A PROOF OF THEOREM 4

Theorem 4. *Suppose we have a dataset of labeled examples* $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$*. Every model* $f : \mathcal{X} \to \mathcal{Y}$ *can provide recourse to* $\boldsymbol{x}$ *if:*

$$\mathsf{FNR}(f) < \frac{1}{n^+} \sum_{i=1}^{n} \mathbb{1}[\boldsymbol{x}_i \in R \ \wedge \ y_i = +1] \tag{5}$$

where $\mathsf{FNR}(f) := \frac{1}{n^+} \sum_{i=1}^{n} \mathbb{1}[f(\boldsymbol{x}_i) = -1 \ \wedge \ y_i = +1]$ *is the false negative rate of* f *on* $\mathcal{D}$ *and where* n^+ *is number of positive examples in* $\mathcal{D}$*, and* $R \subseteq R_A(\boldsymbol{x})$ *is any subset of the reachable set.*

Proof. The proof is based on an application of the pigeonhole principle over the set of positive examples $S^+ := \{\boldsymbol{x}_i \mid y_i = +1, i \in [n]\}$. Given a classifier f, denote the number of true positive and false negative predictions over S^+ as:

$$\mathsf{TP}(f) := \sum_{i=1}^{n} \mathbb{1}[f(\boldsymbol{x}_i) = +1 \wedge y_i = +1] \qquad \mathsf{FN}(f) := \sum_{i=1}^{n} \mathbb{1}[f(\boldsymbol{x}_i) = -1 \wedge y_i = +1].$$

Consider a region of feature space $R \subseteq R_A(\boldsymbol{x})$ for which the number of correct positive predictions exceeds the number of positive examples outside R so that:

$$\mathsf{TP}(f) > n^+ - |S^+ \cap R|.$$

In this case, the pigeonhole principle ensures that the classifier f must assign a correct prediction to at least one of the positive examples in R – i.e., there exists a point $\boldsymbol{x}' \in S^+ \cap R$ such that $f(\boldsymbol{x}') = y_i = +1$. Given $R \subseteq R_A(\boldsymbol{x})$, we have that $\boldsymbol{x} \in R_A(\boldsymbol{x})$. Thus, we can reach $\boldsymbol{x}'$ from $\boldsymbol{x}$ by performing the action $\boldsymbol{a} = \boldsymbol{x}' - \boldsymbol{x}$ – i.e., we can change the prediction from $f(\boldsymbol{x}) = -1$ to $f(\boldsymbol{x} + \boldsymbol{a}) = +1$.

We recover the condition in the statement of the Theorem as follows:

$$\mathsf{TP}(f) > n^+ - |S^+ \cap R| \tag{6}$$

$$\mathsf{FN}(f) < |S^+ \cap R|, \tag{7}$$

$$\mathsf{FNR}(f) < \frac{1}{n^+} \sum_{i=1}^{n} \mathbb{1}[\boldsymbol{x}_i \in R \ \wedge \ y_i = +1] \tag{8}$$

Here, we Eqn. (7) uses the fact that $\mathsf{TP}(f) = n^+ - \mathsf{FN}(f)$, and (8) divides both sides by $\frac{1}{n^+}$. The result follows by applying the definition of the false negative rate. □

B REACHABLE SET GENERATION

In this section, we describe how to formulate the optimization problems in Section 3 as mixed-integer programs. We start by presenting a MIP formulation for the optimization problem we solve in the $\mathsf{FindAction}(\boldsymbol{x}, A(\boldsymbol{x}))$ and $\mathsf{IsReachable}(\boldsymbol{x}, \boldsymbol{x}', A(\boldsymbol{x}))$ routines. Next, we describe how this formulation can be extended to the complex actionability constraints in Table 1.

B.1 MIP FORMULATIONS

Given a point $\boldsymbol{x} \in \mathcal{X}$, an action set $A(\boldsymbol{x})$, and a set of previous optima $\mathcal{A}^{\text{opt}}$, we can formulate $\mathsf{FindAction}(\boldsymbol{x}, A(\boldsymbol{x}))$ as the following mixed-integer program:

