Trustworthy Actionable Perturbations

Jesse Friedbaum [1 2] **Sudarshan Adiga** [2] **Ravi Tandon** [1 2]

Abstract

Counterfactuals, or modified inputs that lead to a different outcome, are an important tool for understanding the logic used by machine learning classifiers and how to change an undesirable classification. Even if a counterfactual changes a classifier's decision, however, it may not affect the true underlying class probabilities, i.e. the counterfactual may act like an adversarial attack and "fool" the classifier. We propose a new framework for creating modified inputs that change the true underlying probabilities in a beneficial way which we call *Trustworthy Actionable Perturbations* (TAP). This includes a novel verification procedure to ensure that TAP change the true class probabilities instead of acting adversarially. Our framework also includes new cost, reward, and goal definitions that are better suited to effectuating change in the real world. We present PAC-learnability results for our verification procedure and theoretically analyze our new method for measuring reward. We also develop a methodology for creating TAP and compare our results to those achieved by previous counterfactual methods.

1. Introduction

As machine learning (ML) classifiers have experienced widespread adoption in applications that have an out-sized impact on individuals' lives (such as credit lending (Leo et al., 2019), college admissions (Martinez Neda et al., 2021) and healthcare (Sauer et al., 2022)), the need to understand classifiers' decision making and how to avoid undesirable classifications has become increasingly important. One of the most important tools for filling this need is the *counterfactual*: a counterfactual for a given input and classifier is a similar input that results in a different classification. Suppose a classifier is designed to determine whether a loan application represents a good or bad credit risk. If the classifier determines a loan to be a bad credit risk, a counterfactual would be a modified loan application that is classified as a good credit risk, e.g. the individual in a loan application is a bad credit risk, but an otherwise identical applicant who is 5 years younger with a \$500 higher monthly income would be a good credit risk. Wachter et al. (2017) first suggested the use of *Counterfactuals Explanations* (CE) to help understand classifier decision making. Subsequent works explored the use of counterfactuals to help individuals change undesirable classifications (Ustun et al., 2019; Karimi et al., 2021; Poyiadzi et al., 2020). Returning to the example of an individual turned down for a loan, this type of counterfactual would not suggest an individual decrease their age (clearly impossible), but rather make practical changes such as pay off all credit card debt and request a 10% smaller loan. These counterfactuals came to be known as *Actionable Counterfactuals* (AC) or *Algorithmic Recourses* (AR). Although these counterfactuals change a classifier's decision, it can not be assumed they will have the same affect on the real world (Freiesleben, 2022), e.g. a change that causes a classifier to determine someone is a good credit risk may not increase the person's odds of paying off the loan in reality. König et al. (2023) point out that a counterfactual could change a classifiers decision without changing the real world if the modifications are not causally linked to the output. For example, having a mailing address in an affluent neighborhood may correlate to higher odds of paying off a loan and changing the address could affect a classifiers decision, but there is no causal link. Accordingly, telling an applicant to change their mailing address to a P.O. box in a wealthy neighborhood would not improve their chances of paying off a loan. König et al. (2023) proposed a framework to ensure modifications are causally linked to the output called *Improvement-Focused Causal Recourse* (ICR).

In this paper, we focus on tackling new challenges for this problem, which have not been addressed in prior work. Trustworthy Actionable Perturbations (TAP) focus on three novel improvements for affecting real world outcomes.

Trustworthiness Against Adversarial Examples: Szegedy et al. (2013) showed that ML classifiers are brittle and small modifications to an input can cause misclassifications in oth-

[1]Program in Applied Mathematics, University of Arizona, Tucson, AZ, USA [2]Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA. Correspondence to: Jesse Friedbaum <friedbaum@math.arizona.edu>, Ravi Tandon <tandonr@arizona.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Figure 1.a) Overview of the framework for creating Trustworthy Actionable Perturbations (TAP). b) Comparison of objectives and features of TAP and Counterfactual Explanations (CE) (Wachter et al., 2017), Actionable Counterfactuals/Algorithmic Recourse (AC/AR) (Ustun et al., 2019; Karimi et al., 2021; Poyiadzi et al., 2020), Improvement-Focused Causal Recourse (ICR) (Kl. et al., 2023).

erwise accurate classi ers. Among the various de nitions of adversarial behavior, we use the de nition: modi cations to a data point are adversarial if they cause a classi er to be far less accurate on modi ed data points than the original data (Diochnos et al., 2018). These modi ed inputs are called *adversarial examples* and the algorithms that create them are called *adversarial attacks*. The algorithms that create counterfactuals are very similar to adversarial attacks and Pawelczyk et al. (2022) showed they produce similar outputs, which leads to the troubling conclusion that many counterfactuals may act as adversarial examples and change the classi er decision (individual is now offered a loan) without changing the true underlying class probabilities (individual is still likely to default on the loan). The adversarial vulnerability of classi ers is separate from causality concerns. For this reason, we introduce a novel two step procedure where (1) we generate a suggested change and (2) we use an independently trained *veri er* to certify that this change is not acting as an adversarial example. We present a methodology for training this veri er and provide analytical results showing that it is PAC-learnable (Theorem 2.3).

Flexible Goal De nition. AC/AR focus solely on the nal classi cation of a data point, but this may not always be suf cient or feasible. For instance a valid AC/AR may lead to a 51% likelihood of paying off a loan, but this may not satisfy the individual. Additionally, a change that improves a cancer patient's odds of survival from 15% to 40% would not constitute a valid AC/AR even though it would be very useful. Accordingly, our framework de nes goals through a *target set* of acceptable outcomes that can be tailored to an individual's needs, and we demonstrate how these target sets can be designed. We note that ICR (Kl. et al., 2023) and one of the AC/AR methods (Dandl et al., 2020) propose the use of goals other than nal classi cation, but our formulation is more exible and applies to multi-class scenarios. We develop a principled measure of reward by de ning a distance to the target set using statistical divergence. We analyze this distance theoretically in Theorem 2.2.

Real World Ef ciency. Previous works on CE and AC/AR reduce the amount of changes made by a counterfactual by minimizing a weighted ℓ_1-norm of the changes (with the exception of Ramakrishnan et al. (2020)), however these norms often fail to represent the real world cost of a change. Alternatively, we minimize a cost measure built speci cally to re ect real world costs of a change. By using this measure of real world cost and principled measure of rewards (distance to target set), TAP can suggest more ef cient advice. We present a few examples of the utility of producing ef cient advice through TAP: (a) Suggest the course of treatment that would double a patient's odds of survival while requiring the least staff hours. (b) List the skills an job applicant could acquire in the least amount of time that would lead to a high probability of receiving an interview. (c) Find the cheapest modi cations to a product that would bring it into a more premium price range. We illustrate through experiments on real world data how the use of application speci c cost functions leads to more ef cient advice.

Figure 1(a) illustrates our framework of Trustworthy Actionable Perturbations (TAP) for using feasible actions, true cost and an individualized goal to create an ef cient change, which is then veri ed to ensure that the change affects the true class probabilities instead of acting adversarially. Our goal to change the true class probabilities (real world outcomes) differs from previous CE and AC/AR works that seek only to change the classi er's decision. We share our goal with ICR which is focused on ensuring that only features causally related to the class are modi ed. Our framework, on the other hand, focuses on ensuring that the changes do not exploit the brittleness of ML classi ers and cause misclassi cations. This can occur regardless of whether modi ed features are causally related to the output. Figure 1(b) provides a summary of the objectives and features of various existing approaches alongside TAP.

2. Trustworthy Actionable Perturbations

Problem Setting and Goals: Suppose there is an unknown distribution $(x; C) \quad D$. Here x is a member of the input space $X \quad R^d$ and $C \ 2 \ f1; :::; kg$ is the class of x. We define the true class probabilities $y(x) := (P(C = 1 jx); :::; P(C = k jx))$. We let Y denote the k-simplex and use a classifier $M : X \to Y$ to estimate $y(x)$. Our goal in designing TAP is as follows: Given an input x with an undesirable classification $M(x)$, find the most efficient real world actions to create a modified input x^* such that the corresponding true probabilities $y(x^*)$ (and not just $M(x^*)$) are more desirable.

Real World Actionability: TAP should only suggest modifications that are feasible in the real world (e.g., not decreasing an individual's age). To this end, we introduce the Actionable Set $A(x)$ of a data point x as the set of all perturbations of x that are feasible in the real world. For example, if X represents loan applications with x_1 the age of the applicant, x_2 the applicant's credit score, x_3 the amount of credit and x_4 the loan duration, the actionable set could be $A(x) = f x^* \ 2 \ X j x^*_1 = x_1; x^*_2 = x_2 g$, i.e. the applicant can change the size and duration of the loan they request, but not their age or credit score. Previous works have examined the complexities of actionability including causal relations between inputs, e.g. one can't increase their education without an increase in age (Mahajan et al., 2019; Karimi et al., 2020b). All of these considerations, as well as a limiting changes to attributes which are believed to have a causal link to the output, can be incorporated into $A(x)$.

Efficiency: The definition of the most efficient change depends on the context of the problem and could involve a well-defined value such as "cost in dollars" or more nebulous value such as "amount of effort required." We characterize this value with a function $d_X : X \quad X \to R$, where $d_X(x; x^*)$ is the cost of changing x to x^*. For example, if x and x^* represent resumes, then $d_X(x; x^*)$ could represent the time it would take to acquire the attributes listed on resume x^*, but not on x. We note this function may not be a true distance measure. For example d_X if represents the difference in financial cost between two courses of medical treatment, then $d_X(x; x^*)$ should be negative when x^* is more affordable than x.

Desirability: We now define what we mean by a desirable outcome—the goal of a TAP. The Target Set T is the set of all elements of Y that would be an acceptable result of a TAP. If we wish to belong to a desirable class w with probability no less than p, the target set would have the form $T = f z \ 2 \ Y j z_w \quad p g$. If our goal is rather to avoid some undesirable class u, T could be of the form $T = f z \ 2 \ Y j z_u \quad q g$ for a fixed q. More generally, if we wish to belong to a set of desirable classes W with probability at least p and we wish to belong to a set of undesirable classes U with probability no greater than q, we would use

$$T = \left\{ z \ 2 \ Y \quad \sum_{i \ 2 \ W} z_i \quad p; \quad \sum_{i \ 2 \ U} z_i \quad q \right\} : \qquad (1)$$

We must quantify how close an TAP comes to achieving its goal in a principled manner. To do this, we first choose a measure of statistical distance $D(y jj z)$ (we use Kullback-Leibler (KL) Divergence). We then denote $d_Y(y; T)$ as the distance of y to the target set T, defined as follows:

$$d_Y(y; T) := \inf_{z \ 2 \ T} D(y jj z): \qquad (2)$$

We may now formally define Trustworthy Actionable Perturbations. Let $$ represent budget —the amount of work we are willing to perform, and represent tolerance —how close the final result is to our target set T.

Definition 2.1 (($; $)-Trustworthy Actionable Perturbation) x^* is an $(;)$-trustworthy actionable perturbation for x and a target set T if

1. $d_X(x; x^*) \quad$
2. $d_Y(y(x^*); T) \quad$
3. $x^* \ 2 \ A(x)$.

In order to verify the second condition we must be able to calculate d_Y. Fortunately, the optimization problem in (2) has a differentiable closed form solution when $D(y jj z)$ is an f-divergence: a broad class of measures including KL-divergence, total-variation (TV) and other commonly used statistical distances. An f-divergence is defined as $D(y jj z) = \sum_{i=1}^{k} z_i f\left(\frac{y_i}{z_i}\right)$, where f is a convex function satisfying $f(1) = 0$ and $f(0) = \lim_{x \to 0^+} f(x)$ (Polyanskiy & Wu, 2024). Theorem 2.2 describes the solution to (2).

Figure 2. Illustration of the partition on Y used to calculate the distance from the target set T in Theorem 2.2. Although the cost function takes different functional form(s) in the four regions, it is continuously differentiable in the entire space.

