and so is the worst-case C-setback. In particular, if adverse perturbations take place, additional intervention to reach $\mathbf{z}'$ may entail a smaller cost than to reach $\mathbf{z}^\star$. Ideally, however, we wish that $\mathbf{z}^\star$ is the best counterfactual example to pursue even if a worst-case C-setback were to happen.

Definition 2.3. (Optimal C-robust counterfactual example) Given a model f, a point $\mathbf{x}$, and a vector $\mathbf{p}$, the optimal counterfactual example $\mathbf{z}^\star := \text{argmin}_{\mathbf{z}}\delta(\mathbf{z}, \mathbf{x})$ is *optimal C-robust* if $\mathbf{z}^\star$ is optimal also under worst-case C-setbacks, i.e.,

$$\text{argmin}_{\mathbf{z}-\mathbf{w}}\delta\left((\mathbf{z}-\mathbf{w}), \mathbf{x}\right) = \mathbf{z}^\star - \mathbf{w}^\star, \tag{3}$$

where $\mathbf{w}^\star$ is the worst-case C-setback of $\mathbf{z}^\star$.

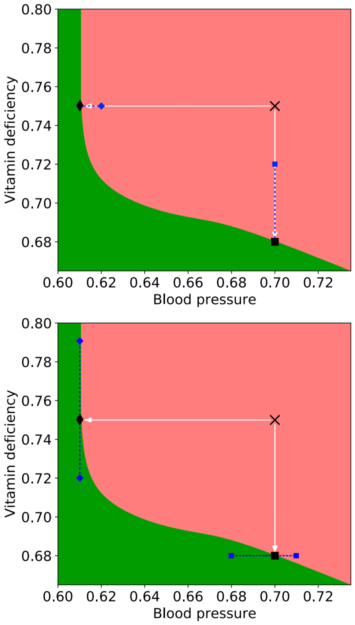

Fig. 1. Examples of C- (top) and $\mathcal{K}$-robustness (bottom). The red and green areas represent *high risk* and *low risk* classifications of a cardiac condition according to a model f. The patient, represented by ×, is at high risk. The closest (and thus optimal) counterfactual example is ■ and concerns treating vitamin deficiency (white arrow pointing down). Another counterfactual example is ♦ and concerns treating blood pressure (white arrow pointing left).
Top panel: Worst-case C-setbacks are shown (blue dashed segments); ■ is *not* optimal C-robust because, under worst-case C-setbacks, reaching ♦ is less costly.
Bottom panel: Perturbations to the features to keep stable are shown (blue dashed segments); ■ is not $\mathcal{K}$-robust because there exist perturbations in blood pressure that lead to high risk; instead, ♦ is $\mathcal{K}$-robust.

This definition gives us a way to assess whether the optimal counterfactual example remains the best to reach also when taking into consideration worst-case C-setbacks. Note that in Eq. (3) setbacks are subtracted from counterfactual examples when computing δ, to account for the fact that the cost should increase (this by construction of $\mathbf{w}$, see the signs in Def. 2.1). To translate the notion of optimal C-robust counterfactual *examples* to *explanations*, we simply say that a counterfactual explanation is optimal C-robust if the respective counterfactual example is optimal C-robust.

We conclude this section by providing an example.

Example 2.4. (C-robustness for treating a cardiac condition) Consider the top panel of Fig. 1, depicting a fictitious situation of a patient who is at high risk of a certain cardiac condition. The patient is represented by $\mathbf{x} = (.70, .75)^\top$, where the feature values concern, in order, blood pressure and vitamin deficiency. Currently, $f(\mathbf{x}) =$ *high risk* (as opposed to *low risk*). The medical doctor attending to the patient finds a counterfactual example $\mathbf{z}^v = (.70, .68)^\top$, i.e., low risk can be obtained by treating vitamin deficiency, and an alternative counterfactual example $\mathbf{z}^b = (.61, .75)^\top$, i.e., low risk can be obtained by treating blood pressure. For the sake of the argument, we assume that the two treatments cannot be performed at the same time. The intervention $\mathbf{z}^v - \mathbf{x} = (0, -.07)^\top$ is less costly than $\mathbf{z}^b - \mathbf{x} = (-.09, 0)^\top$, thus $\mathbf{z}^v$ is the optimal counterfactual example. Is $\mathbf{z}^v$ optimal C-robust? The doctor knows that vitamin deficiency can increase by up to +.04 and blood pressure by up to +.01, due to natural physiological events. Thus, in a worst-case scenario, the cost of treating vitamin deficiency is .11, while the one of treating blood pressure is .10. Hence, $\mathbf{z}^v$ is not optimal C-robust.

2.3 $\mathcal{K}$-robustness

We now consider $\mathcal{K}$, i.e., the set concerning the features that should be kept to their current value. Like before, we assume that a vector of (for now, only absolute and none relative) perturbations $\mathbf{p}$ is given. Differently from before, rather than considering worst-case scenarios, we now consider a notion of *neighborhood* of $\mathbf{z}$ under $\mathbf{p}$ and $\mathcal{K}$:

Definition 2.5. ($\mathcal{K}$-neighborhood and $\mathcal{K}$-neighbors of a counterfactual example) Given a model f, a point $\mathbf{x}$, a respective counterfactual example $\mathbf{z}$, and a vector of possible perturbations $\mathbf{p}$, the $\mathcal{K}$-neighborhood of $\mathbf{z}$ under $\mathbf{p}$ is the set:

$$N := \left\{ \mathbf{z}' \mid \begin{array}{l} z'_i \in [z_i + p_i^{\{-\}}, z_i + p_i^{\{+\}}] \text{ if } i \in \mathcal{K} \\ z'_i = z_i \text{ otherwise} \end{array} \right\}. \tag{4}$$

A point $\mathbf{z}' \in N$ such that $\mathbf{z}' \neq \mathbf{z}$ is called a $\mathcal{K}$-neighbor of $\mathbf{z}$.

In other words, N is a box (i.e., hyper-rectangle) whose boundary is defined by summing to $\mathbf{z}$, for the features in $\mathbf{K}$, all possible combinations of respective elements from $\mathbf{p}$. The dimensionality of N is lower than d because some features are in C and not in $\mathcal{K}$. For example, consider the case of having three features ($d = 3$), meaning that we operate on a three-dimensional space where the decision boundaries of f are surfaces that partition the space. There, with $|C| = 3, |\mathcal{K}| = 0$ (three features to act upon and none to bypass), N will be the empty set; with $|C| = 2, |\mathcal{K}| = 1$, N will be a one-dimensional segment; with $|C| = 1, |\mathcal{K}| = 2$, N will be a two-dimensional rectangle. The case $|C| = 0, |\mathcal{K}| = 3$ cannot possibly exist because, if no feature should change, then $\mathbf{z} = \mathbf{x}$, which violates the definition of counterfactual example ($f(\mathbf{z}) \neq f(\mathbf{x})$). To consider plausibility constraints in Def. 2.5, we can further ask that any $\mathbf{z}' \in N$ satisfies $\mathbf{z}' - \mathbf{x} \in \mathcal{P}$. Relative perturbations can be taken into account similarly to how explained in Sec. 2.2 for building worst-case C-setbacks; i.e., if $p_i^{\{-\}}, p_i^{\{+\}}$ describe perturbations that are relative to z_i, then Eq. (4) should be changed so that z'_i takes values within the interval $[z_i + z_i \times p_i^{\{-\}}, z_i + z_i \times p_i^{\{+\}}]$ (if $i \in \mathcal{K}$). Furthermore, for categorical features, the neighborhood can be built by swapping the categorical feature values of $\mathbf{z}$ with possibilities opportunely listed in $\mathbf{p}$.

We use N to define the concept of $\mathcal{K}$-robustness:

Definition 2.6. ($\mathcal{K}$-robust counterfactual example) Given a model f, a point $\mathbf{x}$, and a vector $\mathbf{p}$, a counterfactual example $\mathbf{z}$ is a *$\mathcal{K}$-robust counterfactual example* if $\nexists \mathbf{z}' \in N(\mathbf{z}, \mathbf{p})$ such that $f(\mathbf{z}') \neq f(\mathbf{z})$.

Informally, this definition says that $\mathbf{z}$ is $\mathcal{K}$-robust if the decision boundary surrounding $\mathbf{z}$ is sufficiently loose with respect to the features to keep as they are that, if perturbations occur to those features, the decision remains the same of $f(\mathbf{z})$. For a counterfactual explanation to be $\mathcal{K}$-robust, we simply ask that the respective counterfactual example is $\mathcal{K}$-robust. The attentive reader may have noticed that, for the case of C-robustness, we only needed to consider worst-case C-setbacks to decide whether a counterfactual example is optimal C-robust. Conversely, in Def. 2.6 we ask that *any* point $\mathbf{z}' \in N$ abides $f(\mathbf{z}') = f(\mathbf{z})$ and not only those on the boundary of N, to determine whether $\mathbf{z}$ is $\mathcal{K}$-robust. In other words, we are claiming that for $\mathcal{K}$-robustness we cannot efficiently consider only some form of worst-case scenarios. Recall that the reason why C-robustness can be assessed only in worst-case conditions follows from Proposition 2.2, which says that there cannot exist points of class $c^\star$ between $\mathbf{x}$ and the optimal counterfactual example $\mathbf{z}^\star$. Unfortunately, a similar result does not exist for perturbations to the features to keep as they are:

Proposition 2.7. *For a general f, information on the classification of a $\mathcal{K}$-neighbor (e.g., that $f(\mathbf{z}') = f(\mathbf{z})$ for $\mathbf{z}'$ on the boundary of N) provides no information about the classification of another $\mathcal{K}$-neighbor (e.g., that $f(\mathbf{z}') \neq f(\mathbf{z}'')$ for $\mathbf{z}''$ in the interior of N).*

Proof. We cannot preclude that the model f is, for example, a neural network. Under the universal approximation theorem [25], f may represent any function. Thus, f may represent a *Swish cheese*-like function, where for example $f(\mathbf{z}') \neq f(\mathbf{z}'')$ with $\mathbf{z}'' := \mathbf{z}' + \mathbf{e}$ and $\mathbf{e} = (\varepsilon_1, \ldots, \varepsilon_d)^\top$ different from the zero-vector, however small $|\varepsilon_i|, \forall i$. □

Because of Proposition 2.7, an exact assessment of $\mathcal{K}$-robustness may not be possible. Unless one has guarantees on the behavior of f (which is often not the case if f is a black-box or proprietary model), all points in N must be checked.