$$\min_{\boldsymbol{a}} \quad \sum_{j\in[d]} a_j^+ + a_j^-$$

$$
\begin{array}{llll}
\text{s.t.} & a_j^+ \geq a_j & j \in [d] & \textit{positive component of } a_j \quad (9a) \\
& a_j^- \geq -a_j & j \in [d] & \textit{negative component of } a_j \quad (9b) \\
& a_j = a_{j,k} + \delta_{j,k}^+ - \delta_{j,k}^- & j \in [d], \boldsymbol{a}_k \in \mathcal{A}^{\text{opt}} & \textit{distance from prior actions} \quad (9c) \\
& \varepsilon_{\text{min}} \leq \sum_{j\in[d]} (\delta_{j,k}^+ + \delta_{j,k}^-) & \boldsymbol{a}_k \in \mathcal{A}^{\text{opt}} & \textit{any solution is } \varepsilon_{min} \textit{ away from } \boldsymbol{a}_k \quad (9d) \\
& \delta_{j,k}^+ \leq M_{j,k}^+ u_{j,k} & j \in [d], \boldsymbol{a}_k \in \mathcal{A}^{\text{opt}} & \delta_{j,k}^+ > 0 \implies u_{j,k} = 1 \quad (9e) \\
& \delta_{j,k}^- \leq M_{j,k}^- (1 - u_{j,k}) & j \in [d], \boldsymbol{a}_k \in \mathcal{A}^{\text{opt}} & \delta_{j,k}^- > 0 \implies u_{j,k} = 0 \quad (9f) \\
& a_j \in A_j(\boldsymbol{x}) & j \in [d] & \textit{separable actionability constraints on } j \quad (9g) \\
& a_j^+, a_j^- \in \mathbb{R}_+ & j \in [d] & \textit{absolute value of } a_j \quad (9h) \\
& \delta_{j,k}^+, \delta_{j,k}^- \in \mathbb{R}_+ & j \in [d] & \textit{signed distances from } a_{j,k} \quad (9i) \\
& u_{j,k} \in \{0, 1\} & j \in [d] & u_{j,k} := 1[\delta_{j,k}^+ > 0] \quad (9j)
\end{array}
$$

The formulation searches for an action in the set $\boldsymbol{a} \in A(\boldsymbol{x})/\mathcal{A}^{\text{opt}}$ by combining two kinds of constraints: (i) constraints to restrict actions $\boldsymbol{a} \in A(\boldsymbol{x})$ and (ii) constraints to rule out actions in $\boldsymbol{a} \in \mathcal{A}^{\text{opt}}$.

The formulation encodes the separable constraints in $A(\boldsymbol{x})$ – i.e., a constraint that can be enforced for each feature. The formulation must be extended with additional variables and constraints to handle constraints as discussed in Appendix B.2. These constraints are handled through the $a_j \in A_j(\boldsymbol{x})$ conditions in Constraint 9g. This constraint can handle a number of actionability constraints that can be passed solver when defining the variables a_j, including *bounds* (e.g., $a_j \in [-x_j, 10 - x_j]$), *integrality* (e.g., $a_j \in \{0, 1\}$ or $a_j \in \{L - x_j, L - x_j + 1, \ldots, U - x_j\}$), and *monotonicity* (e.g., $a_j \geq 0$ or $a_j \leq 0$).

The formulation rules out actions in $\boldsymbol{a} \in \mathcal{A}^{\text{opt}}$ through the "no good" constraints in Constraints (9c) to (9f). Here, Constraint (9d) ensures feasible actions from previous solutions by at least ε_{min}. We set to a sufficiently small number $\varepsilon_{\text{min}} := 10^-6$ by default, but use larger values when working with discrete feature sets (e.g., $\varepsilon_{\text{min}} = 1$ for cases where every actionable feature is binary or integer-valued). Constraints (9e) and (9f) ensure that either $\delta_{j,k}^+ > 0$ or $\delta_{j,k}^- > 0$. These are "Big-M constraints" where the Big-M parameters can be set to represent the largest value of signed distances. Given an action $a_j \in [a_j^{\text{LB}}, a_j^{\text{UB}}]$, we can set $M_{j,k}^+ := |a_j^{\text{UB}} - a_{j,k})$ and $M_{j,k}^- := |a_{j,k} - a_j^{\text{LB}}|$.

The formulation chooses each action in $\boldsymbol{a} \in A(\boldsymbol{x})/\mathcal{A}^{\text{opt}}$ to minimize the L_1 norm. We compute the L_1-norm component-wise as $|a_j| := a_j^+ + a_j^-$ where the variables a_j^+ and a_j^- are set to the positive and negative components of $|a_j|$ in Constraints (9a) and (9b). This choice of objective is meant to induce sparsity among the actions we recover by repeatedly solving Algorithm 1.

