Personalized Generative Recourse via Conjugate Cost Priors and Inverted Sampling

Anonymous authors
Paper under double-blind review

Abstract

Algorithmic recourse aims to provide individuals with actionable changes that yield a favorable prediction while remaining in-distribution and minimizing the cost of change. Recent generative recourse methods balance these criteria more robustly than inference-time optimization, but they fix a single cost function at training time, whereas the cost of a change is user-dependent.

We present ACCORD[1], a generative diffusion model conditioned on the user's cost function, so that a single model can serve recourse tailored to each individual's notion of what a change costs. Rather than tilting a fixed generative model toward the user's cost at inference time, which is expensive and degenerates when that cost departs from the training one, ACCORD amortizes training over a flexible conjugate prior that we design. Our cost prior linearly combines random cost curves and expresses the weighted, asymmetric and range-dependent costs typical of recourse applications. Starting from only unpaired positive and negative instances, we exploit the conjugacy of the prior to invert the order in which (cost, target) training pairs are sampled, avoiding a search over positives for every drawn cost. We analyze the risk of the trained model and show that inverting the sampling lowers risk on costs well supported by the positive instances. Empirically, ACCORD outperforms both search-based recourse methods and diffusion guidance techniques under diverse user-specific costs.

1 Introduction

In algorithmic recourse, given an instance $x \in \mathcal{X}$ that received an unfavorable prediction from a classifier $h : \mathcal{X} \mapsto \{0, 1\}$, the goal is to suggest another instance $x^+ \in \mathcal{X}$ that would achieve the desired prediction from h, while being realistic and minimally different from x. Such a capability is useful (4; 19), in high-stakes ML decision making tasks in domains such as loan approval (18), risk assessment (11), and hiring screening (35). Effective recourse requires that the recommended profile x^+ be **valid** (achieves the desired outcome), **plausible** (lies within the realistic distribution of individuals with favorable outcomes), and **proximal** (minimizes the cost $C(x^+, x)$ of moving from x to x^+).

Early methods addressed these criteria by solving constrained optimization problems at inference time (42) or by searching the latent space of a generative model (17; 32; 10). More recently, generative recourse models (14) have been shown to be more robust at jointly satisfying all three criteria. These methods train a conditional generative model $\mathcal{P}_\theta(x^+ \mid x)$ using instance pairs that jointly encode these criteria, so that recourse at test time reduces to a single forward sample. However, a limitation of the generative approach is that the cost function $C(x^+, x)$ is fixed at training time. In practice, the cost of a recommended change is highly user-dependent: for a loan applicant, changing employment status may be far more costly than improving a credit utilization ratio, while for another applicant the priorities may be reversed. Regulatory frameworks like GDPR (12) reinforce this need by requiring explanations at the individual level (20). One method to handle inference-time costs is resampling from a pre-trained model using existing reward guidance methods (45; 40; 43; 26; 10). This approach is inefficient, however, whenever the inference-time preference differs substantially from a fixed default under which the model was trained.

[1]Amortized Conjugate-COst Recourse Diffusion

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.

In this paper we propose ACCORD, that makes cost a **conditioning variable** of the generative model by incorporating these three components: First, a space of user costs rich enough to cover inference-time preferences, represented by a random cost function C_u parameterized by a set U of random variables. Second, a prior $P(C_u)$ over that space, which a domain expert specifies once through interpretable hyper-parameters. Third, a recourse model $\mathcal{P}_\theta(x^+ \mid x, U)$ trained jointly across the prior, so that at inference a user supplies their own cost and receives recourse tailored to it from a single forward sample, with no retraining and no search.

Training such a model raises two challenges. (1) **Training inefficiency**: We are given only unpaired sets D_h^+ and D_h^-, so training triplets (x, x^+, U) must be constructed. The natural forward order of sampling $x \sim D_h^-$, cost C_U from the prior, and then x^+ from D_h^+ based on x, C_U is expensive. (2) **Coverage degeneracy:** As the number of features increases, or as U grows to make C_U more expressive, jointly sampled cost functions are increasingly likely to induce preferred recourse regions with little or no support in the finite positive dataset D_h^+. Such configurations provide weak or no useful training signal, resulting in inefficient utilization of model capacity.

We propose an inverted sampling order, made efficient by the closed-form marginals and posteriors our conjugate cost family admits. It also tackles coverage degeneracy, visiting each cost in proportion to how well D_h^+ covers it. Well-covered costs get more triplets and lower risk than under forward sampling; poorly covered ones get fewer, but there the training target is itself inaccurate by an amount no sampler or model can remove. The budget thus goes where the data supports it, and empirically inverted triplets train more cheaply while improving recourse quality on every dataset.

Contributions. (1) A cost family for personalized recourse. We propose a family of cost functions in which each feature carries an interpretable preference weight under a Gamma prior and a random level mapping selects among a few base cost curves. The family subsumes the cost functions used in the recourse literature and additionally expresses asymmetric and range-dependent preferences, while remaining conjugate to its prior. **(2) Preference-conditioned architecture.** We present a cost-conditional tabular diffusion model for $\mathcal{P}_\theta$ that accepts the user's cost as an inference-time input, so one model serves heterogeneous preferences. **(3) Efficient inverted triplet sampling.** We derive an algorithm for sampling training triplets that exploits the closed-form cost marginal to avoid inner-loop searches over D_h^+, and prove that it leaves every per-query target unchanged. **(4) Risk analysis.** We characterize the risk of the trained model, showing that inverted sampling lowers it on well-covered costs and that the coverage of a cost by D_h^+ places a floor on the risk of any model. **(5) Empirical evaluation.** ACCORD achieves the best combined validity–plausibility–proximity score across standard recourse benchmarks and sustains this advantage in the personalized setting where users specify non-uniform costs.