If at least one of the features in $\mathcal{K}$ takes values in $\mathbb{R}$, N is uncountable and thus an exact evaluation of $\mathcal{K}$-robustness becomes impossible. Thus, we propose to approximate the assessment of $\mathcal{K}$-robustness with Monte Carlo sampling. Let $\mathbf{1}_{f(\mathbf{z})} : N \rightarrow \{0, 1\}$ be the indicator function that returns 1 for $\mathcal{K}$-neighbors that share the same class of $\mathbf{z}$ (i.e., $f(\mathbf{z})$), and 0 for those that do not. Taken a random sample of m $\mathcal{K}$-neighbors, we define the following score:

$$\mathcal{K}\text{-robustness score}(\mathbf{z}, m) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{f(\mathbf{z})}(\mathbf{z}'_i). \tag{5}$$

If $\mathcal{K}$-robustness score$(\mathbf{z}, m) < 1$, then $\mathbf{z}$ is not $\mathcal{K}$-robust. If $\mathcal{K}$-robustness score$(\mathbf{z}, m) = 1$, we are not guaranteed that $\mathbf{z}$ is $\mathcal{K}$-robust. Still, this score can be used to determine which counterfactual examples are preferable to pursue in that they are associated with a smaller risk that adverse perturbations will cause trouble.

We remark that a lack of $\mathcal{K}$-robustness may be a critical outcome for perturbations that cannot plausibility be corrected, e.g., because the respective features are mutable but not actionable. In fact, unfortunate perturbations to those features may cause all efforts to reach the intended $\mathbf{z}$ to be in vain. If instead the features in $\mathcal{K}$ are actionable and plausibility constraints allow it, the user can decide to intervene again, to reach $\mathbf{z}$ from $\mathbf{z}'$. In such situation, new sets C and $\mathcal{K}$ will be defined for that subsequent step. We assume that the cost associated with such additional intervention is $\delta(\mathbf{z}, \mathbf{x}) = ||\mathbf{z}' - \mathbf{z}||_1 + \lambda||\mathbf{z}' - \mathbf{z}||_0$ (see Eq. (1)).

We conclude this section with the following example:

Example 2.8. ($\mathcal{K}$-robustness for treating a cardiac condition) Consider the bottom panel of Fig. 1, again about the fictitious situation of a patient who is at high risk of a certain cardiac condition. Like in Example 2.4, the optimal counterfactual example is $\mathbf{z}^v = (.70, .68)^\top$ and concerns treating vitamin deficiency. However, the doctor knows that blood pressure can be perturbed by $p_1^{\{-\}} = -.02$ and $p_1^{\{+\}} = +.01$. Since there exist $\mathcal{K}$-neighbors of $\mathbf{z}^v$ for which f deems *high risk*, $\mathbf{z}^v$ is *not* $\mathcal{K}$-robust. The $\mathcal{K}$-robustness score of $\mathbf{z}^v$ is $\frac{.02}{.01+.02} = \frac{2}{3}$ (assuming uniformly random perturbations and sufficiently large m). The doctor then considers another counterfactual example, $\mathbf{z}^b = (.61, .75)^\top$. Even if the vitamin deficiency can fluctuate according to $p_2^{\{-\}} = -.03$ and $p_2^{\{+\}} = +.04$, f classifies all points in the neighborhood of $\mathbf{z}^b$ $N = [(.61, .75 - .03), (.61, .75 + .04)]$ to be low risk, like $\mathbf{z}^b$. Thus, $\mathbf{z}^b$ is $\mathcal{K}$-robust.

3 RELATED WORK

A number of works in literature propose several new desiderata that are largely orthogonal to our notions of robustness but can be important to enhance the practical usability of counterfactual explanations. For example, Dandl et al. [9] consider, besides proximity of $\mathbf{z}$ to $\mathbf{x}$ according to different distances, whether other training points $\mathbf{x}'$ are sufficiently close to $\mathbf{z}$ for it to reasonably belong to the training data distribution. A similar desideratum is considered in [52] and [56]; the latter work employing neural autoencoders to that end. Laugel et al. [38, 39] require that $\mathbf{z}$ can always be reached from a training point $\mathbf{x}'$ without having to cross the decision boundary of f, for $\mathbf{z}$ not to be the result of an artifact in the decision boundary of f. In [29] and [44], counterfactual explanations are studied through the lens of causality. For recent surveys on counterfactual explanations, the reader is referred to [54, 57].

We now focus on works that deal with some notion of robustness and/or perturbations explicitly. The work by Karimi et al. [28] extends [29] to consider possible uncertainties in causal modelling. In [53], it is shown that a malicious actor can, in principle, jointly optimize small perturbations and the model f such that, when applying the perturbations to points of a specific group (e.g., white males), the respective counterfactual explanations are much less costly than normal (in fact, counterfactual explanations are conceptually similar to adversarial examples, see, e.g., [2, 14, 47]). Some works consider forms of robustness of counterfactual explanations with respect to changes of f (e.g., whether $\mathbf{z}$ is still classified as $c^\star$ if f' is used instead of f) [48] or updates to f (e.g., after data distribution shift of temporal or geospatial nature) [13, 49]. In [42], robustness of counterfactual explanations is studied in the context of differentially-private support vector machines. Finally, Dominguez et al. [11] consider whether counterfactual explanations remain valid in presence of uncertainty on $\mathbf{x}$, and also account for causality. To the best of our knowledge, there exists no other work than ours that discerns between, and presents results for, features to change and features to keep as they are ($\mathcal{C}$ and $\mathcal{K}$). Moreover, existing works typically consider whether robustness helps preventing counterfactual explanations from becoming invalid (i.e., whether applying the perturbation causes f to predict a different class than $c^\star$). Here, we further consider that additional intervention may be possible, and assess the associated cost.

4 EXPERIMENTAL SETUP

In this section, we present the experimental setup we used to assess the $\mathcal{C}$- and $\mathcal{K}$-robustness of counterfactual examples, for different data sets.

4.1 Data sets

Table 1 shows summary information on the data sets we consider. For each data set, we make an assumption on the type of user who seeks recourse, e.g., the user could be a private individual seeking to increase his or her income, or a company seeking to improve the productivity of its employees. This allows us to define the desired class $c^\star$ and the set of plausibility constraints $\mathcal{P}$ on what interventions are reasonably plausible. We named the data sets in Table 1 to represent their purpose. Originally, *Credit risk* (abbreviated to Cre) is known as *South German Credit Data* [19], which is a recent update that corrects inconsistencies in the popular *Statlog German Credit Data* [24]. *Income* (Inc) is often called *Adult* or *Census income* [33, 34]. *Housing price* (Hou) is also known as *Boston housing* [23] and is often used for research on fairness and interpretability because one of its features raises ethical concerns [7]. *Productivity* (Pro) concerns the productivity levels of employees producing garments [26]. Lastly, *Recidivism* (Rec) is a data set collected by an investigation of ProPublica about possible racial bias in the commercial software *COMPAS*, which intends to estimate the risk that an inmate will re-offend [36]. Examples of recent works in fair and explainable machine learning that adopted (some of) these data sets (each) are [10, 11, 20, 31, 35, 38, 58].

We pre-process the data sets similarly to how done often in the literature. This includes, e.g., removal of redundant features and of observations with missing values, and limiting the number of observations considered for Rec. Furthermore, we annotate each data set with possible perturbations for each feature, i.e., we define the vector $\mathbf{p}$. Numerical features can have perturbations that increase or decrease the feature value, in absolute or relative terms; we compute relative perturbation with respect to $\mathbf{z}$. For example, for the numerical feature *capital-gain* of Inc, we assume that perturbations can happen that lead up to a relative 5% increase or 10% decrease of that feature, based on the value to achieve for that feature. For categorical features, we define only absolute perturbations, i.e., possible changes of category are not conditioned to the current category. Clearly, many of the choices we made to build $\mathbf{p}$ are subjective; we elaborate on this in Sec. 6. We sample the amount of perturbation using an uniform or normal distribution, as specified in the experiments below. As mentioned before, we also define plausibility constraints $\mathcal{P}$ for each data set. The constraints we define are each specific to a single feature, under the assumption of feature independence. For an i^{th} numerical feature, possible constraints are $z_i - x_i \geq 0$, $z_i - x_i \leq 0$, $z_i - x_i = 0$, and *none*. For an i^{th} categorical feature, possible constraints are $z_i = x_i$ and *none*. We remark that with such type of constraints, it is *always* plausible to reach $\mathbf{z}$ with additional intervention after a $\mathcal{C}$-setback takes place. Full details about our pre-processing and definition of $\mathbf{p}$ and $\mathcal{P}$ are documented in the form of comments in our code, in `robust_cfe/dataproc.py`.

Table 1. Considered data sets, where n and d (resp., d_2) indicate the number of observations and features (only categorical) after pre-processing. The column $c^\star$ is the desired class for the (simulated) user. Plausib. constr. reports the number of plausibility constraints that allow features to only increase ($\geq$), remain equal ($=$), and decrease ($\leq$). The column Perturb. reports the number of perturbations concerning numerical (N) and categorical (C) features. Finally, f's acc. reports the average (across five folds) test accuracy of random forest.

Dataset (abbrev.)	n	d	d_2	Classes	User	$c^\star$	Plausib. constr.	Perturb.	f's acc.
Credit risk (Cre)	1000	20	6	High, low	Individual	Low	≥:3, =:8, ≤:0	N:6, C:0	0.76
Income (Inc)	1900	12	7	High, low	Individual	High	≥:2, =:3, ≤:0	N:4, C:4	0.83
House price (Hou)	510	13	1	High, low	Municipality	Low	≥:0, =:3, ≤:1	N:11, C:0	0.93
Productivity (Pro)	1200	12	5	High, med., low	Company	High	≥:0, =:0, ≤:0	N:5, C:2	0.79
Recidivism risk (Rec)	2000	10	6	High, low	Inmate	Low	≥:2, =:2, ≤:0	N:3, C:2	0.80

Lastly, we adopt a stratified five-fold cross-validation, using random forest as black-box machine learning model f [6]. Each model is obtained by grid-search hyper-parameter tuning, see Appendix A for details. For the discovery of counterfactual examples, we consider observations $\mathbf{x}$ such that $f(\mathbf{x}) \neq c^\star$, from the test sets of the cross-validation.

