RESPONSIVENESS VERIFICATION: WILL PREDICTIONS CHANGE? HOW MUCH? HOW OFTEN?

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning models are often used in applications where their inputs change due to routine interactions, strategic manipulation, or noise. In such settings, models can undermine safety as these changes can lead them to predict over regions of the input space they have not seen. We propose to address these safety risks by measuring responsiveness—the probability that a model output attains a target prediction under feasible interactions defined by the interaction model. We develop algorithms to estimate responsiveness for any machine learning model, and a framework to specify broad classes of interaction models. We pair these algorithms with statistical guarantees that support practical validation. We demonstrate how our tools can promote safety and reliability across domains by detecting preclusion in recidivism prediction, estimating the cost of gaming in content moderation, and testing the robustness of benchmarks for LLMs.

1 INTRODUCTION

Machine learning models are often used in tasks where their inputs change – be it due to user interactions [20, 36, 45], stress tests [27, 51], distributional shifts [14, 33, 37], or noise [48, 57]. In such settings, a model that is accurate on held-out data can still undermine safety when its outputs respond to changes in its inputs in ways we fail to anticipate:

- *Content Moderation:* Consider an online platform that uses a machine learning model to detect bot accounts at scale [38, 47]. Here, bot accounts may be able to evade detection by manipulating their profile or behavior (e.g., reducing followers, or posting more often).
- *Lending:* Consider a lender who uses a machine learning model to underwrite loans. Here, applicants who are denied may be permanently denied access to credit as their prediction may not change under any possible action they can take [25, 50].
- *Benchmarking:* Consider an online retailer adopting a language model to streamline customer support based on its performance in a safety benchmark. Here, the retailer may adopt a different model, or none, if the benchmark scores and rankings change as a result of benign changes to the prompt [58] or context [55].

These failure modes arise because machine learning algorithms are designed to learn *pointwise mappings*. Simply put, we can expect an accurate model to correctly map inputs to their corresponding output, but we cannot expect a provably correct change in outputs when its inputs change.

Existing approaches have sought to mitigate failure modes by developing algorithms to learn models whose predictions will change in predictable ways [e.g., 33, 45, 46]. This approach is challenging because we often lack the labels that give the correct prediction after an input changes, the machinery to specify how inputs can change, and the algorithms to enforce these specifications during training [4, 52]. In effect, many interactions must not only adhere to semantic constraints (e.g., only changing in one direction) but also trigger downstream effects. In practice, these challenges have meant that many of the tools to build models that operate safely under recourse, gaming, and performativity have often sought to achieve safety by compromising fidelity. On the one hand, researchers have sought to design algorithms to learn models that will behave in predictable ways under simple interaction models. On the other, practitioners can resort to extreme measures, such as dropping all mutable features, even when those features contain useful signal.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

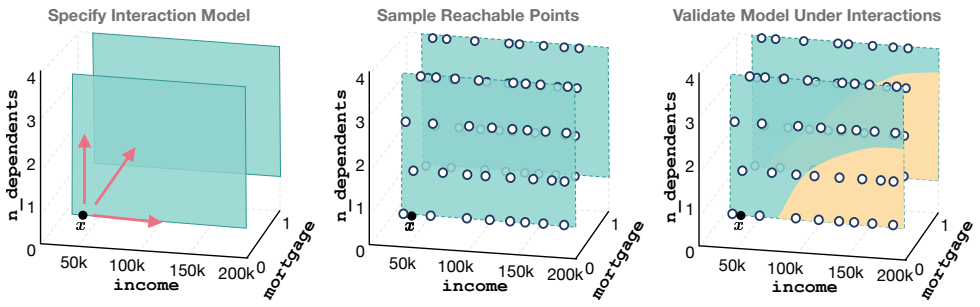


Figure 1: Detecting safety violations in models by measuring responsiveness. Safety violations arise when model outputs are overly or insufficiently responsive to interactions. On the right, we consider detecting preclusion of a model with one discrete (`n_dependents`), one continuous (`income`), and one binary (`mortgage`) feature. `mortgage` and `n_dependents` can only increase.

In this paper, we present an alternative approach: instead of learning models that will behave in predictable ways when their inputs change, we treat models as given and test how their outputs change under interactions. In contrast to learning, testing allows us to work with richer interaction models and can be used at multiple points in the model lifecycle. We capitalize on these advantages by developing theory, methods, and software that make testing rigorous, versatile, and practical. We introduce a foundational notion of *responsiveness* that applies across machine learning models and interaction models, supports decisions, and helps anticipate and mitigate harms. Our main contributions include:

1. We introduce a new approach for model validation in which we measure the responsiveness of its predictions. Our approach can ensure that models operate safely by detecting if, how, and when predictions change under realistic interaction models.
2. We develop machinery for defining feasible interactions through actionability constraints, and downstream effects. Our machinery allows practitioners to faithfully model interactions and to evaluate their specification through sensitivity analysis and ablation studies.
3. We design fast algorithms to estimate the responsiveness of predictions for any model, any dataset, and any interaction model. We pair our estimates with formal guarantees that can guide practical decisions in test design and yield interpretable claims on responsiveness.
4. We demonstrate how responsiveness verification can promote safety across domains, including recidivism prediction, content moderation, and benchmarking LLMs. Our results highlight how responsiveness can be used to detect and mitigate failure modes across domains.
5. We provide a Python library for responsiveness verification in our anonymized repository.

Related Work Our work is motivated by a broad class of safety violations that stem from interactions with the inputs of a model. This class captures a wide range of effects studied in areas such as adversarial robustness [14, 33], strategic classification [8, 20], performative prediction [35, 45], counterfactual invariance [27, 51], and algorithmic recourse [7, 42, 50].

Much of the work in these areas focuses on learning models that are robust to a specific effect – e.g., to gaming in strategic classification [6, 8, 20, 30], or to imperceptible perturbations in adversarial robustness [33, 56, 61, 62]. We propose to address these violations through testing. Testing only needs to check how a fixed model responds to interactions, and can thus accommodate far richer interaction models than we could enforce during training.

On the technical front, our proposal is most closely related to research on algorithmic recourse. Our machinery builds on the model-agnostic approach to recourse verification of Kothari et al. [25], who construct the set of all points reachable through interactions. We generalize their approach to any dataset and replace enumeration with sampling. Sampling makes verification computationally viable for complex interaction models with probabilistic downstream effects. As a result, our approach can verify recourse for a broad class of problems. Existing methods find recourse actions [see e.g., 24, 39, 53], often for specific model classes [39, 50], or certify the feasibility of recourse only over discrete feature spaces [25]. In contrast, our approach can certify the feasibility of recourse over any dataset, requires only black-box access, and returns practical guarantees.

2 FRAMEWORK

Our goal is to study how the output of a model changes due to interactions on its inputs. We consider a model $f : \mathcal{X} \to \mathcal{Y}$ that outputs an outcome $y \in \mathcal{Y}$ from a set of d *features* $\boldsymbol{x} = [x_1, \ldots, x_d] \in \mathcal{X}$. We assume that features encode semantically meaningful characteristics for a task at hand (e.g., `income` as opposed to pixel intensities or token embeddings). We also assume that we may only have *black-box* access to the model—i.e., we can query its output for arbitrary inputs $\boldsymbol{x}$, but may not have its parameters. We consider a subset of instances $\boldsymbol{x} \in \mathcal{X}$ that satisfy specific conditions (e.g., $f(\boldsymbol{x}) = 0$) for our validation tasks.

Understanding how outputs change from interactions over semantically meaningful features is difficult as interactions operate in a constrained space (e.g., an ordinal encoding of `education_level` can only increase). Interactions may also contain indirect changes to features (e.g., increasing `num_posts` may increase `url_count`). Below, we present a formal framework for modeling interactions:

Definition 1: Interaction Model
Given an individual with features $\boldsymbol{x} \in \mathcal{X}$, the interaction model is $I(\boldsymbol{x}) = (A(\boldsymbol{x}), P_{\boldsymbol{x},\cdot})$, where:

- Action set $A(\boldsymbol{x}) \subseteq \mathbb{R}^d$ is the set of feasible actions. We define

$$A(\boldsymbol{x}) := \Big\{ \boldsymbol{a} \in \prod_{j=1}^{d} A_j(\boldsymbol{x}) : G(\boldsymbol{x}, \boldsymbol{a}) \leq 0,\ H(\boldsymbol{x}, \boldsymbol{a}) = 0 \Big\},$$

  where $A_j(\boldsymbol{x}) \subseteq \mathbb{R}$ encodes feature-level constraints, $G(\boldsymbol{x}, \boldsymbol{a}) \leq 0$ encodes joint inequality constraints, and $H(\boldsymbol{x}, \boldsymbol{a}) = 0$ encodes joint equality constraints.
- $P_{\boldsymbol{x},\cdot} := \{P_{\boldsymbol{x},\boldsymbol{a}} : \boldsymbol{a} \in A(\boldsymbol{x})\}$ is a family of downstream effect distributions. Downstream effect distributions $P_{\boldsymbol{x},\boldsymbol{a}}$ may be specified through conditional distributions $P_j(\cdot \mid \boldsymbol{x}, \boldsymbol{a}, \varepsilon_{\mathrm{pa}(j)})$, where $\mathrm{pa}(j)$ denotes the upstream parents of feature j among the downstream effects.

$I(\boldsymbol{x})$ induces a reachable point $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{a} + \boldsymbol{\varepsilon}$, with action $\boldsymbol{a}$ and downstream effect $\boldsymbol{\varepsilon}$.

The interaction model captures how the features of a point can change through two components. The *action set* encodes how we can change the inputs to a model from a given point. We specify it in terms of *actionability constraints* on changes. These include *feature-level* constraints, which restrict each feature independently (e.g., enforcing monotonicity), and *joint* constraints, which restrict multiple features at the same time (e.g., preserving a one-hot encoding). The *downstream effect distribution* encodes what can change as a result of a feasible action, i.e., one that belongs to the action set. It can be specified to capture a variety of post-action effects, from *random effects* (e.g., measurement noise) to *causal effects* (e.g., increasing `num_posts` may increase `url_count`).

We use the interaction model to measure how the output of f responds to interactions at $\boldsymbol{x}$:

Definition 2: Responsiveness
Given a model f, point $\boldsymbol{x}$, interaction model $I(\boldsymbol{x})$, and a set of target outputs $\hat{Y}_{\boldsymbol{x}} \subseteq \mathcal{Y}$, let $\boldsymbol{x}'$ denote a reachable point induced by $I(\boldsymbol{x})$. The *responsiveness* of f at $\boldsymbol{x}$ is:

$$\rho(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{x}'} \Big[\mathbb{I}\big[f(\boldsymbol{x}') \in \hat{Y}_{\boldsymbol{x}} \big] \Big],$$

the probability that an interaction attains a target output.

We can use responsiveness in various tasks and applications (see Table 1). When the space of feasible actions is simple (i.e., we can enumerate through all of them), answering these questions is trivial [25]. However, enumeration is either infeasible (e.g., feature space is not discrete) or computationally expensive. In what follows, we present a Monte Carlo sampling procedure to estimate responsiveness and pair it with statistical guarantees in Section 3.

2.1 SAMPLING UNDER INTERACTIONS

Our framework falsifies, estimates, or tests the responsiveness of model outputs in three steps:

Validation Task	Responsiveness Semantics	Application	Interaction Model
Falsification: Does the output change under interactions?	$\rho^{\max}(\boldsymbol{x}) = \max_{\boldsymbol{x}'} \mathbb{I}[f(\boldsymbol{x}') \in \hat{Y}_{\boldsymbol{x}}]$	Falsify Preclusion in Lending	$a_{\texttt{late_payments}} \geq 0$ $a_{\texttt{checking_bal}} \geq 0$ $\varepsilon_{\texttt{savings_bal}} \sim P(\cdot \mid x, a_{\texttt{checking_bal}})$
Estimation: How often do outputs change under interactions?	$\hat{\rho}_n(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left[f(\boldsymbol{x}'_i) \in \hat{Y}_{\boldsymbol{x}}\right]$	Measure Prevalence of Gaming in Content Moderation	$a_{\texttt{num_tweets}} \geq 0$ $a_{\texttt{followers}} = 0$ $\varepsilon_{\texttt{url_count}} \sim P(\cdot \mid a_{\texttt{num_tweets}})$
Testing: Is an output sufficiently responsive under interactions?	$H_0 : \rho(\boldsymbol{x}) \geq \tau,\ H_1 : \rho(\boldsymbol{x}) < \tau$	Testing Performance Invariance in Safety Benchmarks for LLMs	$a \in \{\texttt{misspell}, \texttt{roleplay}, \texttt{translate}\}$ $\varepsilon \sim P_{\texttt{decode}}(\cdot \mid x, a)$

Table 1: Examples of validation tasks expressed through responsiveness with safety applications.

1. Specify an interaction model $I(\boldsymbol{x}) = (A(\boldsymbol{x}), P_{\boldsymbol{x},\cdot})$
2. Generate a sample $\hat{X}_n(\boldsymbol{x}) := \{\boldsymbol{x}'_i\}_{i=1}^n$ of feasible points under the interaction model $I(\boldsymbol{x})$
3. Estimate responsiveness by querying the outputs of the model over the reachable set:

$$\hat{\rho}_n(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left[f(\boldsymbol{x}'_i) \in \hat{Y}_{\boldsymbol{x}}\right]. \tag{1}$$

This approach has a couple of practical benefits. It is model agnostic: it works for any model class and task because it estimates responsiveness by querying model outputs across $\hat{X}_n(\boldsymbol{x})$. Moreover, we can amortize the cost of sampling by generating the sample $\hat{X}_n(\boldsymbol{x})$ once and reusing it to estimate responsiveness across multiple models (e.g., to account for responsiveness during model selection).

Sampling Algorithm The key challenge in designing a sampling-based approach is that the space of reachable points is highly irregular (see Fig. 1): it mixes discrete and continuous features, enforces joint constraints across features, and includes stochastic downstream effects. Standard samplers do not apply, and naive rejection sampling becomes inefficient as constraints accumulate [26].