2 Related Work

Inference-time guidance in diffusion models. Guidance methods sample from a tilted target $p(x \mid c) \propto p(x)\exp(f(x, c))$ by steering a fixed pre-trained model, using gradients of an auxiliary objective (43; 3; 44), particles and search (40; 26; 15; 45), or a variational approximation to the posterior (28; 29; 25). The first family needs differentiable costs; the second degenerates when the reward is sharply peaked. ACCORD instead conditions on the cost, so no steering is needed at test time; we compare against the first two families in Sec. 4.

Search-based recourse. Wachter (42), GS (24), Dice (30) and Face (34) optimize a user cost per query and so handle inference-time preferences, but none models data density and all can return off-manifold instances (14); Face additionally needs D_h^+ at inference. PROBE (33) adds robustness to input noise but not density. FARE (9) guides MCTS with a learned policy, obtaining realism from action preconditions rather than from P_h^+.

Amortized and personalized recourse. Generative recourse (14) replaces search with one forward sample but fixes the cost at training time. PEAR (39) elicits a cost through Bayesian choice queries and then searches per user; elicitation (38) is complementary to our setting, where the cost is given. iCARMA (7), extending CARMA (27), amortizes *causal* recourse over per-user actionability: its hard and soft constraints correspond to the two

levels of our cost curves. However, it samples preferences independently of the data during training like in the *forward* sampler, and reports that tighter constraints degrade validity and plausibility. ACCORD explains that effect through coverage (Prop. 1) and exploits it, reallocating the triplet budget toward costs the data supports. A more detailed discussion of related work appears in App H

3 Cost-Aware Generative Recourse

The recourse problem. Let $\mathcal{X} \subseteq \mathbf{R}^d$ be a d-dimensional feature space consisting of continuous and categorical attributes denoting the space of user profiles, and let $\mathcal{Y} = \{0, 1\}$ denote the label space, where $y = 1$ is the favorable outcome. Let $P(\mathcal{X}, \mathcal{Y})$ denote their joint distribution. We are given a pre-trained classifier $h : \mathcal{X} \to \{0, 1\}$ that approximates $P(Y{=}1 \mid X)$. Let $P_h^+(X)$ and $P_h^-(X)$ denote the distributions $P(X|Y = 1)\delta(h(X) = 1)$ and $P(X|Y = 0)\delta(h(X) = 0)$ respectively. We are given i.i.d samples D_h^+ and D_h^- from each, and use $\hat{P}_h^+, \hat{P}_h^-$ to denote their corresponding empirical measure. During test time given an instance $x \sim P_h^-(X)$, and a user defined cost function $C_U : (\mathcal{X}, \mathcal{X}) \mapsto \mathbf{R}^+$, our goal is to find an x^+ that is representative of $P_h^+(x^+)$, and proximal i.e, of small cost $C_U(x^+, x)$.

Prior generative recourse methods (14) assume a fixed cost function C and train a model $\mathcal{P}_\theta(x^+ \mid x)$ to match the target $P_x(x^+) \propto P_h^+(x^+) \exp(-C(x^+, x))$ To train the parameters θ of this model, given only D_h^+ and D_h^- instead of paired examples, they construct (x^+, x) pairs by sampling $x \sim D_h^-$ and sampling $x^+ \sim D_h^+$ with probability proportional to reweighting factor $\exp(-C(x^+, x))$. Computing such pairs is computationally expensive but it is performed just once at the start of training in their method.

Overview of ACCORD. ACCORD extends $\mathcal{P}_\theta$ to support generation costs $C_U(x^+, x)$ specified at inference time. We design a space of user costs rich enough to cover inference-time preferences, represented by a random cost function C_u parameterized by a set U of random variables. A domain expert provides a prior $P(C_u)$ over that space through interpretable hyper-parameters. In Section 3.2, we present the design of the space of costs and priors. We condition the generation of recourse on both x and set U of random variables parameterizing C_U and train $\mathcal{P}_\theta$ by sampling triples from given $D_h^-, D_h^+, P(C_u)$. Section 3.1 describes our training method. As we provide flexibility to the end-users to specify their own cost function, they may ask for un-achievable recourse with poor support in P_h^+. We present how we handle such cases during inference in Section 3.3.2.

3.1 Training with Cost Distributions

We train a model $\mathcal{P}_\theta(x^+ \mid x, U)$ to match in expectation over $P(C_U)$ and P_h^- the ideal target

$$P_{x,U}(x^+) \;\propto\; P_h^+(x^+) \exp\big(-C_U(x^+, x)\big) \tag{1}$$

More specifically, our training objective is minimizing the cross entropy with the ideal $P_{x,U}(X)$ and the fitted model $\mathcal{P}_\theta(X \mid x, U)$ and reduces to

$$\hat{\theta} = \text{argmax}_\theta \mathbb{E}_{x\sim\hat{P}_h^-}\, \mathbb{E}_{U\sim P(C_U)}\, \mathbb{E}_{x^+\sim\hat{P}_h^+(x^+)\exp(-C_U(x^+,x))/Z_u}[\log \mathcal{P}_\theta(x^+ \mid x, U))] \tag{2}$$

Implementing the above sampling procedure is computationally challenging. Given D_h^- sampling x is easy, and we assume that $P(C_U)$ allows efficient sampling of U. The main challenge is sampling x^+ from the tilted distribution. Since we only have samples D_h^+ from P_h^+, we use importance sampling with weights proportional to $\exp\big(-C_U(x^+, x)\big)$ for each sampled (x, U). When D_h^+ is large, performing this reweighting for every sampled (x, U) in the inner training loop can be prohibitively expensive, requiring $O(|D_h^-| \cdot |D_h^+| \cdot n)$ time, where n is the number of cost functions sampled per x. We refer to this strategy, sampling (x, U) first and then x^+ by reweighting, as **forward sampling**.