Theorem 2.2. If $D(y\|z)$ is an f-Divergence with twice differentiable f and T is of form (1), then

$$d_Y(y;T) = \begin{cases} 0 & \text{if } y \in A \\ pf\left(\frac{S_W}{p}\right) + (1-p)f\left(\frac{1-S_W}{1-p}\right) & \text{if } y \in B \\ qf\left(\frac{S_U}{q}\right) + (1-q)f\left(\frac{1-S_U}{1-q}\right) & \text{if } y \in C \\ pf\left(\frac{S_W}{p}\right) + qf\left(\frac{S_U}{q}\right) & \\ \quad +(1-p-q)f\left(\frac{1-S_W-S_U}{1-p-q}\right) & \text{if } y \in D \end{cases} \tag{3}$$

where $S_W = \sum_{i \in W} y_i$, $S_U = \sum_{i \in U} y_i$ and the sets A, B, C and D are a partition of Y defined and visualized in Figure 2. Furthermore, $d_Y(y;T)$ is continuously differentiable in y over its entire domain.

Equation (3) in Theorem 2.2 is easily calculable and continuously differentiable despite its piece-wise form, which will be significant when creating TAP (see Section 3). The proof of Theorem 2.2 involves showing that optimization problem (2) is convex and finding a value of z that satisfies the KKT conditions. This proof and additional results about d_Y are found in the Appendix A.1.

Real-world Verifiability of TAP: Note that TAP are defined with respect to the true class probabilities $y(x)$ because TAP should have an effect in the real world. Notwithstanding, $y(x)$ is unknown and we must use $M(x)$ to create our TAP (more details in Section 3), which introduces the risk that we might produce an x' that has the desired effect on $M(x)$ but not on $y(x)$ (like an adversarial example). This is of particular concern because TAP and all other counterfactuals are created by solving an optimization problem of the form

$$x' = \arg\min_{x'} \ \text{loss}(x', w) + \ \text{dist}(x', x); \tag{4}$$

which is precisely how most adversarial examples are created (Pawelczyk et al., 2022). When counterfactuals were first introduced to ML (Wachter et al., 2017), the concern that counterfactuals would act as adversarial examples was dismissed because the adversarial attacks of the time 1) modified many more features than counterfactuals and 2) were targeted almost exclusively at image data whereas counterfactuals were proposed for use on tabular data. Since that time, Gourdeau et al. (2021); Su et al. (2019) demonstrated that adversarial attacks can be effective when changing a very small number of features, and several works (Ballet et al., 2019; Mathov et al., 2020; Cartella et al., 2021; Kumar et al., 2021) have shown that adversarial examples exist on tabular data sets. This implies that verification is necessary to achieve results that can be trusted to change the true class probabilities.

Verifying x' may appear similar to detecting adversarial examples, which has been the object of significant research (Yang et al., 2020; Roth et al., 2019; Fidel et al., 2020; Carlini & Wagner, 2017a) with no satisfactory solution. Fortunately, we have an important advantage over detecting adversarial examples: we know the original data point x and exactly how it was modified, i.e. x, x'. To capitalize on this knowledge, we propose a novel verification procedure using a classifier $V : X \times X \to [0,1]$ which compares two inputs simultaneously and predicts the probability of the inputs belonging to the same class: the value $V(x,x')$ estimates $P(C = C' | x, x')$. Because V has a different classification task from M, attacks targeted against M should not be effective against V, and we can use the discrepancy between estimates of M and V to determine if an x' acts adversarially on M. In order to make this comparison, we use the fact that $P(C = C' | x, x')$ can also be estimated using M by calculating $\sum_{i=1}^{k} M_i(x) M_i(x')$. If x' acts adversarially we would expect $\sum_{i=1}^{k} M_i(x) M_i(x')$ to be very small while $V(x,x')$ is large. If x' is not adversarial we would expect similar values from both $\sum_{i=1}^{k} M_i(x) M_i(x')$ and $V(x,x')$. Accordingly, we define

$$\Delta(x,x') := \left| V(x,x') - \sum_{i=1}^{k} M_i(x) M_i(x') \right|; \tag{5}$$

and verify that an x' is trustworthy only if $\Delta(x,x') < \tau$. In Section 3, we describe how we selected the threshold τ.

Training a Verifier & PAC Learnability: In order to create V, we must have data on which it can be trained. We build this difference training data by creating all possible pairs of elements from our original training data and labeling the pairs by whether they belong to the same class (1 for the same class, 0 for different classes). If the original training data is $\{x^{(i)}, C^{(i)}\}_{i=1}^{n}$, the difference training data is $\{(x^{(i)}, x^{(j)}), z^{(i,j)}\}_{1 \le i,j \le n}$, where $z^{(i,j)} = \mathbb{1}[C^{(i)} = C^{(j)}]$. We use the same architecture for V as M (only changing the number of inputs and outputs), but differing architectures could also be used. Now that we have a method for training V, we show that training in this way leads to a generalizable verifier. To this end, we next present a probably approximately correct (PAC) bound on V's generalization gap which depends on n (number of training samples), k (number of classes), and d (data dimensionality).

Theorem 2.3. Let $R(V)$ be the true risk of a verifier V over data drawn from D and $\hat{R}_S(V)$ be the empirical risk over a sample S of labelled point pairs drawn i.i.d. from D. Both risks are defined using a bounded loss function $\ell(\cdot,\cdot) \le B$. Also let V be selected from a function class $\mathcal{V}$. Then for any $\delta \in (0,1)$, with probability $(1-\delta)$, the following bound on the generalization gap holds.

$$\sup_{V \in \mathcal{V}} \left| R(V) - \hat{R}_S(V) \right| = O\left(\sqrt{\frac{k}{n^2 k^{2n}}} \right)^{1/d} \tag{6}$$

Here the terms with explicit dependence on δ have been suppressed because they are dominated by the term in (6).

The precise generalization bound is presented (43) in the Appendix A.2.

To prove Theorem 2.3, we construct a de nition of risk that ts this new learning scenario (i.e., learning if two samples are from the same class or not, as opposed to conventional classi cation). This risk takes into account that we expect large imbalances between the number of point pairs from the same class and from different classes. In order to obtain the bounds on the generalization gap, we expand this risk into a sum of terms which can be bounded with existing Rademacher complexity PAC-methods. Finally, we bound the growth of these Rademacher complexity terms as a function of n, k and d to arrive at (6). The complete proof, including detailed de nitions of $R(V)$ and $\hat{R}_S(V)$ as well as additional discussion, is presented in the Appendix A.2.

Remark 2.4. The bound in Theorem 2.3 is small as long as $n \gg k^2$ and n is exponentially larger than d. The relation between n and k is crucial because it implies that the denominator $\sqrt{n^2 - k^2 n} \approx n^2 - k$. This differs from typical PAC bounds where the primary requirement is n be exponentially larger than d (Theorem 4.3 in Gottlieb et al. (2016)) and have mild dependence on the number of classes k. The key implication of this result is when using a veri er as described in this paper, as the data sets used increase in number of classes k, it is esseptial that the amount of training data increases at a rate of $\gg k$.

3. Generating TAP

Two Step Creation Method: We now present and discuss the general optimization framework for creating TAP. Ideally, we would like to solve the following optimization problem: $\arg\min_{x^* \in A(x)} d_Y(y^*, T) + d_X(x^*, x)$, where the scalar parameter balances the effort (-reward) trade-off. Solving this optimization would be guaranteed to create an effective TAP; unfortunately $y(x^*)$ is unknown and we cannot solve this problem directly. Instead propose the following two-step procedure where: in Step 1, we treat $M(x^*)$ as a surrogate for $y(x^*)$, and in Step 2, we use a veri cation algorithm to ensure that x^* is not just fooling the classi er.

$$\text{Step 1}: \arg\min_{x^* \in A(x)} d_Y(M(x^*), T) + d_X(x^*, x) \quad (7)$$

$$\#$$

$$\text{Step 2}: \text{Verify } M(x^*) \quad y(x^*) \text{ i.e. } (x, x^*) \quad (8)$$

$$\#$$

$$\text{TAP}$$

Solving Step 1: We solve (7) using gradient descent which requires us to use differentiable models M and formulate d_X in a differentiable manner (d_Y is differentiable according to Theorem 2.2). We modify our gradient descent to address two challenges. (1) We must insure that our solution is actionable $x^* \in A(x)$. (2) Our solution x^* must follow any formatting rules associated with the data set (for instance, Boolean variables must be either 0 or 1, categorical features must respect one-hot encoding, etc.). A perturbation that follows these formatting rules is called coherent. To solve these two difficulties, we first assume $A(x) = \{x^* | l_i \le x^*_i \le u_i, 1 \le i \le d\}$ for some set of lower bounds $\{l_i\}_{i=1}^d$ and upper bounds $\{u_i\}_{i=1}^d$. An attribute is immutable if $l_i = u_i$. We ensure actionability by setting all elements of the gradient corresponding to immutable features to zero and adding a large penalty $a(x^*)$ term to the objective function which punishes points for leaving the actionable set. To ensure coherence, we project the result of our gradient descent onto the coherent space by using a function $\text{cond}: R^m \to X$ which performs the appropriate value rounding to make an input coherent. We found it useful to introduce a second penalty term $p(x^*)$ which requires that any one-hot encoded features sum to 1. This ensures our answers never stray too far from a coherent point and improves robustness. Details on a, p and cond are found in the Appendix A.3.3. In practice we also found it useful to replace regular gradient descent with the ADAM algorithm (Kingma & Ba, 2014).

Algorithm 1 Generating TAP

Input: Classi ers M & V, point x, target family T, learning rate α, veri cation-cut off τ
$x^* \leftarrow x$
while x^* not converged **do**
  $g \leftarrow \nabla_{x^*}(d_Y(M(x^*), T) + d_X(x^*, x) + a(x^*) + p(x^*))$
  $g_j \leftarrow 0$ for all immutable features j.
  $x^* \leftarrow x^* - \alpha g$
end while
$x^* = \text{cond}(x^*)$ (project onto the coherent space)
$\tau' = d_X(x^*, x); d_Y(M(x^*), T)$
if τ' and τ requirements NOT met **then**
  Adjust τ (see text for explanation)
  Return to while loop
end if
if $V(x, x^*) \ne \sum_{i=1}^k M_i(x) M_i(x^*)$ **then**
  Adjust problem parameters (see text for explanation)
  Restart algorithm
end if
return x^*

Solving Step 2: In Section 2, we discussed the necessity of veri cation and suggested that an TAP can be trusted if $\tau(x, x^*) = V(x, x^*) \approx \sum_{i=1}^k M_i(x) M_i(x^*)$ is smaller than a threshold τ. Our process for choosing τ starts with deciding on an acceptable risk of eliminating a truly effective TAP (we use 10%). To find the corresponding τ to this risk, we calculate $\tau(x^{(i)}, x^{(j)})$ for a sufficiently large number of pairs $(x^{(i)}, x^{(j)})$ from the testing data such that

Figure 3. Table containing details on data sets used for testing.

$C^{(i)} \in C^{(j)}$. Finally, we pick γ such that only the desired percentage of $(x^{(i)}, x^{(j)})$ values (e.g. 10%) are above γ. The verification procedure is now reduced to eliminating any x that results in $(x, x) > \gamma$.

Adjusting for Suitability and Verifiability: When creating TAP we will often have a particular budget δ (or tolerance ϵ) bound we need to satisfy. To find a suitable TAP we repeat Step 1 of our process adjusting γ until the desired budget or tolerance is met: increasing γ to decrease δ and decreasing γ to decrease ϵ. It may also be appropriate to use a variety of γ values and plot the δ and ϵ values of each resulting TAP (see Figure 4). The user may then select a TAP they see as offering particularly good value. When a TAP fails the verification step, there are a few recourses. (1) Sometimes it is sufficient to decrease γ, putting greater emphasis on reaching the target set. (2) "Shrink" the target set (increase the value of ρ and decrease the value of ϕ) in order to force the algorithm to find more effective changes. (3) Add a random perturbation to x in order to move the starting point away from the adversarial example. The entire procedure is described in Algorithm 1.

4. Experimental Results

Data Sets: We compare TAP, counterfactuals and adversarial attacks on four data sets from different fields; data set details are found in Figure 3 and the Appendix A.3.1. The associated code can be found at https://github.com/JesseFriedbaum/TAP_code.