We present an algorithm that samples reachable points efficiently by partitioning the feature space into conditionally independent parts (Algorithm 1). Each part in $\mathcal{J} := \{S_1, \ldots, S_k\}$ contains features coupled by a constraint or downstream effect, so that $A(\boldsymbol{x}) = \prod_{S \in \mathcal{J}} A_S(\boldsymbol{x}_S)$, where A_S is the action set for part S. This strategy can dramatically improve the speed and scalability of rejection sampling in our setting, since each feasibility check involves only a small number of features.

For each part $S \in \mathcal{J}$, we sample an action $\boldsymbol{a}_S$ from the proposal distribution $q_S(\cdot \mid \boldsymbol{x}_S)$ (Section 2.1) and pass it to $\mathsf{Rejector}(\boldsymbol{x}_S, \boldsymbol{a}_S)$, which checks whether $\boldsymbol{a}_S$ obeys the actionability constraints within the part. If $\boldsymbol{a}_S$ is feasible, we sample the downstream effect $\boldsymbol{\varepsilon}_S$ from $P_S(\boldsymbol{x}_S, \boldsymbol{a}_S)$. Once we have samples from all parts, we add $\boldsymbol{x} + \boldsymbol{a} + \boldsymbol{\varepsilon}$ to the reachable set $\hat{X}$ (Section 2.1) and repeat until we have n samples.

Algorithm 1 Rejection Sampling for Reachable Points

Input point $\boldsymbol{x}$, interaction model $I(\boldsymbol{x}) = (A(\boldsymbol{x}), P_{\boldsymbol{x},\cdot})$, proposal distribution $q(\cdot \mid \boldsymbol{x})$, sample size n
Initialize $\mathcal{J} \leftarrow \big\{\{j\} \mid j \in [d]\big\}$, $\hat{X} \leftarrow \{\}$
for each constraint/effect c in $I(\boldsymbol{x})$ **do**
  $\mathcal{M}_c \leftarrow \{S \in \mathcal{J} \mid S \cap \mathsf{scope}(c) \neq \varnothing\}$ *get parts related to constraint/effect*
  $\mathcal{J} \leftarrow (\mathcal{J} \setminus \mathcal{M}_c) \cup \big\{\bigcup_{S \in \mathcal{M}_c} S\big\}$ *combine related parts*
while $|\hat{X}| < n$ **do**
  $\boldsymbol{x}' \leftarrow \boldsymbol{x}$ *start from current point*
  for each part $S \in \mathcal{J}$ **do**
    Sample action part $\boldsymbol{a}_S \sim q_S(\cdot \mid \boldsymbol{x}_S)$ until $\mathsf{Rejector}(\boldsymbol{x}_S, \boldsymbol{a}_S)$ passes *sample feasible action in partition*
    $\boldsymbol{\varepsilon}_S \sim P_S(\boldsymbol{x}_S, \boldsymbol{a}_S)$ *sample downstream effect within partition*
    $\boldsymbol{x}'_S \leftarrow \boldsymbol{x}_S + \boldsymbol{a}_S + \boldsymbol{\varepsilon}_S$ *compute reachable sub-point*
  $\hat{X} \leftarrow \hat{X} \cup \{\boldsymbol{x}'\}$
Output $\hat{X}_n$, a sample of n reachable points

$\mathsf{Rejector}(\boldsymbol{x}_S, \boldsymbol{a}_S)$ solves a MIP to check if $\boldsymbol{a}_S \in A_S(\boldsymbol{x}_S)$ (Appendix B).

$$\min_{\boldsymbol{c}} \ 0 \quad \text{s.t. } \boldsymbol{c} = \boldsymbol{a}_S,\ \boldsymbol{a}_S \in A_S(\boldsymbol{x}_S)$$

3 Statistical Certification

A sample of reachable points $\hat{X}_n(\boldsymbol{x})$ allows us to perform validation tasks with responsiveness in a wider range of use cases. However, it does not yield definitive conclusions about safety; it yields probabilistic claims. Classical statistical estimation and testing procedures provide explicit interpretations of our probabilistic claims. These interpretations help practitioners make informed decisions when configuring the validation task (e.g., number of samples). Below, we describe the formal procedure for *testing* and leave formal falsification and estimation procedures to Appendix A.

3.1 Hypothesis Testing

We set up a hypothesis test to see if the responsiveness of a prediction exceeds or falls beneath a threshold value $\tau > 0$. This is useful in settings where we may expect predictions to exhibit a minimum or maximum degree of responsiveness. For instance, in lending, we may want to detect preclusion—i.e., if individuals can change the features of a model to be approved for a loan—by testing if $\rho(\boldsymbol{x}) < \tau$ for a small τ (e.g., 0.01).

Proposition 3. *Given a model $f : \mathcal{X} \to \mathcal{Y}$, we wish to test if its responsiveness at a point $\boldsymbol{x}$ is below a threshold value $\tau > 0$:*

$$H_0 : \ \rho(\boldsymbol{x}) \geq \tau \quad \textit{vs.} \quad H_1 : \ \rho(\boldsymbol{x}) < \tau$$

Given responsiveness estimate $\hat{\rho}$ constructed from a sample of n reachable points $\hat{X}_n(\boldsymbol{x})$ and a confidence level $\alpha \in (0, 1)$, we can claim that $f(\boldsymbol{x})$ is unresponsive whenever:

$$\bar{\rho}_\alpha(n, \hat{\rho}) := \mathsf{B}_{1-\alpha}(n\hat{\rho} + 1, n - n\hat{\rho}) < \tau,$$

where $\mathsf{B}_\alpha(a, b)$ denotes the αth quantile of a Beta distribution with parameters a, b.

We can set the size of a reachable set to control the probability of correctly rejecting H_0—detecting unresponsive points. Assuming $\rho(\boldsymbol{x}) \leq \tau - \delta$ for some margin $\delta \in (0, \tau)$, we can ensure:

$$\Pr\big(\textit{Reject } H_0 \mid H_0 \textit{ is False}\big) \geq 1 - \beta,$$

by estimating responsiveness with a sample of at least n^* reachable points where

$$n^* := 1 + \max\left\{n : \Pr(\bar{\rho}_\alpha(n, K/n) < \tau) < 1 - \beta\right\}, \textit{ where } K \sim \mathsf{Binomial}(n, \tau - \delta)$$

We can also deduce the minimum number of samples required for a valid test without making any assumptions on power $1 - \beta$ and the margin δ:

Remark 4. *Consider testing if the responsiveness of $f(\boldsymbol{x})$ exceeds a threshold value $\tau > 0$ via a hypothesis test in Prop. 3. Given the probability of incorrectly claiming $f(\boldsymbol{x})$ as unresponsive $\alpha > 0$, we need at least $n > \log \alpha / \log(1-\tau)$ reachable points to reject H_0.*

3.2 Stress-Testing Interaction Models

The interaction model $I(x)$ encodes the assumptions of practitioners explicitly. They must decide which features are actionable, what constraints govern joint changes, and whether there are downstream effects. These choices determine the set of feasible interactions and, consequently, responsiveness. In practice, these choices are non-trivial, and it is difficult to gauge their impact on results a priori. Below, we present two post-hoc analyses of how these choices affect responsiveness.

Sensitivity Analysis In some cases, practitioners can disagree or be unsure about what the "correct" interaction model is. In this case, we consider a family of interaction models $\mathcal{I}$ and see if responsiveness is sensitive to changes in the interaction model. Given $\mathcal{I}$, we can calculate:

$$\rho^{\text{all}}(\boldsymbol{x}) = \max_{I \in \mathcal{I}} \rho_I(\boldsymbol{x}) \quad \text{or} \quad \rho^{\text{any}}(\boldsymbol{x}) = \min_{I \in \mathcal{I}} \rho_I(\boldsymbol{x})$$

where $\rho_I(\boldsymbol{x})$ is the responsiveness of $f(\boldsymbol{x})$ under interaction model $I(\boldsymbol{x})$. Suppose $\hat{Y}_{\boldsymbol{x}}$ is the set of safe outputs (e.g., prediction not gamed). $\rho^{\text{all}}(\boldsymbol{x})$ informs us if there is a *consensus* of harm—i.e.,

$f(\boldsymbol{x})$ is unresponsive under all interaction models in $\mathcal{I}$. This is useful when practitioners disagree on the interaction model. If we have a consensus, the disagreement is not a significant issue. $\rho^{\text{any}}(\boldsymbol{x})$ informs us if there is a *possibility* of harm—i.e., $f(\boldsymbol{x})$ is unresponsive under at least one interaction model in $\mathcal{I}$. This is useful when the correct interaction model is ambiguous. If there is a possibility of harm, practitioners must refine the exact interaction model and resolve ambiguities. Our sensitivity analysis for testing shares the same statistical certifications as Section 3 (see Appendix A.2).

Identifying Drivers of (Un)responsiveness When a model output is unresponsive, we often want to know *which constraints* in the interaction model are responsible. For instance, one might suspect that a particular constraint (e.g., immutability of race) is preventing the output from changing, or may want to evaluate whether a new constraint (e.g., a policy rule) would materially affect responsiveness. We address this by systematically relaxing or removing individual constraints and measuring the resulting change in responsiveness. More formally, given the original action set $A(\boldsymbol{x})$, we relax or remove the constraint of interest to obtain a relaxed action set $A_{\text{rel}} \supseteq A(\boldsymbol{x})$, and measure: $\Delta\rho := \rho_{A_{\text{rel}}}(\boldsymbol{x}) - \rho_{A(\boldsymbol{x})}(\boldsymbol{x})$. A large positive $\Delta\rho$ indicates that the ablated constraint was a key driver of unresponsiveness. Practitioners can repeat this across constraints to rank their contributions and identify actionable targets for model revision or policy changes. We detail the full procedure in Algorithm 2 in the Appendix and demonstrate it in Section 4.

3.3 Efficient Sampling with Side Information

By default, we sample actions uniformly from $A(\boldsymbol{x})$. This way, responsiveness ensures coverage across all feasible actions, rather than under specific assumptions about which actions are most likely. Coverage over actions is important in safety-critical applications because the actions that expose failures may not be known a priori; without reliable side information, we refrain from prematurely downweighting feasible actions that may later be consequential. When additional information on informative regions of the reachable set is available, we can use *importance sampling*: let $p(\cdot \mid \boldsymbol{x})$ denote the uniform density over $A(\boldsymbol{x})$, and $q(\cdot \mid \boldsymbol{x})$ be a proposal distribution that concentrates probability on actions that practitioners believe are more informative. We reweight each sample to correct for the change in distribution:

$$\hat{\rho}_q(\boldsymbol{x}) := \tfrac{1}{n} \sum_{i=1}^{n} \tfrac{p(\boldsymbol{a}_i \mid \boldsymbol{x})}{q(\boldsymbol{a}_i \mid \boldsymbol{x})} \cdot \mathbb{I}\big[f(\boldsymbol{x}'_i) \in \hat{Y}_{\boldsymbol{x}}\big].$$

This approach can accelerate the detection of failure modes in targeted settings by concentrating samples on actions that might be more likely to reveal safety violations. However, we do not use importance sampling in our framework by default because: (1) specifying a good proposal distribution q is difficult, as it must be defined for each point $\boldsymbol{x}$; and (2) q must have support over the full action set (i.e., $q(\boldsymbol{a} \mid \boldsymbol{x}) > 0$ for all $\boldsymbol{a} \in A(\boldsymbol{x})$) since actions where q has zero mass are never sampled and cannot be corrected by reweighting, biasing the estimator.

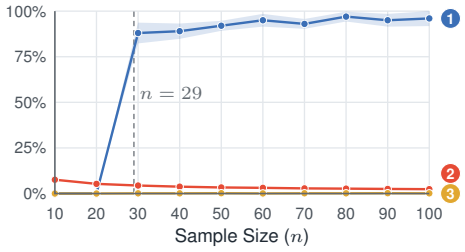
Either issue can lead to unreliable validation results. Hence, we use uniform sampling as the default and view importance sampling as an extension for settings with reliable side information.

4 Experiments and Demonstrations

We now demonstrate how our machinery can promote safety and reveal failure modes when developing or evaluating models using real-world data from salient applications. We include additional details in Appendices F to H, an additional demonstration on testing responsiveness in organ transplant models in Appendix E, and code in our anonymized repository.

4.1 Convergence and Reliability

We start with a simple experiment to empirically demonstrate the reliability of our estimates and statistical guarantees in Section 3. We work with the `german` dataset [9] to train an XGBoost classifier, and estimate the responsiveness of its predictions using $n = 10, 20, \ldots, 100$ points (see Appendix D for the interaction model and detailed setup). Since the `german` dataset is comprised of discrete features, we are able to compute the ground-truth value of responsiveness by enumeration. We



Metric	What it measures	At $n = 30$
① Detection Rate	Certifies an unresponsive prediction ($\rho \leq \tau - \delta$) as unresponsive	88.0%
② Estimation Error	Average gap between estimated and true responsiveness	4.4%
③ False Alarm Rate	Certifies a responsive prediction ($\rho \geq \tau$) as unresponsive	0.1%

Figure 2: Estimation error and test reliability as we increase sample size to estimate responsiveness of XGBoost model predictions on the `german` dataset [9]. The vertical line marks $n = 29$, the minimum number of samples required by Remark 4 to certify unresponsiveness with $\alpha = 0.05, \tau = 0.1, \delta = 0.05$.

compare our estimates to ground-truth values of $\rho(\boldsymbol{x})$, showing how the estimation error, false alarm rate, and detection rate change as the sample size grows in Fig. 2. Fig. 2 shows that at $n = 30$, the average estimation error is 4.4%, the false alarm rate is 0.1%, and the detection rate is 88.0%. These results show that the sample-size guarantee is conservative enough to keep false alarms rare while providing statistical power at modest sample sizes.

4.2 Preclusion in Recidivism Prediction

Many recidivism prediction models use features like `age` and `sex` to improve accuracy across subpopulations of defendants [see e.g., 10, 21, 28]. In practice, the features can lead models to predict recidivism *by default*. Lawless et al. [29] show how a risk score developed by the Pennsylvania Sentencing Commission predicts that every male defendant under 21 will recidivate, regardless of their criminal history. This behavior can be seen as a failure in responsiveness where defendants who are predicted to recidivate cannot attain a different prediction even if they were to "clear" their criminal history. Failures like this rarely surface under existing practices. In this case, the risk score was a simple score that was designed to be scrutinized by a wide range of experts and that had undergone periodic audits by third parties [44]. Even so, the failure went undetected.