3.1.1 Inverted Sampling for Efficiently Generating Training Triplets

We change the sampling order to $x \to x^+ \to U$ to avoid any U-dependent pass over D_h^+ as:

Step 1. Draw $x \sim \hat{P}_h^-(x)$, i.e., uniformly from D_h^-.

Step 2. Draw x^+ from the *cost-marginalized pairing kernel distribution:*

$$\tilde{Q}(x^+ \mid x) \;\propto\; \hat{P}_h^+(x^+)\, M(x^+, x), \qquad M(x^+, x) \;:=\; \mathbb{E}_{U\sim P(C_U)}\left[e^{-C_U(x^+,x)}\right] \tag{3}$$

The $M(x^+, x)$ can be interpreted as how compatible x^+, x are in expectation over the cost functions. This pass is *independent of* U and is shared across all triplets with the same x.

$$\textbf{Step 3.}\ \text{Draw } n \text{ samples of } U \text{ from posterior } \tilde{P}(U \mid x^+, x) = \frac{P(C_U)\exp\big(-C_U(x^+, x)\big)}{M(x^+, x)} \tag{4}$$

Inverted sampling reduces the time complexity of data construction from $O(|D_h^-| \cdot |D_h^+| \cdot n)$ to $O\big(|D_h^-| \cdot (|D_h^+| + n)\big)$ provided we can efficiently compute the marginals and posteriors above. We show how we achieve that in the next section. We theoretically analyze the impact of such inverted sampling in Section 3.3.

3.2 Designing Cost Function Distribution

Our goal is to design a parametric cost function along with a prior distribution $P(C_U)$ that satisfies the following properties: (1) Providing flexibility for expressing common types of cost functions during inference. (2) Intuitive for the domain expert to choose the hyper-parameters of the cost prior. (3) Efficiently trainable, which for above inverted triplet sampling implies that the marginals in Equation 3 and posterior in Equation 4 be efficiently computable. Satisfying all these preferences can be non-trivial. We develop our proposal in two versions.

3.2.1 Version 1: Linear cost preferences

A common method of handling user-preferences is by allowing users to linearly reweight a fixed dictionary of interpretable basis costs that encode how x^+ differs from x. We denote a basis cost as $\phi : \mathcal{X}\times\mathcal{X} \to \mathbf{R}^k$. Examples of such costs include: (1) $\phi_j^{\text{mag}}(x^+, x) = w_j\,|x_j^+ - x_j^-|$ where w_j normalizes each continuous feature by its population median absolute deviation. (2) $\phi_j^{\text{dir}}(x^+, x) = (x_j^- - x_j^+)^+$, and (3) $\phi_j^{\text{bin}}(x^+, x) = \mathbf{1}[x_j^+ \neq x_j^-]$. User preference is in term of linear weights $\lambda \in \mathbf{R}_+^k$ assigned to the cost map. The cost prior $P(C_U)$ is specified as distributions over λ. We place independent Gamma priors on each component: $\lambda_j \overset{\text{ind}}{\sim} \text{Gamma}(\alpha_j,\, \beta_j)$ with shape $\alpha_j > 0$ and rate $\beta_j > 0$. The hyper-parameters are interpretable: the ratio α_j/β_j measures the mean cost of one unit change of ϕ_j, and $\frac{1}{\alpha_j}$ measures the relative uncertainty of this estimate. This allows a domain expert to select these hyper-parameters. Taken together, this recourse divergence measure family is

$$C_U(x^+, x \mid \lambda) \;=\; \lambda^\top \phi(x^+, x), \qquad \lambda \in \mathbf{R}_+^k \sim \prod_{j=1}^{k} \text{Gamma}(\alpha_j,\ \beta_j)\,. \tag{5}$$

With this choice of C_U, U becomes $\{\lambda\}$ and the marginals and posterior of Eqns 3,4 have close forms since the Gamma density forms a conjugate prior with this divergence measure.

$$M(x^+, x) \;=\; \int_{\lambda\geq 0} \prod_j \frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)} \lambda_j^{\alpha_j - 1} e^{-\beta_j \lambda_j} e^{-\lambda_j^\top \phi_j(x^+, x)}\, d\lambda, \;=\; \prod_{j=1}^{k} \left(\frac{\beta_j}{\beta_j + \phi_j(x^+, x)}\right)^{\alpha_j} \tag{6}$$

$$\tilde{P}(\lambda \mid x^+, x) \;=\; \frac{P(\lambda)\,\exp\big(-\lambda^\top \phi(x^+, x)\big)}{M(x^+, x)} \;=\; \prod_j \text{Gamma}\big(\alpha_j,\ \beta_j + \phi_j(x^+, x)\big) \tag{7}$$

While the above family subsumes common recourse cost functions such as Weighted L1., it only lets the user set the price of change in each dimension, but not preferred ranges. For example, consider this cost where user has a tolerable window $[l_u, r_u]$ of change in L1, outside which he imposes a steeper penalty:

$$C_U(x_j^+, x_j) = |x_j^+ - x_j| \quad \textbf{if } l_u \leq x_j^+ - x_j \leq r_u, \textbf{else} \;\; B. \tag{8}$$

Such preferences are common in practical applications in finance. A linear combination of fixed basis costs can express Eq. 8 only by discretizing the change: fix a grid $t_0 < t_1 < \cdots < t_b$ spanning the range of $\Delta = x_j^+ - x_j$, define one basis per bin, $\phi_{j,b}(x^+, x) = \mathbf{1}\big[\Delta \in (t_{b-1}, t_b]\big]$,

and let the user encode the window through the weights: $\lambda_{j,b} \approx |t_b|$ for bins inside $[l_u, r_u]$ and $\lambda_{j,b} = B$ outside. This requires user to specify b λs, ties them together (the bins of one window must move jointly as the window moves), and burdens the user with a joint prior over a whole cost surface. Our final family instead makes the location of preferred ranges user-specific while keeping a single weight per feature.