MIP Formulation for $\mathsf{IsReachable}$ Given a point $\boldsymbol{x} \in \mathcal{X}$, an action set $A(\boldsymbol{x})$, we can formulate the optimization problem for $\mathsf{IsReachable}(\boldsymbol{x}, \boldsymbol{x}', A(\boldsymbol{x}))$ as a special case of the MIP in (9) in which we set $\mathcal{A}^{\text{opt}} = \emptyset$ and include the constraint $\boldsymbol{a} = \boldsymbol{x} - \boldsymbol{x}'$. Given that the objective function does not affect the feasibility of the optimization problem, we can set the objective to 1 when solving the problem for $\mathsf{IsReachable}$. In this case, any feasible solution would certify that $\boldsymbol{x}'$ is reachable from $\boldsymbol{x}$ using the actions in $A(\boldsymbol{x})$. Thus, we can return $\mathsf{IsReachable}(\boldsymbol{x}, \boldsymbol{x}', A(\boldsymbol{x})) = 1$ if the MIP is feasible and $\mathsf{IsReachable}(\boldsymbol{x}, \boldsymbol{x}', A(\boldsymbol{x})) = 0$ if it is infeasible.

B.2 Encoding Actionability Constraints

We describe how to extend the MIP formulation in (9) to encode salient classes of actionability constraints. Our software includes an ActionSet API that allows practitioners to specify these constraints across each MIP formulation.

Encoding Preservation for Categorical Attributes Many datasets contain subsets of features that reflect the underlying value of a categorical attribute. For example, we may encode a categorical attribute with $K = 3$ categories such `marital_status` $\in \{single, married, other\}$ using a subset of $K - 1 = 2$ dummy variables such as `married` and `single`. In such cases, actions on the dummy variables must obey non-separable actionability constraints to preserve the encoding – i.e., to ensure that a person cannot be `married` and `single` at the same time.

We can enforce these conditions by adding the following constraints to the MIP Formulation in (9):

$$L \leq \sum_{j \in \mathcal{J}} x_j + a_j \leq U \tag{10}$$

Here, $\mathcal{J} \subseteq [d]$ is the index set of features with encoding constraints, and L and U are lower and upper limits on the number of features in $\mathcal{J}$ that must hold to preserve an encoding.

Given a standard one-hot encoding of a categorical variable with K categories, $\mathcal{J}$ would contain the indices of $K - 1$ dummy variables for the $K - 1$ categories other than the reference category. We would ensure that all actions preserve this encoding by setting $L = 0$ and $U = 1$.

Implications and Deterministic Causal Effects Datasets often include features where actions on one feature will induce changes in the values and actions for other features. For example, in Table 1, changing `is_employed` from `FALSE` to `TRUE` would change the value of `work_hrs_per_week` from 0 to a value ≥ 0.

We capture these conditions by adding variables and constraints that capture logical implications in action space. In the simplest case, these constraints would relate the values for a pair of features $j, j' \in [d]$ through an if-then condition such as: "if $a_j \geq v_j$ then $a'_j = v_{j'}$". In such cases, we could capture this relationship by adding the following constraints to the MIP Formulation in (9):

$$Mu \geq a_j - v_j + \epsilon \tag{11}$$

$$M(1-u) \geq v_j - a_j \tag{12}$$

$$uv_{j'} = a_{j'} \tag{13}$$

$$u \in \{0, 1\}$$

The constraints shown above capture the "if-then" condition by introducing a binary variable $u := \mathbb{1}[a_j \geq v_j]$. The indicator is set through the Constraints (11) and (12) where $M := a_j^{\text{UB}} - v_j$ and $\epsilon = 1e-5$. If the implication is met, then $a_{j'}$ is set to $v_{j'}$ through Constraint (13). We apply this approach to encode a number of salient actionability constraints shown in Table 1 by generalizing the constraint shown above to a setting where: (i) the "if" and "then" conditions to handle subsets of features, and (ii) the implications link actions on mutable features to actions on an immutable feature (i.e. so that actions on a mutable feature `years_since_last_application` will induce changes in an immutable feature `age`).

Generalized Reachability Constraints We end with a general-purpose solution to enforce arbitrary actionability constraints on discrete features. These constraints can be used, for example, to preserve a one-hot encoding of ordinal features (e.g., `max_degree_BS` and `max_degree_MS`) or a thermometer encoding (e.g., `monthly_income_geq_2k`, `monthly_income_geq_5k`, `monthly_income_geq_10k`).

We can formulate custom reachability constraints for the relevant features $\mathcal{J} \subset [d]$ given two parameters:

1. Set of Viable Values: V, a set of all values that can be assigned to the features in J.

2. Reachability Matrix: $E \in \{0,1\}^{k \times k}$, a matrix where $e_{i,j} = \mathbb{1}[v_i \text{ is reachable from } v_j]$ for all $v_i, v_j \in V$.