Adult Income (Kohavi & Becker, 1996): This data set contains demographic information on Americans labelled by whether they had a high income. The actionable set $A(x)$ allows individuals to increase their education, change jobs and adjust their weekly work hours. The cost function d_x sums the expected number of years to improve education, a one-year cost to change jobs and the square of the change in hours worked (weighted so an additional 3 hours of work per week is equal to a year spent on education).

Law School Success (Wightman, 1998): This data set contains information on law school students labelled by whether they passed the BAR exam. $A(x)$ allows changes to law school grades (through more studying) and the region where the exam is taken. The cost function d_x sums the increase in grades and the physical distance travelled to take the BAR. Moving to an adjacent region (Far West to North West) is weighted equal to increasing grades one standard deviation.

Diabetes Prediction (for Disease Control & , CDC): The individuals in this data set are labelled by whether they have diabetes. We define $A(x)$ to allow changes in health habits, BMI, education and income. We use a weighted 2-norm for d_x to represent the relative difficulty of making changes. For example, starting to get regular physical activity is weighted the same as dropping one BMI.

German Credit (Hofmann, 1994): This data set contains loan applications. In $A(x)$, we allow for changes to the loan duration and size and funds in the checking and savings accounts. We use d_x to measure the total difference in Deutsche Marks (DM) over all elements of the application.

Other Methods: We compare our results against counterfactuals created using the original method proposed to create counterfactuals (Wachter et al., 2017) and the diverse counterfactuals (DICE) method in (Mothilal et al., 2020), the most cited methods in the literature. These methods use an ℓ_p norm based cost function that often fails to reflect real world costs (see examples on the next page). We also compare TAP against the Carlini & Wagner (2017b) ℓ_2 adversarial attack, one of the most well known and effective adversarial attacks. The counterfactuals belong to the same actionable set as the TAP, but the adversarial examples are not limited to an actionable set and may not be coherent.

Models: Gradient boosted tree algorithms (Friedman, 2001) are considered state of the art architectures for tabular data classification (Shwartz-Ziv & Armon, 2022). Unfortunately,

Figure 4. Cost-Benefit plots of TAP and counterfactuals for an individually from the Law School data set with grades measured in standard deviations from the mean (a) and an individual in the Adult Income data set (b).

these models are not differentiable and cannot be used with our framework. Instead we use neural networks which we tuned until they provide accuracy on par with gradient boosted tree models on the same data set. Details on our models' structure and training are given in Appendix A.3.2.

Representative Examples of TAP and Trade-off between cost/desirability: We first examine two representative examples of how TAP behave differently than counterfactuals for specific individuals. Figure 4 shows a plot of the values of TAP and counterfactuals for one individual in the Law School data set and one individual in the Adult Income data set. We examine the results from the Law School dataset: The TAP labelled TAP-1 suggests only a mild 0.2 (standard deviation) increase in grades and the relatively short move from the Far West to the Great Lakes region resulting in a small 11% increase in the chance of passing the BAR. On the other hand, TAP-2 suggest a larger increase in grades and a longer move which results in a much larger 34% increase to the odds of success. Finally the counterfactual CF-1 suggest an enormous increase in grades and a massive cross country move to achieve 51% increase in the odds of success. Turning our attention to the Adult Income example: TAP-3 suggests a relatively simple increase in education to the masters level resulting in 20% increase to the odds of a high income. Alternatively, TAP-4 achieves an 71% increase by suggesting far more changes including a professional degree and becoming self-employed. The counterfactual CF-2 does not suggest becoming self-employed and produces a smaller 67% increase in the odds of high income despite also suggesting a professional degree and a drastic 16 hour increase in the hours worked per week.

These examples illustrates two trends: 1) TAP offer both low-cost/low-reward (large/small- δ) and high-cost/high-reward options, whereas counterfactual methods (Wachter et al., 2017; Mothilal et al., 2020) offer only high-cost options. This is because TAP are defined by distance to the target set, but counterfactuals are defined as belonging to the desirable class. That rules out any advice that doesn't result in the desirable class being the most likely class. 2) Counterfactuals are prone to suggesting very high-cost outliers. This has two main causes: (a) The ℓ_1-norm used to create the counterfactuals does not accurately represent real world effort. For example this norm considers any move in region to cost the same regardless of actual distance. (b) Because counterfactuals do not use a target set, they are prone to "overshooting" the desired goal. For example CF_1 resulted in a 95% chance of passing the BAR when our goal was only 85%.

Comparison of TAP vs. Other Approaches: We now compare TAP, counterfactuals (Wachter et al., 2017; Mothilal et al., 2020) and CW attacks (Carlini & Wagner, 2017b) over the entire data sets. In Figure 5: Each bar chart refers to a particular data set and desired distance δ to the target set T. Each bar shows the percentage of individuals that a method was able to move inside the goal δ at a variety

Figure 5.a) & b) show average success rate for moving individuals within a variety of distances (ℓ_1) to the target set. The y-axis shows the percentage of individuals within the goal distance, and the x-axis, represents different values of ϵ. c) Summarizes success values for all data sets. The upper (red) value for each row is the success rate before the verification procedure and the lower (green) value is the success rate after verification with a 10% chance of rejecting valid examples.

of costs. (Bar charts for all data sets are found in the Appendix A.3.4.) The table summarizes this information for all data sets with the upper (red) value in each cell representing the data before the verification procedure and the lower (green) value the success rate after the verification procedure. Consider the bar chart on the top middle which refers to the German Credit data and a goal of $\epsilon = 0.5$ from the target (the same information as the last three columns of the table). At a $= 0$ Deutsche Marks (DM) cost, TAP are able to move 73% of individuals within the goal range by closing empty accounts. Counterfactuals do not match this success until the cost $= 7,000$ DM, and CW attacks never achieve more than a 31% success rate. TAP outperform counterfactuals in all of the test scenarios.

Impact and Effectiveness of Verifier: The first important take away from the success rates after verification is that the verifier was 100% effective at eliminating Carlini Wagner adversarial examples (visible in the bottom row of the table in Figure 5 c), implying that the verification method does indeed eliminate inputs that fool the classifier. Importantly, the verification procedure also removes a significant number of TAP and counterfactuals. Consider the second column of Figure 5 c: Out of all TAP generated, 14% appeared effective but were eliminated by the verification procedure. Counterfactual methods fared even worse with 20% to 27% of counterfactuals eliminated. This reinforces the necessity of a verification procedure.

Concluding Remarks & Future Work: In this work, we proposed Trustworthy Actionable Perturbations (TAP) which leverage ML classifiers to find efficient actions to

achieve real world results. Our proposed framework introduces a novel veri cation procedure, exible de nition of goals, and principled reward measure for use in generating counterfactuals. We demonstrated their effectiveness when compared to other methods on data sets from multiple elds. Finally we note that our framework is exible enough to incorporate contributions from previous works on counterfactuals such as individualized cost measures (De Toni et al., 2023), causal relations between inputs (Mahajan et al., 2019; Karimi et al., 2020b), causal relationships to the output (König et al., 2023), and advanced optimization methods (Guidotti et al., 2018; Karimi et al., 2020a).

Impact Statement

As the use of AI and ML expands into critical applications such as healthcare, criminal justice, and hiring, the importance of explaining decisions deemed unfavorable and providing recourse to such users has grown signi cantly. In this context, our paper introduces a novel contribution aimed at making recourse mechanisms more trustworthy. We present a exible framework, Trustworthy Actionable Perturbations (TAP), designed to generate cost-effective recourse which can ensure that the recourse being provided to users results in real-world changes. TAP can be useful to both end-users and institutions that suggest the recourse. The technical tools and the analytical results developed in the paper (including a exible target set, and a novel pairwise veri cation procedure) can also nd use and lead to new insights for other problems such as cost-sensitive learning and adversarial defense.

Acknowledgements

We thank the anonymous ICML reviewers and the area chairs for their insightful suggestions. This work was supported by NSF grants CAREER 1651492, CCF-2100013, CNS-2209951, CNS-1822071, CNS-2317192, and by the U.S. Department of Energy, Of ce of Science, Of ce of Advanced Scienti c Computing under Award Number DE-SC-ERKJ422, and NIH Award R01-CA261457-01A1.

References

Ballet, V., Renard, X., Aigrain, J., Laugel, T., Frossard, P., and Detyniecki, M. Imperceptible adversarial attacks on tabular data. arXiv preprint arXiv:1911.03274, 2019.

Bartlett, P. L. and Mendelson, S. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. Journal of Machine Learning Research, 3(Nov):463–482, 2002.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM workshop on arti cial intelligence and security, pp. 3–14, 2017a.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57, 2017b.

Cartella, F., Anunciacao, O., Funabiki, Y., Yamaguchi, D., Akishita, T., and Elshocht, O. Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. arXiv preprint arXiv:2101.08030, 2021.

Dandl, S., Molnar, C., Binder, M., and Bischl, B. Multi-objective counterfactual explanations. In International Conference on Parallel Problem Solving from Nature, pp. 448–469. Springer, 2020.

De Toni, G., Viappiani, P., Teso, S., Lepri, B., and Passerini, A. Personalized algorithmic recourse with preference elicitation. Transactions on Machine Learning Research, 2023.

Diochnos, D., Mahloujifar, S., and Mahmoody, M. Adversarial risk and robustness: General de nitions and implications for the uniform distribution. Advances in Neural Information Processing Systems, 31, 2018.

Fidel, G., Bitton, R., and Shabtai, A. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. In 2020 international joint conference on neural networks (IJCNN), pp. 1–8. IEEE, 2020.

for Disease Control, C. and (CDC), P. Behavioral risk factor surveillance system survey data (brfss), 2015. URL https://www.cdc.gov/brfss/index.html .

Freiesleben, T. The intriguing relation between counterfactual explanations and adversarial examples. Minds and Machines, 32(1):77–109, 2022.

Friedman, J. H. Greedy function approximation: a gradient boosting machine. Annals of statistics, pp. 1189–1232, 2001.

Gottlieb, L.-A., Kontorovich, A., and Krauthgamer, R. Adaptive metric dimensionality reduction. Theoretical Computer Science, 620:105–118, 2016.

Gourdeau, P., Kanade, V., Kwiatkowska, M., and Worrell, J. On the hardness of robust classi cation. Journal of Machine Learning Research, 22(273):1–29, 2021.

Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., and Giannotti, F. Local rule-based explanations of black box decision systems. arXiv preprint arXiv:1805.10820, 2018.

Hofmann, H. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

Karimi, A.-H., Barthe, G., Balle, B., and Valera, I. Model-agnostic counterfactual explanations for consequential decisions. In International Conference on Arti cial Intelligence and Statistics, pp. 895–905. PMLR, 2020a.

Karimi, A.-H., Von Kügelgen, J., Schölkopf, B., and Valera, I. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. Advances in neural information processing systems, 33:265–277, 2020b.

Karimi, A.-H., Schölkopf, B., and Valera, I. Algorithmic recourse: from counterfactual explanations to interventions. In Proceedings of the 2021 ACM conference on fairness, accountability, and transparency, pp. 353–362, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

Kohavi, R. and Becker, B. Uci adult data set. UCI machine learning repository, 1996.

König, G., Freiesleben, T., and Grosse-Wentrup, M. Improvement-focused causal recourse (icr). Proceedings of the AAAI Conference on Arti cial Intelligence, volume 37, pp. 11847–11855, 2023.

Kumar, N., Vimal, S., Kayathwal, K., and Dhama, G. Evolutionary adversarial attacks on payment systems. In 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 813–818. IEEE, 2021.

Leo, M., Sharma, S., and Maddulety, K. Machine learning in banking risk management: A literature review. Risks, 7(1):29, 2019.

Mahajan, D., Tan, C., and Sharma, A. Preserving causal constraints in counterfactual explanations for machine learning classi ers. arXiv preprint arXiv:1912.03277, 2019.

Martinez Neda, B., Zeng, Y., and Gago-Masague, S. Using machine learning in admissions: Reducing human and algorithmic bias in the selection process. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, pp. 1323–1323, 2021.

Mathov, Y., Levy, E., Katzir, Z., Shabtai, A., and Elovici, Y. Not all datasets are born equal: On heterogeneous data and adversarial examples. arXiv preprint arXiv:2010.03180, 2020.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. Foundations of Machine Learning. MIT press, 2018.