3.2.2 Our Proposal: Linearly weighted Random Feature Maps

Instead of fixed cost functions, we allow each cost function to be a **random curve** $f_j(x^+; x)$ mapping each (x^+, x) pair to one of a small number of cost functions. A random cost curve $f_j(x^+; x)$ is defined in terms of the following fixed and random components:

1. A **fixed set of cost functions** $c_j^{(\ell)}(x^+; x) \mapsto \mathbf{R}^+$ $j = 1 \dots L_j$. Each cost function could vary smoothly with x^+ (e.g. $c^{(1)}(x^+; x) = |x_j^+ - x_j|$) or be a constant $c^{(\ell)}(x^+; x) = B$.
2. A **set of random variables** ω_j with a prior $P(\cdot; \Omega_j)$ with hyper-parameters Ω_j. We discuss the conditions required for admissible priors after Eqn 11.
3. A **level mapping function** $m_j(x^+; x, \omega_j) \in \{1, \dots, L_j\}$ that assigns each pair (x^+, x) to a cost function so that $f_j(x^+; x) = c_j^{(m_j(x^+; x, \omega_j))}(x^+; x)$.

We combine these with Gamma prior preferences λ_j to define our cost function family as:

$$C_U(x^+, x \mid \lambda, \omega) \;=\; \sum_j \lambda_j c_j^{(m_j(x^+; x, \omega_j))}(x^+; x),\; \lambda_j \sim \text{Gamma}(\alpha_j,\; \beta_j)\,,\; \omega_j \sim P(\cdot \mid \Omega_j). \tag{9}$$

Here set U extends to $\{\lambda, \omega\}$. We will show that this formulation allows C_U to subsume many recourse cost function classes proposed in the prior literature as special cases, while also supporting discrete preferred ranges. For example, the windowed L_1 cost function in Eq. 8 can be expressed as follows, where we use $\omega_{j1} - \omega_{j2}$ for l_u to ensure $l_u < r_u$:

$$\begin{aligned} &c_j^{(1)}(x^+; x) = |\Delta| \\ &c_j^{(2)}(x^+; x) = B \\ &P(\omega_{j1}; \Omega_{j1}) \sim \text{Exp}(\Omega_{j1}), \qquad \omega_{j2} \sim \text{Exp}(\Omega_{j2}), \qquad \Delta = x_j^+ - x_j \end{aligned} \qquad m_j\left(x^+; x, \omega_j\right) = \begin{cases} 1, & \text{if } \omega_{j1} - \omega_{j2} \le \Delta \le \omega_{j1} \\ 2, & \text{otherwise} \end{cases} \tag{10}$$

Now the preference prior is needed for a single λ_j per dimension. Likewise for categorical members, user's can choose their own support set. More examples in Appendix A.

Marginalizing cost. The marginals of Equation 3 now becomes

$$M(x^+, x) = \mathbb{E}_\omega\, \mathbb{E}_\lambda \left[e^{-\lambda C_U(x^+, x \mid \lambda, \omega)}\right] = \prod_{j=1}^d \sum_{\ell=1}^{L_j} \underbrace{P_{\omega_j}\big(m_j(x^+; x, \omega_j) = \ell\big)}_{g_j^{(\ell)}(x^+; x)} \underbrace{\left(\frac{\beta_j}{\beta_j + c_j^{\ell}(x^+; x)}\right)^{\alpha_j}}_{M_j^{\ell}(x^+, x)}$$

The term $g_j^{(\ell)}(x^+; x)$ denotes the probability that the cost is defined using the ℓth function under the distribution $P(\omega_j; \Omega_j)$. Note $\sum_{\ell=1}^{L_j} g_j^{(\ell)}(x^+; x) = 1$.

Sampling from cost posterior (Eqn 4) can be achieved in the following way:

$$\omega_j \sim P_{\Omega_j}(\omega \mid x^+, x) \propto P(\omega_j; \Omega_j)\, M_j^{(m_j(x^+; x, \omega_j))}, \quad \lambda_j \sim \Gamma\Big(\alpha_j,\; \beta_j + c_j^{(m_j(x^+; x, \omega_j))}(x^+; x)\Big) \tag{11}$$

Thus, the only requirement in the choice of $P(\omega; \Omega)$ are efficient computation of $g_j^{(\ell_j)}(x^+; x)$ and sampling from the posterior $P_{\Omega_j}(\omega \mid x^+, x)$. Our instantiations use exponential priors on window endpoints for continuous features and independent inclusion probabilities on preferred sets for categorical features, both of which yield closed forms (App. A). For the example in Eq. 10, suppressing the subscript j, the closed-form formulae are:

$$P_{\Omega_1}(\omega_1 \mid x, x^+) \propto \text{Exp}(\omega_1) \begin{cases} M^2, & \omega_1 < \Delta, \\ M^2 + e^{-\Omega_2(\omega_1 - \Delta)}(M^1 - M^2), & \omega_1 \ge \Delta \end{cases} \quad P_{\Omega_2}(\omega_2 \mid \omega_1, x, x^+) \propto \text{Exp}(\omega_2) \begin{cases} M^2, & \omega_2 < \omega_1 - \Delta, \\ M^1, & \omega_2 \ge \omega_1 - \Delta \end{cases}$$

These are mixtures of *exponential* distributions; proof and sampling procedure in App. A.1.