4.2 Counterfactual search algorithm

We do not assume that f is differentiable (in fact, random forest is not) nor to have access to its gradients. Thus, algorithms that require gradients to discover counterfactual examples (e.g., like those in [11, 60]) are not an option. We initially considered three gradient-free algorithms from the literature to search for counterfactual examples/explanations, namely Growing Spheres (GrSp) [37], LOcal Rule-based Explanations (LORE) [20] (these two have been used recently, e.g., in [5, 39]), and the implementation of the Nelder-Mead method (NeMe) [16, 46] by SciPy [59]. More details on these algorithms are given in Appendix B. Another interesting algorithm is the one by [55], which relies on integer-programming. However, we did not consider this algorithm because it only recombines the feature values that are present in the data and does not generate new ones.

Since the considered algorithms have some limitations (see Appendix B), we implemented our own counterfactual search algorithm which is a genetic algorithm that we named Counterfactual Genetic Search (CoGS)[2]. We built CoGS to be capable of handling both categorical and numerical features, consider plausibility constraints of the form described above (Sec. 4.1), and account for $\mathcal{C}$- and $\mathcal{K}$-robustness (as explained in Sec. 4.3 below). CoGS is written in Python due to the popularity of the language and heavily relies on matricial operations with Numpy [22] for fast computations. We found that, under relatively standard settings (detailed in Appendix B.2), CoGS outperforms the other algorithms in terms of success in discovering counterfactual examples (i.e., such that $f(\mathbf{z}^\star) = c^\star$), execution speed, and proximity of the *best-found* $\mathbf{z}^\star$ to $\mathbf{x}$ (none of the algorithms guarantees optimality of $\mathbf{z}^\star$). These results are reported in Appendix C.1. We thus consider only results for CoGS in the remainder of the paper. Since CoGS is stochastic, we repeat its execution five times, and take the best-found counterfactual example out of the five attempts to be $\mathbf{z}^\star$.

4.3 Loss

We use the following loss to drive the search of counterfactual examples (where $f(\mathbf{z})$ and $c^\star$ are treated as integers):

$$\frac{1}{2}\gamma(\mathbf{z},\mathbf{x}) + \frac{1}{2}\frac{||\mathbf{z}-\mathbf{x}||_0}{d} + ||f(\mathbf{z}) - c^\star||_0, \text{where } \gamma(\mathbf{z},\mathbf{x}) = \frac{1}{d}\left(\sum_i^{d_1} \frac{|z_i - x_i|}{\max_i - \min_i} + \sum_j^{d_2} ||z_j - x_j||_0\right). \quad (6)$$

[2] https://github.com/marcovirgolin/cogs

The term γ in the equation above is Gower's distance [12, 18], where features indexed by i are numerical and those indexed by j are categorical (with values treated as integers); the maximal and minimal values of a numerical feature, $\max_i$ and $\min_i$, can be taken from the (training) data set or, as done in our case, are provided as extra annotations of the data sets. The term $||\mathbf{z} - \mathbf{x}||_0/d$ promotes sparsity of intervention and, like Gower's distance, ranges from zero to one. The third and last term requires the execution of the machine learning model f, and simply returns zero when $f(\mathbf{z}) = c^\star$ and one when $f(\mathbf{z}) \neq c^\star$.

When plausibility constraints $\mathcal{P}$ are enforced, CoGS only searches for feasible counterfactual examples that meet $\mathcal{P}$ (see Appendix B.1). Moreover, when optimizing for C-robustness, worst-case C-setbacks are computed on the fly for the candidate $\mathbf{z}$ and their contribution is used to update the contribution of γ to the loss function. When optimizing for $\mathcal{K}$-robustness, we compute the $\mathcal{K}$-robustness score with Eq. (5) and add $\frac{1}{2}$ (1 − $\mathcal{K}$-robustness score) to the loss. In the results presented below, we use $m = 64$ to compute the $\mathcal{K}$-robustness score; an analysis on the impact of m is provided in Appendix C.2.

5 EXPERIMENTAL RESULTS

We provide experimental results to answer what we believe to be some of the most important research questions: (RQ1) *Do we need to account for robustness to discover robust counterfactual examples?* (RQ2) *Does a lack of robustness compromise the feasibility of correcting perturbations with additional intervention?* (RQ3) *Are robust counterfactual explanations advantageous in terms of additional intervention cost?* These questions are addressed, in order, in the next subsections. Because of space limitations, a number of additional results is reported in Appendix C. We always apply $\mathcal{P}$ in the following experiments. We remark that in all our experiments, CoGS always succeeded in discovering a counterfactual example for which f predicts $c^\star$.

5.1 (RQ1) Do we need to account for robustness to discover robust counterfactual examples?

Table 2 shows the frequency with which robust counterfactual examples are discovered accidentally. To realize this, we compare the best-found counterfactual example $\mathbf{z}^\star$ that is discovered by CoGS when robustness *is not* accounted for, and the one that is found when C- or $\mathcal{K}$-robustness *is* accounted for. If the two match, then we say that a robust counterfactual example can in fact be discovered by accident. Since numerical feature values may differ only slightly between two best-found counterfactual examples, we consider the values to match if they are sufficiently close to each other, according to a tolerance level of 5% of the range of that feature; results with different tolerance levels are shown in Appendix C.3.

The first row of Table 2 reports how often the (supposedly) optimal $\mathbf{z}^\star$ happens to be optimal C-robust, i.e., it matches the counterfactual example that is discovered when accounting for C-robustness in the loss. As it can be seen, the result depends strongly on the data set in consideration. For example on Inc, best-found counterfactual examples are very rarely optimal C-robust (less than 5% of the times). Conversely, on Hou, almost 85% of them is. The middle row shows whether best-found counterfactual examples happen to be $\mathcal{K}$-robust (or better, have high $\mathcal{K}$-robustness score). Like for C-robustness, the result depends on the data set. Importantly, the data sets where the frequencies are high for C-robustness and $\mathcal{K}$-robustness are not necessarily the same. On Inc, best-found counterfactual examples are rarely optimal C-robust, but match relatively often (0.40 on average) with counterfactual examples discovered when penalizing low $\mathcal{K}$-robustness scores. This should not be surprising because C- and $\mathcal{K}$-robustness are orthogonal to each other under the assumption of feature independence. The last row shows how often best-found counterfactual examples happen to be both C- and $\mathcal{K}$-robust. The frequencies are clearly always lower than for the previous triplets of

Table 2. Mean ± standard deviation of the frequency with which the best-found counterfactual example is accidentally $\mathcal{C}$- or $\mathcal{K}$-robust.

Robustness	Cre	Inc	Hou	Pro	Rec
Only $\mathcal{C}$	0.42 ± 0.07	0.04 ± 0.02	0.84 ± 0.09	0.57 ± 0.06	0.37 ± 0.07
Only $\mathcal{K}$	0.44 ± 0.03	0.40 ± 0.02	0.63 ± 0.17	0.37 ± 0.06	0.08 ± 0.03
Both $\mathcal{C}, \mathcal{K}$	0.27 ± 0.03	0.00	0.54 ± 0.21	0.26 ± 0.05	0.06 ± 0.04

rows. Hou is the only data set for which the frequency of discovering a counterfactual example that happens to be both $\mathcal{C}$- and $\mathcal{K}$-robust by chance is relatively large (above 50%). In general, that is not the case.

5.2 (RQ2) Does a lack of robustness compromise the feasibility of correcting perturbations with additional intervention?

At this point, current works on the robustness of counterfactual explanations typically consider the extent by which robustness helps preventing the invalidation of counterfactual explanations (see Sec. 3). In other words, they consider whether the point $\mathbf{z}'$ that is given by perturbing $\mathbf{z}^\star$ is still classified as $c^\star$, when $\mathbf{z}^\star$ is or is not robust. For completeness, we report on this in Appendix C.4. However, current works do not consider whether an additional intervention that allows to correct the perturbation and obtain $c^\star$ might exist.

Fig. 2 shows the frequency with which reaching $\mathbf{z}^\star$ remains possible after random perturbations take place. The frequency is computed by applying, for any given $\mathbf{z}^\star$, 100 perturbations that are sampled uniformly at random from the categorical possibilities for categorical features, and normally (with st.dev. of 0.1) or uniformly within the numerical intervals for numerical features, as defined in $\mathbf{p}$. As expected due to how we defined $\mathcal{P}$, it is always possible to contrast perturbations to $\mathcal{C}$ (i.e., $\mathcal{C}$-setbacks, see Sec. 4.1). Instead, perturbations concerning $\mathcal{K}$ can lead to a $\mathbf{z}'$ such that $\mathbf{z}^\star - \mathbf{z}' \notin \mathcal{P}$. We do not report a result for perturbations concerning both $\mathcal{C}$ and $\mathcal{K}$ because, by construction, it is the same as the result for perturbations concerning only $\mathcal{K}$. Like for the results of Sec. 5.1, the extent by which perturbations to $\mathcal{K}$ reduce the possibility for further intervention depends on the data set. On Pro, all perturbations can be contrasted by an additional intervention because there are no plausibility constraints (see Table 1). Conversely, on Rec, perturbations to $\mathcal{K}$ can often make it impossible to reach the intended $\mathbf{z}^\star$, unless $\mathcal{K}$-robustness is accounted for. In fact, accounting for $\mathcal{K}$-robustness generally improves the chances that further intervention is possible, at times substantially (e.g., on Inc, Hou, and Rec). Cre represents the only exception to this, as accounting for

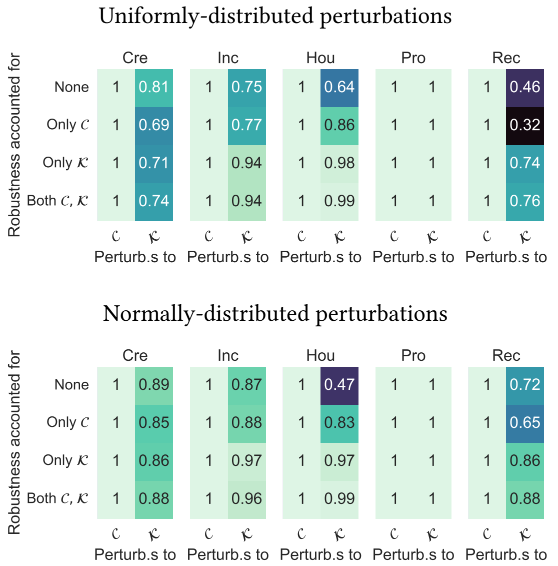

Fig. 2. Mean frequency with which a plausible additional intervention exists, to contrast the perturbations and reach the intended $\mathbf{z}^\star$ (uniformly-distributed categorical changes and normally- or uniformly-distributed numerical changes). Darker colors represent worse cases.