Mothilal, R. K., Sharma, A., and Tan, C. Explaining machine learning classi ers through diverse counterfactual explanations. In Proceedings of the 2020 conference on fairness, accountability, and transparency, pp. 607–617, 2020.

Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In Proceedings of the AAAI conference on arti cial intelligence, volume 29, 2015.

Pawelczyk, M., Agarwal, C., Joshi, S., Upadhyay, S., and Lakkaraju, H. Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis. In International Conference on Arti cial Intelligence and Statistics, pp. 4574–4594. PMLR, 2022.

Polyanskiy, Y. and Wu, Y. Information theory: From coding to learning. 2024.

Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., and Flach, P. Face: feasible and actionable counterfactual explanations. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 344–350, 2020.

Ramakrishnan, G., Lee, Y. C., and Albarghouthi, A. Synthesizing action sequences for modifying model decisions. In Proceedings of the AAAI Conference on Arti cial Intelligence, volume 34, pp. 5462–5469, 2020.

Roth, K., Kilcher, Y., and Hofmann, T. The odds are odd: A statistical test for detecting adversarial examples. In International Conference on Machine Learning, pp. 5498–5507. PMLR, 2019.

Sauer, C. M., Dam, T. A., Celi, L. A., Faltys, M., de la Hoz, M. A., Adhikari, L., Ziesemer, K. A., Girbes, A., Thoral, P. J., and Elbers, P. Systematic review and comparison of publicly available icu data sets—a decision guide for clinicians and data scientists. Critical care medicine, 50 (6):e581–e588, 2022.

Shwartz-Ziv, R. and Armon, A. Tabular data: Deep learning is not all you need. Information Fusion, 81:84–90, 2022.

Su, J., Vargas, D. V., and Sakurai, K. One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation, 23(5):828–841, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

Ustun, B., Spangher, A., and Liu, Y. Actionable recourse in linear classi cation. In Proceedings of the conference on fairness, accountability, and transparency, pp. 10–19, 2019.

Wachter, S., Mittelstadt, B. D., and Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the gdpr.Cybersecurity2017.

Wightman, L. F. Lsac national longitudinal bar passage study. lsac research report series. 1998.

Yang, P., Chen, J., Hsieh, C.-J., Wang, J.-L., and Jordan, M. Ml-loo: Detecting adversarial examples with feature attribution. InProceedings of the AAAI Conference on Arti cial Intelligence, volume 34, pp. 6639–6647, 2020.

A. Appendix

The Appendix is organized as follows:

A.1. Proof of Theorem 2.2 (Analysis of statistical distance d_Y to the target set T)

Recall that our target sets have the form

$$T = \left(\sum_{i \in W} z_i \geq p ; \sum_{i \in U} z_i \geq q \right);$$

where either W or U could be empty. Also recall

$$d_Y(y; T) = \min_{z \in T} D_f(y \| z) = \min_{z \in T} \sum_{i=1}^{k} z_i f\left(\frac{y_i}{z_i}\right) : \tag{9}$$

We must prove three facts: A) $d_Y(y; T)$ has the closed form found in equation (3), B) This function is continuous, C) the derivative of the function is continuous. We begin by proving the closed form equation.

Our proof of the closed form of $d_Y(y; T)$ will be made easier by introducing notation $N = (W \cup U)^C$ as the neutral classes that are neither desirable nor undesirable. We will use the fact that $S_W + S_U + S_N = 1$ to rewrite (3) as

$$d_Y(y; T) = \begin{cases} 0 & \text{if } S_W \geq p \text{ and } S_U \geq q \\ p f\left(\frac{S_W}{p}\right) + (1-p) f\left(\frac{S_U + S_N}{1-p}\right) & \text{if } S_W < p \text{ and } S_U \geq (1-S_W)\frac{q}{1-p} \\ q f\left(\frac{S_U}{q}\right) + (1-q) f\left(\frac{S_W + S_N}{1-q}\right) & \text{if } S_U > q \text{ and } S_W \geq (1-S_U)\frac{p}{1-q} \\ p f\left(\frac{S_W}{p}\right) + q f\left(\frac{S_U}{q}\right) + (1-p-q) f\left(\frac{S_N}{1-p-q}\right) & \text{if } S_U > (1-S_W)\frac{q}{1-p} \\ & \text{and } S_W < (1-S_U)\frac{p}{1-q} \end{cases};$$

where $S_W = \sum_{i \in W} y_i$, $S_U = \sum_{i \in U} y_i$ and $S_N = \sum_{i \in N} y_i$.

The case where $y \in T$ is obvious so we consider only the case where $y \notin T$, First note that f-divergence $D_f(y \| z)$ is convex in z. Furthermore T is a convex set. Therefore any z satisfying the KKT conditions is a minimizer. The KKT

conditions for this problem can be written as

$$\nabla L(z) = 0 \tag{10}$$

$$\sum_{i=1}^{k} z_i = 1 \tag{11}$$

$$p - \sum_{i \in W} z_i \le 0 \tag{12}$$

$$\sum_{i \in U} z_i - q \le 0 \tag{13}$$

$$\lambda_1, \lambda_2 \ge 0 \tag{14}$$

$$\lambda_1 \left(p - \sum_{i \in W} z_i \right) = 0 \tag{15}$$

$$\lambda_2 \left(q - \sum_{i \in U} z_i \right) = 0, \tag{16}$$

where the Lagrangian is defined by

$$L(z) = \sum_{i=1}^{k} z_i f\left(\frac{y_i}{z_i}\right) + \sum_{i=1}^{k} z_i + \lambda_1 \left(p - \sum_{i \in W} z_i \right) + \lambda_2 \left(\sum_{i \in U} z_i - q \right).$$

Note that we have neglected to explicitly state the requirement that $z_i \ge 0$ for all i. This is because our eventual solution will satisfy these bounds anyways, and omitting these bounds will drastically simplify our calculations. We now rewrite (10) as

$$f\left(\frac{y_i}{z_i}\right) - \frac{y_i}{z_i} f'\left(\frac{y_i}{z_i}\right) + \lambda_1 = 0 \qquad i \in W \tag{17}$$

$$f\left(\frac{y_i}{z_i}\right) - \frac{y_i}{z_i} f'\left(\frac{y_i}{z_i}\right) + 1 + \lambda_2 = 0 \qquad i \in U \tag{18}$$

$$f\left(\frac{y_i}{z_i}\right) - \frac{y_i}{z_i} f'\left(\frac{y_i}{z_i}\right) + 1 = 0 \qquad i \in N \tag{19}$$

We now propose a solution can be found where that the ratios $\frac{y_i}{z_i}$ are constant in each of the sets W, U, N. That is

$$\begin{aligned} z_i &= C_W y_i && i \in W \\ z_i &= C_U y_i && i \in U \\ z_i &= C_N y_i && i \in N. \end{aligned}$$

In that case we can satisfy conditions (17), (18) and (19) (originally (10)) by setting

$$\begin{aligned} \lambda &= C_N^{-1} f'(C_N^{-1}) - f(C_N^{-1}) \\ \lambda_1 &= \lambda + f(C_W^{-1}) - C_W^{-1} f'(C_W^{-1}) \\ \lambda_2 &= -f(C_U^{-1}) + C_U^{-1} f'(C_U^{-1}). \end{aligned}$$

We can now reformulate (14) so that it is easier to analyze. We will first define $h(x) = xf'(x) - f(x)$. Note that because $f(x)$ is convex $h'(x) = xf''(x) \ge 0$ for all $x \ge 0$ and $h(x)$ is increasing. We can then rewrite our formulas for λ, λ_1 and λ_2 ((17), (18) and (19)) as

$$\begin{aligned} \lambda &= h(C_N^{-1}) \\ \lambda_1 &= h(C_N^{-1}) - h(C_W^{-1}) \\ \lambda_2 &= h(C_U^{-1}) - h(C_N^{-1}). \end{aligned}$$

Then $_1 \quad 0$ becomes

$$h(C_N{}^1) \quad h(C_W{}^1)$$
$$C_N{}^1 \quad C_W{}^1$$
$$C_N \quad C_W ;$$

and $_2 \quad 0$ similarly becomes $C_N \quad C_U$. This implies (14) is equivalent to

$$C_U \quad C_N \quad C_W : \tag{20}$$

Our choice of , $_1$ and $_2$ (de ned in (17), (18) and (19)) satisfy (10), so we must now nd values of C_W, C_U and C_N that satisfy (11) through (16). We will consider 3 cases illustrated in Figure 6.

Figure 6. The three cases visualized in probability space.

Case: 1 Suppose $S_W < p$ and $S_U \quad (1 \quad S_W) \quad \frac{q}{1 \quad p}$.

Let $C_W = \frac{p}{S_W}$ and $C_U = C_N = \frac{1 \quad p}{S_U + S_N}$. This implies $_2 = 0$ which satis es (16) and $_2 \quad 0$ (half of (14)). This also implies $\sum_{i=2W} z_i = p$ satisfying (12) and (15). We will use the fact $S_U + S_N = 1 \quad S_W$ in our proof of condition (13).

$$\sum_{i2S_U} z_i = \sum_{i2S_U} C_U y_i = \frac{1 \quad p}{S_U + S_N} S_U$$
$$\frac{1 \quad p}{S_U + S_N}(1 \quad S_W) \quad \frac{q}{1 \quad p} = q$$

This proves (13) is satis ed.

Because $S_W < p$ we have

$$C_W = \frac{p}{S_W} > 1 > \frac{1 \quad p}{1 \quad S_W} = \frac{1 \quad p}{S_U + S_N} = C_N$$

This implies $_1 > 0$ and satis es the other half of (14).

We have now shown all the KKT conditions are satisfied and we have found a minimizer. We now plug these values into (9) to find a closed form for the distance.

$$\begin{aligned} d_Y(y^*, T) &= \min_{z \in T} \sum_{i=1}^{k} z_i f\left(\frac{y_i}{z_i}\right) \\ &= \sum_{i \in W} \frac{p y_i}{S_W} f\left(\frac{S_W}{p}\right) + \sum_{i \in W} \frac{(1-p) y_i}{S_U + S_N} f\left(\frac{S_U + S_N}{1-p}\right) \\ &= p f\left(\frac{S_W}{p}\right) + (1-p) f\left(\frac{S_U + S_N}{1-p}\right). \end{aligned}$$

Case: 2 Suppose $S_U > q$ and $S_W \leq (1 - S_U)\frac{p}{1-q}$.

Let $C_U = \frac{q}{S_U}$ and $C_W = C_N = \frac{1-q}{S_W + S_N}$. This implies $\lambda_1 = 0$ which satisfies (15) and $\lambda_1 \geq 0$ (half of (14)). We also have $\sum_{i \in U} z_i = q$ satisfying (13) and (16). We now prove condition (12) is satisfied.

$$\begin{aligned} \sum_{i \in S_W} z_i &= \sum_{i \in S_W} C_W y_i = \frac{1-q}{S_W + S_N} S_W \\ &\leq \frac{1-q}{S_W + S_N} (1 - S_U)\frac{p}{1-q} = p \end{aligned}$$

Finally we prove $C_N \geq C_U$ implying $\lambda_2 \geq 0$ which satisfies the other half of (14)

$$C_U = \frac{q}{S_U} < 1 < \frac{1-q}{1-S_U} = \frac{1-q}{S_W + S_N} = C_N$$

Now that we have proven that this is a minimizer we will again plug solution into (9) to find the distance value.

$$\begin{aligned} d_Y(y^*, T) &= \min_{z \in T} \sum_{i=1}^{k} z_i f\left(\frac{y_i}{z_i}\right) \\ &= \sum_{i \in U} \frac{q y_i}{S_U} f\left(\frac{S_U}{q}\right) + \sum_{i \in U} \frac{(1-q) y_i}{S_W + S_N} f\left(\frac{S_W + S_N}{1-p}\right) \\ &= q f\left(\frac{S_U}{q}\right) + (1-q) f\left(\frac{S_W + S_N}{1-q}\right). \end{aligned}$$

Case: 3 Suppose $S_U > (1 - S_W)\frac{q}{1-p}$ and $S_W < (1 - S_U)\frac{p}{1-q}$.