3.3 Cost Coverage: Analysis and Consequences

The extent to which a user cost can be served with finite positive data is a function of its *coverage*: the fraction of positive mass that survives the tilt e^{-C_U}. During training coverage

decides which sampled costs yield informative triplets, and at inference it decides whether the user's request is achievable at all. Both effects worsen as the number of features or the parameters in U grow, since the space of preferred ranges grows combinatorially while D_h^+ stays fixed. We analyze both theoretically, then discuss its training-time (Sec. 3.3.1) and inference-time (Sec. 3.3.2) consequences in practice.

The two samplers. Forward sampling (Eq. 2) draws a cost from the prior and then a positive example from the tilted distribution, so its training triplets follow

$$P_{\text{fwd}}(x^+, U \mid x) = \frac{\hat{P}_h^+(x^+)\, P(C_U)\, e^{-C_U(x^+,x)}}{\widehat{Z}(x,U)}, \qquad \widehat{Z}(x,U) := \mathbb{E}_{\hat{P}_h^+}\left[e^{-C_U(\cdot,x)}\right], \tag{12}$$

where $\widehat{Z}(x,U) \in (0,1]$ is*coverage* of cost u at x. Inverted sampling (Sec. 3.1.1) yields

$$P_{\text{inv}}(x^+, U \mid x) = \frac{\hat{P}_h^+(x^+)\, P(C_U)\, e^{-C_U(x^+,x)}}{\widehat{\bar{Z}}(x)}, \qquad \widehat{\bar{Z}}(x) := \mathbb{E}_{U\sim P(C_U)}\left[\widehat{Z}(x,U)\right], \tag{13}$$

whose normalizer is prior-average coverage. Each produces T triplets, from which $\theta \in \Theta$ is fit by minimizing $\mathcal{L}_\bullet(\theta) := \mathbb{E}_x\, \mathbb{E}_{(x^+,U)\sim P_\bullet(\cdot|x)}[-\log \mathcal{P}_\theta(x^+ \mid x, U)]$, $\bullet \in \{\text{fwd}, \text{inv}\}$.

Proposition 1. *Assume costs are finite and non-negative. Fix a query (x, U).*

(A) Exactness. *The two samplers induce the same conditional over x^+ and differ only in how often each cost is visited:*

$$P_{inv}(x^+ \mid x, U) = P_{fwd}(x^+ \mid x, U) = \widehat{P}_{x,U}(x^+), \quad \frac{P_{inv}(U \mid x)}{P_{fwd}(U \mid x)} = \rho(x,U) := \frac{\widehat{Z}(x,U)}{\widehat{\bar{Z}}(x)}. \tag{14}$$

Hence $\mathcal{L}_{\text{inv}}$ is $\mathcal{L}_{\text{fwd}}$ with each query reweighted by $\rho(x,U) > 0$. If $\mathcal{P}_\Theta$ is flexible enough that the per-query objectives admit a minimizer θ^0 that is common across all (x,U) on the support of the training joint, both samplers recover θ^0 as $T \to \infty$, even with D_h^+, D_h^- finite.

(B) Finite sample risk. *When trained on finite T triples, the fraction of triples allocated to u by the inverted sampler is $\rho(x,U)$ times more than forward. Consequently, for any per-query risk $r(n)$ that is non-increasing in the number n of triplets at the query, inverted sampling has risk no larger than forward sampling whenever $\widehat{Z}(x,U) \geq \widehat{\bar{Z}}(x)$.*

(C) Coverage floor. *For finite $\mathcal{X}$ and every D_h^+,*

$$\left\|\widehat{P}_{x,U} - P_{x,U}\right\|_2 \;\geq\; \widehat{\text{ESS}}(x,U)^{-1/2} - \|P_{x,U}\|_2, \quad \widehat{\text{ESS}}(x,U) \leq |D_h^+|\, \widehat{Z}(x,U)\, e^{\bar{C}(x,U)} \tag{15}$$

where $\bar{C}(x,u) := \mathbb{E}_{\widehat{P}_{x,u}}[C_U(\cdot,x)]$ is the mean cost under the plug-in target. The plug-in $\widehat{P}_{x,U}$ cannot adequately represent the ideal target $P_{x,U}$ when coverage and consequently ESS is small. Since both samplers train towards $\widehat{P}_{x,U}$, small cost coverage puts a floor on the risk by any model. (Proofs in App. C.)

3.3.1 Training-time consequence: informative coupling

Forward sampling draws U from the prior, independently of the data, so it may often samples costs that have little or no support in D_h^+. Such triplets are uninformative and under-utilize model capacity. Inverted sampling draws U from the cost-aware posterior (Eq. 4) instead; by Proposition 1(A) this leaves every per-query target unchanged and only reweights the prior by $\rho(x,u)$, the ratio of the cost's coverage to the prior-average coverage. Well-covered costs are therefore visited more often, and by Proposition 1(B) served with lower risk for the same triplet budget T. We verify the claim empirically in Sec. 4 (RQ2).

3.3.2 Inference-time consequence: feasibility

At inference, end-users may specify costs whose preferred regions have limited support under P_h^+. As per Proposition 1(C) this drives $\widehat{\text{ESS}}(x,U)$ small, and no model can meet the user's risk tolerance. The desirable behavior in this regime is to fall back to unpersonalized recourse rather than fabricating samples off the data manifold. This fallback is appropriate because

deciding *which* of the user's preferred ranges to relax, while deviating minimally from the rest, is itself a non-trivial joint optimization that the data cannot resolve. We propose to tell the user which regime they are in during inference by training a small auxiliary model $\mathcal{M}_\psi(x, U)$ to predict $\log \widehat{\mathrm{ESS}}(x, U)$ from queries visited during training (Algorithm 1). At inference, a request with $\mathcal{M}_\psi$ below a threshold is flagged as low-support, signaling that the returned recourse is close to an unpersonalized draw from P_h^+; above it, the recourse is both personalized and on-manifold. We evaluate this estimator in Sec. 4 (RQ5).