In what follows, we show how our machinery would flag this failure and trace it to its root cause. We work with a dataset from the U.S. Department of Justice [49] which contains labeled examples for $n = 29{,}400$ defendants from New York State. Here, each example consists of $d = 14$ features describing criminal history, demographics, sentence characteristics, and program participation. The label is set to $y_i = 1$ if defendant i is rearrested within three years of release. We fit logistic regression and XGBoost models and test if they assign fixed predictions even when defendants "clear" their criminal history.

Our interaction model permits `prior_arrest` counts to decrease toward zero, program-participation features to increase, and treats demographic and sentence-related features as immutable. It also en-

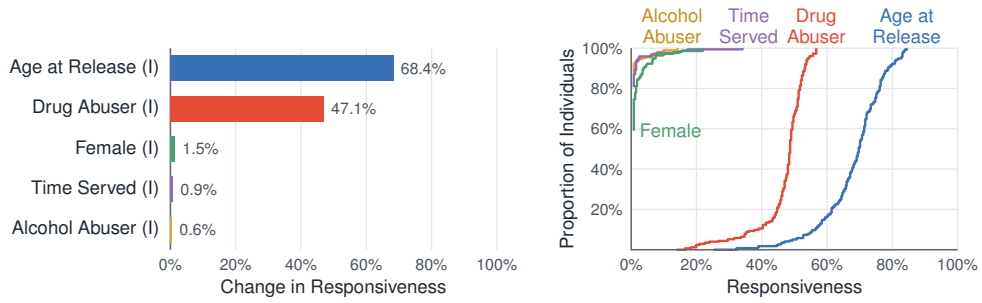


Figure 3: Constraints driving preclusion in the recidivism prediction model. We run an ablation analysis of the interaction model: we relax one constraint at a time and measure how the responsiveness of individuals with fixed predictions ($n = 5{,}084$) changes. Left: mean change in responsiveness after relaxing each constraint; "(I)" marks an immutability constraint. Right: distribution of responsiveness after each relaxation. Relaxing the immutability of `age_at_release` and `drug_abuser` largely resolves fixed predictions, indicating that the classifier relies on these demographic features rather than criminal history.

		Models Considered	Susceptibility to Gaming		Best Model	
Procedure	**Description**	**Robust / Total**	**Reported**	**Actual**	**Test AUC**	**Δ**
Baseline	Ignore gaming	901/901	63.0%	60.0%	0.763	—
Manual	Use only immutable features	370/370	**0.0%**	**0.0%**	0.570	0.193
Convex	Cap gaming via convex relaxation	687/901	**0.0%**	57.1%	0.754	0.009
Exact	Cap gaming via exact model	76/901	9.9%	9.9%	0.727	0.036

Table 2: Model selection on `twitterbot` under four procedures. Each procedure selects the model with the highest validation AUC among those that pass its gaming check, and we report the test performance of that model. *Baseline* returns an accurate model whose predictions are gameable. *Manual* prevents gaming but loses accuracy. *Convex* returns a model that appears robust but is gameable. Our method (*Exact*) returns a robust model that loses little accuracy. We report the following measures: *Robust / Total*, models that pass the check out of those considered; *Reported*, the proportion of gameable predictions estimated by the procedure; *Actual*, the proportion of gameable predictions under the exact interaction model; *Test AUC*; and Δ, the drop in test AUC relative to *Baseline*. All tests use $\tau = 0.05$ and $\alpha = 0.05$.

forces internal consistency among criminal-history features: for example, setting `prior_arrests` $= 5 \to 0$ requires `n_prior_felony`, as well as the other prior-history counts, to be set to zero. Given this model, we test for fixed predictions by checking whether $\rho(\boldsymbol{x}) \leq \tau$ with $\tau = 0.01$, and $\alpha = 0.05$.

Our tests reveal that 17.3% of defendants who are predicted to recidivate are assigned fixed predictions. One of the benefits of our approach is that we can readily understand the scope of failure by testing responsiveness across subpopulations or models. In this case, we find that the effect generalizes across training and test samples and arises when we work with simpler models such as logistic regression (since interactions are unchanged). We can pair such analysis with formal guarantees by repeating the sampling process or applying multiple testing correction.

We use Algorithm 2 to trace the root causes of fixed predictions by relaxing individual constraints and measuring the resulting change in responsiveness. This analysis identifies effective points of intervention at both the population and individual level. As seen in Fig. 3, relaxing immutability constraints on demographic and behavioral features (`age_at_release` and `drug_abuser`) produces the largest gains in responsiveness on average, indicating that the model relies on these features rather than criminal history, and removing them would largely resolve the failure.

4.3 Gaming in Content Moderation

Modern content moderation systems often rely on machine learning models to limit misinformation or harassment at scale [13, 15]. In such settings, we may build models to predict if an account belongs to a "bot" or "human" to guide follow-up actions [e.g., human review, 31]. Bot detection is difficult at scale: we need accurate models using many features, yet those features must be robust against bots that "game" the system to skirt detection.

We consider a bot detection task from account characteristics in the `twitterbot` dataset [12]. To build realistic attack models, we capture feasible interactions through non-separable constraints (e.g., forcing `num_tweets` to be 0 when `inactivity` is 1). We also assume some features (e.g., `followers`) are not actionable. Given the interaction model, we use an approach inspired by [60]: constructing classifiers that are robust to manipulation by penalizing or excluding certain features. We train a pool of penalized regression models over a large grid of l_1 and l_2 parameters using `glmnet` [11].

Table 2 details three approaches to highlight practical challenges when building a well-performing model that also limits gaming. Although models trained under the *Manual* procedure were all robust, they performed poorly with a Test AUC of 0.570. We also see that verifying responsiveness using a convex relaxation of the reachable set returns a well-performing model that *appears* robust. In fact, the Test AUC of the model under the *Convex* regime (0.754) exceeds that of the model chosen under *Exact* (0.727). However, we see that the *Convex* procedure under-reports responsiveness: the *perceived* proportion of responsive points is 0, when the proportion is actually 57.1% under a faithful interaction model.

These results (1) show that there may exist a well-performing model that is robust to gaming without additional adversarial training and (2) highlight the importance of validating responsiveness with

		I_{misspell}		I_{roleplay}		I_{chinese}		I_{french}	
Model	**Compliance**	Δ	($< 40.0\%$) **p**	Δ	($< 40.0\%$) **p**	Δ	($< 40.0\%$) **p**	Δ	($< 40.0\%$) **p**
OLMo 2 7B	22.7%	+31.7 p.p.	0.998	−10.0 p.p.	< 0.001	+31.7 p.p.	0.998	+33.0 p.p.	0.998
Llama 3.1 8B	27.7%	−1.9 p.p.	< 0.001	−0.6 p.p.	< 0.001	+1.9 p.p.	0.005	+1.9 p.p.	0.005
Qwen 2.5 7B	36.8%	+11.2 p.p.	1.000	−0.7 p.p.	< 0.001	−1.6 p.p.	< 0.001	+5.0 p.p.	1.000
Phi-3.5-mini	42.4%	+20.1 p.p.	1.000	+6.3 p.p.	1.000	+19.0 p.p.	1.000	+6.1 p.p.	1.000

Table 3: Model selection under responsiveness on `sorrybench`. We evaluate the rate of compliance with harmful prompts (lower is better) for several LLMs. For each interaction model, we evaluate responsiveness as compliance under interactions, e.g., misspelling, or translation to Chinese, and report the difference as Δ. We conduct responsiveness tests at $\tau = 40\%$ and $\alpha = 0.05$, and report the p-values. We use Benjamini-Hochberg multiplicity correction within each LLM's set of tests.

respect to accurate interactions. Procedures like *Convex* can lead to unaccounted risk, where a model that is perceived as robust remains vulnerable to gaming.

4.4 Evaluating Large Language Model Safety under Responsiveness

We conclude with a demonstration of how responsiveness can be used in a broader range of settings and model classes. Here, we consider benchmarking in large language models (LLMs). Although models can be highly sensitive to changes in their inputs [see, e.g., 32], standard practices in LLM evaluations are often static. One particular downstream failure mode is selecting a model that might seem optimal in a standard static evaluation, but whose performance degrades under interventions. Responsiveness provides us with statistical tools to rigorously test model performance in this scenario.

We consider a task of selecting a language model based on its compliance with harmful requests. To this end, we evaluate OLMo 2 7B [54], Llama 3.1 8B [17], Qwen 2.5 7B [3], and Phi-3.5 mini [19], in terms of their rate of compliance with harmful requests on the `sorrybench` dataset [58]. We also measure their compliance rate under an interaction model which modifies the prompt to induce compliance. We consider several such interaction models: I_{misspell}—the request is modified with misspellings, I_{roleplay}—the request is framed as roleplay, I_{chinese} and I_{french}—the request is translated from English to Chinese and French, respectively. For all interaction models, we consider the downstream effect ε of the randomness in LLM sampling. The function $f(\boldsymbol{x})$ we evaluate is an indicator of whether a model complied with a request or not, as judged by the `sorrybench` judge model, and $\hat{Y} = \{\texttt{comply}\}$. We evaluate *average* responsiveness.

$$\mathbb{E}_{x\sim D}[\rho(\boldsymbol{x})] = \mathbb{E}_{\boldsymbol{x}\sim D}\mathbb{E}_{\boldsymbol{x}'\sim X_I(\boldsymbol{x})}\left[\mathbb{I}[f(\boldsymbol{x}') = \texttt{comply}]\right].$$

We find that the base compliance rate of the OLMo model is the best, at 22.7%. Then, we aim to test if, under interactions, the compliance rate is at most $\tau = 40\%$. First, we use the sample size calculation in Prop. 3 to determine how many samples from the joint distribution of $(\boldsymbol{x}, \boldsymbol{x}')$, for $\boldsymbol{x} \sim D, \boldsymbol{x}' \sim X_I(\boldsymbol{x})$, we need to test this at power 95%. E.g., we find that we need 79 samples for OLMo and 159 samples for Llama. For models with lower performance, we find that we need more than 2000 samples, so we focus on OLMo and Llama, which have the best base performance. Finally, we show the results of testing in Table 3. The key finding is that even though OLMo has a lower base compliance rate on this benchmark and thus appears safer, its compliance under responsiveness increases substantially. In particular, we cannot ensure the compliance rate of OLMo is within the τ threshold.

5 Concluding Remarks

Machine learning is increasingly used in high-stakes systems with inputs that change through interactions from external actors. Yet our auditing practices and guarantees for model behavior under such interactions remain nascent [41]. Our work offers a concrete starting point for practitioners to reliably detect failures in how predictions respond to interactions.

Our approach requires practitioners to specify an interaction model based on domain knowledge, documentation, or policy rules. These specifications are assumptions that determine the constraints and downstream distributions. To guard against misspecified assumptions, we recommend repeating our tests over plausible interaction models. In practice, the outcome from this sensitivity analysis can help identify violations that are robust to assumptions surrounding actionability. Alternatively, it could also reveal that we fail to make any reliable claims over the plausible models.

AI Use Statement

We used Claude for assistance writing and managing our codebase. All generated code was tested manually by the authors. We also used Claude for grammar checks and readability improvements. All language edits were manually reviewed by the authors.

References

[1] Attia, Anthony, Jamie Webb, Katherine Connor, Chris JC Johnston, Michael Williams, Tim Gordon-Walker, Ian A Rowe, Ewen M Harrison, and Ben M Stutchfield. Effect of recipient age on prioritisation for liver transplantation in the uk: a population-based modelling study. *The Lancet Healthy Longevity*, 5(5): e346–e355, 2024.

[2] Attia, Antony, Ian A Rowe, Ewen M Harrison, Tim Gordon-Walker, and Ben M Stutchfield. Implausible algorithm output in uk liver transplantation allocation scheme: importance of transparency. *The Lancet*, 401(10380):911–912, 2023.

[3] Bai, Jinze, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and others. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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

[5] Ben-David, Arie. Monotonicity Maintenance in Information-Theoretic Machine Learning Algorithms. *Machine Learning*, 19(1):29–43, 1995.

[6] Chen, Yiling, Yang Liu, and Chara Podimata. Learning strategy-aware linear classifiers. In *Advances in Neural Information Processing Systems*, volume 33, pages 15265–15276. Curran Associates, Inc., 2020.

[7] Cheon, Harry, Anneke Wernerfelt, Sorelle Friedler, and Berk Ustun. Feature responsiveness scores: Model-agnostic explanations for recourse. In *The Thirteenth International Conference on Learning Representations*, 2025.

[8] Dong, Jinshuo, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 55–70. ACM, 2018.

[9] Dua, Dheeru and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[10] Eaglin, Jessica M. Constructing recidivism risk. *Emory LJ*, 67:59, 2017.

[11] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[12] Gilani, Zafar, Ekaterina Kochmar, and Jon Crowcroft. Classification of twitter accounts into automated agents and human users. In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, pages 489–496, 2017.

[13] Gillespie, Tarleton. Content moderation, ai, and the question of scale. *Big Data & Society*, 7(2): 2053951720943234, 2020.

[14] Goodfellow, Ian J, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

[15] Gorwa, Robert, Reuben Binns, and Christian Katzenbach. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society*, 7(1): 2053951719897945, 2020.

[16] Gotlieb, Neta, Amirhossein Azhie, Divya Sharma, Ashley Spann, Nan-Ji Suo, Jason Tran, Ani Orchanian-Cheff, Bo Wang, Anna Goldenberg, Michael Chassé, and others. The promise of machine learning applications in solid organ transplantation. *NPJ digital medicine*, 5(1):89, 2022.

[17] Grattafiori, Aaron, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and others. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[18] Gupta, Maya, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Alexander Mangylov, Wojciech Moczydlowski, and Alexander Van Esbroeck. Monotonic calibrated interpolated look-up tables. *Journal of Machine Learning Research*, 17(109):1–47, 2016.

[19] Haider, Emman, Daniel Perez-Becker, Thomas Portet, Piyush Madan, Amit Garg, Atabak Ashfaq, David Majercak, Wen Wen, Dongwoo Kim, Ziyi Yang, and others. Phi-3 safety post-training: Aligning language models with a" break-fix" cycle. *arXiv preprint arXiv:2407.13833*, 2024.