Given these parameters, we constrain the reachability of features $j \in J$ by adding the following constraints to the MIP formulation in (9):

$$a_j = \sum_{k \in E[i]} e_{i,k} a_{j,k} u_{j,k} \tag{14}$$

$$1 = \sum_{k \in E[i]} u_{j,k} \tag{15}$$

$$u_{j,k} \leq e_{i,k} \tag{16}$$

$$u_{j,k} \in \{0,1\}$$

Here, $u_{j,k} := \mathbb{1}[\boldsymbol{x}' \in V]$ indicates that we choose an action to attain point $\boldsymbol{x}' \in V$. Constraint (14) defines the set of reachable points from i, while Constraint (14) ensures that only one such point can be selected. Here, $e_{i,k}$ is i^{th} row of E for point i and $a_{j,k} := x'_j - x_j$ is the action on feature j to reach point $\boldsymbol{x}' \in V$ from point $\boldsymbol{x}$.

We show an example of how to formulate reachability constraints to preserve a thermometer encoding in Fig. 4.

V			
`NetFractionRevolvingBurdenGeq90`	`NetFractionRevolvingBurdenGeq60`	`NetFractionRevolvingBurdenLeq30`	E
0	0	0	$[1,1,0,0]$
1	0	0	$[0,1,0,0]$
0	1	0	$[1,1,1,0]$
0	1	1	$[1,1,1,1]$

Figure 4: V denotes valid combinations of features. For these features, we wanted to produce actions that would reduce `NetFractionRevolvingBurden` for consumers. E shows which points can be reached. For example, $[1,1,0,0]$ represents point $[0,0,0]$ can be reached, and point $[1,0,0]$ can be reached, but no other points can be reached.

C SUPPLEMENTAL MATERIAL FOR EXPERIMENTS

C.1 ACTIONABILITY CONSTRAINTS FOR THE german DATASET

We show a list of all features and their separable actionability constraints in Table 3.

Name	Type	LB	UB	Actionability	Sign
Age	$\mathbb{Z}$	19	75	No	
Male	$\{0,1\}$	0	1	No	
Single	$\{0,1\}$	0	1	No	
ForeignWorker	$\{0,1\}$	0	1	No	
YearsAtResidence	$\mathbb{Z}$	0	7	Yes	+
LiablePersons	$\mathbb{Z}$	1	2	No	
Housing=Renter	$\{0,1\}$	0	1	No	
Housing=Owner	$\{0,1\}$	0	1	No	
Housing=Free	$\{0,1\}$	0	1	No	
Job=Unskilled	$\{0,1\}$	0	1	No	
Job=Skilled	$\{0,1\}$	0	1	No	
Job=Management	$\{0,1\}$	0	1	No	
YearsEmployed≥1	$\{0,1\}$	0	1	Yes	+
CreditAmt≥1000K	$\{0,1\}$	0	1	No	
CreditAmt≥2000K	$\{0,1\}$	0	1	No	
CreditAmt≥5000K	$\{0,1\}$	0	1	No	
CreditAmt≥10000K	$\{0,1\}$	0	1	No	
LoanDuration≤6	$\{0,1\}$	0	1	No	
LoanDuration≥12	$\{0,1\}$	0	1	No	
LoanDuration≥24	$\{0,1\}$	0	1	No	
LoanDuration≥36	$\{0,1\}$	0	1	No	
LoanRate	$\mathbb{Z}$	1	4	No	
HasGuarantor	$\{0,1\}$	0	1	Yes	+
LoanRequiredForBusiness	$\{0,1\}$	0	1	No	
LoanRequiredForEducation	$\{0,1\}$	0	1	No	
LoanRequiredForCar	$\{0,1\}$	0	1	No	
LoanRequiredForHome	$\{0,1\}$	0	1	No	
NoCreditHistory	$\{0,1\}$	0	1	No	
HistoryOfLatePayments	$\{0,1\}$	0	1	No	
HistoryOfDelinquency	$\{0,1\}$	0	1	No	
HistoryOfBankInstallments	$\{0,1\}$	0	1	Yes	+
HistoryOfStoreInstallments	$\{0,1\}$	0	1	Yes	+
CheckingAcct_exists	$\{0,1\}$	0	1	Yes	+
CheckingAcct≥0	$\{0,1\}$	0	1	Yes	+
SavingsAcct_exists	$\{0,1\}$	0	1	Yes	+
SavingsAcct≥100	$\{0,1\}$	0	1	Yes	+

Table 3: Separable actionability constraints for the german dataset.