4 Implementation and Experiments

Our goal is to now empirically evaluate ACCORD to answer the following research questions

1. **RQ1**: How effective is cost conditioning compared to inference time steering methods in terms of overall recourse quality and adherence to user preferences?
2. **RQ2**: Performance comparison under good/bad coverage, validating theoretical claims.
3. **RQ3**: How do forward and inverted sampling strategies compare on speed and quality?
4. **RQ4**: How robust is ACCORD to prior mis-specification?
5. **RQ5**: How good is ESS estimator in rejecting unachievable cost targets (in App G.5).

Architecture of $\mathcal{P}_\theta$: a conditional tabular diffusion model. We implement $\mathcal{P}_\theta$ as a continuous-time diffusion model with an EDM-style (21) process for numerical features and an absorbing discrete diffusion process (2) for categorical features. The categorical diffusion uses a log-linear noise schedule, while the numerical diffusion follows the power-mean noise schedule of EDM. We use a Transformer-based denoiser following the architecture of TabDiff (37) The conditioning variable for $\mathcal{P}_\theta$ is a cost map over the domain of each feature, where the cost at each point represents the cost of changing x to that point. Further details about the diffusion model are provided in Appendix E.

Datasets. We evaluate across two data regimes: (i) a synthetic, 2D Gaussian Mixture Model (GMM) dataset with 10 modes each for x and x^+, which permits both gradient-based and derivative-free guidance and admits many closed-form operations for rigorous evaluation and (ii) three real-world datasets (Adult Income (5), COMPAS (1), HELOC (13)) with mixed continuous and discrete features, which restrict comparisons to derivative-free methods due to their non-differentiable cost functions. More details in Appendix D)

Baselines. Our baselines covering varied categories of inference-time guidance methods:

- **TFG** (43): A derivative-based guidance approach, applicable only to the GMM dataset.
- **TTS** (45) **& Search Variants (BFS, DFS)**: On GMM, we evaluate standard TTS, which combines local derivative-guided search with global particle sampling. For real-world datasets where gradients are unavailable, we evaluate its derivative-free tree-search variants, BFS and DFS (45), which supersede prior search guidance like DAS (22).
- **SVDD** (26): A particle search-based guidance method, included across both GMM and real-world regimes given its strong baseline performance.
- **RERD** (40): An evolutionary refinement method evaluated on real-world datasets, which iteratively injects noise followed by reward-guided denoising.
- **Forward**: ACCORD but with forward sampling of triples.
- **Traditional Recourse Methods**: On real-world datasets, we additionally compare against search-based non-generative recourse techniques: **Wachter** (42) (gradient-based cost minimization) and **PROBE** (33) (margin-maximizing recourse under noise).

Evaluation Metrics. We measure generated recourse instances x^+ on (1) **Preferred Range Fraction (PRF)**, the fraction lying inside user-preferred ranges. (2) **InDist$^+$**, how much in the P_h^+ distribution is x^+. For real-data we estimate on D_h^+ Local Outlier Factor (6). (3) **Validity (Val)**, the fraction that got favorable label from $h(x^+)$. (4) **Cost**, the cost $C_U(x, x^+)$ averaged over all test points and normalised by $\sum_j \mathbb{E}[\lambda_j]$. (5) **Rscore** measures how well generated x^+ matches the target twisted distribution. For synthetic data where the gold target is available, we measure it as as exp(−wasserstein distance). For real data lacking that we approximate it as Val $\times$ InDist$^+$ PRF exp(−Valid Cost)

Implementation Details. To ensure a fair comparison, all diffusion-based baselines fit $P(x^+ \mid x)$ using the same diffusion base model as ACCORD (excluding U conditioning),

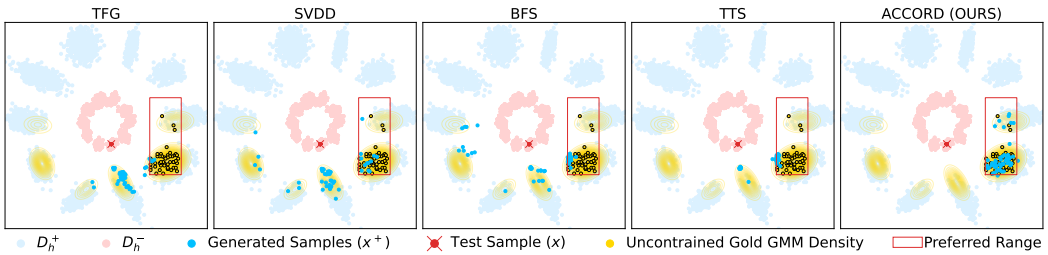


Figure 1: ACCORD against baselines on sampled recourse x^+ (cyan points) for a test instance x (red point) with $\lambda = (0.13, 0.39)$ and preferred range ω (red rectangle). ACCORD shows best conformance to user preference while staying around D_h^+ (sky blue points) and twisted target (yellow region).

trained on pairs generated following standard generative recourse procedures (14). For each dataset, we train a Random Forest (RF) classifier to serve as an oracle model and use the labels sampled from it for further training and evaluation of $\mathcal{P}_\theta$. This model is further used to evaluate the validity of the generated recourses. Additionally, we train another classifier model $h(\cdot)$ to categorize recourse points as x and x^+, similar to approach used in (14). Further implementation details, including model architectures, baseline configurations, and hyperparameters, are detailed in Appendix D F.

Choice of Hyperparameters for Cost Prior $P(C_U)$. User U is parameterized by $\{\lambda, \omega\}$, the cost prior $P(C_U)$ is governed by the prior distributions over λ and ω. We model λ using a Gamma prior. The parameter λ represents user preferences, where smaller values denote attributes users prefer to modify. Using identical prior hyperparameters across all features assumes identical preference for every attribute. To address this, we dynamically calibrate β for each mutable feature. In our experiments we use the cost function defined in equation 8, so ω is parametrized by $[l_u, r_u]$. We use exponential distribution as the prior over l_u and r_u whose parameters are set such that whole range of x^+ can be covered. For evaluation we sample U= λ, ω from the posterior as motivated by the coverage concerns in section 3.3.2.