Let $C_W = \frac{p}{S_W}$, $C_U = \frac{q}{S_U}$ and $C_N = \frac{1-p-q}{S_N}$ in which case $\sum_{i \in W} z_i = p$ (satisfying (12) and (15)), $\sum_{i \in U} z_i = q$ (satisfying (13) and (16)). The choice of C_N ensures that (11) is satisfied:

$$\begin{aligned} \sum_{i=1}^{M} z_i &= \sum_{i \in C_W} z_i + \sum_{i \in C_U} z_i + \sum_{i \in C_N} z_i \\ &= C_W S_W + C_U S_U + C_N S_N = 1 \end{aligned}$$

To show that (14) is satisfied. We note $S_U > (1 - S_W)\frac{q}{1-p}$ implies $C_N > C_U$ and $S_W < (1 - S_U)\frac{p}{1-q}$ implies

$C_N < C_W$. this proves (20) which is equivalent to (14) Plugging these minimizing values into (9) yields

$$\begin{aligned} d_Y(\vec{y}; T) &= \min_{z \in T} \sum_{i=1}^{k} z_i f\left(\frac{y_i}{z_i}\right) \\ &= \sum_{i \in W} \frac{p y_i}{S_W} f\left(\frac{S_W}{p}\right) + \sum_{i \in U} \frac{q y_i}{S_U} f\left(\frac{S_U}{q}\right) + \sum_{i \in W} \frac{(1-p-q) y_i}{S_N} f\left(\frac{S_N}{1-p-q}\right) \\ &= p f\left(\frac{S_W}{p}\right) + q f\left(\frac{S_U}{q}\right) + (1-p-q) f\left(\frac{S_N}{1-p-q}\right) : \end{aligned}$$

This proves the closed form in equation (3) and we may now proceed to show that this function is continuous. To prove continuity we need only show continuity the piece-wise boundaries which we will evaluate one at a time.

Boundary 1: $S_W = p$. The two functions that share this boundary are 0 and $p f\left(\frac{S_W}{p}\right) + (1-p) f\left(\frac{1-S_W}{1-p}\right)$. Plugging the boundary into the latter function yields

$$p f\left(\frac{S_W}{p}\right) + (1-p) f\left(\frac{1-S_W}{1-p}\right) = p f\left(\frac{p}{p}\right) + (1-p) f\left(\frac{1-p}{1-p}\right) = 0 :$$

The two functions are equal on the boundary and the boundary is continuous.

Boundary 2: $S_U = q$. The two functions that share this boundary are 0 and $q f\left(\frac{S_U}{q}\right) + (1-q) f\left(\frac{1-S_U}{1-q}\right)$. Plugging the boundary into the latter function yields

$$q f\left(\frac{S_U}{q}\right) + (1-q) f\left(\frac{1-S_U}{1-q}\right) = q f\left(\frac{q}{q}\right) + (1-q) f\left(\frac{1-q}{1-q}\right) = 0 :$$

The two functions are equal on the boundary and the boundary is continuous.

Boundary 3: $S_U = (1-S_W)\frac{q}{1-p}$. The two functions that share this boundary are $p f\left(\frac{S_W}{p}\right) + (1-p) f\left(\frac{1-S_W}{1-p}\right)$ and $p f\left(\frac{S_W}{p}\right) + q f\left(\frac{S_U}{q}\right) + (1-p-q) f\left(\frac{S_N}{1-p-q}\right)$. Plugging the boundary into the latter function yields

$$p f\left(\frac{S_W}{p}\right) + q f\left(\frac{S_U}{q}\right) + (1-p-q) f\left(\frac{1-S_W-S_U}{1-p-q}\right) = p f\left(\frac{S_W}{p}\right) + (1-p) f\left(\frac{1-S_W}{1-p}\right) :$$

The two functions are equal on the boundary and the boundary is continuous.

Boundary 4: $S_W = (1-S_U)\frac{p}{1-q}$. The two functions that share this boundary are $q f\left(\frac{S_U}{q}\right) + (1-q) f\left(\frac{1-S_U}{1-q}\right)$ and $p f\left(\frac{S_W}{p}\right) + q f\left(\frac{S_U}{q}\right) + (1-p-q) f\left(\frac{S_N}{1-p-q}\right)$. Plugging the boundary into the latter function yields

$$q f\left(\frac{S_U}{q}\right) + p f\left(\frac{S_W}{p}\right) + (1-p-q) f\left(\frac{1-S_U-S_W}{1-p-q}\right) = q f\left(\frac{S_U}{q}\right) + (1-q) f\left(\frac{1-S_U}{1-q}\right) :$$

The two functions are equal on the boundary and the boundary is continuous. We have now shown continuity on all boundaries and the function is continuous. Now, to show that the derivative of the function is continuous, we need only show the all partial derivatives exist and agree on the boundaries. We use the closed form equation found in the body of the paper (which is equivalent to the one found in the beginning of the proof) but suppress S_N. This makes it easier to differentiate with respect to $y_i, i \in W \cup U$.

$$d_Y(\vec{y}; T) = \begin{cases} 0 & \text{if } S_W \le p \text{ and } S_U \le q \\ p f\left(\frac{S_W}{p}\right) + (1-p) f\left(\frac{1-S_W}{1-p}\right) & \text{if } S_W < p \text{ and } S_U \le (1-S_W)\frac{q}{1-p} \\ q f\left(\frac{S_U}{q}\right) + (1-q) f\left(\frac{1-S_U}{1-q}\right) & \text{if } S_U > q \text{ and } S_W \le (1-S_U)\frac{p}{1-q} \\ p f\left(\frac{S_W}{p}\right) + q f\left(\frac{S_U}{q}\right) + (1-p-q) f\left(\frac{1-S_W-S_U}{1-p-q}\right) & \text{if } S_U > (1-S_W)\frac{q}{1-p} \\ & \text{and } S_W < (1-S_U)\frac{p}{1-q} \end{cases}$$

We now take the derivative with respect to a desirable class, $\frac{\partial}{\partial s_{1|W}} d_Y(\hat{y}; T) =$

$$\begin{cases} 0 & \text{if } S_W > p \text{ and } S_U < q \\ f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W}{1 - p}\right) & \text{if } S_W < p \text{ and } S_U < (1 - S_W)\frac{q}{1-p} \\ 0 & \text{if } S_U > q \text{ and } S_W > (1 - S_U)\frac{p}{1-q} \\ f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W - S_U}{1 - p - q}\right) & \text{if } S_U > (1 - S_W)\frac{q}{1-p} \\ & \text{and } S_W < (1 - S_U)\frac{p}{1-q} \end{cases}$$

Now we need only ensure all pieces agree on the boundaries to show that the derivative exists and is continuous.

Boundary 1: $S_W = p$. The two functions that share this boundary are 0 and $f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W}{1 - p}\right)$. Plugging the boundary into the latter function yields

$$f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W}{1 - p}\right) = f'\left(\frac{p}{p}\right) - f'\left(\frac{1 - p}{1 - p}\right) = 0.$$

Then setting the derivative at the boundary to 0 makes the derivative on this boundary continuous.

Boundary 2: $S_U = q$. The two functions that share this boundary are 0, and setting the derivative at the boundary to 0 makes the derivative on this boundary continuous.

Boundary 3: $S_U = (1 - S_W)\frac{q}{1-p}$. The two functions that share this boundary are $f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W}{1 - p}\right)$ and $f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W - S_U}{1 - p - q}\right)$. Plugging the boundary into the latter function yields

$$f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W - S_U}{1 - p - q}\right) = f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W}{1 - p}\right).$$

Then setting the derivative at the boundary to $f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W}{1 - p}\right)$ makes the derivative on this boundary continuous.

Boundary 4: $S_W = (1 - S_U)\frac{p}{1-q}$. The two functions that share this boundary are 0 and $f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W - S_U}{1 - p - q}\right)$. We rewrite the boundary as $S_U = \frac{1-q}{p} S_W + 1$ and plug it into the latter function.

$$f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W - S_U}{1 - p - q}\right) = f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W - \frac{1-q}{p} S_W + 1}{1 - p - q}\right) = 0$$

Then setting the derivative at the boundary to 0 makes the derivative on this boundary continuous.

This yields the continuous partial derivative

$$\frac{\partial}{\partial s_{1|W}} d_Y(\hat{y}; T) = \begin{cases} 0 & \text{if } S_W \quad p \text{ and } S_U \quad q \\ f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W}{1 - p}\right) & \text{if } S_W < p \text{ and } S_U \quad (1 - S_W)\frac{q}{1-p} \\ 0 & \text{if } S_U > q \text{ and } S_W \quad (1 - S_U)\frac{p}{1-q} \\ f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1 - S_W - S_U}{1 - p - q}\right) & \text{if } S_U > (1 - S_W)\frac{q}{1-p} \\ & \text{and } S_W < (1 - S_U)\frac{p}{1-q} \end{cases} \quad (21)$$

We now take the derivative with respect to a undesirable class, $\frac{\partial}{\partial s_{1|U}} d_Y(\hat{y}; T) =$

$$\frac{@}{@x_{12U}} d_Y(x;T) = \begin{cases} 0 & \text{if } S_W > p \text{ and } S_U < q \\ 0 & \text{if } S_W < p \text{ and } S_U < (1 - S_W)\ \frac{q}{1-p} \\ f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_U}{1-q}\right) & \text{if } S_U > q \text{ and } S_W > (1 - S_U)\ \frac{p}{1-q} \\ f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_W-S_U}{1-p-q}\right) & \text{if } S_U > (1 - S_W)\ \frac{q}{1-p} \\ & \text{and} S_W < (1 - S_U)\ \frac{p}{1-q} \end{cases}$$

Now we need only ensure that there is agreement on the boundaries.

Boundary 1: $S_W = p$. The two functions that share this boundary are 0, and setting the derivative at the boundary to 0 makes the derivative on this boundary continuous.

Boundary 2: $S_U = q$. The two functions that share this boundary are 0 and $f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_U}{1-q}\right)$. Plugging the boundary into the latter function yields

$$f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_U}{1-q}\right) = f'\left(\frac{q}{q}\right) - f'\left(\frac{1-q}{1-q}\right) = 0.$$

Then setting the derivative at the boundary to 0 makes the derivative on this boundary continuous.

Boundary 3: $S_U = (1 - S_W)\ \frac{q}{1-p}$. The two functions that share this boundary are 0 and $f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_W-S_U}{1-p-q}\right)$. We rewrite the boundary as $S_W = 1 - \frac{1-p}{q} S_U$ and plug it into the latter function.

$$f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_W-S_U}{1-p-q}\right) = f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_U-1+\frac{1-p}{q}S_U}{1-p-q}\right) = 0.$$

Then setting the derivative at the boundary to 0 makes the derivative on this boundary continuous.

Boundary 4: $S_W = (1 - S_U)\ \frac{p}{1-q}$. The two functions that share this boundary are $f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_U}{1-q}\right)$ and $f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_W-S_U}{1-p-q}\right)$. Plugging the boundary into the latter function yields

$$\begin{aligned} f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_W-S_U}{1-p-q}\right) &= f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_U-(1-S_U)\ \frac{p}{1-q}}{1-p-q}\right) \\ &= f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_U}{1-q}\right) \end{aligned}$$

Then setting the derivative at the boundary to $f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_U}{1-q}\right)$ makes the derivative on this boundary continuous.

This yields the continuous partial derivative

$$\frac{@}{@x_{12U}} d_Y(x;T) = \begin{cases} 0 & \text{if } S_W \quad p \text{ and } S_U \quad q \\ 0 & \text{if } S_W < p \text{ and } S_U \quad (1 - S_W)\ \frac{q}{1-p} \\ f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_U}{1-q}\right) & \text{if } S_U > q \text{ and } S_W \quad (1 - S_U)\ \frac{p}{1-q} \\ f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_W-S_U}{1-p-q}\right) & \text{if } S_U > (1 - S_W)\ \frac{q}{1-p} \\ & \text{and} S_W < (1 - S_U)\ \frac{p}{1-q} \end{cases} : \qquad (22)$$

We now present a corollary to Theorem 2.2 that shows explicitly that d_Y decreases with added probability to the desirable classes and increases with added probability to the undesirable classes.

Corollary A.1. If T is of form (1) and f is twice differentiable, then $d_Y(\mathbf{y}, T)$ is decreasing in y_i if $i \in W$ and is increasing if $i \in U$.