The non-separable actionability constraints for this dataset include:

1. DirectionalLinkage: Actions on YearsAtResidence will induce to actions on ['Age']. Each unit change in YearsAtResidence leads to:1.00-unit change in Age
2. DirectionalLinkage: Actions on YearsEmployed≥1 will induce to actions on ['Age']. Each unit change in YearsEmployed≥1 leads to:1.00-unit change in Age
3. ThermometerEncoding: Actions on [CheckingAcctexists, CheckingAcct≥0] must preserve thermometer encoding of CheckingAcct., which can only increase. Actions can only turn on higher-level dummies that are off, where CheckingAcctexists is the lowest-level dummy and CheckingAcct≥0 is the highest-level-dummy.
4. ThermometerEncoding: Actions on [SavingsAcctexists, SavingsAcct≥100] must preserve thermometer encoding of SavingsAcct., which can only increase. Actions can only turn on

higher-level dummies that are off, where SavingsAcctexists is the lowest-level dummy and SavingsAcct≥100 is the highest-level-dummy.

C.2 Actionability Constraints for the heloc Dataset

We show a list of all features and their separable actionability constraints in Table 4.

Name	Type	LB	UB	Actionability	Sign
ExternalRiskEstimate≥40	$\{0,1\}$	0	1	No	
ExternalRiskEstimate≥50	$\{0,1\}$	0	1	No	
ExternalRiskEstimate≥60	$\{0,1\}$	0	1	No	
ExternalRiskEstimate≥70	$\{0,1\}$	0	1	No	
ExternalRiskEstimate≥80	$\{0,1\}$	0	1	No	
YearsOfAccountHistory	$\mathbb{Z}$	0	50	No	
AvgYearsInFile≥3	$\{0,1\}$	0	1	Yes	
AvgYearsInFile≥5	$\{0,1\}$	0	1	Yes	
AvgYearsInFile≥7	$\{0,1\}$	0	1	Yes	
MostRecentTradeWithinLastYear	$\{0,1\}$	0	1	Yes	
MostRecentTradeWithinLast2Years	$\{0,1\}$	0	1	Yes	
AnyDerogatoryComment	$\{0,1\}$	0	1	No	
AnyTrade120DaysDelq	$\{0,1\}$	0	1	No	
AnyTrade90DaysDelq	$\{0,1\}$	0	1	No	
AnyTrade60DaysDelq	$\{0,1\}$	0	1	No	
AnyTrade30DaysDelq	$\{0,1\}$	0	1	No	
NoDelqEver	$\{0,1\}$	0	1	No	
YearsSinceLastDelqTrade≤1	$\{0,1\}$	0	1	Yes	
YearsSinceLastDelqTrade≤3	$\{0,1\}$	0	1	Yes	
YearsSinceLastDelqTrade≤5	$\{0,1\}$	0	1	Yes	
NumInstallTrades≥2	$\{0,1\}$	0	1	Yes	
NumInstallTradesWBalance≥2	$\{0,1\}$	0	1	Yes	
NumRevolvingTrades≥2	$\{0,1\}$	0	1	Yes	
NumRevolvingTradesWBalance≥2	$\{0,1\}$	0	1	Yes	
NumInstallTrades≥3	$\{0,1\}$	0	1	Yes	
NumInstallTradesWBalance≥3	$\{0,1\}$	0	1	Yes	
NumRevolvingTrades≥3	$\{0,1\}$	0	1	Yes	
NumRevolvingTradesWBalance≥3	$\{0,1\}$	0	1	Yes	
NumInstallTrades≥5	$\{0,1\}$	0	1	Yes	
NumInstallTradesWBalance≥5	$\{0,1\}$	0	1	Yes	
NumRevolvingTrades≥5	$\{0,1\}$	0	1	Yes	
NumRevolvingTradesWBalance≥5	$\{0,1\}$	0	1	Yes	
NumInstallTrades≥7	$\{0,1\}$	0	1	Yes	
NumInstallTradesWBalance≥7	$\{0,1\}$	0	1	Yes	
NumRevolvingTrades≥7	$\{0,1\}$	0	1	Yes	
NumRevolvingTradesWBalance≥7	$\{0,1\}$	0	1	Yes	
NetFractionInstallBurden≥10	$\{0,1\}$	0	1	Yes	
NetFractionInstallBurden≥20	$\{0,1\}$	0	1	Yes	
NetFractionInstallBurden≥50	$\{0,1\}$	0	1	Yes	
NetFractionRevolvingBurden≥10	$\{0,1\}$	0	1	Yes	
NetFractionRevolvingBurden≥20	$\{0,1\}$	0	1	Yes	
NetFractionRevolvingBurden≥50	$\{0,1\}$	0	1	Yes	
NumBank2NatlTradesWHighUtilizationGeq2	$\{0,1\}$	0	1	Yes	+

Table 4: Separable actionability constraints for the heloc dataset.