4.1 RQ1: ACCORD AGAINST INFERENCE TIME STEERING

We qualitatively demonstrate the effectiveness of ACCORD in Fig. 1. ACCORD captures the true bounded target density within the Preferred Range far more faithfully than competing methods. While TTS generates samples near the target region, it exhibits a strong proximity bias toward the test instance x, concentrating its samples along the outer boundary and failing to cover the full distribution inside the preferred range. In contrast, TFG, SVDD, and BFS fail to adequately enforce the boundary constraints, frequently placing generated samples outside the Preferred Range box on unconstrained modes. ACCORD successfully places generated samples x^+ directly onto the high-density gold regions inside the preferred range, accurately capturing both primary and secondary modes of the bounded distribution.

In case of real-world datasets Table 9 shows that ACCORD achieves superior adherence to user preferences compared to inference-time steering methods. Unlike FACE and DICE, ACCORD not only respects user preferences but also achieves higher validity. Moreover, search-based methods like these require D_h^+ during inference, making them significantly slower.

4.2 RQ2: PERFORMANCE ACROSS DIFFERENT COST COVERAGE REGIONS

To evaluate whether inverted sampling effectively accounts for cost coverage, we evaluate its performance across two distinct regions of the cost space: (i) *Covered Region*, where the user-preferred ranges have sufficient support in P_h^+, and (ii) *Low-Coverage Region*, where the user-preferred ranges have little to no support in P_h^+.

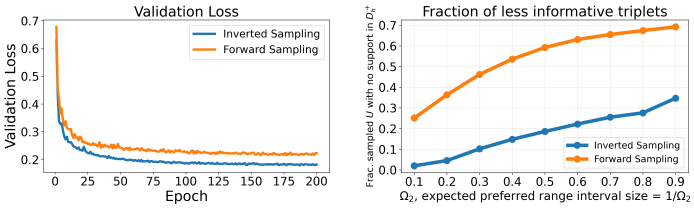
Table 2c shows that inverted sampling improves performance in the Covered Region by allocating the training budget in proportion to cost coverage. In contrast, in the Low-Coverage Region, FOW maintains higher validity than the other methods. This behavior can be explained by the higher fraction of low-coverage costs among the triplets sampled by forward sampling, as shown in Fig. 2b. Such samples provide a stronger signal to generate recourses that remain close to P_h^+, thereby preserving validity even when the specified user preferences have little support. However, these recourses are generally infeasible under the specified user constraints, which are flagged by the auxiliary model $\mathcal{M}_\psi$.

	ADULT				COMPAS				HELOC			
Method	**RScore↑**	**PRF↑**	**Val↑**	**Cost ↓**	**RScore↑**	**PRF↑**	**Val↑**	**Cost ↓**	**RScore↑**	**PRF↑**	**Val↑**	**Cost ↓**
RERD	0.91	0.18	0.91	16.61	1.32	0.62	0.97	0.87	1.03	0.08	0.95	9.30
BFS	1.11	0.39	0.90	12.75	1.57	0.88	0.98	0.30	1.17	0.22	0.96	7.88
DFS	0.96	0.24	0.91	16.56	1.47	0.77	0.98	0.51	1.07	0.13	0.95	8.88
SVDD	0.93	0.24	0.91	14.99	1.19	0.51	0.90	1.03	1.03	0.08	0.96	8.75
FACE	0.74	0.53	0.29	7.41	1.73	0.98	0.85	0.03	1.45	0.85	0.65	0.44
DICE	0.76	0.53	0.34	8.02	1.11	0.75	0.49	0.79	1.08	0.53	0.58	2.17
FOW	1.22	0.52	0.91	7.73	1.62	0.93	0.98	0.14	1.07	0.13	0.95	6.99
ACCORD	1.27	0.57	0.90	7.01	1.61	0.91	0.99	0.17	1.24	0.30	0.96	4.81

Table 1: Comparing ACCORD against baselines across datasets. ACCORD's recourse quality (Rscore) is within the top two across datasets. The FACE method searches for recourse from amongst training examples, which may not be practical both in terms of storage and time overheads during inference.

4.3 **RQ3**: Comparing forward and inverted sampling

The validation curves in Fig. 2a show that inverted sampling converges faster and consistently achieves lower validation loss than forward sampling. This improved convergence translates into better recourse quality, as reflected by the performance gains of inverted sampling in Table 2c. In Table 5 (in Appendix G.2), the timing breakdowns show that INV sampling while incurring a higher offline setup time, significantly reduces per-query time resulting in faster training (2× in ADULT dataset) as compared to FOW sampling.



(a) (b)

Data	Method	Cov.	Low-Cov.
ADULT	FOW	1.41	0.46
	INV	1.48	0.52
COMP	FOW	1.63	1.59
	INV	1.68	1.56
HEL	FOW	1.23	1.05
	INV	1.35	1.18
GMM	FOW	0.18	0.03
	INV	0.31	0.09

(c)

Figure 2: **(a)** Validation loss across epochs on GMM showing that inverted sampling converges faster and at a lower loss than Forward; **(b)** Fraction of uninformative triplets across preferred-range sizes $1/\Omega_2$ is significantly higher with Forward sampling; and **(c)** Recourse quality (Rscore) across Covered and Low-Coverage Regions.

4.4 **RQ4**: Robustness to prior mis-specification

We evaluate the sensitivity of ACCORD to the hyperparameters (α, β) of the preference prior distribution λ. As shown in Fig. 4 (Appendix G.3), performance remains highly stable across configurations, demonstrating that our approach is robust to the specific choice of prior hyperparameters. We also show the robustness of ACCORD when the training and prior distribution mismatch in 7 in App. G.4.