To prove Corollary A.1, we need only show equation (3) is decreasing in y_i for $i \in W$ and increasing in y_i for $i \in U$, we need only prove that the partial derivative (21) is non-positive and the partial derivative (22) is non-negative. We will rely heavily on the fact that f' is increasing because f is convex.

We start with (21):

$$\frac{\partial}{\partial y_{i \in W}} d_Y(\mathbf{y}, T) = \begin{cases} 0 & \text{if } S_W \geq p \text{ and } S_U \geq q \\ f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1-S_W}{1-p}\right) & \text{if } S_W < p \text{ and } S_U \geq (1-S_W)\frac{q}{1-p} \\ 0 & \text{if } S_U > q \text{ and } S_W \geq (1-S_U)\frac{p}{1-q} \\ f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1-S_W-S_U}{1-p-q}\right) & \text{if } S_U > (1-S_W)\frac{q}{1-p} \\ & \text{and } S_W < (1-S_U)\frac{p}{1-q} \end{cases}.$$

Clearly the first and third cases are non-positive, so we proceed to the second case.

Because $S_W < p$, we have $\frac{S_W}{p} < 1 < \frac{1-S_W}{1-p}$ and

$$\begin{aligned} f'\left(\frac{S_W}{p}\right) &< f'\left(\frac{1-S_W}{1-p}\right) \\ f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1-S_W}{1-p}\right) &< 0. \end{aligned}$$

Next we prove the partial derivative is negative in the fourth case.

$$\begin{aligned} S_W &< (1-S_U)\frac{p}{1-q} \\ S_W - qS_W &< p - pS_U \\ S_W - qS_W - pS_W &< p - pS_U - pS_W \\ \frac{S_W}{p} &< \frac{1-S_U-S_W}{1-p-q} \\ f'\left(\frac{S_W}{p}\right) &< f'\left(\frac{1-S_U-S_W}{1-p-q}\right) \\ f'\left(\frac{S_W}{p}\right) - f'\left(\frac{1-S_U-S_W}{1-p-q}\right) &< 0 \end{aligned}$$

This shows that (21) is non-positive and (3) is decreasing in y_i for $i \in W$.

We now consider (22):

$$\frac{\partial}{\partial y_{i \in U}} d_Y(\mathbf{y}, T) = \begin{cases} 0 & \text{if } S_W \geq p \text{ and } S_U \geq q \\ 0 & \text{if } S_W < p \text{ and } S_U \geq (1-S_W)\frac{q}{1-p} \\ f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_U}{1-q}\right) & \text{if } S_U > q \text{ and } S_W \geq (1-S_U)\frac{p}{1-q} \\ f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1-S_W-S_U}{1-p-q}\right) & \text{if } S_U > (1-S_W)\frac{q}{1-p} \\ & \text{and } S_W < (1-S_U)\frac{p}{1-q} \end{cases}.$$

Clearly the first two cases are non-negative, so we consider the third case.

Because $S_U > q$, we have $\frac{S_U}{q} > 1 > \frac{1 - S_U}{1 - q}$ and

$$f'\left(\frac{S_U}{q}\right) > f'\left(\frac{1 - S_U}{1 - q}\right)$$
$$f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1 - S_U}{1 - q}\right) > 0.$$

We can no prove the fourth case is positive.

$$S_U > (1 - S_W)\frac{q}{1 - p}$$
$$S_U - pS_W > q - qS_W$$
$$S_U - pS_W - qS_U > q - pS_W - qS_U$$
$$\frac{S_U}{q} > \frac{1 - S_W - S_U}{1 - p - q}$$
$$f'\left(\frac{S_U}{q}\right) > f'\left(\frac{1 - S_W - S_U}{1 - p - q}\right)$$
$$f'\left(\frac{S_U}{q}\right) - f'\left(\frac{1 - S_W - S_U}{1 - p - q}\right) > 0$$

This shows that (22) is non-negative and (3) is increasing for $i \in U$.

Additional Analysis on d_Y The following lemma generalizes the result of Corollary A.1 to any f-divergence if we are restricted to the binary classification case. That is, exhibits expected behavior of a reward measure on any f-divergence if we restrict ourselves to the binary classification setting (reward goes down as the probability of being the undesirable class goes up).

Lemma A.2. In the binary classification setting, if $T = \{z \in Y \mid z_1 \geq p\}$, then $d_Y(y, T)$ is decreasing (not necessarily strictly) in y_1 for $D(y \| z)$ any f-divergence.

We now present the proof of Lemma A.2. Recall $d_Y(y, T) = \min_{z \in T} D_f(y \| z)$. For binary probability distributions a and b, the f-divergence has the simple form

$$D_f(b \| a) = a_1 f\left(\frac{b_1}{a_1}\right) + (1 - a_1) f\left(\frac{1 - b_1}{1 - a_1}\right) \tag{23}$$

for a convex function f with $f(1) = 0$. We show a relationship between this formula and a secant line. To refer to the secant line of a function $g(x)$ from point $x = \alpha$ to $x = \beta$ evaluated at γ, we will use the notation $S_g(\alpha; \beta; \gamma)$. When using this notation we will assume that $\alpha \leq \beta$.

We assume $a_1 > b_1$ and show that $D_f(b \| a)$ is equivalent to the secant line of $f(x)$ from $x = \frac{b_1}{a_1}$ to $\frac{1 - b_1}{1 - a_1}$ evaluated at 1. (Note $\frac{b_1}{a_1} < 1 < \frac{1 - b_1}{1 - a_1}$.) We show this simply using the point slope form.

$$S_f\left(\frac{b_1}{a_1}; \frac{1 - b_1}{1 - a_1}; x\right) = x - \frac{1 - b_1}{1 - a_1} \frac{f\left(\frac{1 - b_1}{1 - a_1}\right) - f\left(\frac{b_1}{a_1}\right)}{\frac{1 - b_1}{1 - a_1} - \frac{b_1}{a_1}} + f\left(\frac{1 - b_1}{1 - a_1}\right)$$
$$S_f\left(\frac{b_1}{a_1}; \frac{1 - b_1}{1 - a_1}; 1\right) = 1 - \frac{1 - b_1}{1 - a_1} \frac{f\left(\frac{1 - b_1}{1 - a_1}\right) - f\left(\frac{b_1}{a_1}\right)}{\frac{1 - b_1}{1 - a_1} - \frac{b_1}{a_1}} + f\left(\frac{1 - b_1}{1 - a_1}\right)$$
$$= a_1 f\left(\frac{b_1}{a_1}\right) + (1 - a_1) f\left(\frac{1 - b_1}{1 - a_1}\right)$$
$$= D_f(b \| a)$$

Now that $D_f(b \| a)$ is related to a secant line we prove a few facts about secant lines of convex functions. If g is convex, then $S_g(\cdot\,;\,\cdot\,;\,\cdot)$ is decreasing in $\cdot$ and increasing in $\cdot$ whenever $< \cdot <$. Recall that if g is convex, then by definition for any $v_1 < v_2 < v_3$, we have

$$\frac{g(v_2) - g(v_1)}{v_2 - v_1} \quad \frac{g(v_3) - g(v_1)}{v_3 - v_1} \quad \frac{g(v_3) - g(v_2)}{v_3 - v_2}. \tag{24}$$

Then for any $< \tilde{}$ we have

$$S_g(\cdot\,;\,\cdot\,;\,) = (\quad)m + g(\quad) \tag{25}$$

$$S_g(\cdot\,;\,\tilde{\cdot}\,;\,) = (\quad)\tilde{m} + g(\quad) \tag{26}$$

for $\tilde{m}$ m. It follows that for any

$$S_g(\cdot\,;\,\cdot\,;x) \quad S_g(\cdot\,;\,\tilde{\cdot}\,;x); \tag{27}$$

and $S_g(\cdot\,;\,\cdot\,;x)$ is increasing in $\cdot$.

A similar argument shows that $S_g(\cdot\,;\,\cdot\,;x)$ is decreasing in $\cdot$ when .

We will use these facts to analyze $d_Y(y\text{-};T) = \min_{z2T} D_f(y\text{-}\|z)$. The f-divergence between identical distributions is zero, so we have $d_Y(y\text{-};T) = 0$ whenever $y\text{-}_1$ p. When $y\text{-}_1 < p$ we have $\frac{y\text{-}_1}{z_1} < 1 < \frac{1\ y\text{-}_1}{1\ z_1}$ and

$$\begin{aligned} d_Y(y\text{-};T) &= \min_{z2T} D_f(y\text{-}\|z) \\ &= \min_{z2T} S_f\left(\frac{y\text{-}_1}{z_1}; \frac{1\ y\text{-}_1}{1\ z_1}; 1\right); \end{aligned}$$

which is decreasing in $\frac{y\text{-}_1}{z_1}$ and increasing in $\frac{1\ y\text{-}_1}{1\ z_1}$, so to achieve the minimum we use the smallest possible z_1, i.e. $z_1 = p$. We may now simplify

$$d_Y(y\text{-};T) = \begin{cases} S_f\left(\frac{y\text{-}_1}{p}; \frac{1\ y\text{-}_1}{1\ p}; 1\right) & \text{if } y\text{-} < p \\ 0 & \text{if } y\text{-} \quad p \end{cases}.$$

Note that this is continuous at $y\text{-} = p$ because $S_f(1;1;1) = f(1) = 0$. With this closed form solution for $d_Y(y\text{-};T)$ we may finish the proof.

We have already shown that $S_f\left(\frac{y\text{-}_1}{p}; \frac{1\ y\text{-}_1}{1\ p}; 1\right)$ is decreasing in $\frac{y\text{-}_1}{p}$ and increasing in $\frac{1\ y\text{-}_1}{1\ p}$, so increasing $y\text{-}_1$ will decrease $S_f\left(\frac{y\text{-}_1}{p}; \frac{1\ y\text{-}_1}{1\ p}; 1\right)$ and $d_Y(y\text{-};T)$ is decreasing in $y\text{-}_1$.

A.2. Proofs of Theorem 2.3 (PAC generalization bounds for Veri er)

Let us de neD_i as the distribution of the datax conditioned on the event that it is drawn from classi. We de ne a loss function` : f 0; 1g [0; 1] ! R as follows:

$$`(z; v) = zl(v) + (1 \quad z)l(1 \quad v); \tag{28}$$

wherel is some differentiable function (e.g.,log() which would lead to the cross-entropy loss). Furthermore, we assume that the output of the loss` is upper bounded by a constantB and is Lipschitz. The veri er output isV (x; x) estimates probability thatx andx belong to the same class.

Using this loss, we now de ne the true risk(V) of a veri er V as

$$R(V) = \underbrace{\frac{1}{k(k \quad 1)} \sum_{i \in j} \mathop{E}_{x^{(i)} D_i; x^{(j)} D_j} [`(0; V(x^{(i)}; x^{(j)}))]}_{R^{(\text{diff})}(V)} + \underbrace{\frac{1}{k} \sum_{i=1}^{k} \mathop{E}_{(x;x) D_i} [`(1; V(x; x))]}_{R^{(\text{same})}(V)} : \tag{29}$$

The veri er faces two types of inputs that it should be able to distinguish: (a) pairs of inputs that can come from the same class (i.e.,x; x D_i for some class) and (b) pairs of inputs that can belong to different classes (i.e., x^(i) D_i and x^(j) D_j for some pair of classes). This formulation of risk assigns equal value to identifying pairs from the same class and pairs form different classes because both of R^(diff)(V) (accuracy on pairs form different classes) and R^(same)(V) (accuracy on pairs form the same class) are normalized by dividing by the total number of terms in the sum. Speci cally, we normalize the total risk for misclassifying pairs from different classes by(k 1), which is the number of distinct ordered pairs of classes we can form out ofk classes. Similarly, we normalize the total risk of misclassifying pairs from same classes by k. Furthermore, both R^(diff)(V) and R^(same)(V) assign equal importance to each possible type of class combination (which class the rst element of the pair comes form and which class the second element of the pair comes from).