The non-separable actionability constraints for this dataset include:

1. DirectionalLinkage: Actions on NumRevolvingTradesWBalance≥2 will induce to actions on ['NumRevolvingTrades≥2']. Each unit change in NumRevolvingTradesWBalance≥2 leads to: 1.00-unit change in NumRevolvingTrades≥2
2. DirectionalLinkage: Actions on NumInstallTradesWBalance≥2 will induce to actions on ['NumInstallTrades≥2']. Each unit change in NumInstallTradesWBalance≥2 leads to: 1.00-unit change in NumInstallTrades≥2
3. DirectionalLinkage: Actions on NumRevolvingTradesWBalance≥3 will induce to actions on ['NumRevolvingTrades≥3']. Each unit change in NumRevolvingTradesWBalance≥3 leads to: 1.00-unit change in NumRevolvingTrades≥3

4. DirectionalLinkage: Actions on NumInstallTradesWBalance≥3 will induce to actions on [‘NumInstallTrades≥3’]. Each unit change in NumInstallTradesWBalance≥3 leads to: 1.00-unit change in NumInstallTrades≥3
5. DirectionalLinkage: Actions on NumRevolvingTradesWBalance≥5 will induce to actions on [‘NumRevolvingTrades≥5’]. Each unit change in NumRevolvingTradesWBalance≥5 leads to: 1.00-unit change in NumRevolvingTrades≥5
6. DirectionalLinkage: Actions on NumInstallTradesWBalance≥5 will induce to actions on [‘NumInstallTrades≥5’]. Each unit change in NumInstallTradesWBalance≥5 leads to: 1.00-unit change in NumInstallTrades≥5
7. DirectionalLinkage: Actions on NumRevolvingTradesWBalance≥7 will induce to actions on [‘NumRevolvingTrades≥7’]. Each unit change in NumRevolvingTradesWBalance≥7 leads to: 1.00-unit change in NumRevolvingTrades≥7
8. DirectionalLinkage: Actions on NumInstallTradesWBalance≥7 will induce to actions on [‘NumInstallTrades≥7’]. Each unit change in NumInstallTradesWBalance≥7 leads to: 1.00-unit change in NumInstallTrades≥7
9. DirectionalLinkage: Actions on YearsSinceLastDelqTrade≤1 will induce to actions on [‘YearsOfAccountHistory’]. Each unit change in YearsSinceLastDelqTrade≤1 leads to: -1.00-unit change in YearsOfAccountHistory
10. DirectionalLinkage: Actions on YearsSinceLastDelqTrade≤3 will induce to actions on [‘YearsOfAccountHistory’]. Each unit change in YearsSinceLastDelqTrade≤3 leads to: -3.00-unit change in YearsOfAccountHistory
11. DirectionalLinkage: Actions on YearsSinceLastDelqTrade≤5 will induce to actions on [‘YearsOfAccountHistory’]. Each unit change in YearsSinceLastDelqTrade≤5 leads to: -5.00-unit change in YearsOfAccountHistory
12. ReachabilityConstraint: The values of [MostRecentTradeWithinLastYear, MostRecentTradeWithinLast2Years] must belong to one of 4 values with custom reachability conditions.
13. ThermometerEncoding: Actions on [YearsSinceLastDelqTrade≤1, YearsSinceLastDelqTrade≤3, YearsSinceLastDelqTrade≤5] must preserve thermometer encoding of YearsSinceLastDelqTradeleq., which can only decrease. Actions can only turn off higher-level dummies that are on, where YearsSinceLastDelqTrade≤1 is the lowest-level dummy and YearsSinceLastDelqTrade≤5 is the highest-level-dummy.
14. ThermometerEncoding: Actions on [AvgYearsInFile≥3, AvgYearsInFile≥5, AvgYearsInFile≥7] must preserve thermometer encoding of AvgYearsInFilegeq., which can only increase. Actions can only turn on higher-level dummies that are off, where AvgYearsInFile≥3 is the lowest-level dummy and AvgYearsInFile≥7 is the highest-level-dummy.
15. ThermometerEncoding: Actions on [NetFractionRevolvingBurden≥10, NetFractionRevolvingBurden≥20, NetFractionRevolvingBurden≥50] must preserve thermometer encoding of NetFractionRevolvingBurdengeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where NetFractionRevolvingBurden≥10 is the lowest-level dummy and NetFractionRevolvingBurden≥50 is the highest-level-dummy.
16. ThermometerEncoding: Actions on [NetFractionInstallBurden≥10, NetFractionInstallBurden≥20, NetFractionInstallBurden≥50] must preserve thermometer encoding of NetFractionInstallBurdengeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where NetFractionInstallBurden≥10 is the lowest-level dummy and NetFractionInstallBurden≥50 is the highest-level-dummy.
17. ThermometerEncoding: Actions on [NumRevolvingTradesWBalance≥2, NumRevolvingTradesWBalance≥3, NumRevolvingTradesWBalance≥5, NumRevolvingTradesWBalance≥7] must preserve thermometer encoding of NumRevolvingTradesWBalancegeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where NumRevolvingTradesWBalance≥2 is the lowest-level dummy and NumRevolvingTradesWBalance≥7 is the highest-level-dummy.
18. ThermometerEncoding: Actions on [NumRevolvingTrades≥2, NumRevolvingTrades≥3, NumRevolvingTrades≥5, NumRevolvingTrades≥7] must preserve thermometer encoding of

NumRevolvingTradesgeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where `NumRevolvingTrades≥2` is the lowest-level dummy and `NumRevolvingTrades≥7` is the highest-level-dummy.

19. ThermometerEncoding: Actions on [`NumInstallTradesWBalance≥2`, `NumInstallTradesWBalance≥3`, `NumInstallTradesWBalance≥5`, `NumInstallTradesWBalance≥7`] must preserve thermometer encoding of NumInstallTradesWBalancegeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where `NumInstallTradesWBalance≥2` is the lowest-level dummy and `NumInstallTradesWBalance≥7` is the highest-level-dummy.
20. ThermometerEncoding: Actions on [`NumInstallTrades≥2`, `NumInstallTrades≥3`, `NumInstallTrades≥5`, `NumInstallTrades≥7`] must preserve thermometer encoding of NumInstallTradesgeq., which can only decrease. Actions can only turn off higher-level dummies that are on, where `NumInstallTrades≥2` is the lowest-level dummy and `NumInstallTrades≥7` is the highest-level-dummy.

C.3 Actionability Constraints for the `givemecredit` Dataset

We present a list of all features and their separable actionability constraints in Table 5.

Name	Type	LB	UB	Actionability	Sign
`Age≤24`	$\{0,1\}$	0	1	No	
`Age_bt_25_to_30`	$\{0,1\}$	0	1	No	
`Age_bt_30_to_59`	$\{0,1\}$	0	1	No	
`Age≥60`	$\{0,1\}$	0	1	No	
`NumberOfDependents=0`	$\{0,1\}$	0	1	No	
`NumberOfDependents=1`	$\{0,1\}$	0	1	No	
`NumberOfDependents≥2`	$\{0,1\}$	0	1	No	
`NumberOfDependents≥5`	$\{0,1\}$	0	1	No	
`DebtRatio≥1`	$\{0,1\}$	0	1	Yes	+
`MonthlyIncome≥3K`	$\{0,1\}$	0	1	Yes	+
`MonthlyIncome≥5K`	$\{0,1\}$	0	1	Yes	+
`MonthlyIncome≥10K`	$\{0,1\}$	0	1	Yes	+
`CreditLineUtilization≥10.0`	$\{0,1\}$	0	1	Yes	
`CreditLineUtilization≥20.0`	$\{0,1\}$	0	1	Yes	
`CreditLineUtilization≥50.0`	$\{0,1\}$	0	1	Yes	
`CreditLineUtilization≥70.0`	$\{0,1\}$	0	1	Yes	
`CreditLineUtilization≥100.0`	$\{0,1\}$	0	1	Yes	
`AnyRealEstateLoans`	$\{0,1\}$	0	1	Yes	+
`MultipleRealEstateLoans`	$\{0,1\}$	0	1	Yes	+
`AnyCreditLinesAndLoans`	$\{0,1\}$	0	1	Yes	+
`MultipleCreditLinesAndLoans`	$\{0,1\}$	0	1	Yes	+
`HistoryOfLatePayment`	$\{0,1\}$	0	1	No	
`HistoryOfDelinquency`	$\{0,1\}$	0	1	No	

Table 5: Separable actionability constraints for the `heloc` dataset.