$\mathcal{K}$-robustness results in lower performance than ignoring any notion of robustness. This suggests that the decision boundary learned by f on this data set may not be very smooth, making the use of our $\mathcal{K}$-robustness score insufficient. Finally, accounting for C-robustness *alone* may (e.g., on Hou) or may not (e.g., on Cre) happen to improve resilience to perturbations to $\mathcal{K}$, depending on underlying correlations that are present between the features of the data set and what was learned by f. Importantly, accounting for C-robustness *together* with accounting for $\mathcal{K}$-robustness does not compromise the gains obtained by the latter. Overall, these results show that accounting for robustness can be crucial to ensure that, if perturbations happen, additional intervention to obtain $c^\star$ remains possible.

5.3 (RQ3) Are robust counterfactual explanations advantageous in terms of additional intervention cost?

We present the following results in terms of a *relative cost*, namely, the ratio between the cost of intervention to reach $\mathbf{z}^\star$ when random perturbations take place (i.e., initial the cost of reaching $\mathbf{z}^\star$ from $\mathbf{x}$ plus the cost of reaching $\mathbf{z}^\star$ from the perturbed $\mathbf{z}'$), and the cost that is expected in the ideal case, when no perturbations are applied (i.e., the cost of reaching $\mathbf{z}^\star$ from $\mathbf{x}$). We compute this relative cost when the notions of robustness are or are not accounted for. The cost for the ideal case is computed without considering robustness. The cost is modeled by $\frac{1}{2}\gamma(\mathbf{z}, \mathbf{x}) + \frac{1}{2}\frac{||\mathbf{z}-\mathbf{x}||_0}{d}$ (i.e., the first part of Eq. (6)). Note that for $\mathbf{z}'$ obtained due to perturbations to the features in C, $||\mathbf{z}' - \mathbf{z}||_0 = 0$. Moreover, if $f(\mathbf{z}') = c^\star$, we assume no additional intervention to be needed, and thus no additional cost.

Fig. 3 shows that when no robustness is accounted for (the left-most triplets of boxes in each plot), the relative cost can become dramatically large. In other words, additional intervention to correct the perturbations can be extremely costly. Whether the relative cost increases mostly due to perturbations to C (blue boxes) or to $\mathcal{K}$ (orange boxes) depends on the data set. For example, perturbations to $\mathcal{K}$ have the largest effect on Rec, while those to C have the largest effect on Inc (by far). Remarkably, when one accounts for the notion of robustness that is meant to deal with the right of perturbations, the relative cost decreases dramatically (except in some cases for $\mathcal{K}$). We remark that since the ideal cost is computed when no notion of robustness is accounted for, part of the relative cost for when a notion of robustness is accounted for comes from the fact that robust counterfactual examples are farther away from $\mathbf{x}$ than non-robust ones. We show this in more detail in Appendix C.6. On all data sets, accounting for C-robustness (second blue box from the left in each plot) counters perturbations to C very well. On Inc in particular, the relative cost improves by two orders of magnitude. On the other hand, accounting for $\mathcal{K}$-robustness is effective in contrasting perturbations to $\mathcal{K}$ (third orange box from the left in each plot) on Hou, Pro, and Rec, but not really on Cre and Inc. Again, this is likely a limitation of using a simple heuristic such as the $\mathcal{K}$-robustness score to deal with $\mathcal{K}$-robustness. Accounting for robustness in C (resp., $\mathcal{K}$) does not, in general, lead to smaller relative cost under perturbations to $\mathcal{K}$ (resp., C). Lastly, accounting for both C- and $\mathcal{K}$-robustness (right-most triplets of boxes in each plot) offers protection (lower relative cost) from situations in which both types of perturbations take place, which is perhaps the most reasonable scenario in real life. On all data sets, the distribution of relative costs for when perturbations to both C and $\mathcal{K}$ take place and both C- and $\mathcal{K}$-robustness are accounted for (right-most green box in each plot) is better than the distribution for when the same perturbations take place but no notion of robustness is accounted for (left-most green box in each plot).

These results confirm that even though robust counterfactual explanations are, in principle, more costly to pursue than non-robust ones (see Appendix C.6), if random perturbations take place, robust counterfactual explanations require less additional intervention than non-robust ones.

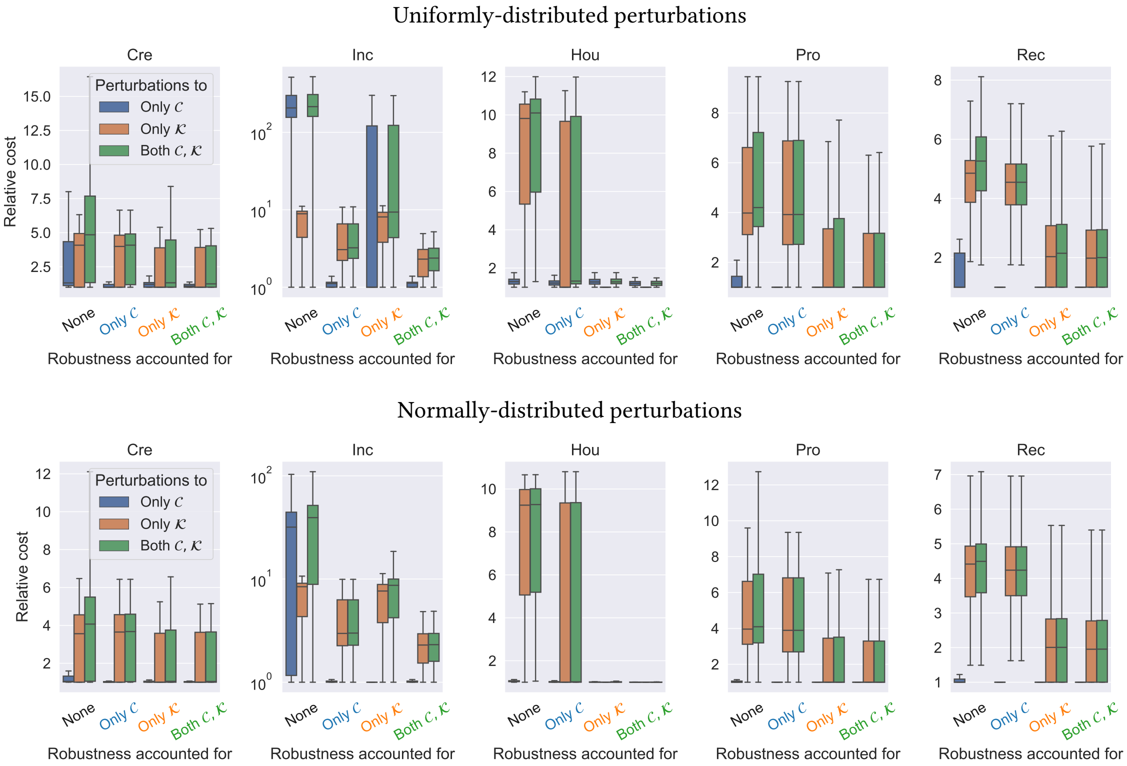

Fig. 3. Relative cost in terms of different configurations of accounting for robustness with respect to not accounting for any notion of robustness, under adverse perturbations (uniformly-distributed categorical changes and normally- or uniformly-distributed numerical changes) for features in $\mathcal{C}$, $\mathcal{K}$, and both $\mathcal{C}$ and $\mathcal{K}$. Under adverse perturbations, the relative cost for when no notion of robustness is accounted for (label None) is typically much larger than the one for when the right notion of robustness is accounted for (matching color between box and label). The vertical axis for Inc is in logarithmic scale.

6 DISCUSSION

Our experimental results provide a positive answer to all three research questions. Most often than not, counterfactual explanations are *not* robust, be it in terms of the features whose value is prescribed to be changed ($\mathcal{C}$-robustness), or those whose value is prescribed to be kept as is ($\mathcal{K}$-robustness). Moreover, non-robust counterfactual explanations are more susceptible to make it impossible for the user to contrast perturbations with additional intervention, and the cost of additional intervention is larger for non-robust counterfactual explanations than for robust ones. Ultimately, it is clear that accounting for robustness is important.

Our experimental results confirm that perturbations and notions of robustness that concern the features in $\mathcal{C}$ are largely orthogonal to those that concern the features in $\mathcal{K}$. Thanks to Proposition 2.2, the discovery of counterfactual examples that entail the minimal cost under worst-case $\mathcal{C}$-setbacks can be guaranteed. The extra intervention cost for considering the worst-case is far inferior to the extra cost entailed by non-worst-case perturbations Sec. 5.3, even considering the possible lack of optimality of the discovered $\mathbf{z}^\star$ (for more on this, see also Appendix C.4). Instead, due to Proposition 2.7, a fast and worst-case-proof modeling of robustness cannot be achieved for $\mathcal{K}$-robustness. Thus, we proposed to control for $\mathcal{K}$-robustness using an approximation, i.e., the $\mathcal{K}$-robustness score. Our results show that, for the data sets considered, seeking counterfactual examples that maximize the $\mathcal{K}$-robustness score is often but not always sufficient to obtain a good resilience to perturbations to the features in $\mathcal{K}$. Therefore, future work should consider whether a better heuristic can be used than randomly sampling points in the $\mathcal{K}$-neighborhood N of a counterfactual example. If instead information on f is available, that information can be used to provide guarantees on N (see, e.g., Theorem 2 in [11] for linear f).