To calculate our empirical risk we will assume we are given k setsS^(i) 1 i k, each containingn=k samples drawn independently from the corresponding D_i as de ned above. We index these sets as follows:

$$S^{(i)} = \{x^{(i)}_{(q)}\}_{q=1}^{n=k}; \quad i = 1; 2; : : : ; k: \tag{30}$$

We de ne the entire dataset S as

$$S = \bigcup_{i=1}^{k} S^{(i)} \tag{31}$$

We de ne our empirical risk for training the veri er over the set S as follows:

$$\hat{R}_S(V) = \underbrace{\frac{1}{k(k \quad 1)} \sum_{i \in j} \frac{1}{\frac{n}{k}^2} \sum_{q=1}^{\frac{n}{k}} \sum_{r=1}^{\frac{n}{k}} `(0; V(x^{(i)}_q; x^{(j)}_r))}_{\hat{R}_S^{(\text{diff})}(V)} + \underbrace{\frac{1}{k} \sum_{i=1}^{k} \frac{1}{\frac{n}{k}^2 n} \sum_{q=1}^{\frac{n}{k}} \sum_{r=1; r \in q}^{\frac{n}{k}} `(1; V(x^{(i)}_q; x^{(i)}_r))}_{\hat{R}_S^{(\text{same})}(V)} : \tag{32}$$

where $\hat{R}_S^{(\text{diff})}(V)$ denotes the empirical risk of the veri er on inputs from different classes, and $\hat{R}_S^{(\text{same})}(V)$ denotes the empirical risk of the veri er on inputs from the same class. It is straightforward to verify that $\hat{R}_S(V)$ is an unbiased estimator of the true riskR(V), i.e.,E($\hat{R}_S(V)$) = R(V).

Let us de ne worst case generalization gap for a given data set S as

$$(S) = \sup_{V 2V} \quad R(V) \quad \hat{R}_S(V) \ ; \tag{33}$$

whereV denotes the hypothesis class from which the veri er(V) is selected. To bound this generalization gap, we will use the notion of Rademacher complexity which measures the correlation between the function class and the random labels to upper bound the generalization gap (Mohri et al., 2018). The Rademacher complexity of a hypothesis class over a particular data set is formally de ned as:

De nition A.3. The empirical Rademacher complexity of a function class F with respect to the set $S = \{a_i\}_{i=1}^n$ is given by the following equation:

$$R_S(F) = \frac{1}{n} E_{\sigma}\left[\sup_{f \in F} \sum_{i=1}^{n} \sigma_i f(a_i)\right]; \tag{34}$$

where σ_i's are i.i.d. Rademacher random variables, i.e., $\Pr(\sigma_i = 1) = \Pr(\sigma_i = -1) = \frac{1}{2}$.

In the following steps, we upper bound the generalization gap (33) as using Rademacher complexity. We first bound the generalization gap using triangle inequality as follows:

$$\Phi(S) = \sup_{V \in \mathcal{V}} R^{(\text{diff})}(V) + R^{(\text{same})}(V) - \hat{R}_S^{(\text{diff})}(V) - \hat{R}_S^{(\text{same})}(V) \tag{35}$$

$$\sup_{V \in \mathcal{V}} R^{(\text{diff})}(V) - \hat{R}_S^{(\text{diff})}(V) + \sup_{V \in \mathcal{V}} R^{(\text{same})}(V) - \hat{R}_S^{(\text{same})}(V) \tag{36}$$

The above bound first decomposes the generalization gap into the sum of two generalization gaps, the first over the pair of samples coming from different classes; and the second over the samples drawn from the same class. To proceed we will need a few additional definitions: we define $\mathcal{D}_i \otimes \mathcal{D}_j$ to represent the distribution over pairs $(x^{(i)}, x^{(j)})$ where $x^{(i)}$ is drawn from $\mathcal{D}_i$ and $x^{(j)}$ is drawn from $\mathcal{D}_j$ independently. We also define the sets

$$S^{(i)} \otimes S^{(j)} = \begin{cases} \{f(x_{(q)}^{(i)}, x_{(r)}^{(j)})\}_{1 \le q,r \le k} & i \neq j \\ \{f(x_{(q)}^{(i)}, x_{(r)}^{(j)})\}_{1 \le q,r \le k; q \neq r} & i = j \end{cases}; \tag{37}$$

and enumerate the elements of each set by $S^{(i)} \otimes S^{(j)} = \{u_q^{i,j}\}_{q=1}^{(n/k)^2}$ when $i \neq j$. When $i = j$ the enumeration takes the form $S^{(i)} \otimes S^{(i)} = \{u_q^{i,i}\}_{q=1}^{(n/k)^2 - n}$.

Using our definitions of true and empirical risk, we can now upper bound the above sum as follows,

$$\begin{aligned}
\Phi(S) \le & \sup_{V \in \mathcal{V}} \frac{1}{k(k-1)} \sum_{i \neq j} \left[E_{x^{(i)} \sim \mathcal{D}_i, x^{(j)} \sim \mathcal{D}_j} \left[\ell(V(x^{(i)}, x^{(j)}), 0)\right] - \frac{k}{n}^2 \sum_{q=1}^{n/k} \sum_{r=1}^{n/k} \ell(V(x_{(q)}^{(i)}, x_{(r)}^{(j)}), 0) \right] \\
& + \sup_{V \in \mathcal{V}} \frac{1}{k} \sum_{k=1} \left[E_{x,x' \sim \mathcal{D}_i} [\ell(V(x,x'), 1)] - \sum_{q=1}^{n/k} \sum_{r=1, r \neq q}^{n/k} \ell(V(x_{(q)}^{(i)}, x_{(r)}^{(i)}), 1) \right] \\
& \frac{1}{k(k-1)} \sum_{i \neq j} \sup_{V \in \mathcal{V}} E_{u \sim \mathcal{D}_i \otimes \mathcal{D}_j} \ell(V(u), 1) - \frac{k}{n}^2 \sum_{q=1}^{(n/k)^2} \ell(V(u_q^{(i,j)}), 1) \\
& + \frac{1}{k} \sum_{i=1}^{k} \sup_{V \in \mathcal{V}} E_{u \sim \mathcal{D}_i \otimes \mathcal{D}_i} [\ell(V(u), 1)] - \frac{k}{n}^2 \sum_{q=1}^{(n/k)^2 - n} \ell(V(u_q^{(i,i)}), 1)
\end{aligned} \tag{38}$$

where the second inequality follows by bounding the absolute value of a sum by the sum of the absolute values (across both the “diff” and “same” terms).

We now apply the standard Rademacher complexity PAC-bound (Mohri et al., 2018; Bartlett & Mendelson, 2002) to each of the supremums in (38). This implies that the following inequalities hold with probability $1 - \delta$ for any $\delta \in (0, 1)$. (The probability in this case stems from he random selection/draw of S.)

$$\begin{aligned}
\Phi(S) \le & \frac{1}{k(k-1)} \sum_{i \neq j} \left[2R_{S^{(i)} \otimes S^{(j)}}(V) + \frac{6kB\ell}{n} \sqrt{\frac{\log(2/\delta)}{2}} \right] + \frac{1}{k} \sum_{i=1}^{k} \left[2R_{S^{(i)} \otimes S^{(i)}}(V) + \sqrt{\frac{6kB\ell}{n^2 - k^2 n}} \sqrt{\frac{\log(2/\delta)}{2}} \right] \\
= & \frac{2}{k(k-1)} \sum_{i \neq j} R_{S^{(i)} \otimes S^{(j)}}(V) + \frac{2}{k} \sum_{i=1}^{k} R_{S^{(i)} \otimes S^{(i)}}(V) + \frac{6kB\ell}{n} \sqrt{\frac{\log(2/\delta)}{2}} + \sqrt{\frac{6kB\ell}{n^2 - k^2 n}} \sqrt{\frac{\log(2/\delta)}{2}} \\
& \frac{2}{k(k-1)} \sum_{i \neq j} R_{S^{(i)} \otimes S^{(j)}}(V) + \frac{2}{k} \sum_{i=1}^{k} R_{S^{(i)} \otimes S^{(i)}}(V) + \sqrt{\frac{12kB\ell}{n^2 - k^2 n}} \sqrt{\frac{\log(2/\delta)}{2}}
\end{aligned} \tag{39}$$

where the final inequality comes form replacing $\frac{6kB\sqrt{}}{n}$ with the larger $p\frac{6kB\sqrt{}}{n^2 \quad k^2n}$. Equation (39) can be interpreted as the sum of three terms: the first term is the average Rademacher complexity over the datasets corresponding to pairs which are drawn from different classes; the second term is the average Rademacher complexity over the datasets corresponding to pairs which are drawn from same classes; the third term is a standard term which shows the dependence as well as $n; k$).

We now apply the bound on empirical Rademacher complexity

$$R_Q(V) = O \quad jQj \quad ^{1=d'} \tag{40}$$

with d' the dimension of the elements of Q (Gottlieb et al., 2016). To apply this we will recall the dimension of the elements of $S^{(i)} \quad S^{(j)}$ is $2d$, and $jS^{(i)} \quad S^{(j)}j = \frac{n}{k}^2$ when $i \neq j$, and $jS^{(i)} \quad S^{(i)}j = \frac{n}{k}^2 \quad n = \frac{n^2 \quad k^2n}{k^2}$. Applying our Rademacher complexity bound yields

$$(S) \quad \frac{2}{k(k \quad 1)} \sum_{i \neq j} O \quad \frac{k}{n}^{1=d'} + \frac{2}{k} \sum_{i=1}^{k} O \quad p\frac{k}{n^2 \quad k^2n}^{1=d'} + p\frac{12kB\sqrt{}}{n^2 \quad k^2n} \sqrt{\frac{\log(2=\ )}{2}} \tag{41}$$

$$=2\,O \quad \frac{k}{n}^{1=d'} + 2\,O \quad p\frac{k}{n^2 \quad k^2n}^{1=d'} + p\frac{12kB\sqrt{}}{n^2 \quad k^2n} \sqrt{\frac{\log(2=\ )}{2}} \tag{42}$$

$$4O \quad p\frac{k}{n^2 \quad k^2n}^{1=d'} + p\frac{12kB\sqrt{}}{n^2 \quad k^2n} \sqrt{\frac{\log(2=\ )}{2}}: \tag{43}$$

The bound in (43) is our final PAC bound true with probability 1 . However, we expect the containing term to be dominated by the other term because $(\text{since } \frac{k}{n^2 \quad k^2n}) < 1$ and d is expected to be much larger than 1.

A.3. Additional Implementation Details

In this section we give additional details on how we implemented our methods to create the experimental results found in this paper. We also provide code for replicating our results at https://github.com/JesseFriedbaum/TAP_code .

A.3.1. Data Set and Cost Function Details

Here we give additional description of each data set and the corresponding the cost function d_X used in our experiments. As noted in Section 3 we must ensure d_X is differentiable. When dealing with categorical features costs are by nature discrete (and not differentiable). We show how we were able to write these costs in a differentiable form. Suppose $z \in \mathbb{R}^\ell$ is a one-hot encoding of a categorical feature and define the transition cost matrix A such that $A_{i,j}$ as the cost of changing from category i to category j. Then $z^T Az$ represents the costs of changing this categorical feature and is differentiable in z.

Adult Income Prediction Dataset (Kohavi & Becker, 1996) This widely used data set contains information from the 1994 U.S. census, with individuals labelled by whether their annual income was over $50,000 ($100,000 in 2023 adjusted for in ation). We define our target set T as over 80% probability high income. Our actionable set allows changes in job type, education and number of hours worked with all other attributes immutable. The cost function d_X includes the expected number of years to improve education (e.g. two years to go from associate's degree to bachelors degree), a one-year cost to change employer type and the 2-norm of the change in hours worked per week (weighted so 3 hours per week is equivalent to a year spent on education). Here Trustworthy Actionable Perturbations suggest the best way to improve an individuals odds of making a large income with the least time and effort.

Specifically d_X is the sum cost from changes (1) hours worked per week (2) change in employment type (3) change in education and (4) change in field of work.

The cost from a change in hours is given by $\frac{h^2}{10}$ where h is the change in weekly hours worked. This will mean 3 extra hours of work are approximately equivalent to one year of schooling.

The cost from a change in employer (the options are government, private, self-employed and other) is always 1 (always equivalent to a year spent on education).