The non-separable actionability constraints for this dataset include:

1. ThermometerEncoding: Actions on [`MonthlyIncome≥3K`, `MonthlyIncome≥5K`, `MonthlyIncome≥10K`] must preserve thermometer encoding of MonthlyIncomegeq., which can only increase.Actions can only turn on higher-level dummies that are off, where `MonthlyIncome≥3K` is the lowest-level dummy and `MonthlyIncome≥10K` is the highest-level-dummy.
2. ThermometerEncoding: Actions on [`CreditLineUtilization≥10.0`, `CreditLineUtilization≥20.0`, `CreditLineUtilization≥50.0`, `CreditLineUtilization≥70.0`, `CreditLineUtilization≥100.0`] must preserve thermometer encoding of CreditLineUtilizationgeq., which can only decrease. Actions can only turn

off higher-level dummies that are on, where CreditLineUtilization≥10.0 is the lowest-level dummy and CreditLineUtilization≥100.0 is the highest-level-dummy.

3. ThermometerEncoding: Actions on [AnyRealEstateLoans, MultipleRealEstateLoans] must preserve thermometer encoding of continuousattribute., which can only decrease. Actions can only turn off higher-level dummies that are on, where AnyRealEstateLoans is the lowest-level dummy and MultipleRealEstateLoans is the highest-level-dummy.
4. ThermometerEncoding: Actions on [AnyCreditLinesAndLoans, MultipleCreditLinesAndLoans] must preserve thermometer encoding of continuousattribute., which can only decrease. Actions can only turn off higher-level dummies that are on, where AnyCreditLinesAndLoans is the lowest-level dummy and MultipleCreditLinesAndLoans is the highest-level-dummy.

C.4 Results on Classifier Performance

In Table 6, we report the performance of models on all datasets using all algorithms. We split each dataset into a training sample (80%, used for training and hyperparameter tuning) and a hold-out sample (20%, used to evaluate out-of-sample performance).

		AUC		Error	
Dataset	Model	Train	Test	Train	Test
heloc	LR	0.7723	0.7882	0.2774	0.2774
	XGB	0.7721	0.7880	0.2783	0.2783
	RF	0.8593	0.7853	0.2877	0.2877
german	LR	0.8193	0.7602	0.2350	0.2350
	XGB	0.8191	0.7614	0.2300	0.2300
	RF	0.9708	0.7937	0.2350	0.2350
givemecredit	LR	0.8411	0.8441	0.2390	0.2390
	XGB	0.8412	0.8442	0.2380	0.2380
	RF	0.8752	0.7928	0.2619	0.2619

Table 6: Overview of model performance

C.5 Results on the Illusion of Feasibility

We present the results of an ablation study to show how recourse may appear to be feasible when we fail to consider complex actionability constraints. Here, we repeat the experiments in Section 4 for the heloc dataset over three classes of nested actionability constraints:

- Simple, a separable action set which only includes constraints to conditions on the immutability, integrality, and soundness of features.
- Separable, a separable action set which includes all conditions in Simple and adds monotonicity constraints to ensure that certain features can only increase or decrease.
- Actual, a non-separable action set which includes all conditions in Simple and Separable. Note that this corresponds to the action set that we use in our main study.

We present the results from our procedure for all three action sets in Table 7.

Model Type	Metrics	Actual			Separable		Simple	
		Reach	AR	DiCE	AR	DiCE	AR	DiCE
LR	Certifies No Recourse	22.2%	—	—	—	—	—	—
	Outputs Action	77.8%	85.9%	57.6%	85.9%	57.0%	99.9%	65.6%
	↳ Loopholes	0.0%	41.1%	34.4%	41.1%	34.8%	95.5%	50.3%
	Outputs No Action	22.2%	14.1%	42.4%	14.1%	43.0%	0.1%	34.4%
	↳ Blindspots	0.0%	0.0%	21.0%	0.0%	21.7%	0.0%	14.6%
XGB	Certifies No Recourse	22.3%	NA	—	NA	—	NA	—
	Outputs Action	77.7%		57.3%		57.5%		60.5%
	↳ Loopholes	0.0%		42.1%		42.0%		46.7%
	Outputs No Action	22.3%		42.7%		42.5%		39.5%
	↳ Blindspots	0.0%		21.1%		21.1%		18.2%
RF	Certifies No Recourse	31.3%	NA	—	NA	—	NA	—
	Outputs Action	68.7%		49.3%		49.3%		59.2%
	↳ Loopholes	0.0%		29.5%		29.5%		44.8%
	Outputs No Action	31.3%		50.7%		50.7%		40.8%
	↳ Blindspots	0.0%		19.8%		19.7%		15.7%

Table 7: Feasibility of recourse across model classes, and various actionability constraints on the heloc dataset. We determine the ground-truth feasibility of recourse using reachable sets (Reach), and use these results to evaluate the reliability of verification with baseline methods for recourse provision (AR and DiCE). We describe the metrics in the caption of Table 2.