A limitation of this work is that we assume the features to be independent from each other, which, although commonly made in research literature, is hardly completely met in real-world problems. Assuming feature independence does not impact the validity of counterfactual explanations ($\mathbf{z}^\star$ is necessarily of class $c^\star$ under the model f, no matter how we arrive at $\mathbf{z}^\star$), but can make finding them less efficient (as the search does not exploit correlations) and can make the cost of intervention result larger than it may be in practice (as improving in one feature, e.g., salary, may cause improving in another, e.g., savings [29]). Regarding perturbations, some perturbation configurations may not be feasible if some features are correlated (e.g., it may not be possible that perturbations impact salary but not savings). It is important to note, however, than by accounting for robustness in a worst-case scenario (as commonly done in literature and also here), results in ensuring that a violation of the assumption of feature independence does not lead to a violation of robustness (since the provided protection entails a hyper-cube for the $L1$-norm, or a hyper-sphere for the $L2$-norm), but only an additional expected cost. Thus, our definitions of robustness could be adapted to work within causal frameworks that model feature inter-dependencies, such as those introduced in [28, 29], to make accounting for robustness less costly. However, causal models typically need to be specified by hand, and are application-specific.

In literature, typically one or a few types of distribution are assumed for perturbations that should be countered by robustness (e.g., normal [49], adversarially-generated [11], and others [42]). Here, we assumed that the magnitude of perturbation that is possible for the features is sampled uniformly or normally, within some intervals. Of course, assuming this represents a simplification. Certain real-world application likely entail perturbations for different features to follow different likelihoods (e.g., some perturbations may be uniformly distributed, some normal distributed, and some others be distributed even differently).

There is a number of further aspects worth mentioning when one wishes to implement a research work like on counterfactual explanations into practice, including this work. For example, we use the $L1$-norm within Gower's distance across the board to measure intervention cost. In fact, literature works typically choose one distance measure and stick to it thorough the paper (e.g., ours, or Gower's with $L2$-norm instead, or other variants, see Sec. 3). Of course, realistic implementation of intervention cost needs to use the a refined distance which may include mixing different types of norms, based on the features at play. Next, other notions of robustness that concern a different type of perturbations than the one we considered here may also need to be incorporated, such as uncertainties arising from updates to f [48, 49]. Moreover, other important desiderata such as those listed in Sec. 3 should be taken into account at the same time, possibly using a multi-objective search algorithm [9] that allows the user to make a post-hoc decision on what intervention best suits him or her. It is clear that all of these aspects need to come from expert knowledge and are application-specific.

Lastly, we had to make subjective choices to define perturbations ($\mathbf{p}$) and plausibility constraints ($\mathcal{P}$) in the data sets. We made these choices as best as we could, based on reading the meta-information in web sources and the papers that describe the data sets. We have no doubt that domain experts would make much better choices than ours. Nevertheless, we argue that this is not an important limitation because, as long as the community agrees that our choices are sufficiently sensible, they suffice to provide experimental evidence that robustness is an important desideratum to account for and, importantly, allow other researchers to build upon the code provided with this paper, to study the phenomenon further.

7 CONCLUSION

We have presented novel notions of robustness for counterfactual explanations, to improve their practical adoption. The notions are two and concern adverse perturbations to the features that a counterfactual explanation prescribes to change ($\mathcal{C}$-robustness) and to keep as they are ($\mathcal{K}$-robustness), respectively. We have annotated five existing data sets with reasonable perturbations and plausibility constraints and developed a competitive counterfactual search algorithm to search for (robust) counterfactual explanations. Our experimental results show that, most often than not, counterfactual explanations do not happen to be robust by accident. Consequently, if adverse perturbations take place, counterfactual explanations may require a much larger cost to be realized than anticipated, or even make it impossible for the user to achieve recourse. Fortunately, our definitions of robustness can be incorporated in the search process, and robust counterfactual explanations can be discovered. We have shown that under any general black-box machine learning model, $\mathcal{C}$-robustness can be accounted for efficiently and effectively, while the same is not always true for $\mathcal{K}$-robustness. Overall, robust counterfactual explanations are resilient against invalidation and require much smaller additional intervention to contrast perturbations.

ACKNOWLEDGMENTS

We thank dr. Stef C. Maree for early insightful discussions. This publication is part of the project Robust Counterfactual Explanations (with project number EINF-2512) of the research program Computing Time on National Computer Facilities which is (partly) financed by the Dutch Research Council (NWO). This work made use of the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-2512.

REFERENCES

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on eXplainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160.

[2] Vincent Ballet, Xavier Renard, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, and Marcin Detyniecki. 2019. Imperceptible adversarial attacks on tabular data. *arXiv preprint arXiv:1911.03274* (2019).

[3] Solon Barocas, Andrew D Selbst, and Manish Raghavan. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 80–89.

[4] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences* 116, 32 (2019), 15849–15854.

[5] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. 2021. Benchmarking and survey of explanation methods for black box models. *arXiv preprint arXiv:2102.13076* (2021).

[6] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32.

[7] Michael Carlisle. 2019. Racist data destruction? https://medium.com/@docintangible/racist-data-destruction-113e3eff54a8.

[8] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. 2019. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32.

[9] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. 2020. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature*. Springer, 448–469.

[10] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring Adult: New Datasets for Fair Machine Learning. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

[11] Ricardo Dominguez-Olmedo, Amir-Hossein Karimi, and Bernhard Schölkopf. 2021. On the Adversarial Robustness of Causal Algorithmic Recourse. *arXiv preprint arXiv:2112.11313* (2021).

[12] Marcello D'Orazio. 2021. Distances with mixed type variables some modified Gower's coefficients. *arXiv preprint arXiv:2101.02481* (2021).

[13] Andrea Ferrario and Michele Loi. 2020. A series of unfortunate counterfactual events: The role of time in counterfactual explanations. *arXiv preprint arXiv:2010.04687* (2020).

[14] Timo Freiesleben. 2021. The intriguing relation between counterfactual explanations and adversarial examples. *Minds and Machines* (2021), 1–33.

[15] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29 (2001), 1189–1232.

[16] Fuchang Gao and Lixing Han. 2012. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications* 51, 1 (2012), 259–277.

[17] Bryce Goodman and Seth Flaxman. 2017. European Union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine* 38, 3 (2017), 50–57.

[18] John C Gower. 1971. A general coefficient of similarity and some of its properties. *Biometrics* (1971), 857–871.

[19] Ulrike Grömping. 2019. South German Credit Data: Correcting a Widely Used Data Set. https://archive.ics.uci.edu/ml/datasets/South+German+Credit+%28UPDATE%29. Report 4/2019, Reports in Mathematics, Physics and Chemistry, Department II, Beuth University of Applied Sciences Berlin.

[20] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820* (2018).

[21] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *Comput. Surveys* 51, 5 (2018), 1–42.

[22] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362.

[23] David Harrison Jr and Daniel L Rubinfeld. 1978. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management* 5, 1 (1978), 81–102.

[24] Hans Hofmann. 1994. Statlog German Credit Data. https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data).

[25] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 5 (1989), 359–366.

[26] Abdullah Al Imran, Md Shamsur Rahim, and Tanvir Ahmed. 2021. Mining the productivity data of the garment industry. *International Journal of Business Intelligence and Data Mining* 19, 3 (2021), 319–342.

[27] Anna Jobin, Marcello Ienca, and Effy Vayena. 2019. The global landscape of AI ethics guidelines. *Nature Machine Intelligence* 1, 9 (2019), 389–399.

[28] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. 2021. Algorithmic Recourse: From Counterfactual Explanations to Interventions. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) *(FAccT '21)*. Association for Computing Machinery, New York, NY, USA, 353–362. https://doi.org/10.1145/3442188.3445899

[29] Amir-Hossein Karimi, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic recourse under imperfect causal knowledge: A probabilistic approach. *arXiv preprint arXiv:2006.06831* (2020).

[30] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems* 30 (2017), 3146–3154.

[31] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*. PMLR, 2564–2572.

[32] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. 2016. Examples are not enough, learn to criticize! Criticism for interpretability. *Advances in Neural Information Processing Systems* 29 (2016).

[33] Ron Kohavi. 1996. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Vol. 96. 202–207.

[34] Ronny Kohavi and Barry Becker. 1996. Census income. https://archive.ics.uci.edu/ml/datasets/adult.

[35] William La Cava and Jason H. Moore. 2020. Genetic Programming Approaches to Learning Fair Classifiers. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (Cancún, Mexico) *(GECCO '20)*. Association for Computing Machinery, New York, NY, USA, 967–975.

[36] Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. 2016. How we analyzed the COMPAS recidivism algorithm. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

[37] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2018. Comparison-based inverse classification for interpretability in machine learning. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, 100–111.

[38] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2019. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. *arXiv preprint arXiv:1907.09294* (2019).

[39] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2019. Unjustified classification regions and counterfactual explanations in machine learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 37–54.

[40] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[41] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*. 4768–4777.

[42] Rami Mochaourab, Sugandh Sinha, Stanley Greenstein, and Panagiotis Papapetrou. 2021. Robust Counterfactual Explanations for Privacy-Preserving SVM. In *International Conference on Machine Learning (ICML 2021), Workshop on Socially Responsible Machine Learning*.

[43] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 607–617.

[44] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 607–617.

[45] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2021. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment* 2021, 12 (2021), 124003.

[46] John A Nelder and Roger Mead. 1965. A simplex method for function minimization. *Comput. J.* 7, 4 (1965), 308–313.

[47] Martin Pawelczyk, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. 2021. Exploring Counterfactual Explanations Through the Lens of Adversarial Examples: A Theoretical and Empirical Analysis. *arXiv preprint arXiv:2106.09992* (2021).