The possible levels of education are (1) any schooling, (2) High School Degree, (3) Professional Degree, (4) some college, (5) Associate's Degree, (6) Bachelors Degree, (7) Master's Degree, (8) Doctorate Degree. The cost transition matrix associated

with the level of education (as ordered above) is

$$A_{\text{Education}} = \begin{bmatrix} 0 & 2 & 10 & 3 & 4 & 6 & 8 & 11 \\ L & 0 & 8 & 1 & 2 & 4 & 6 & 9 \\ L & L & 0 & L & L & L & 2 & 5 \\ L & L & 7 & 0 & 1 & 3 & 5 & 8 \\ L & L & 6 & L & 0 & 2 & 4 & 7 \\ L & L & 4 & L & L & 0 & 2 & 5 \\ L & L & 4 & L & L & L & 0 & 3 \\ L & L & 4 & L & L & L & L & 0 \end{bmatrix}; \tag{44}$$

where L is a large number meant to prevent suggestions that lead to a decrease in education, which is impossible (we use $L = 1,000$). These numbers represent the expected number of years required to gain the specified degree (i.e. the cost of going from a high school degree to a bachelors degree is $A_{2,6} = 4$).

Finally the options for fields of work are (1) Service, (2) Sales, (3) Blue-Collar (4) White Collar, (5) Professional, (6) Other. The cost transition matrix associated with the field of work (as ordered above) is

$$A_{\text{Profession}} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 1 \\ 1 & 0 & 1 & 2 & 3 & 1 \\ 1 & 1 & 0 & 1 & 2 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{45}$$

This represents a minimum cost of 1 for any change in field of work with higher costs when moving to a relatively more selective field (i.e., service to professional).

Law School Success Prediction Dataset (Wightman, 1998) This data set contains demographic information and academic records for over 20,000 law school students labelled by whether or not a student passed the BAR exam. Our target set is an 85% chance of passing the BAR. To create $A(x)$, we suppose the law school performance is merely a projection that can be changed through more studying, allowing us to change the law school grades and the location where the students take the BAR. The cost function d_X sums the increase in grades and the physical distance travelled to take the BAR where moving to an adjacent region (e.g. Far West to North West) is weighted the same as increasing grades by one standard deviation.

Specifically d_Y sums the increase in grades and the physical distance travelled to take the BAR where moving to an adjacent region (e.g. Far West to North West) is weighted the same as increasing grades one standard deviation. This set up returns the optimal combination of studying harder and moving location to take the BAR. In this data set d_X is sum of the change in grades (in standard deviations from the mean) and distance traveled. The country was divided into eight regions: (1) Far West, (2) Great Lakes, (3) Mid-South, (4) Mountain West, (5) Mid-West, (6) North East, (7) New England, (8) North West. We use the transition cost matrix

$$A_{\text{Region}} = \begin{bmatrix} 0 & 3 & 4 & 1 & 2 & 6 & 5 & 1 \\ 3 & 0 & 1 & 2 & 1 & 2 & 1 & 3 \\ 4 & 1 & 0 & 2 & 1 & 2 & 1 & 5 \\ 1 & 2 & 2 & 0 & 1 & 4 & 3 & 2 \\ 2 & 1 & 1 & 1 & 0 & 3 & 2 & 3 \\ 6 & 2 & 2 & 4 & 3 & 0 & 1 & 5 \\ 5 & 1 & 1 & 3 & 2 & 1 & 0 & 5 \\ 1 & 3 & 5 & 2 & 3 & 5 & 5 & 0 \end{bmatrix} \tag{46}$$

Moves to adjacent regions result in a cost of 1, while the highest cost of 6 is incurred by moving from Far West to New England or back.

Diabetes Prediction Dataset (for Disease Control & , CDC) This data set contains information on the demographics, health conditions and health habits of 250,000 individuals labelled by whether an individual is diabetic extracted from the Behavioral Risk Factor Surveillance System (BRFSS), a health-related telephone survey that is collected annually by the CDC.. We define $A(x)$ to allow changes in health habits, BMI, education and income. We use a weighted 2-norm for d_X to represent the relative difficulty of making changes. For example, starting to get regular physical activity is weighted the

same as dropping one BMI point. Increasing education, income and health insurance were weighted as more difficult that simply adjusting health habits.

German Credit Dataset (Hofmann, 1994) This commonly used data set contains information on 1,000 loan applications in Germany labelled by their credit risk. The actionable set allows for changes in the loan request (time and size) as well as the funds in the applicants checking and savings account and whither the applicant has a telephone. The target is greater than 80% of being a good credit risk. The cost function is the direct measuring the total difference in Deutsche Marks (DM) between all elements of the application. No cost was assigned to closing empty accounts. The change in length of loan is converted to DM through the individual's monthly disposable income. Finally we set a at cost of 50DM to acquire a telephone.

A.3.2. Model Details

We used fully connected feed forward neural networks. Each network used 3 hidden layers with ReLu activation functions between each layer. We tuned the parameters of the neural networks until we achieved accuracy on par with common tree based classifiers (random forests and histogram boosted trees). Accuracy results are presented in table A.3.2. For all data sets except the German Credit data set each hidden layer had 60 nodes. The German Credit data set required 120 nodes per layer. Additionally, for the German Credit data set only, we used dropout regularization with 20% on each hidden layer. We trained these models using the ADAM optimizer to minimize cross entropy loss. We used an 80 10 10 train-validate-test data split and implemented early stopping with the validation data. All Trustworthy Actionable Perturbations, counterfactuals and adversarial examples were created for the testing data. We used identical architectures for , except for doubling the input size. Accuracy data may be found in table 3.

	Adult Income	Law School Success	Diabetes Prediction	German Credit
Random Forest	73%	64%	62%	74%
Histogram Gradient Boosted Trees	81%	77%	75%	69%
Neural Network	80%	77%	75%	75%

We also tested the calibration of our networks by calculating the expected calibration error (ECE) (Naeini et al., 2015). We used 15 bins and record the results in table A.3.2

	Adult Income	Law School Success	Diabetes Prediction	German Credit
ECE (15 bins)	16%	15%	7%	21%

A.3.3. Objective Function Details

In our implementation we formulated the actionablility penalty term as

$$b(x) = G \sum_{i=1}^{m} \max\{0, x_i - u_i\} + \max\{0, l_i - x_i\} \tag{47}$$

with G a sufficiently large constant. We formulated our coherence penalty term as

$$p(x) = P \sum_{i=1}^{c} \left(1 - \sum_{j \in C_i} x_j \right)^2; \tag{48}$$

with P another appropriately large constant. The conditioner function simply rounded integer and Boolean values to the nearest integer value. For one-hot encoded features categorical features, the category with the largest value set to one and all other categories set to zero.

A.3.4. Additional Experimental Results

Here we show success bar charts similar to those found in figure 7 compare the efficacy of Trustworthy Actionable Perturbations, counterfactuals (Wachter et al., 2017; Mothilal et al., 2020) and adversarial examples from the Carlini Wagner ℓ_2 attack (Carlini & Wagner, 2017b) for all data sets. These are similar to Figure 5, but include all data sets and an increased number of cost () values.

Each bar chart refers to a particular data set and desired distance to the target set $\mathcal{T}$. Inside of each chart, the bars show the percentage of individuals that a method was able to successfully move inside the goal for a variety of costs α. Figure 7 shows data before the verification procedure has been performed and 7 shows the data after all. In these tests, the Trustworthy Actionable Perturbations (in blue) outperform the counterfactuals (in green and orange) in nearly all cases except for when both methods achieved 100% success or the very high-cost (large α) high reward ($\gamma = 0$) scenarios. Carlini Wagner attacks (red) are only effective at large γ values because they are designed to move a data point just barely inside the target class. The Carlini Wagner attacks are not required to be actionable (or even feasible), so they do not constitute useful advise. The verifier is able to recognize that these adversarial examples are untrustworthy in all cases.

Figure 7. Performance comparison over entire datasets before verification: The graphs show average success rate for moving individuals within a variety of distances ϵ (to the target set. The y-axis shows the percentage of individuals within the goal distance, and the x-axis, represents different costs v (values) to achieve the goal. These values were obtained before applying the verification procedure.

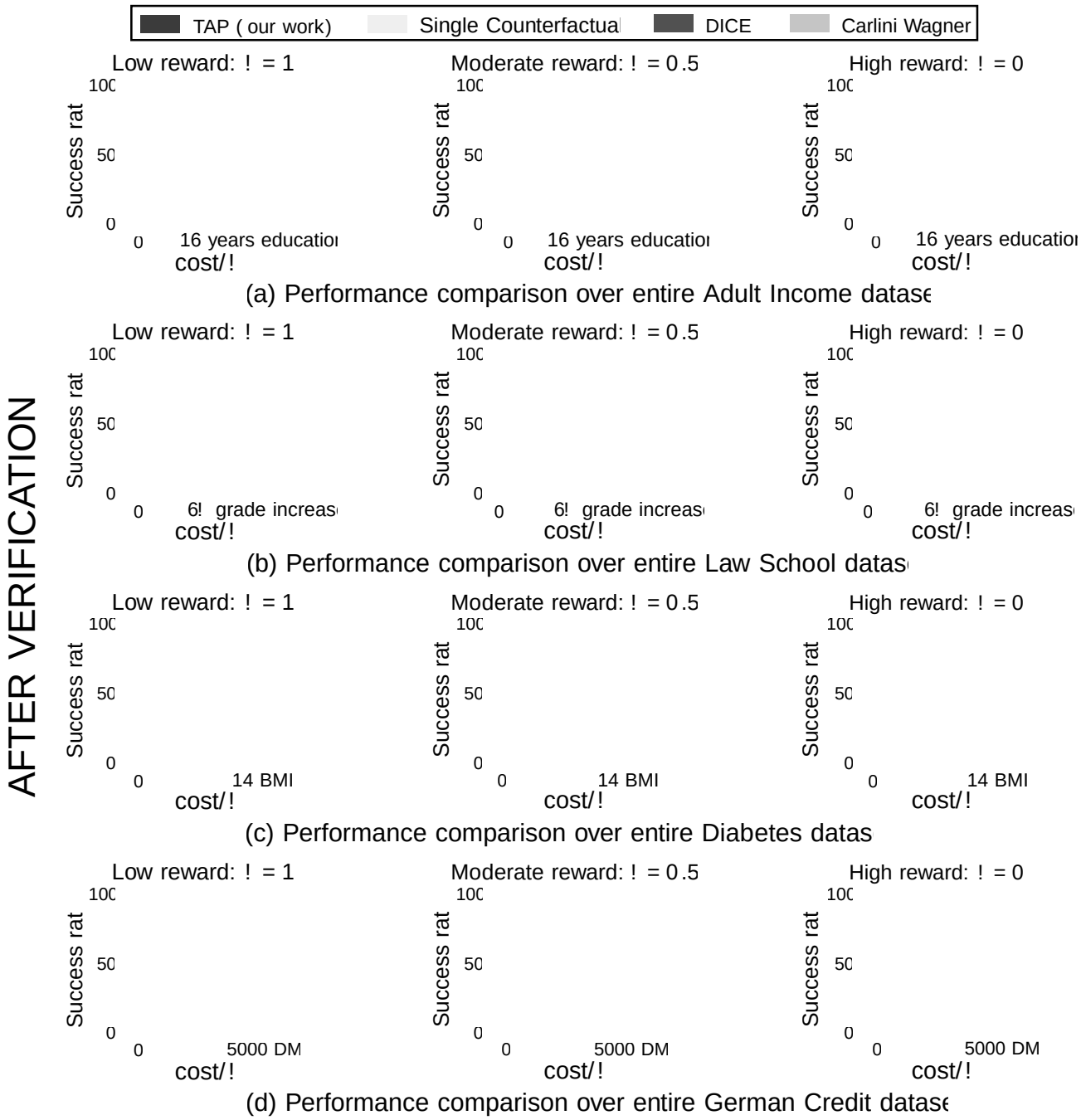

Figure 8. **Performance comparison over entire datasets after verification:** The graphs show average success rate for moving individuals within a variety of distances (δ) to the target set. The y-axis shows the percentage of individuals within the goal distance, and the x-axis, represents different costs (ϵ values) to achieve the goal. These values were obtained after applying the verification procedure with a 10% chance of eliminating valid inputs.