[20] Hardt, Moritz, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 111–122. ACM, 2016.

[21] Hester, Rhys. Prior record and recidivism risk. *American Journal of Criminal Justice*, 44:353–375, 2019.

[22] Holm, Sture. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[23] Hu, Lily, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 259–268, New York, NY, USA, 2019. ACM.

[24] Karimi, Amir-Hossein, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 353–362, 2021.

[25] Kothari, Avni, Bogdan Kulynych, Tsui-Wei Weng, and Berk Ustun. Prediction without preclusion: Recourse verification with reachable sets. In *The Twelfth International Conference on Learning Representations*, 2024.

[26] Kroese, Dirk P, Thomas Taimre, and Zdravko I Botev. *Handbook of Monte Carlo mMthods*. John Wiley & Sons, 2011.

[27] Kusner, Matt J, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

[28] Latessa, Edward J, Richard Lemke, Matthew Makarios, and Paula Smith. The creation and validation of the ohio risk assessment system (oras). *Fed. Probation*, 74:16, 2010.

[29] Lawless, Connor, Tsui-Wei Weng, Berk Ustun, and Madeleine Udell. Understanding fixed predictions via confined regions, 2025. URL https://arxiv.org/abs/2502.16380.

[30] Levanon, Sagi and Nir Rosenfeld. Strategic classification made practical. In *International Conference on Machine Learning*, pages 6243–6253. PMLR, 2021.

[31] Link, Daniel, Bernd Hellingrath, and Jie Ling. A human-is-the-loop approach for semi-automated content moderation. In *ISCRAM*, 2016.

[32] Lu, Yao, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, 2022.

[33] Madry, Aleksander, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.

[34] Marx, Charles, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In *Proceedings of Machine Learning and Systems 2020*, pages 9215–9224. 2020.

[35] Mendler-Dünner, Celestine, Juan Perdomo, Tijana Zrnic, and Moritz Hardt. Stochastic optimization for performative prediction. *Advances in Neural Information Processing Systems*, 33:4929–4939, 2020.

[36] Miller, John, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *International Conference on Machine Learning*, pages 6917–6926. PMLR, 2020.

[37] Miller, John P, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International conference on machine learning*, pages 7721–7735. PMLR, 2021.

[38] Milli, Smitha, Micah Carroll, Yike Wang, Sashrika Pandey, Sebastian Zhao, and Anca D. Dragan. Engagement, user satisfaction, and the amplification of divisive content on social media, 2024. URL https://arxiv.org/abs/2305.16941.

[39] Mothilal, Ramaravind K, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[40] Murgia, Madhumita. Algorithms are deciding who gets organ transplants. are their decisions fair?, 2023.

[41] Nader, Ralph. *Unsafe at Any Speed: The Designed-in Dangers of the American Automobile*. Pocket Books, 1966.

[42] Pawelczyk, Martin, Teresa Datta, Johan Van Heuvelden , Gjergji Kasneci, and Himabindu Lakkaraju. Probabilistically robust recourse: Navigating the trade-offs between costs and robustness in algorithmic recourse. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sC-PmTsiTB.

[43] Pearl, Judea. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

[44] Pennsylvania Bulletin, . Sentence Risk Assessment Instrument, April 2017.

[45] Perdomo, Juan, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, pages 7599–7609. PMLR, 2020.

[46] Ross, Alexis, Himabindu Lakkaraju, and Osbert Bastani. Learning models for actionable recourse. *Advances in Neural Information Processing Systems*, 34:18734–18746, 2021.

[47] Shao, Chengcheng, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. The spread of low-credibility content by social bots. *Nature Communications*, 9(1), November 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-06930-7. URL http://dx.doi.org/10.1038/s41467-018-06930-7.

[48] Tzeng, Jing-Tong, Jeng-Lin Li, Huan-Yu Chen, Chun-Hsiang Huang, Chi-Hsin Chen, Cheng-Yi Fan, Edward Pei-Chuan Huang, and Chi-Chun Lee. Improving the robustness and clinical applicability of automatic respiratory sound classification using deep learning–based audio enhancement: Algorithm development and validation. *JMIR AI*, 4:e67239, March 2025. ISSN 2817-1705. doi: 10.2196/67239. URL http://dx.doi.org/10.2196/67239.

[49] United States Department Of Justice, . Recidivism of prisoners released in 1994, 2002. URL https://www.icpsr.umich.edu/web/NACJD/studies/3355/versions/V8.

[50] Ustun, Berk, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. pages 10–19, 2019. doi: 10.1145/3287560.3287566.

[51] Veitch, Victor, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545*, 2021.

[52] Venkatasubramanian, Suresh and Mark Alfano. The philosophical basis of algorithmic recourse. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 284–293, 2020.

[53] Wachter, Sandra, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. 2017.

[54] Walsh, Evan Pete, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, and others. 2 olmo 2 furious (colm's version). In *Second Conference on Language Modeling*, 2025.

[55] Wang, Angelina, Daniel E. Ho, and Sanmi Koyejo. The inadequacy of offline llm evaluations: A need to account for personalization in model behavior, 2025. URL https://arxiv.org/abs/2509.19364.

[56] Wong, Eric and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, 2018.

[57] Xia, Xiaobo, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in neural information processing systems*, 33:7597–7610, 2020.

[58] Xie, Tinghao, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, and others. SORRY-Bench: Systematically evaluating large language model safety refusal. In *The Thirteenth International Conference on Learning Representations*, 2025.

[59] Zeng, Jiaming, Berk Ustun, and Cynthia Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(3):689–722, 2017.

[60] Zhang, Fei, Patrick PK Chan, Battista Biggio, Daniel S Yeung, and Fabio Roli. Adversarial feature selection against evasion attacks. *IEEE transactions on cybernetics*, 46(3):766–777, 2015.

[61] Zhang, Hongyang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically Principled Trade-off between Robustness and Accuracy. In *ICML*, 2019.

[62] Zhang, Huan, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane S. Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *ICLR*, 2020.

RESPONSIVENESS VERIFICATION: WILL PREDICTIONS CHANGE? HOW MUCH? HOW OFTEN?

SUPPLEMENTARY MATERIALS

A Formal Procedure for Additional Responsiveness Validation Tasks **15**
A.1 Statistical Certification for Estimation . . . 15
A.2 Statistical Certification for Sensitivity Analysis . . . 15

B Description of the Mixed Integer Program **17**
B.1 Encoding Additive Linkage Constraints . . . 17
B.2 Encoding Directional Linkage Constraints . . . 18
B.3 Encoding If-Then Constraints . . . 18
B.4 Encoding One-Hot Constraints . . . 19
B.5 Encoding Ordinal Constraints . . . 19
B.6 Encoding Reachability Constraints . . . 19
B.7 Encoding Switch Constraints . . . 20
B.8 Encoding Thermometer-Encoding Constraints . . . 20

C Supplementary Material for Section 2.1 **21**
C.1 Algorithm for Constraint Relaxation . . . 21

D Supporting Material for the `german` Study **22**
D.1 Interaction Model . . . 22
D.2 Additional Results . . . 22

E Monotonicity Violations in Organ Transplants **24**
E.1 Demonstration . . . 24
E.2 Components of the Transplant Benefit Score . . . 25
E.3 Description of Dataset . . . 25
E.4 Interaction Model . . . 26
E.5 Additional Results . . . 27

F Preclusion in Recidivism Prediction **29**
F.1 Description of Dataset . . . 29
F.2 Interaction Model . . . 29
F.3 Additional Results . . . 29

G Preventing Gaming in Content Moderation **30**
G.1 Description of Dataset . . . 30
G.2 Interaction Model . . . 30
G.3 Additional Results . . . 31

H LLM Safety Evaluation **31**
H.1 Description of the Dataset and LLM Details . . . 31
H.2 Interaction Models . . . 31
H.3 Statistical Test Details . . . 31

I Omitted Formal Results **32**
I.1 Proof of Remark 4 . . . 32

A Formal Procedure for Additional Responsiveness Validation Tasks

A.1 Statistical Certification for Estimation

We begin with estimation, which supports decisions across the machine learning pipeline. Consider a lender or insurer who builds a model to approve loans or price premiums. They may fear that applicants can manipulate features to obtain favorable predictions without affecting true risk [20, 23]. In such settings, model developers can estimate responsiveness to quantify the potential risks of manipulation and decide whether to remove manipulable features. By constructing a confidence interval around $\hat{\rho}_n(\boldsymbol{x})$, they can bound the probability of making such decisions in error.

Proposition 5. *Given a model $f : \mathcal{X} \to \mathcal{Y}$, a point $\boldsymbol{x}$ and confidence level $\alpha \in (0, 1)$, we wish to construct a confidence interval C with:*

$$\Pr\big(\rho(\boldsymbol{x}) \in C_\alpha\big) \geq 1 - \alpha.$$

We can construct a confidence interval that meets this condition based on an estimate $\hat{\rho}$ from n samples:

$$C_\alpha = \left[\mathsf{B}_{\frac{\alpha}{2}}(n\hat{\rho}, n - n\hat{\rho} + 1), \mathsf{B}_{1-\frac{\alpha}{2}}(n\hat{\rho} + 1, n - n\hat{\rho})\right]$$

where $\mathsf{B}_\alpha(a, b)$ denotes the αth quantile of a Beta distribution with parameters a, b.

In this setting, we can restrict $C_\alpha = C_\alpha(n, \hat{\rho})$ to a desired width L by using a sample of at least n^* reachable points where:

$$n^* := \min\left\{n \in \mathbb{N} : \max_{s \in [n]} \left|C_\alpha\left(n, \frac{s}{n}\right)\right| \leq L\right\}.$$

The results in Prop. 5 let us to gauge the number of samples practitioners need to achieve a desired level of precision. For a fixed level of risk α, the sample size n determines the level of uncertainty around the estimate $\hat{\rho}_n(\boldsymbol{x})$—increasing n decreases uncertainty: the estimate $\hat{\rho}_n(\boldsymbol{x})$ is expected to converge to ρ. This is because $\hat{\rho}_n(\boldsymbol{x})$ is the empirical mean of i.i.d Bernoulli variables; consistency follows immediately from the Weak Law of Large Numbers.

For instance, if we know that we would like to estimate with a precision of given $\alpha = 0.05$ and desired width 0.1, we can calculate that $N^{\min}(0.05, 0.1) = 402$. To half the width at the same confidence level, we would need at least $N^{\min}(0.05, 0.05) = 1574$ samples.

Example 6. *Consider a classification task where an online platform uses a model $f : \mathcal{X} \to \{0, 1\}$ to detect bot accounts. To assess the potential cost of undetected bot accounts, the platform estimates responsiveness of bot predictions under f:*

- *Target population $\{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) = 1\}$*
- *$\hat{Y}_{\boldsymbol{x}} = \{0\}$, classified as organic user*
- *$A(\boldsymbol{x})$: actionability constraints*
- *$\alpha = 0.05$—95% probability that C contains true ρ*
- *$L = 0.05$—C can be at most 0.05 wide*

Under these specifications, $N^{min}(0.05, 0.05) = 1574$.

Falsification is equivalent to setting $\tau = 0$ for testing. If we observe one point in the sample that attains the target output, we can immediately falsify. When there are no samples that attain the target output, we can run testing with a very small τ (e.g., 0.001).

A.2 Statistical Certification for Sensitivity Analysis

Constraint Analysis Suppose practitioners disagree on the actionability constraints in the interaction model (e.g., in predicting loan amount, `num_dependants` can change after a certain amount of

time vs. a more condensed timeline assumes it cannot change). In this case, we can test if interaction models in $\mathcal{I}$ come to a *consensus*. More formally, we consider the task results using:

$$\rho^{\text{all}}(\boldsymbol{x}) = \max_{I \in \mathcal{I}} \rho_I(\boldsymbol{x})$$

If the results hold for all $I \in \mathcal{I}$, we can conclude that the conflict in specification is not significant.

Distribution Analysis Suppose there is some ambiguity in the exact details of the interaction model (e.g., range of possible values for the parameters of the downstream distribution). In this case, we can test if there exists a *possibility* where we falsify or flag the prediction as unresponsive. More formally, we consider the task results using:

$$\rho^{\text{any}}(\boldsymbol{x}) = \min_{I \in \mathcal{I}} \rho_I(\boldsymbol{x})$$

If there exists $I \in \mathcal{I}$, where we falsify a prediction or claim that it is unresponsive, we can conclude that it is possible to have a safety violation under a realistic interaction model.

Hypothesis Testing $\mathcal{I}$ may not be countable (e.g., interaction models over an interval of downstream distribution parameters). We propose two ways to draw conclusions in testing depending on rejecting H_0 (i.e., sufficient evidence that $\rho(\boldsymbol{x}) < \tau$) entails for safety:

Flag: There exists an interaction model $I(\boldsymbol{x}) \in \mathcal{I}$ where we reject H_0. More formally, we check if

$$\min_{I \in \mathcal{I}} p_I^{\text{adj}} < \alpha$$

where p_I^{adj} is the p-value of the test under I while controlling the family-wise error rate (e.g., using Bonferroni's or Holm's p-value adjustment [22]. We would typically use this if $\hat{Y}_{\boldsymbol{x}}$ is a safe outcome (e.g., predicted to pay back loan), meaning rejecting H_0 is a safety violation.

Certify: Check we reject H_0 for all interaction models $I(\boldsymbol{x}) \in \mathcal{I}$. More formally, we check if

$$\max_{I \in \mathcal{I}} p_I < \alpha$$

where p_I is the p-value of the test under I. We would typically use this if $\hat{Y}_{\boldsymbol{x}}$ is an unsafe outcome (e.g., bots predicted as human users), meaning that rejecting H_0 is safe.

Using the examples from the introduction, when testing for preclusion, we want to *flag* fixed predictions, whereas in gaming, we want to *certify* that the model is not susceptible to gaming across interaction models.