[48] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. 2020. On counterfactual explanations under predictive multiplicity. In *Conference on Uncertainty in Artificial Intelligence*. PMLR, 809–818.

[49] Kaivalya Rawal, Ece Kamar, and Himabindu Lakkaraju. 2021. Algorithmic Recourse in the Wild: Understanding the Impact of Data and Model Shifts. *arXiv preprint arXiv:2012.11788* (2021).

[50] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1135–1144.

[51] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215.

[52] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2020. 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*. 166–172.

[53] Dylan Slack, Sophie Hilgard, Himabindu Lakkaraju, and Sameer Singh. 2021. Counterfactual Explanations Can Be Manipulated. *arXiv preprint arXiv:2106.02666* (2021).

[54] Ilia Stepin, Jose M Alonso, Alejandro Catala, and Martín Pereira-Fariña. 2021. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access* 9 (2021), 11974–12001.

[55] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 10–19.

[56] Arnaud Van Looveren and Janis Klaise. 2019. Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584* (2019).

[57] Sahil Verma, John Dickerson, and Keegan Hines. 2020. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596* (2020).

[58] Marco Virgolin, Andrea De Lorenzo, Francesca Randone, Eric Medvet, and Mattias Wahde. 2021. Model Learning with Personalized Interpretability Estimation (ML-PIE). In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 1355–1364.

[59] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272.

[60] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology* 31 (2017), 841.

APPENDIX OF
ON THE ROBUSTNESS OF COUNTERFACTUAL EXPLANATIONS TO ADVERSE PERTURBATIONS

A HYPER-PARAMETER OPTIMIZATION OF RANDOM FOREST

To obtain a black-box model f for a given cross-validation fold, we train a random forest model optimized with grid-search hyper-parameter tuning (with five-fold cross-validation on the training set). The hyper-parameter settings we considered are listed in Table 3. Furthermore, we one-hot encode categorical features when training and querying the random forest model (see the code `robust_cfe/blackbox_with_preproc.py`).

The performance of tuned random forest on all folds in shown in Table 4.

Table 3. Hyper-parameter settings considered for tuning random forest.

Name	Range
No. trees	$\{50, 500\}$
Min. samples split	$\{2, 8\}$
Max. features	$\{\sqrt{d}, d\}$

Table 4. Test accuracy of hyper-parameter-tuned random forests acting as black-boxm models f for the considered data sets across five-fold cross-validation.

Fold	Cre	Inc	Hou	Pro	Rec
0	0.71	0.86	0.93	0.79	0.80
1	0.78	0.82	0.90	0.77	0.82
2	0.78	0.79	0.91	0.78	0.78
3	0.74	0.82	0.91	0.82	0.77
4	0.76	0.83	0.97	0.78	0.80
Avg.	0.76	0.83	0.93	0.79	0.80

B DETAILS ON COUNTERFACTUAL SEARCH ALGORITHMS

We performed preliminary experiments with CoGS, Growing Spheres (GrSp) [64], LOcal Rule-based Explanations (LORE) [62], and the implementation of the Nelder-Mead method (NeMe) [16, 66] by SciPy [67].

CoGS, GrSp and NeMe return a point, i.e., a *best-found* counterfactual example $\mathbf{z}^\star$ (no algorithm, including LORE described below, guarantees optimality). GrSp and NeMe can only handle numerical features. Therefore, we let these algorithms handle categorical features as if they were numerical ones, and transform numerical values back to categories at the end of the optimization. This is done by encoding categories with integers (e.g., 0, 1, 2, . . .), and rounding numerical values to the nearest integer.

LORE handles both numerical and categorical features, but it is different from CoGS, GrSp and NeMe in that it does not return a counterfactual example $\mathbf{z}^\star$. LORE returns a set of counterfactual explanations encoded as rules, such as "`AGE>3.4 & SALARY_CATEGORY=HIGH`". For LORE, we build the point $\mathbf{z}^\star$ by taking the shortest rule from the returned set of rules, and *applying* the rule to the starting point $\mathbf{x}$. Note that CoGS is similar to LORE in that also LORE adopts a genetic algorithm. However, LORE uses a genetic algorithm as an intermediate step, which is followed by fitting a decision tree on the points explored by the genetic algorithm. The rules LORE returns are then extracted from the decision tree.

We found (confirmed by a discussion with the authors) that applying LORE's rules may results in points that are not actually classified as $c^\star$. When that happens, we perform up to 15 attempts at generating $\mathbf{z}^\star$ from the (shortest returned) rule, by focusing on numerical features that are prescribed to be $>$, $\geq$ (or $<$, $\leq$) than a certain value. In particular, in applying such part of the rule to $\mathbf{x}$, we add (or subtract) to the prescribed value a term ϵ, which is initially set to 10^{-3} and is doubled at every attempt.

B.1 More details on CoGS

We built CoGS to be a relatively standard genetic algorithm, adapted for the search of points neighboring $\mathbf{x}$ (especially in terms of $L0$-norm. First, an initial *population* of candidate solutions is generated by sampling feature values uniformly within an interval for numerical features, and from the possible categories for categorical features. With probability of $2/d$ (d being the total number of features), the feature value of a candidate solution is *copied* from $\mathbf{x}$ rather than sampled. Every iteration of the algorithm (in the jargon of evolutionary computation, *generation*), offspring solutions are produced from the current population by *crossover* and *mutation*.

Our version of crossover produces two offspring solutions by simply swapping the feature values of two random parents uniformly at random. Our version of mutation produces one offspring solution from one parent solution by randomly altering its feature values. A feature value is altered with probability of $1/d$ (else, it is left untouched). If the feature to alter is categorical, then the category is swapped with another category, uniformly at random. If the feature to alter is numerical, firstly a random number r is sampled uniformly at random between $-s_{mut}/2$ and $+s_{mut}/2$, where $s_{mut} \in (0, 1]$ is a hyper-parameter that represents the maximal extent of allowed mutations; secondly, the original feature value is changed by adding $r \times (\max_i - \min_i)$, where $\max_i$ and $\min_i$ are, respectively, the maximum and minimum values that are possible for that feature. If plausibility constraints ($\mathcal{P}$) are used, then mutation is restricted to plausible changes (e.g., the feature that represents age can only increase). If mutation makes a numerical feature obtain a value bigger than $\max_i$ (resp., smaller than $\min_i$), then the value of that feature is set to $\max_i$ (resp., $\min_i$).

The quality (*fitness*) of offspring solutions is then evaluated using the loss function (Eq. (6)), and finally survival of the fittest (in our case, tournament selection [65]) is applied to form a new population, to be used as starting point for the next generation.

Our code is written in Python and relies heavily on NumPy [63] (e.g., the population is encoded as a matrix and crossover and mutation operate upon it with matrix operations) for the sake the speed.

B.2 Settings of the counterfactual search algorithms

We use mostly default settings for all the algorithms (with respect to their code bases, last accessed Jan 21 2022). We set all algorithms to minimize the same distance (Gower's plus the $L0$-norm, as per Eq. (6)). Table 5 reports the main settings, which are mostly set to their default values (not reported ones are always set to their default value). Note that we set the genetic algorithm of LORE to have a much smaller search budget than CoGS, making the comparison not really fair (although LORE performs further operations which include building a decision tree, while CoGS does not). However, as can be seen Fig. 4, LORE is orders of magnitude slower to execute than CoGS, and we found increasing its search budget to simply be prohibitive. We chose to run CoGS for 100 generations because this led to commensurate runtimes to those of GrSp and NeMe (actually, CoGS is faster, see Appendix C.1.1).

Table 5. Settings of the considered counterfactual search algorithms. For NeMe, we only set the maximum number of iterations to 100. For an explanation of the meaning of the settings for methods other than CoGS, we refer to the respective papers and code bases.

CoGS	
Setting	Value
Population size	1000
Num. generations	100
Tournament size	2

GrSp	
Setting	Value
Num. in layer	2000
First radius	0.1
Decrease radius	10
Sparse	True

LORE	
Setting	Value
Population size	1000
Num. generations	10
Discrete use probabilities	False
Continuous function estim.	False

C ADDITIONAL RESULTS

C.1 Comparison of the search algorithms

Like for the results reported in the main body of the paper, we repeat the execution of each algorithm five times and consider the best-found $\mathbf{z}^\star$ out of the five repetitions. We search for a counterfactual example for each $\mathbf{x}$ in the test sets from the five cross-validation, for $\mathbf{x}$ such that $f(\mathbf{x}) \neq c^\star$. Since LORE takes much longer to execute than the other algorithms (see Fig. 4), we perform three repetitions instead of five, and consider only the first five $\mathbf{x}$ in each test set of the five folds. Since only CoGS supports plausibility constraints, we do not use them in this comparison.

C.1.1 Runtimes. Fig. 4 shows the runtime of the algorithms across the different data sets, irrespective of whether they succeed or fail to find a counterfactual example, i.e., a point for which f predicts $c^\star$. The experiments were run on a cluster where the computing nodes can have slightly different CPUs, thus we invite to consider the order of magnitude of the runtimes rather than the exact numbers. What algorithm-data set pair runs on what CPU is assigned randomly by the cluster. The figure shows that CoGS is the fastest algorithm (or better, implementation), but GrSp and NeMe are competitive. LORE is much slower to execute than the other algorithms. The bottleneck in LORE is the implementation of its genetic algorithm (using the library *deap* [61]).

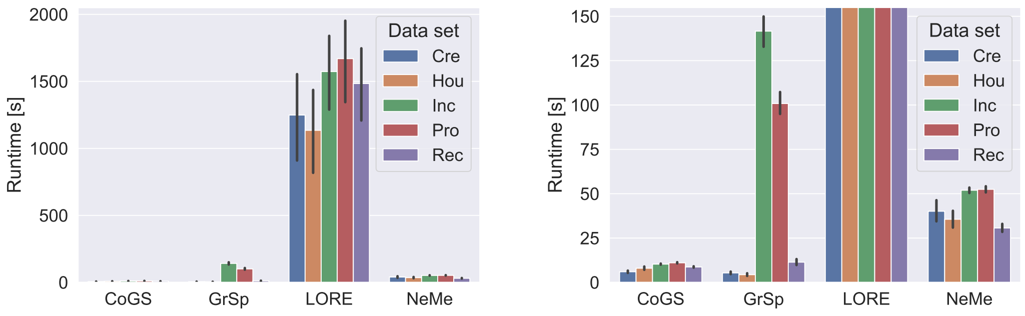

Fig. 4. Runtimes (means and 95% confidence intervals) of the counterfactual search algorithms for the considered data sets. The right plot is a zoomed-in version of the left one.