B Description of the Mixed Integer Program

Our MIP checks if the sampled interaction $\boldsymbol{a}'$ obeys the constraints in $A(\boldsymbol{x})$. Although each a_j for $j \in [d]$ obeys separable constraints: integrality, monotonicity and bounds, we must ensure that $\boldsymbol{a}'$ does not violate joint constraints. Given $\boldsymbol{x}$, a sampled interaction $\boldsymbol{a}'$ and a set of constraints $A(\boldsymbol{x})$, the routine solves a problem of form:

$$\min_{\boldsymbol{a},\boldsymbol{c}} \ \|\boldsymbol{a}\|_1 \quad \text{s.t. } \boldsymbol{c} = \boldsymbol{a}',\ \boldsymbol{c} \in A(\boldsymbol{x}) \tag{2}$$

We implement Eq. (2) as a mixed-integer program that consists of a baseline formulation—enforcing separable constraints like bounds and monotonicity—and additional constraints to enforce joint conditions and deterministic downstream effects. The baseline formulation has the form:

$$\min \quad \sum_{j\in[d]} \left(a_j^+ + a_j^-\right) \tag{3a}$$

$$\text{s.t.} \quad c_j = a'_j \qquad j \in [d] \quad \textit{match sampled aggregate interaction } \boldsymbol{a}' \tag{3b}$$

$$a_j, c_j \in [\underline{A}_j(\boldsymbol{x}), \bar{A}_j(\boldsymbol{x})] \qquad j \in [d] \quad \textit{separable actionability bounds} \tag{3c}$$

$$a_j^+, a_j^- \in \mathbb{R}_+ \qquad j \in [d] \quad \textit{positive, negative components of } a_j \tag{3d}$$

$$a_j = a_j^+ - a_j^- \qquad j \in [d] \quad \textit{absolute value reconstruction} \tag{3e}$$

$$c_j = a_j \qquad j \in [d] \quad \textit{baseline aggregate change} \tag{3f}$$

$$\sigma_j \in \{0, 1\} \qquad j \in [d] \quad \textit{sign of } a_j \tag{3g}$$

$$a_j^+ \geq a_j \qquad j \in [d] \quad \textit{positive component of } a_j \tag{3h}$$

$$a_j^- \geq -a_j \qquad j \in [d] \quad \textit{negative component of } a_j \tag{3i}$$

$$a_j^+ \leq \bar{A}_j(\boldsymbol{x})\sigma_j \qquad j \in [d] \quad \textit{only 1 of } a_j^+ \textit{ or } a_j^- \textit{ can be positive} \tag{3j}$$

$$a_j^- \leq -\underline{A}_j(\boldsymbol{x})(1-\sigma_j) \qquad j \in [d] \quad \textit{only 1 of } a_j^+ \textit{ or } a_j^- \textit{ can be positive} \tag{3k}$$

$$\boldsymbol{a} \in A(\boldsymbol{x}) \qquad \textit{joint actionability constraints} \tag{3l}$$

where $\underline{A}_j(\boldsymbol{x}), \bar{A}_j(\boldsymbol{x})$ denote the lower and upper bounds of actions for feature j according to feature-level constraints.

The baseline formulation in Eq. (3) minimizes the l_1 norm of $\boldsymbol{a}$, splitting $\boldsymbol{a}$ into positive and negative parts $a_j^+, a_j^- \geq 0$ (3e), of which only one can be non-zero through (3j)–(3k). This allows us to use this baseline formulation for both sampling and enumeration. Here, σ_j is a boolean sign variable: $\sigma_j = 1$ allows positive movement and $\sigma_j = 0$ allows negative movement.

(3b) stipulates that we intervene with sampled aggregate interaction $\boldsymbol{a}'$—i.e., find an interaction $\boldsymbol{a}$ whose aggregate change $\boldsymbol{c}$ satisfies the remaining constraints and is equal to $\boldsymbol{a}'$. In the baseline model, (3f) sets $\boldsymbol{c} = \boldsymbol{a}$; deterministic downstream effects can replace this equality for linked target features. The remaining constraints enforce separable (constraint (3c), (3j), (3k)) and non-separable actionability constraints (constraint (3l)).

Below we provide all actionability constraints and their explicit formulations.

B.1 Encoding Additive Linkage Constraints

Additive linkage captures relationships in which an action on a source feature must equal (or upper/lower bound) a weighted sum of *aggregate changes* in a set of target features. We use this construction to express conservation-like relationships between a source intervention and the total resulting change across one or more targets.

Given a source feature $k \in [d]$, a non-empty set of target features $T \subseteq [d] \setminus \{k\}$, a scalar source coefficient $\alpha_k \in \mathbb{R} \setminus \{0\}$, and target coefficients $\boldsymbol{\alpha}_T \in (\mathbb{R} \setminus \{0\})^{|T|}$, we add the following constraint to Eq. (3):

$$\alpha_k a_k \ - \sum_{l\in T} \alpha_l c_l \ \square \ 0 \tag{4}$$

where a_k is the action on the source, c_l is the aggregate change in target l (as defined in Eq. (6)), and $\square \in \{=, \geq, \leq\}$ specifies whether the linkage is enforced as equality or inequality. Note that the use of c_l allows additive linkage to compose with directional linkage on the same target features.

B.2 Encoding Directional Linkage Constraints

We often encounter features where intervening on them has a direct (and sometimes deterministic) effect on other features. For example, in Table 8, joint constraint 4 stipulates that `urls_count` increases at most as the change in `num_tweets`. Here, the "source variable"—the source of the effect—is `num_tweets` and the "target variable"—the feature affected—is `urls_count`.

We capture this effect, called *Directional Linkage*, by adding additional constraints to Eq. (3). Given source feature $k \in [d]$, a non-empty set of target features $T \subseteq [d] \setminus \{k\}$, and a scale vector $\tilde{\boldsymbol{s}} \in (\mathbb{R} \setminus \{0\})^{|T|+1}$ with one entry per feature in $\{k\} \cup T$, we define the effective per-target scales $s_l := \tilde{s}_l / \tilde{s}_k$ for each $l \in T$. We then introduce one auxiliary variable b_l per target capturing the induced change, and add the following constraints to Eq. (3):

$$b_l - s_l \cdot a_k = 0 \tag{5}$$

$$c_l - a_l - b_l = 0 \tag{6}$$

for each target feature $l \in T$, where b_l is the change in feature l induced by the source action a_k, and c_l represents the aggregate change in l.

The auxiliary variable b_l is bounded by the propagation of the source action bounds $[\,-\underline{A}_k(\boldsymbol{x}),\ \bar{A}_k(\boldsymbol{x})\,]$ through the scale s_l, namely

$$\min\big(s_l \bar{A}_k(\boldsymbol{x}),\ -s_l \underline{A}_k(\boldsymbol{x})\big) \ \le\ b_l \ \le\ \max\big(s_l \bar{A}_k(\boldsymbol{x}),\ -s_l \underline{A}_k(\boldsymbol{x})\big), \tag{7}$$

which handles both signs of s_l. By default, the linkage also *widens* the aggregate-change bounds on c_l by the corresponding b_l bounds, so that c_l can absorb the induced effect on top of any direct action a_l. When the linkage is configured to preserve the original target bounds, this widening step is skipped and the original c_l bounds are kept.

We can also substitute the equality in Eq. (5) with inequalities. The aforementioned example with `num_tweets` and `urls_count` is a case where the relationship is an inequality ($\le$) and $s_l = 1$.

B.3 Encoding If-Then Constraints

If-then constraints enforce conditional relationships of the form "if the action on feature k pushes $\boldsymbol{x}_k$ above a threshold, then the post-action value of feature l must satisfy a consequent condition". This is useful for situations such as requiring a minimum balance level once a checking account is opened, or requiring a minimum program completion level once enrollment changes.

We capture an antecedent action threshold v^{if} on feature k, with the antecedent active when $a_k \ge v^{\text{if}}$, and a consequent on the *post-action feature value* $x_l + a_l$ given as either $x_l + a_l \ge v^{\text{then}}$ or $x_l + a_l = v^{\text{then}}$. We introduce a binary indicator $u \in \{0, 1\}$ that activates when the antecedent holds, and add the following big-M constraints to Eq. (3):

$$M^{\text{fwd}} u - a_k \ge -v^{\text{if}} + \epsilon \tag{8a}$$

$$-M^{\text{bwd}} u + a_k \ge v^{\text{if}} - M^{\text{bwd}} \tag{8b}$$

$$a_l - M_l^{\text{lb}} u \ge \tilde{v}^{\text{then}} - M_l^{\text{lb}} \tag{8c}$$

where $\tilde{v}^{\text{then}} = v^{\text{then}} - x_l$ rewrites the consequent feature value as a required action magnitude, $\epsilon > 0$ is a small tolerance, and the big-M constants are tightened from the bounding box on $\boldsymbol{a}$:

$$M^{\text{fwd}} = \bar{A}_k(\boldsymbol{x}) - v^{\text{if}} + \epsilon, \quad M^{\text{bwd}} = v^{\text{if}} + \underline{A}_k(\boldsymbol{x}), \quad M_l^{\text{lb}} = \tilde{v}^{\text{then}} + \underline{A}_l(\boldsymbol{x}).$$

(8a) and (8b) together force $u = \mathbb{I}[a_k \ge v^{\text{if}}]$, while (8c) enforces the consequent lower bound on a_l (equivalent to the lower bound on $x_l + a_l$) only when $u = 1$. Note that our implementation always uses this action-threshold trigger for the antecedent, regardless of whether the antecedent condition is specified with equality or inequality sense. When the consequent uses an equality sense, we additionally add an upper bound:

$$a_l + M_l^{\text{ub}} u \le \tilde{v}^{\text{then}} + M_l^{\text{ub}} \tag{9}$$

with $M_l^{\text{ub}} = \bar{A}_l(\boldsymbol{x}) - \tilde{v}^{\text{then}}$.

B.4 Encoding One-Hot Constraints

One-hot constraints maintain the encoding of a categorical attribute that has been split into a collection of binary dummies. Given a categorical attribute Z with m levels encoded as boolean features $x_{j_1}, \ldots, x_{j_m}$ where $x_{j_k} = \mathbb{I}[Z = z_k]$, valid feature vectors must satisfy a cardinality requirement on the dummies.

Given the index set $J \subseteq [d]$ of dummies and a cardinality limit $L \in [|J|]$, we add a single linear constraint to Eq. (3):

$$\sum_{j \in J} a_j \,\square\, L - \sum_{j \in J} x_j \tag{10}$$

where $\square$ is $=$ when the encoding is *exact* (exactly L dummies must be active) and $\leq$ when the encoding is *at most* L. The right-hand side rewrites the cardinality requirement on $\boldsymbol{x} + \boldsymbol{a}$ in terms of the action $\boldsymbol{a}$, so that any feasible action preserves the encoding.

B.5 Encoding Ordinal Constraints

Ordinal constraints maintain the encoding of an ordinal attribute with m levels that has been one-hot encoded. They differ from one-hot constraints in that we may also wish to restrict transitions between levels (e.g., an ordinal attribute representing tenure can only increase).

Ordinal constraints are a special case of the reachability constraint encoding instantiated as follows. The set of feasible values V contains every length-m binary vector with exactly one entry equal to one. We construct the reachability matrix $E \in \{0,1\}^{|V| \times |V|}$ from a step direction $\sigma \in \{-1, 0, 1\}$:

$$E_{ij} = \begin{cases} \mathbb{I}[i \leq j] & \text{if } \sigma = +1 \text{ (level can only increase)} \\ \mathbb{I}[j \leq i] & \text{if } \sigma = -1 \text{ (level can only decrease)} \\ 1 & \text{if } \sigma = 0 \text{ (free)} \end{cases}$$

where V is ordered by ascending level. We then add Eqs. (11) to (13) from the reachability constraint subsection below using these V and E.

B.6 Encoding Reachability Constraints

Reachability constraints generalize one-hot, ordinal and thermometer encodings to arbitrary subsets of features that are restricted to a finite set of jointly feasible values, with optional restrictions on which values can be reached from a given starting point. Many encoding constraints in our framework reduce to a reachability constraint with an appropriate value set V and reachability matrix E.

Given an ordered set of feasible values $V \subseteq \mathbb{R}^{|J|}$ over a feature subset $J \subseteq [d]$ and a reachability matrix $E \in \{0,1\}^{|V| \times |V|}$ with $E_{ij} = 1$ iff v_j is reachable from v_i, we add the following constraints to Eq. (3). Let $i^* \in [|V|]$ denote the index of the value matching $\boldsymbol{x}_J$ (i.e., $v_{i^*} = \boldsymbol{x}_J$), and let $e_k = E_{i^*,k}$ be the reachability indicator for value v_k from the current point. We introduce indicators $u_k \in \{0,1\}$ for $k \in [|V|]$ and require:

$$\sum_{k \in [|V|]} u_k = 1 \tag{11}$$

$$u_k \leq e_k \qquad \forall\, k \in [|V|] \tag{12}$$

$$a_j = \sum_{k \in [|V|]} e_k \,(v_{k,j} - x_j)\, u_k \qquad \forall\, j \in J \tag{13}$$

(11) ensures that exactly one feasible value is selected, (12) forbids selecting unreachable values—without it, the model could pick $u_k = 1$ for some v_k with $e_k = 0$, which would zero out the right-hand side of (13) and falsely report $\boldsymbol{a}_J = 0$ while "selecting" an unreachable destination—and (13) pins each component of $\boldsymbol{a}$ on J to the action that moves $\boldsymbol{x}_J$ to the selected value v_k. In our CPLEX implementation, (12) is enforced directly by setting the upper bound of each indicator u_k to e_k.

B.7 Encoding Switch Constraints

Switch constraints capture mutability that is conditional on a binary "switch" feature. Concretely, when the switch is on, a designated set of target features cannot change; when the switch is off, those targets may (and optionally must) change. For example, in the `german` dataset, `CheckingAcct≥0` should only be permitted to vary when `CheckingAcct_exists` is `True`.

By default, our implementation treats the switch state at the input point $\boldsymbol{x}$ as a precondition: the constraint requires that $\boldsymbol{x}$ already satisfies the implied target invariant (targets are zero when the switch is on at $\boldsymbol{x}$, and—if force-change is enabled—non-zero when the switch is off at $\boldsymbol{x}$). The MIP encoding below is added only when this precondition is met; a permissive mode that skips this check is also supported.