C.1.2 Success in discovering counterfactual examples. Table 6 shows the frequency with which the counterfactual search algorithms succeed in finding a counterfactual example, i.e., a point for which f predicts $c^\star$. CoGS succeeds systematically, whereas the other algorithms do not. GrSp performs second-best overall. In particular, GrSp always found a counterfactual example on Hou, which is a data set with a single categorical feature. Since GrSp is intended to operate solely with numerical features, this results nicely supports the hypothesis that GrSp works well when (almost all) features are numerical. Although LORE supports both numerical and categorical features, it does not perform better than GrSp on most data sets; at least for the limited number of runs conducted with LORE due to excessive runtime, as explained before. Lastly, NeMe often performs substantially worse than all other algorithms.

Table 6. Mean ± standard deviation across five cross-validation folds of the frequency with which the counterfactual search algorithms succeed in finding a counterfactual example. Plausibility constraints are not considered here because not all algorithms support them.

Alg.	Cre	Inc	Hou	Pro	Rec
CoGS	1.00	1.00	1.00	1.00	1.00
GrSp	0.34 ± 0.12	0.58 ± 0.10	1.00	0.69 ± 0.09	0.24 ± 0.14
LORE	0.23 ± 0.18	0.11 ± 0.10	0.56 ± 0.24	0.09 ± 0.14	0.25 ± 0.24
NeMe	0.08 ± 0.03	0.05 ± 0.02	0.04 ± 0.05	0.03 ± 0.01	0.14 ± 0.02

C.1.3 Quality of discovered counterfactual examples. Lastly, we consider the relative change in loss for the best-found counterfactual example with respect to the loss obtained by CoGS, only for success cases. Recall that the loss (Eq. (6) in the main body of the paper) is:

$$\mathcal{L}(\mathbf{z}) = \frac{1}{2}\gamma(\mathbf{z},\mathbf{x}) + \frac{1}{2}\frac{||\mathbf{z}-\mathbf{x}||_0}{d} + ||f(\mathbf{z}) - c^\star||_0.$$

Here, the last term $||f(\mathbf{z}) - c^\star||_0$ is always null here because only success cases are considered. The relative change in loss with respect to CoGS for another algorithm *Alg* is:

$$\frac{\mathcal{L}_{Alg}(\mathbf{z}) - \mathcal{L}_{\text{CoGS}}(\mathbf{z})}{\mathcal{L}_{\text{CoGS}}(\mathbf{z})}.$$

Fig. 5 shows the relative change in loss of GrSp, LORE, and NeMe with respect to CoGS. GrSp and LORE typically (but not always) find points that have larger loss than those found by CoGS. NeMe performs very similarly to CoGS, however NeMe seldom succeeds (cfr. Table 6). This suggests that NeMe can explore a small neighborhood of $\mathbf{x}$ particularly well, but fails if counterfactual examples are relatively distant from $\mathbf{x}$.

C.2 Setting m for $\mathcal{K}$-robustness

We report results on setting the hyper-parameter m for computing $\mathcal{K}$-robustness scores (see Eq. (5)). To do this, we run CoGS accounting for $\mathcal{K}$-robustness in the loss function, for $m \in [0, 4, 16, 64]$, and consider the discovered $\mathbf{z}^\star$. Note that $m = 0$ corresponds to *not* accounting for $\mathcal{K}$-robustness. For that $\mathbf{z}^\star$, we compute a "ground-truth"-like $\mathcal{K}$-robustness score, by using 1000 samples.

Fig. 6 shows the results obtained for this experiment. We also consider the case in which C-robustness is also accounted for. If $\mathcal{K}$-robustness is not accounted for ($m = 0$), then the ground-truth $\mathcal{K}$-robustness scores of the discovered counterfactual examples oscillates between 0.5 and 0.8 on average (see, resp., Rec and Cre). As soon as a few samples are considered ($m = 4$), ground-truth $\mathcal{K}$-robustness scores increase substantially (see, e.g., Cre). Further increasing m has diminishing returns (note that m is increased exponentially). Accounting for C-robustness is largely orthogonal, meaning, it has no effect in terms of $\mathcal{K}$-robustness scores. A limited decrease in $\mathcal{K}$-robustness can however be observed on Inc when accounting for C-robustness.

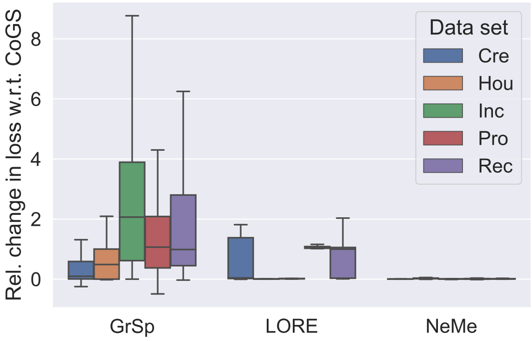

Fig. 5. Boxplots of relative change in loss with respect to CoGS for GrSp, LORE, and NeMe, on the different data sets for success cases.

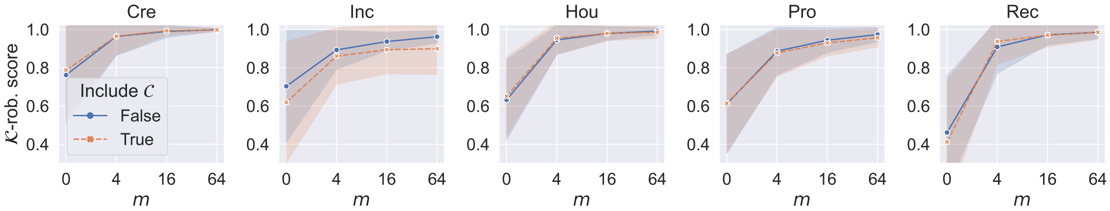

Fig. 6. Mean $\mathcal{K}$-robustness scores for increasing m, with and without including optimization for $\mathcal{C}$-robustness. Shaded areas represent standard deviations.

C.3 Accidental discovery of robust counterfactual examples for different tolerance levels

Table 7 provides further results for Sec. 5.1. In particular, the table shows how the frequency of accidental discovery of counterfactual examples that are robust changes based on the tolerance level used to decide whether two numerical features match. As reasonable to expect, the larger the tolerance level, the more a $\mathbf{z}^\star$ discovered when not accounting for robustness matches the respective one that is discovered when accounting for robustness. However, on some data sets, the accidental discovery of robust counterfactual examples remains rare even with a large tolerance level.

C.4 Invalidity of counterfactual explanations

We now show whether the fact that best-found counterfactual explanations are typically not robust is associated with a greater chance that perturbations can make them invalid, i.e., such that $f(\mathbf{z}') \neq c^\star$ where $\mathbf{z}'$ is the point to which $\mathbf{z}^\star$ is shifted by the perturbation. Here, we do not consider whether further intervention may or may not be possible. Fig. 7 shows the average frequency with which perturbations cause invalidity. The frequencies are computed by applying, to each $\mathbf{z}^\star$, 100 perturbations that are sampled uniformly at random for categorical features (from the categorical possibilities) and uniformly or normally (with st.dev. of 0.1) for numerical features (within the numerical intervals). The figure shows that when no notion of robustness is accounted for, perturbations generally have a larger chance of causing invalidity of the counterfactual explanation.

Table 7. Mean ± standard deviation of the frequency with which the best-found (among five search repetitions) counterfactual example when not accounting for robustness is accidentally $\mathcal{C}$- or $\mathcal{K}$-robust. For numerical features, we consider them to match in value if they are within a tolerance level (Tol.) of 1%, 5% or 10% of the range for that feature.

Robustness	Tol.	Cre	Inc	Hou	Pro	Rec
Only $\mathcal{C}$	1%	0.40 ± 0.06	0.02 ± 0.00	0.76 ± 0.10	0.53 ± 0.05	0.27 ± 0.06
	5%	0.42 ± 0.07	0.04 ± 0.02	0.84 ± 0.09	0.57 ± 0.06	0.37 ± 0.07
	10%	0.43 ± 0.07	0.05 ± 0.02	0.85 ± 0.09	0.58 ± 0.06	0.40 ± 0.09
Only $\mathcal{K}$	1%	0.37 ± 0.01	0.06 ± 0.02	0.33 ± 0.24	0.26 ± 0.05	0.04 ± 0.04
	5%	0.44 ± 0.03	0.40 ± 0.08	0.63 ± 0.17	0.37 ± 0.06	0.08 ± 0.03
	10%	0.46 ± 0.04	0.58 ± 0.07	0.67 ± 0.16	0.46 ± 0.07	0.12 ± 0.02
Both $\mathcal{C}, \mathcal{K}$	1%	0.23 ± 0.04	0.00 ± 0.00	0.21 ± 0.21	0.19 ± 0.06	0.03 ± 0.03
	5%	0.27 ± 0.03	0.00 ± 0.00	0.54 ± 0.21	0.26 ± 0.05	0.06 ± 0.04
	10%	0.30 ± 0.05	0.00 ± 0.00	0.60 ± 0.19	0.34 ± 0.06	0.08 ± 0.04

Regarding $\mathcal{C}$-robustness and respective perturbations to (features in) $\mathcal{C}$ (i.e., $\mathcal{C}$-setbacks), recall that accounting for this notion of robustness results in finding a $\mathbf{z}^\star$ that has the lowest intervention cost when the worst-case $\mathcal{C}$-setback were to happen, and further intervention is subsequently needed. However, $\mathbf{z}^\star$ is still a point on the boundary of f (exactly so if $\mathbf{z}^\star$ is truly optimal) and thus, under optimality guarantees, $f(\mathbf{z}^\star + \tilde{\mathbf{w}}) \neq c^\star$ (Proposition 2.2); This means that, unless $\tilde{w}_i = 0, \forall i$ (i.e., there is no $\mathcal{C}$-setback), all entries regarding perturbations to $\mathcal{C}$ should result in invalidity (i.e., all entries for perturbations to $\mathcal{C}$ should report 1). This does not always happen in the figure because some discovered $\mathbf{z}^\star$ are not optimal and the randomly generated $\mathcal{C}$-setback $\tilde{\mathbf{w}}$ can be relatively small, or because there exists no $\mathcal{C}$-setback for the discovered counterfactual explanation. For example on Inc we do not define a perturbation that allows the user to become younger, but increasing age (to gain seniority and thus higher salary) is an explanation that CoGS discovers. Moreover, on Hou, the lower frequency of invalidity for perturbations to $\mathcal{C}$ when using normally-distributed magnitudes (smaller on average) over uniformly-distributed ones (larger on average) indicates that CoGS often fails to truly find a point $\mathbf{z}^\star$ on the boundary, and $f(\mathbf{z}^\star + \tilde{\mathbf{w}}) = c^\star$.

When $\mathcal{K}$-robustness is accounted for, the best-found counterfactual explanation is supposed to be in a region such that the decision boundary is relatively loose with respect to the features in $\mathcal{K}$. Consequently, accounting for $\mathcal{K}$-robustness counters quite well respective perturbations that afflict features in $\mathcal{K}$. At times, gains are almost optimal (see Inc, Hou, and Pro) since the frequency of invalidity becomes almost null. Cre represents the only exception to this, as discussed in Sec. 5.2.

Lastly, since accounting for $\mathcal{C}$-robustness can be in conflict with accounting with $\mathcal{K}$-robustness, the frequencies of invalidity can raise when both notions are considered (e.g., on Inc).

C.5 Additional required runtime to account for robustness

Fig. 8 shows the additional runtime incurred between runs of CoGS that account for some notion of robustness and runs that do not account for it. The figure shows that accounting for $\mathcal{C}$-robustness comes at no significant extra cost in runtime. This follows from the fact that we can use Proposition 2.2 and thus only need to compute the worst-case $\mathcal{C}$-setback. Conversely, accounting for $\mathcal{K}$-robustness can come at a relatively large additional cost in runtime, which appears to be linear in m (note that m grows exponentially in the plots). Fortunately, the experimental results of Appendix C.2 suggest that small values of m are often sufficient to obtain good $\mathcal{K}$-robustness scores.

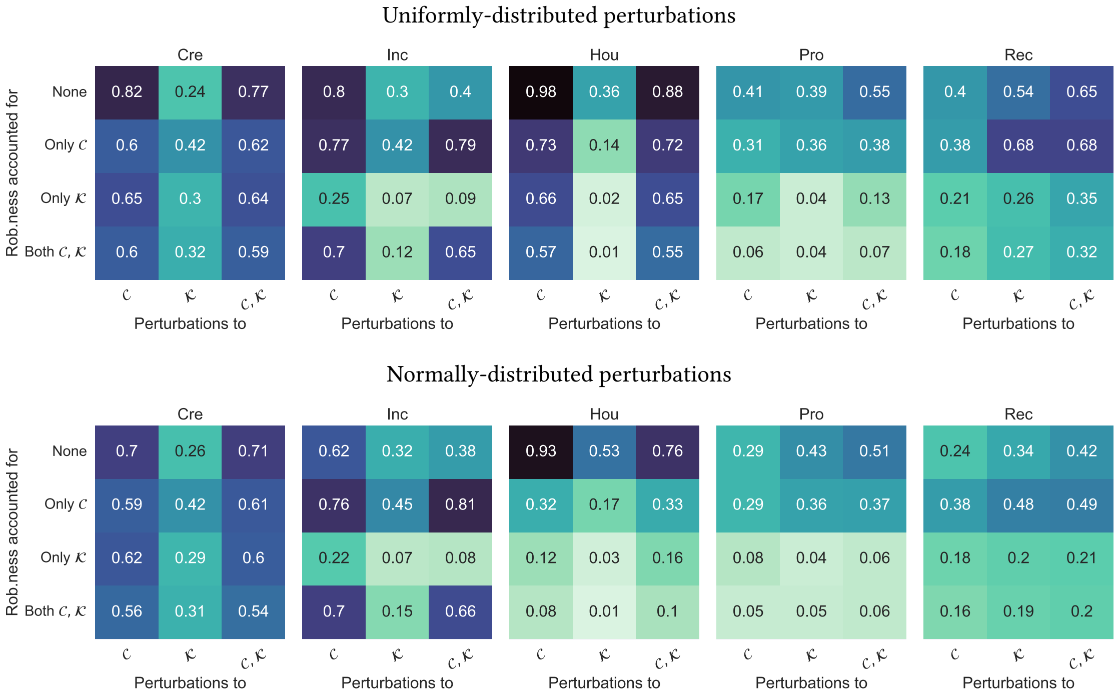

Fig. 7. Mean frequency of invalidity of counterfactual explanations under different types of perturbations and when accounting for different types of robustness. Darker colors represent worse scenarios, i.e., larger average invalidity.

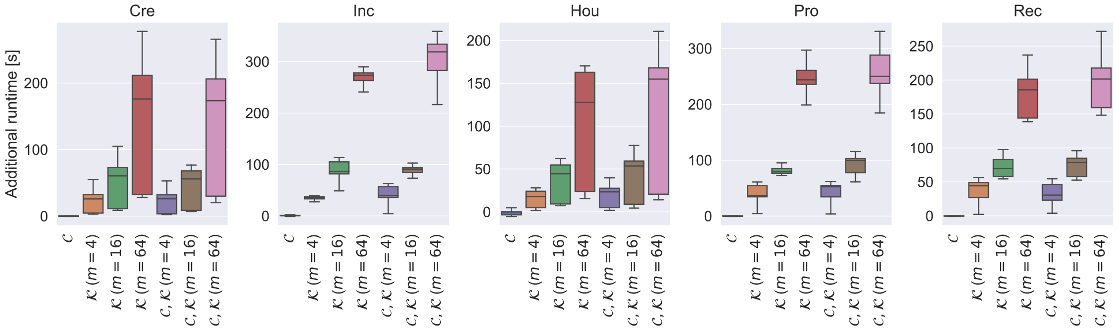

Fig. 8. Additional runtime of CoGS for different configurations of accounting for robustness with respect to the runtime when not accounting for robustness.

C.6 Additional cost from accounting for robustness

We consider how much additional cost comes from solely accounting for robustness during the search, i.e., when no perturbations take place. In fact, optimal robust counterfactual examples $\mathbf{z}^{\star}$ need necessarily be equally- or farther-away

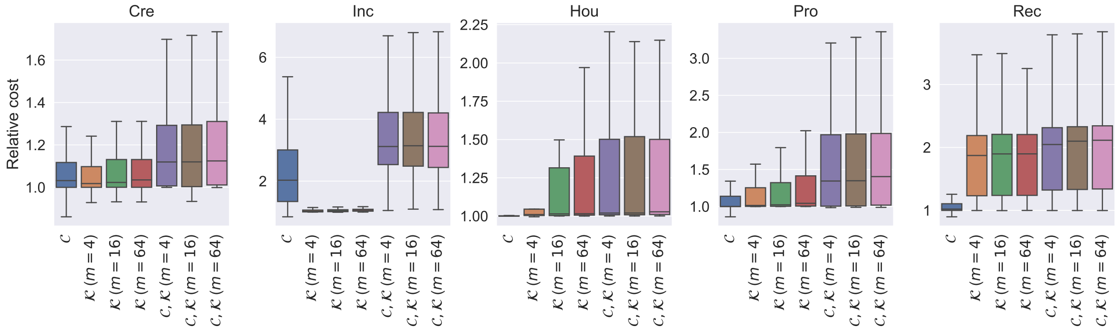

Fig. 9. Relative cost associated with best-found counterfactual examples in terms of different configurations of accounting for robustness with respect to not accounting for any notion of robustness. Values below one are due to the lack of optimality guarantees of CoGS.

from $\mathbf{x}$ than non-robust ones by construction (see Sec. 4.3). These results complement those of Sec. 5.3, where instead perturbations were applied.

Fig. 9 shows the relative cost for the different ways of accounting for robustness, over accounting for none. We remark that values smaller than 1 are *not* theoretically possible under optimality guarantees of $\mathbf{z}^\star$ (which CoGS, like GrSp, LORE and NeMe, does not have). Below-one relative costs are anyway rare and small in magnitude. On average, accounting for robustness requires additional cost whose magnitude depends on the data set (cfr., e.g., Cre and Inc). Across the different data sets, there is no clear trend on whether accounting for C- or $\mathcal{K}$- leads to larger relative cost. For example on Inc, accounting for C-robustness results, on average, in twice the cost than not accounting for any form of robustness, while the additional cost that comes with accounting for $\mathcal{K}$-robustness is negligible. The vice versa holds for Rec. We do not find major differences based on the setting of m for computing the $\mathcal{K}$-robustness score, except for the tails of the respective distributions on Hou, and slightly less so on Pro. Accounting for C- and $\mathcal{K}$-robustness at the same time leads to larger costs than accounting for only one of the two, as reasonable to expect. On average, the cost that comes from accounting for robustness alone is limited (up to 6.5 times of the non-robust cost, see Inc), especially in light of the results found for when perturbations take place, described in Sec. 5.3 (additional intervention due to perturbations can lead to 100 times larger costs for non-robust counterfactual explanations, see Inc on Fig. 3).

APPENDIX REFERENCES

[61] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. 2012. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research* 13 (jul 2012), 2171–2175.

[62] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820* (2018).

[63] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362.

[64] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2018. Comparison-based inverse classification for interpretability in machine learning. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, 100–111.

[65] Brad L Miller and David E Goldberg. 1995. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems* 9, 3 (1995), 193–212.

[66] John A Nelder and Roger Mead. 1965. A simplex method for function minimization. *Comput. J.* 7, 4 (1965), 308–313.

[67] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272.