Given a boolean switch feature $k \in [d]$, an "on value" $\nu \in \{0, 1\}$, a set of target features $T \subseteq [d] \setminus \{k\}$, and per-target step bounds $A_l^+, A_l^- \geq 0$ derived from $\bar{A}_l(\boldsymbol{x}), \underline{A}_l(\boldsymbol{x})$, we introduce a binary indicator $w \in \{0, 1\}$ that tracks the *post-action* switch state. We then add the following constraints to Eq. (3):

$$w = \begin{cases} x_k + a_k & \text{if } \nu = 1 \\ 1 - x_k - a_k & \text{if } \nu = 0 \end{cases} \tag{14a}$$

$$a_l^+ + A_l^+ \, w \leq A_l^+ \qquad \forall \, l \in T \tag{14b}$$

$$a_l^- + A_l^- \, w \leq A_l^- \qquad \forall \, l \in T \tag{14c}$$

where a_l^+, a_l^- are the positive and negative parts of a_l from (3e). (14a) forces $w = 1$ exactly when the post-action switch state $x_k + a_k$ matches the on-value ν, and (14b)–(14c) together fix $a_l = 0$ for every target $l \in T$ whenever $w = 1$.

When we additionally require that targets must change when the switch is off (i.e., $w = 0$), we add a single forcing constraint:

$$\sum_{l \in T} \left(a_l^+ + a_l^- \right) + \delta \, w \geq \delta \tag{15}$$

where δ is slightly less than the minimum step size across targets in T. When $w = 1$, this constraint is slack; when $w = 0$, it forces the total absolute movement on the targets to be at least δ.

B.8 Encoding Thermometer-Encoding Constraints

Datasets often include features that are based on thresholds. These features are often encoded like unary codes, a number of ones followed by zeros. For example, in Table 8, `age_of_account_geq` has a thermometer encoding with thresholds at 180, 365, 730 and 1825 days. Hence there are five possible encoding values:

1. $[0, 0, 0, 0]$: account is less than 180 days old
2. $[1, 0, 0, 0]$: account is older than 180 days but less than 365 days old
3. $[1, 1, 0, 0]$: account is older than 365 days but less than 730 days old
4. $[1, 1, 1, 0]$: account is older than 730 days but less than 1825 days old
5. $[1, 1, 1, 1]$: account is more than 1825 days old

Given an ordered set of feasible values V as above, we define a reachability matrix $E \in \{0, 1\}^{|V| \times |V|}$, where the (i, j)-th entry of E is 1 when we can reach the j-th element of V from its i-th element and 0 otherwise. There are three possibilities for E: an upper triangular matrix of ones (when the underlying value can only increase), a lower triangular matrix of ones (when it can only decrease), or an all-ones matrix (when the value can move freely). For example, `age_of_account_geq` additionally has a monotonicity constraint—age can only increase—so E is an upper triangular matrix of ones (i.e., one can reach $[1, 0, 0, 0]$ from $[0, 0, 0, 0]$, but not the other way around).

A thermometer-encoded feature subset is then encoded in the MIP exactly as a reachability constraint: we add Eqs. (11) to (13) from the reachability constraint subsection above with the value set V and reachability matrix E defined here. As with reachability, the upper bound $u_k \leq e_k$ in (12) is enforced directly via the indicator's variable bound.

C Supplementary Material for Section 2.1

C.1 Algorithm for Constraint Relaxation

Algorithm 2 Paired Reachability Probe

Input: Point $\boldsymbol{x}$, sample size n, original action set $A(\boldsymbol{x})$, relaxed action set A_{rel}, classifier f

$y_0 \leftarrow f(\boldsymbol{x})$
$\mathcal{P} \leftarrow$ Sample (n proposals from A_{rel} around $\boldsymbol{x}$)
$\mathsf{feas} \leftarrow \mathbf{1}_n$
$\mathsf{feas}^{\text{rel}} \leftarrow \mathbf{1}_n$
for each partition p in $A(\boldsymbol{x})$ **do**
    $\mathsf{feas} \leftarrow \mathsf{feas} \wedge \text{IsFeasible}(p, A(\boldsymbol{x}), \mathcal{P}[p])$
    $\mathsf{feas}^{\text{rel}} \leftarrow \mathsf{feas}^{\text{rel}} \wedge \text{IsFeasible}(p, A_{\text{rel}}, \mathcal{P}[p])$
$\hat{y} \leftarrow -\mathbf{1}_n, \quad \hat{y}^{\text{rel}} \leftarrow -\mathbf{1}_n$
$\hat{y}[\mathsf{feas}] \leftarrow f(\mathcal{P}[\mathsf{feas}])$
$\hat{y}^{\text{rel}}[\mathsf{feas}^{\text{rel}}] \leftarrow f(\mathcal{P}[\mathsf{feas}^{\text{rel}}])$
return $\left(y_0, \mathsf{feas}, \mathsf{feas}^{\text{rel}}, \hat{y}, \hat{y}^{\text{rel}}\right)$

D SUPPORTING MATERIAL FOR THE german STUDY

In Fig. 2, we conduct a study on sample size n to (1) validate our responsiveness estimation and testing procedure outlined in Section 2.1, and (2) determine their reliability under various sample sizes. We work with a discrete dataset, german, where we can fully enumerate reachable sets using the enumeration procedure from Kothari et al. [25]. The enumerated reachable sets provide ground truth responsiveness proportions. We compare our results to determine the error of our *estimation* procedure and the precision (in the main body, we refer to it as "reliability" for simplicity) of our *testing* procedure under XGBoost (XGB).

The german dataset is a credit dataset originally compiled in 1994 that is publicly available through the UCI Machine Learning Repository [9]. It contains $n = 1{,}000$ de-identified instances, each representing a credit applicant. It includes $d = 20$ categorical or discrete features, providing insights into aspects such as loan history, demographic information (including gender, age, and marital status), occupation, and past payment behavior. The objective is to predict whether an applicant is a "good" $(y_i = 1)$ or "bad" $(y_i = 0)$ credit customer. We note that the dataset does not have missing values, and have adapted some feature names for clarity.

D.1 INTERACTION MODEL

We consider an interaction model where each applicant can intervene on current features like account balances, but not history nor credit related features. For example, Housing=Owner is not actionable since one cannot go from renting to buying without additional loans. This interaction model is conservative and is intended to capture indisputable actionability constraints. In total, our dataset contains 36 features of which 9 are actionable and 10 are mutable. There a total of four constraints: two Directional Linkage, and two Thermometer Encoding constraints:

- Directional Linkage constraints in this interaction model govern downstream effects on Age from 1) YearsAtResidence and 2) YearsEmployed$\geq$1, which form a partition.
- Thermometer Encoding constraints enforce conceptual requirements in this dataset - 1) requiring CheckingAcct$\geq$0=True to be reachable only if CheckingAcct_exists is also True, and 2) requiring SavingsAcct$\geq$100=True to be reachable only if SavingsAcct_exists is True.

These lead to 31 partitions.

We present a list of all features and their corresponding feature-level constraints in Table 4 and list the non-separable joint constraints below it.

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

D.2 ADDITIONAL RESULTS

We report statistics on model performance statistics for all models Table 5. We include statistics for the LR model as a baseline.

Name	Type	LB	UB	Actionable	Sign	Joint Constraints	Partition ID
Age	$\mathbb{Z}$	19	75	No		1, 2	0
YearsAtResidence	$\mathbb{Z}$	0	7	Yes	+	1	0
YearsEmployed≥1	$\{0,1\}$	0	1	Yes	+	2	0
CheckingAcct_exists	$\{0,1\}$	0	1	Yes	+	3	30
CheckingAcct≥0	$\{0,1\}$	0	1	Yes	+	3	30
SavingsAcct_exists	$\{0,1\}$	0	1	Yes	+	4	31
SavingsAcct≥100	$\{0,1\}$	0	1	Yes	+	4	31
Male	$\{0,1\}$	0	1	No		–	1
Single	$\{0,1\}$	0	1	No		–	2
ForeignWorker	$\{0,1\}$	0	1	No		–	3
LiablePersons	$\mathbb{Z}$	1	2	No		–	4
Housing=Renter	$\{0,1\}$	0	1	No		–	5
Housing=Owner	$\{0,1\}$	0	1	No		–	6
Housing=Free	$\{0,1\}$	0	1	No		–	7
Job=Unskilled	$\{0,1\}$	0	1	No		–	8
Job=Skilled	$\{0,1\}$	0	1	No		–	9
Job=Management	$\{0,1\}$	0	1	No		–	10
CreditAmt≥1000K	$\{0,1\}$	0	1	No		–	11
CreditAmt≥2000K	$\{0,1\}$	0	1	No		–	12
CreditAmt≥5000K	$\{0,1\}$	0	1	No		–	13
CreditAmt≥10000K	$\{0,1\}$	0	1	No		–	14
LoanDuration≤6	$\{0,1\}$	0	1	No		–	15
LoanDuration≥12	$\{0,1\}$	0	1	No		–	16
LoanDuration≥24	$\{0,1\}$	0	1	No		–	17
LoanDuration≥36	$\{0,1\}$	0	1	No		–	18
LoanRate	$\mathbb{Z}$	1	4	No		–	19
HasGuarantor	$\{0,1\}$	0	1	Yes	+	–	20
LoanRequiredForBusiness	$\{0,1\}$	0	1	No		–	21
LoanRequiredForEducation	$\{0,1\}$	0	1	No		–	22
LoanRequiredForCar	$\{0,1\}$	0	1	No		–	23
LoanRequiredForHome	$\{0,1\}$	0	1	No		–	24
NoCreditHistory	$\{0,1\}$	0	1	No		–	25
HistoryOfLatePayments	$\{0,1\}$	0	1	No		–	26
HistoryOfDelinquency	$\{0,1\}$	0	1	No		–	27
HistoryOfBankInstallments	$\{0,1\}$	0	1	Yes	+	–	28
HistoryOfStoreInstallments	$\{0,1\}$	0	1	Yes	+	–	29

Table 4: Interaction Model for the processed german dataset. **Type** indicates the feature type ($\mathbb{Z}$ for integer, $\{0,1\}$ for binary). **LB**, **UB** are the lower and upper bounds for the feature. **Actionable** indicates whether the feature is globally actionable. Non-actionable features are highlighted. **Sign** indicates monotonicity constraints—whether feature can only increase (+) or decrease (-). **Joint Constraints** are a list non-separable constraint indices (listed below table) it is tied to (if any). **Partition ID** indicates which partition the feature belongs to.

–	XGB	
	Train	**Test**
AUC	0.819	0.7615
ECE	0.0%	10.0%
Error	21.9%	23.0%
n	800	200
$n_{\textbf{pos}}$	560	140
p	70.0%	70.0%
$n_{\textbf{clf_pos}}$	615	120
$n_{\textbf{clf_neg}}$	185	80

Table 5: Overview of model performance for XGB models for the german dataset

.

E Monotonicity Violations in Organ Transplants

E.1 Demonstration

Statistical models in organ transplant allocation [16] have recently attracted scrutiny for their potential to preclude access to transplants [1, 40]. Most notably, in 2018, the UK introduced a patient matching scheme for liver transplants using the Transplant Benefit Score (TBS). Attia et al. [2] found that TBS can predict *higher survival after the patient gets cancer*, which is biologically implausible and could result in the implicit de-prioritization of cancer patients [2]. We show how our machinery could have preemptively flagged such behavior during model development.

The TBS system comprises four Cox proportional hazard regression models. We focus on its *need* component which predicts survival without a transplant:

$$f(\boldsymbol{x}) = \sum_{t=1}^{T} \sum_{s \in \{\mathsf{c},\mathsf{nc}\}} \mathbb{I}[\boldsymbol{x}_{\mathsf{cancer}} = s] \cdot S_0^{\mathsf{c}}(t)^{\exp(\boldsymbol{\beta}^{\mathsf{c}} \cdot (\boldsymbol{x} - \boldsymbol{\mu}^{\mathsf{c}}))}$$

where $S_0^{\mathsf{c}}, S_0^{\mathsf{nc}} : \mathbb{N} \to [0, 1]$ for $t \in [T]$ for some $T \in \mathbb{N}$ are baseline hazard functions, and the vectors $\boldsymbol{\beta}^{\mathsf{c}}, \boldsymbol{\beta}^{\mathsf{nc}}$ and $\boldsymbol{\mu}^{\mathsf{c}}, \boldsymbol{\mu}^{\mathsf{nc}}$ are the model parameters and data normalizers for patients with cancer (c) and without (nc), respectively.

We apply our machinery to test if the model predicts higher survival after a patient gets cancer. This tests a monotonicity condition [5, 18] that can be violated because the TBS uses separate models for patients with and without cancer. We test responsiveness on a cohort of 1,000 patients without cancer from a clinical distribution in Attia et al. [2], where we ensure a uniform distribution of ages between 30 and 80 to be able to evaluate the effect of demographics on monotonicity violations. We define responsiveness $\rho(\boldsymbol{x})$ to capture how often the model violates monotonicity by setting target predictions as $\hat{Y}_{\boldsymbol{x}} = \{y \mid y < f(\boldsymbol{x})\}$, i.e., range of values that would *not* violate monotonicity. We test if responsiveness is below $\tau = 0.01$ at $\alpha = 0.05$ using $n = 30$ samples for each patient. To showcase our framework under multiple interaction models, we follow the *flagging* approach in Appendix A.2 by identifying the patients where the need model violates monotonicity under at least one interaction model after applying Holm's p-value adjustment. We use the family $\mathcal{I}$ of three models:

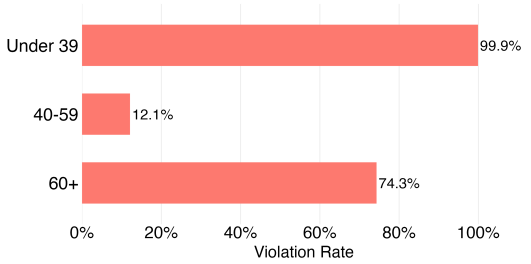


Figure 4: Proportion of predictions that violate monotonicity of predicted survival to changes in cancer status across a cohort of simulated patients, in any of the three plausible interaction sets. See Fig. 6 in the Appendix for a breakdown by interaction model.

Deterministic: Each action changes the cancer status by setting values of the `primary_disease`, `max_tumor_size`, and `tumor_number` features.

Random Effects: In addition to direct cancer features, we simulate plausible natural variation in lab values, such as levels of `bilirubin` and `sodium`.

Causal Effects: Instead of simulating variation in lab values, we assume that cancer has downstream causal effects on other variables, such as increasing the probability of `hospitalization` and `renal_dysfunction`.

In Fig. 4, we see that the *need* model violates monotonicity under at least one interaction model for a considerable number of patients. The effect varies significantly across age groups: the model violates monotonicity conditions for only 12% of patients between 40 and 59. In contrast, violation rates for patients under 39 and over 60 are 99.9% and 74% respectively.

When considering the violations in each interaction model separately, we see differences across the models (see Appendix for details). In the 40–59 age group, within the *random effects* model, the rate of violations is only around 10%, but rises to over 50% rate in the other models. The *flagging* procedure takes this variation into account. In our setting, we did not have enough power to consider most of the detected violations in the *deterministic* and *causal effects* models as significant after p-value adjustment, resulting in the final 12% violation rate in at least one of the models.

The detected violations implicitly de-prioritize cancer patients. Our tools flag this concrete safety issue at the system level, enabling model developers to test responsiveness individually for each patient, and generate test cases for iterative model development.

E.2 Components of the Transplant Benefit Score

Our study evaluates the transplant benefit score, which is publicly available at `https://github.com/SurgicalInformatics/transplantbenefit/`.

The TBS consists of four Cox regression models with the following corresponding survival functions:

$$f^{\mathsf{c}}_{\text{need}}(\boldsymbol{x}) = \sum_{t=1}^{T} S^{\mathsf{c}}_{0,\text{need}}(t)^{\exp(\boldsymbol{\beta}^{\mathsf{c}\top}_{\text{need}}(\boldsymbol{x}-\boldsymbol{\mu}^{\mathsf{c}}_{\text{need}}))},$$

$$f^{\mathsf{nc}}_{\text{need}}(\boldsymbol{x}) = \sum_{t=1}^{T} S^{\mathsf{nc}}_{0,\text{need}}(t)^{\exp(\boldsymbol{\beta}^{\mathsf{nc}\top}_{\text{need}}(\boldsymbol{x}-\boldsymbol{\mu}^{\mathsf{nc}}_{\text{need}}))}$$

$$f^{\mathsf{c}}_{\text{utility}}(\boldsymbol{x}) = \sum_{t=1}^{T} S^{\mathsf{c}}_{0,\text{utility}}(t)^{\exp(\boldsymbol{\beta}^{\mathsf{c}\top}_{\text{utility}}(\boldsymbol{x}-\boldsymbol{\mu}^{\mathsf{c}}_{\text{utility}}))},$$

$$f^{\mathsf{nc}}_{\text{utility}}(\boldsymbol{x}) = \sum_{t=1}^{T} S^{\mathsf{nc}}_{0,\text{utility}}(t)^{\exp(\boldsymbol{\beta}^{\mathsf{nc}\top}_{\text{utility}}(\boldsymbol{x}-\boldsymbol{\mu}^{\mathsf{nc}}_{\text{utility}}))}$$

where c and nc indicate models applied to patients with cancer and without, respectively, $S_{0,\cdot} : \mathbb{N} \to [0,1]$ for $t \in [T]$ for some $T \in \mathbb{N}$ are pre-defined baseline hazard functions, and the vectors β and μ are the corresponding model parameters and data normalizers, respectively. The final TBS score is computed as $f_{\text{TBS}}(\boldsymbol{x}) = f^{\boldsymbol{x}_{\mathsf{cancer}}}_{\text{utility}}(\boldsymbol{x}) - f^{\boldsymbol{x}_{\mathsf{cancer}}}_{\text{need}}(\boldsymbol{x})$.

E.3 Description of Dataset

We follow the methodology in the Attia et al. [2], who define a suitable probability distribution as part of a study on organ transplant prioritization for cancer patients. Attia et al. specify a probabilistic model to simulate patients by fitting a distribution to match a real cohort a liver transplant patients. We follow their approach to generate $n = 1{,}000$ simulated patients with $d = 32$ features. We start with the default patient case in this implementation to set the baseline characteristics in our cohort and modify certain variables as described below.

Static Variables We simulate the demographics as follows:

$$\boldsymbol{x}_{\mathsf{gender}} \sim \mathsf{Bernoulli}(0.5) \quad \text{where } 0 = \mathsf{man}, 1 = \mathsf{woman} \tag{16}$$

$$\boldsymbol{x}_{\mathsf{age}} \sim \mathsf{Unif}[\{30, 31, \ldots, 80\}]. \tag{17}$$

We also simulate other lab values as follows:

$$\boldsymbol{x}_{\mathsf{albumin}} \sim \mathsf{Unif}[30, 40] \tag{18}$$

$$\boldsymbol{x}_{\mathsf{potassium}} \sim \mathsf{Unif}[3.5, 5.0] \tag{19}$$

We detail the model for sampling other clinical variables next.

Liver Health Parameters We set up the following structural causal model (SCM) [43] for the liver parameters: bilirubin, sodium, international normalized ratio (INR), and creatinine. We use this probabilistic model both to generate the initial patient cohort, and to simulate the random effects due to natural variation in the reachable sets.

Let $\mathbf{U} = (U_{\mathsf{bili}}, U_{\mathsf{Na}}, U_{\mathsf{INR}}, U_{\mathsf{creat}})$ denote the vector of exogenous noise variables, and let $(\boldsymbol{x}_{\text{bili}}, \boldsymbol{x}_{\text{Na}}, \boldsymbol{x}_{\text{INR}}, \boldsymbol{x}_{\text{creat}})$ denote the vector of correlated endogenous variables representing the four liver parameters.

Exogenous Variables. We set $\mathbf{U} \sim \mathcal{N}(0, \mathbf{\Sigma})$ with:

$$\mathbf{\Sigma} = \begin{bmatrix} 1.0 & 0.447 & 0.320 & -0.257 \\ 0.447 & 1.0 & 0.370 & -0.043 \\ 0.320 & 0.370 & 1.0 & -0.091 \\ -0.257 & -0.043 & -0.091 & 1.0 \end{bmatrix} \tag{20}$$

Structural Equations. The endogenous variables are determined by the following structural equations:

$$\boldsymbol{x}_{\texttt{bili}} = \min(200, \max(15, \exp(0.5 \cdot U_{\texttt{bili}} + 3.5))) \tag{21}$$
$$\boldsymbol{x}_{\texttt{Na}} = \min(145, \max(125, 5 \cdot U_{\texttt{Na}} + 137)) \tag{22}$$
$$\boldsymbol{x}_{\texttt{INR}} = \min(2.4, \max(0.9, \exp(0.3 \cdot U_{\texttt{INR}} - 0.2) + 0.8)) \tag{23}$$
$$\boldsymbol{x}_{\texttt{creat}} = \min(200, \max(45, \exp(0.4 \cdot U_{\texttt{creat}} + 4.2))) \tag{24}$$

where:

- $\boldsymbol{x}_{\texttt{bili}}$ represents bilirubin levels (clipped to $[15, 200]$)
- $\boldsymbol{x}_{\texttt{Na}}$ represents sodium levels (clipped to $[125, 145]$)
- $\boldsymbol{x}_{\texttt{INR}}$ represents international normalized ratio (clipped to $[0.9, 2.4]$)
- $\boldsymbol{x}_{\texttt{creat}}$ represents creatinine levels (clipped to $[45, 200]$)

We choose the parameters to approximately match the reported statistics in a simulated cohort from Attia et al. [2]. We show the statistical properties of our generated cohort in Fig. 5.

E.4 Interaction Model

We detail the features and the considered interaction model in the table:

Name	Type	LB	UB	Actionable	Sign	Joint Constraints	Partition ID
rinpatient_tbs	$\{0,1\}$	0	1	No		–	3
rregistration_tbs	$\mathbb{Z}$	1	7	No		–	4
rwaiting_time_tbs	$\mathbb{Z}$	0	3650	No		–	5
rage_tbs	$\mathbb{Z}$	30	80	No		–	6
rgender_tbs	$\{0,1\}$	0	1	No		–	7
rdisease_primary_tbs	$\mathbb{Z}$	1	9	Yes		1, 2	1
rdisease_secondary_tbs	$\mathbb{Z}$	1	9	No		–	8
rdisease_tertiary_tbs	$\mathbb{Z}$	1	9	No		–	9
previous_tx_tbs	$\{0,1\}$	0	1	No		–	10
rprevious_surgery_tbs	$\{0,1\}$	0	1	No		–	11
rbilirubin_tbs	$\mathbb{Z}$	15	200	No		–	2
rinr_tbs	$\mathbb{R}$	0.9	2.4	No		–	2
rcreatinine_tbs	$\mathbb{Z}$	45	200	No		–	2
rrenal_tbs	$\{0,1\}$	0	1	No		–	12
rsodium_tbs	$\mathbb{Z}$	125	145	No		–	2
rpotassium_tbs	$\mathbb{R}$	3.5	5.0	No		–	13
ralbumin_tbs	$\mathbb{Z}$	30	40	No		–	14
rencephalopathy_tbs	$\{0,1\}$	0	1	No		–	15
rascites_tbs	$\{0,1\}$	0	1	No		–	16
rdiabetes_tbs	$\{0,1\}$	0	1	No		–	17
rmax_afp_tbs	$\mathbb{Z}$	0	1000	No		–	18
rtumour_number_tbs	{'0 or 1'*, '2', '3+'}			Yes		1, 2	1
rmax_tumour_size_tbs	$\mathbb{R}$	0	20	Yes		1, 2	1
dage_tbs	$\mathbb{Z}$	18	80	No		–	19
dcause_tbs	$\mathbb{Z}$	1	4	No		–	20
dbmi_tbs	$\mathbb{R}$	15	50	No		–	21
ddiabetes_tbs	$\mathbb{Z}$	1	3	No		–	22
dtype_tbs	$\{0,1\}$	0	1	No		–	23
bloodgroup_compatible_tbs	$\{0,1\}$	0	1	No		–	24
splittable_tbs	$\{0,1\}$	0	1	No		–	25

* This feature value is treated as no tumours if the primary disease does not indicate cancer, rdisease_primary_tbs $\neq 1$, and as one tumour otherwise.

1. IfThenConstraint: If rtumour_number_tbs $\in \{$'2','3+'$\}$, then rdisease_primary_tbs = 1 (cancer)
2. IfThenConstraint: If rdisease_primary_tbs = 1, then rmax_tumour_size_tbs > 0.

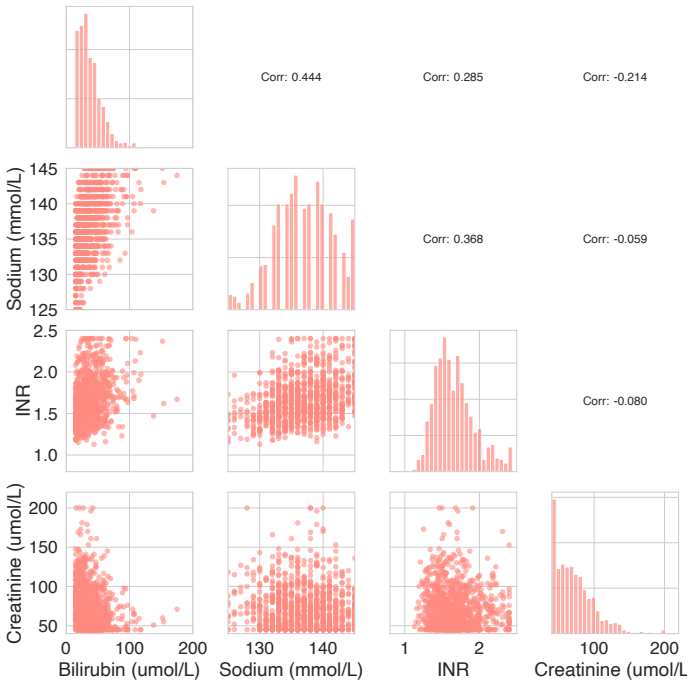


Figure 5: Pairwise relationships of the four liver parameter distributions according to our probabilistic model. These statistics are similar to those obtained by Attia et al. [2].

Next, we describe three interaction models constructed by setting different downstream effect models P.

Deterministic In this baseline interaction model, we assume that cancer diagnosis does not affect any non-cancer features. Under this model, counterfactual patients with cancer are identical to their non-cancer counterparts except for the cancer-specific features described above: primary disease diagnosis, tumor number, and maximum tumor size.

Random Effects We model natural variation in labs by applying directional perturbations to liver function parameters:

$$\boldsymbol{x}^{(1)}_{\texttt{bili}} = \texttt{clip}(\boldsymbol{x}^{(0)}_{\texttt{bili}} \cdot \gamma_{\texttt{bili}}, 15, 200), \quad \gamma_{\texttt{bili}} \sim \mathsf{Unif}[1.1, 1.5] \tag{25}$$

$$\boldsymbol{x}^{(1)}_{\texttt{INR}} = \texttt{clip}(\boldsymbol{x}^{(0)}_{\texttt{INR}} + \delta_{\texttt{INR}}, 0.9, 2.4), \quad \delta_{\texttt{INR}} \sim \mathsf{Unif}[0.1, 0.3] \tag{26}$$

$$\boldsymbol{x}^{(1)}_{\texttt{Na}} = \texttt{clip}(\boldsymbol{x}^{(0)}_{\texttt{Na}} - \delta_{\texttt{Na}}, 125, 145), \quad \delta_{\texttt{Na}} \sim \mathsf{Unif}[2, 5] \tag{27}$$

$$\boldsymbol{x}^{(1)}_{\texttt{AFP}} \sim \mathsf{Unif}[100, 1000] \tag{28}$$

Causal Effects Advanced liver disease with cancer may lead to clinical complications that affect other patient features. We model two such effects:

Renal dysfunction: With probability 0.3, patients develop renal complications, setting $\texttt{rrenal_tbs} = 1$ and increasing creatinine by a factor $\gamma_{\texttt{creat}} \sim \mathsf{Unif}[1.2, 1.8]$.

Hospitalization: With probability 0.4, patients require hospitalization, setting $\texttt{rinpatient_tbs} = 1$.

E.5 Additional Results

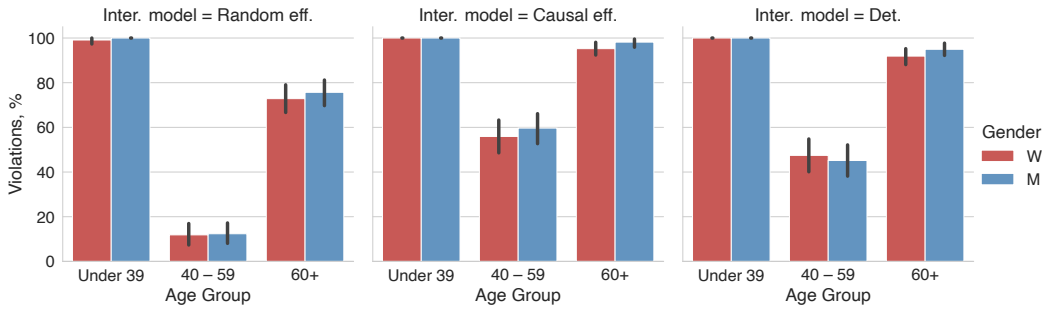


Figure 6: Monotonicity violations by interaction model, with a breakdown by age group and gender.

F PRECLUSION IN RECIDIVISM PREDICTION

F.1 DESCRIPTION OF DATASET

We work with a large sample of defendants from New York state derived from the "Recidivism of Prisoners Released in 1994" dataset released by the U.S. Department of Justice [49] and used in prior work [see e.g.,. 34, 59]. Our goal is to build a logistic regression model to predict whether prisoner i is rearrested within 3 years of release from prison ($y_i = 1$). Our processed dataset contains a total of $n = 29{,}400$ and a set of $d = 14$ features with varying degrees of mutability listed in Table 6.

F.2 INTERACTION MODEL

We consider a deterministic interaction model where each defendant can perform hypothetical actions that clear their criminal history (see below for detailed examples). In this case, our model contains a set of eight separable constraints for each feature which we use to ensure actionability, monotonicity, integrality, and bounds show in Table 6.

Name	Type	LB	UB	Actionable	Sign	Joint Constraints	Partition ID
drug_abuser	$\{0,1\}$	0	1	No		–	0
drug_treatment	$\{0,1\}$	0	1	Yes	+	–	1
alcohol_abuser	$\{0,1\}$	0	1	No		–	2
alcohol_treatment	$\{0,1\}$	0	1	Yes	+	–	3
edu_program_participation	$\{0,1\}$	0	1	Yes	+	–	4
voc_program_participation	$\{0,1\}$	0	1	Yes	+	–	5
age_at_release	$\mathbb{R}$	17.3	83.9	No		–	6
female	$\{0,1\}$	0	1	No		–	7
time_served	$\mathbb{Z}$	1	8	No		–	8
prior_arrests	$\mathbb{Z}$	0	30	Yes	–	1, 2, 3, 4	9
n_prior_felony	$\mathbb{Z}$	0	61	Yes	–	1	9
n_prior_misdemeanor	$\mathbb{Z}$	0	185	Yes	–	2	9
n_prior_general_violence	$\mathbb{Z}$	0	24	Yes	–	3	9
n_prior_prb_or_fine	$\mathbb{Z}$	0	34	Yes	–	4	9

Table 6: Interaction Model for the processed `rearrest_NY` dataset. **Type** indicates the feature type ($\mathbb{Z}$ for integer, $\{0,1\}$ for binary). **LB**, **UB** are the lower and upper bounds for the feature. **Actionable** indicates whether the feature is globally actionable. Non-actionable features are highlighted. **Sign** indicates monotonicity constraints—whether feature can only increase (+) or decrease (-). **Constraints** are a list non-separable constraint indices (listed below table) it is tied to (if any). **Partition ID** indicates which partition the feature belongs to.

1. IfThenConstraint: If `priorarrests` = 0.0, then `npriorfelony` = 0.0
2. IfThenConstraint: If `priorarrests` = 0.0, then `npriormisdemeanor` = 0.0
3. IfThenConstraint: If `priorarrests` = 0.0, then `npriorgeneralviolence` = 0.0
4. IfThenConstraint: If `priorarrests` = 0.0, then `npriorprborfine` = 0.0

F.3 ADDITIONAL RESULTS

This table includes additional model training and performance statistics.

Dataset	Interaction model	Model	Train AUC	Train error	Test AUC	Test error
rNYap	criminal_history_clearance	logreg	0.7287	0.2893	0.6901	0.3063

Table 7: Overview of model performance for the `rearrest_NY` dataset.

G Preventing Gaming in Content Moderation

G.1 Description of Dataset

We work with the `twitterbot` dataset, which was originally curated by Gilani et al. [12]. The dataset defines a binary classification task where we wish to predict if a user account on Twitter belongs to a *human* ($y_i = 1$) or a *bot* ($y_i = 0$). The dataset contains a total of $n = 3{,}431$ instances and $d = 20$ features that encode semantically meaningful characteristics about their interactions and login history—e.g., `age_of_account_in_days` for account age, `user_tweeted` for the number of user tweets, and `source_identity` for source of user interaction (mobile, web, etc.).

In this case, the dataset contains a limited number of features given that all features are not readily available or shared across accounts. We process the dataset to define a subset of additional features as follows: (1) we include additional dummies to indicate "missing" values for `num_tweets`, `num_retweets` and `num_replies`; (2) we binarize features by using a adding a thermometer encoding to `num_followers` and `age_of_accounts_in_days`, setting thresholds that reflect salient milestones for follows and membership history; (3) we multi-hot encoded `source_identity`.

G.2 Interaction Model

We specify an interaction model to specify the kinds of changes that each user can perform on the 20 features in the dataset. We present a list of all features and their corresponding feature-level constraints in Table 8 and list joint actionability constraints below. Our model only considers interactions on features that a user can change themselves and ignores changes on features that could change but cannot be manipulated.

Name	Type	LB	UB	Actionable	Sign	Joint Constraints	Partition ID
`followers≥1k`	$\{0,1\}$	0	1	No		4	0
`followers≥100k`	$\{0,1\}$	0	1	No		4	0
`followers≥1M`	$\{0,1\}$	0	1	No		4	0
`followers≥10M`	$\{0,1\}$	0	1	No		4	0
`num_tweets`	$\mathbb{Z}$	0	35000	Yes	+	1, 5	5
`no_tweets`	$\{0,1\}$	0	1	Yes	−	1	5
`urls_count`	$\mathbb{Z}$	0	13013	No		5	5
`num_retweets`	$\mathbb{Z}$	0	3000	Yes	+	2	6
`no_retweets`	$\{0,1\}$	0	1	Yes	−	2	6
`num_replies`	$\mathbb{Z}$	0	6991	Yes	+	3	7
`no_replies`	$\{0,1\}$	0	1	Yes	−	3	7
`age_of_account≥180_days`	$\{0,1\}$	0	1	Yes		–	1
`age_of_account≥365_days`	$\{0,1\}$	0	1	Yes		–	2
`age_of_account≥730_days`	$\{0,1\}$	0	1	Yes		–	3
`age_of_account≥1825_days`	$\{0,1\}$	0	1	Yes		–	4
`follower_friend_ratio`	$\mathbb{R}$	0.0	13364332.2	Yes	−	–	8
`source_web`	$\{0,1\}$	0	1	No		–	10
`source_mobile`	$\{0,1\}$	0	1	No		–	11
`source_app`	$\{0,1\}$	0	1	No		–	12
`source_news`	$\{0,1\}$	0	1	No		–	15

Table 8: Interaction Model for the processed `twitterbot` dataset. **Type** indicates the feature type ($\mathbb{Z}$ for integer, $\{0, 1\}$ for binary). **LB**, **UB** are the lower and upper bounds for the feature. **Actionable** indicates whether the feature is globally actionable. Non-actionable features are highlighted. **Sign** indicates monotonicity constraints—whether feature can only increase (+) or decrease (-). **Joint Constraints** are a list non-separable constraint indices (listed below table) it is tied to (if any). **Partition ID** indicates which partition the feature belongs to.

1. IfThenConstraint: If `notweets` = 0.0, then `numtweets` > 1.0
2. IfThenConstraint: If `noretweets` = 0.0, then `numretweets` > 1.0
3. IfThenConstraint: If `noreplies` = 0.0, then `numreplies` > 1.0
4. DirectionalLinkage: Actions on `numtweets` will induce to actions on ['`urlscount`']. Each unit change in `numtweets` leads to at least 1.00-unit change in `urlscount`

Procedure	Description	# Models	# Robust	Train	Test	Valid	Train	Test	Valid	Train	Test	Valid
Manual	Train Models with Immutable Features	370	370	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.531	0.570	0.581
Convex	Consider Responsiveness w.r.t Convex Perturbation Check	901	687	0.3%	0.0%	0.9%	56.2%	57.1%	55.9%	0.743	0.754	0.759
Exact	Evaluate Responsiveness w.r.t Exact Actions	901	76	9.6%	9.9%	9.3%	9.6%	9.9%	9.3%	0.722	0.727	0.734

Table 9: Full train, test, validation set results for the model with the highest validation AUC among *Considered* models: $\leq 10\%$ "Bot" predictions with certified responsiveness $\geq \tau = 0.05$. *% Responsive* show % of "Bot" predictions with responsiveness $\geq \tau = 0.05$ under the procedure's reachable set (*Perceived*) and the exact reachable set (*True*).

G.3 Additional Results

H LLM Safety Evaluation

H.1 Description of the Dataset and LLM Details

We evaluate four open-weight instruction-tuned LLMs accessed via Hugging Face: `OLMo-2-1124-7B-Instruct` [54], `Llama-3.1-8B-Instruct` [17], `Qwen2.5-7B-Instruct` [3], and `Phi-3.5-mini-instruct` [19]. The harmful-prompt corpus is `sorrybench` (release 202503), consisting of $n_{\texttt{prompts}} = 440$ prompts spanning 44 fine-grained safety categories [58].

Completions are produced with `vLLM` in `bfloat16` at temperature $T = 0.7$ and `max_tokens` $= 1024$, using each model's chat template. For every prompt under every interaction we draw multiple completions across multiple sampling seeds; together these realize the stochastic downstream effect ε that accounts for LLM sampling noise.

The classifier $f(\boldsymbol{x})$ is the `sorrybench` fine-tuned judge `ft-mistral-7b-instruct-v0.2-sorry-bench-202406`, queried at $T = 0$ with the official scoring template. The judge emits a single token in $\{0, 1\}$, where 1 denotes compliance ($\hat{Y} = \{\texttt{comply}\}$) and 0 refusal.

H.2 Interaction Models

We adopt four prompt-level perturbations from the `sorrybench` mutation suite as our interaction models I:

- I_{misspell}: writing-style mutation introducing systematic misspellings.
- I_{roleplay}: rewriting the request as a roleplay scenario.
- I_{chinese}, I_{french}: translation of the English prompt to Chinese (`zh-cn`) and French.

For a base prompt $\boldsymbol{x}$, the corresponding mutation corresponds to an action $\boldsymbol{a} \in A(\boldsymbol{x})$; the downstream effect ε is the LLM sampling distribution, as described before.

H.3 Statistical Test Details

With $\rho = \mathbb{E}_{\boldsymbol{x}\sim D}\mathbb{E}_{\boldsymbol{x}'\sim X_I(\boldsymbol{x})}\left[\mathbb{I}[f(\boldsymbol{x}') = \texttt{comply}]\right]$ and threshold $\tau = 40\%$, we test:

$$H_0 : \rho \geq \tau \qquad \text{vs.} \qquad H_1 : \rho < \tau.$$

For each (model, interaction) pair, we pool compliance indicators into a single sufficient statistic $k \sim \mathsf{Binom}(n, \rho)$ and report the exact one-sided Clopper-Pearson inversion test p-value

$$p \;=\; \Pr(K \leq k),$$

where $K \sim \mathsf{Binom}(n, \tau)$. The required sample size $n^\star$ is the smallest n such that the exact binomial test of H_0 has power at least $1 - \beta = 95\%$ at the alternative $\rho = \tau - \delta$ at level $\alpha = 0.05$. We anchor the alternative at each model's full-cohort base compliance rate, so $\delta = \tau - \rho^{(0)}$. This yields $n^\star = 79$ for OLMo ($\rho^{(0)} = 22.7\%$, $\delta = 17.3$ p.p.) and $n^\star = 159$ for Llama ($\rho^{(0)} = 27.7\%$, $\delta = 12.3$ p.p.), as reported in the main body.

I Omitted Formal Results

I.1 Proof of Remark 4

Proof. From the definition of the exact Binomial confidence interval, we have that:

$$\bar{\rho}_\alpha(n, \hat{\rho}) = \mathsf{B}_{1-\alpha}(n\hat{\rho} + 1,\ n - n\hat{\rho}) \tag{29}$$

provides a one-sided guarantee $\Pr(\rho(\boldsymbol{x}) \leq \bar{\rho}_\alpha(n, \hat{\rho})) \geq 1 - \alpha$.

The cumulative distribution of the Beta distribution is given by:

$$\mathrm{F}(x; a, b) = \frac{\mathrm{B}(x; a, b)}{\mathrm{B}(a, b)}$$

where $\mathrm{B}(x; a, b)$ is the incomplete beta function, defined as:

$$\mathrm{B}(x; a, b) = \int_0^x t^{a-1}(1-t)^{(b-1)}\, dt$$

and $\mathrm{B}(a, b) = \mathrm{B}(1; a, b)$.

Suppose $\hat{\rho}(\boldsymbol{x}) = 0$. Then our parameters for the beta distribution are $a = 1$, $b = n$. Hence,

$$\mathrm{F}(x; 1, n) = 1 - (1 - x)^n$$

Since the quantile function is the inverse of the CDF, we have

$$\mathsf{B}_{1-\alpha}(1, n) = 1 - \alpha^{\frac{1}{n}}$$

To reject H_0, we need $\mathsf{B}_{1-\alpha}(1, n) = 1 - \alpha^{\frac{1}{n}} < \tau$. By rearranging the inequality, we have

$$n > \frac{\ln(\alpha)}{\ln(1 - \tau)}$$

■