5 Conclusion and Future Work

In this work, we introduced ACCORD, a generative recourse model conditioned on the user's cost, so that one model serves heterogeneous preferences without retraining or search. Our cost family pairs interpretable per-feature weights with random cost curves, expressing weighted, asymmetric and range-dependent costs while staying conjugate to its prior; we exploit that conjugacy to invert the order in which training triplets are drawn, removing the inner search over positives. Our analysis shows that inverting leaves every per-query target unchanged, lowers risk on well-covered costs, and that the coverage of a cost by D_h^+ floors the risk of any model. Experiments show consistent gains over guidance-based and search-based baselines. Open directions include extending our framework to other cost families admitting conjugate structure; interfacing with users to elicit well-covered preference parameters, and accounting for causal dependence between features.

Use of Generative AI

We have not used generative AI tools for generating synthetic data, formulating mathematical claims or proofs, designing methodology or experiments, implementing methods, or interpreting results.

We used generative AI tools (a large language model) to assist with English-language grammar correction, paraphrasing, and text compression for improving clarity and conciseness in the writing. All AI-assisted text was reviewed and verified by the authors for accuracy and fidelity to the original intended meaning.

References

[1] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. How we analyzed the compas recidivism algorithm, 2016.

[2] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993. Curran Associates, Inc., 2021.

[3] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

[4] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023.

[5] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[6] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, page 93–104, New York, NY, USA, 2000. Association for Computing Machinery.

[7] Lena Marie Budde, Ayan Majumdar, Richard Uth, Markus Langer, and Isabel Valera. From universal to individualized actionability: Revisiting personalization in algorithmic recourse, 2026.

[8] Seung Hyun Cheon, Anneke Wernerfelt, Sorelle Friedler, and Berk Ustun. Feature responsiveness scores: Model-agnostic explanations for recourse. In *International Conference on Learning Representations*, volume 2025, pages 41382–41413, 2025.

[9] Giovanni De Toni, Bruno Lepri, and Andrea Passerini. Synthesizing explainable counterfactual policies for algorithmic recourse with program synthesis. *Machine Learning*, 112(4):1389–1409, 2023.

[10] Michael Downs, Jonathan L Chu, Yaniv Yacoby, Finale Doshi-Velez, and Weiwei Pan. Cruds: Counterfactual recourse using disentangled subspaces. *ICML WHI*, 2020:1–23, 2020.

[11] Doaa Abu Elyounes. Bail or jail? judicial versus algorithmic decision-making in the pretrial system. *CoLuM. SCi. & TECH. L. REv.*, 21:376, 2019.

[12] European Parliament and Council of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation), 2016. Accessed: [Insert Date].

[13] FICO. Fico xml challenge. `https://community.fico.com/s/explainable-machine-learning-challenge`, 2018. [Accessed 14-09-2024].

[14] Prateek Garg, Lokesh Nagalapatti, and Sunita Sarawagi. From search to sampling: Generative models for robust algorithmic recourse. In *The Thirteenth International Conference on Learning Representations*, 2025.

[15] Yingqing Guo, Yukang Yang, Hui Yuan, and Mengdi Wang. Training-free guidance beyond differentiability: Scalable path steering with tree search in diffusion and flow models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[17] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615*, 2019.

[18] Hari Prasad Josyula, Sunil Rajaram Landge, Srinivasa Rao Gunturu, Ketan Gupta, and T Kiruthiga. Leveraging ai and ml to automate financial predictions and recommendations. In *2024 2nd International Conference on Disruptive Technologies (ICDT)*, pages 452–457. IEEE, 2024.

[19] Margot E. Kaminski. "the right to explanation, explained". *btlj*, 2019.

[20] Amir-Hossein Karimi, Gilles Barthe, Bernhard Scholkopf, and Isabel Valera. A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 55:1 – 29, 2022.

[21] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[22] Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

[23] Avni Kothari, 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.

[24] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Inverse classification for comparison-based interpretability in machine learning. *arXiv preprint arXiv:1712.08443*, 2017.

[25] Jaewoo Lee, Minsu Kim, Sanghyeok Choi, Inhyuck Song, Sujin Yun, Hyeongyu Kang, Woocheol Shin, Taeyoung Yun, Kiyoung Om, and Jinkyoo Park. Diffusion alignment as variational expectation-maximization. In *The Fourteenth International Conference on Learning Representations*, 2026.

[26] Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gökcen Eraslan, Surag Nair, Tommaso Biancalani, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *CoRR*, abs/2408.08252, 2024.

[27] Ayan Majumdar and Isabel Valera. Carma: A practical framework to generate recommendations for causal algorithmic recourse at scale. In *Proceedings of the 2024 ACM conference on fairness, accountability, and transparency*, pages 1745–1762, 2024.

[28] Abbas Mammadov, Hyungjin Chung, and Jong Chul Ye. Amortized posterior sampling with diffusion prior distillation, 2025.

[29] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. *CoRR*, abs/1905.07697, 2019.

[31] Art B. Owen. *Monte Carlo Theory, Methods and Examples.* 2013. Available at `https://artowen.su.domains/mc/`.

[32] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of the web conference 2020*, pages 3126–3132, 2020.

[33] Martin Pawelczyk, Teresa Datta, Johannes van-den Heuvel, Gjergji Kasneci, and Himabindu Lakkaraju. Probabilistically robust recourse: Navigating the trade-offs between costs and robustness in algorithmic recourse. In *11th International Conference on Learning Representations (ICLR)*, 2023.

[34] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl Bie, and Peter Flach. Face: Feasible and actionable counterfactual explanations. pages 344–350, 02 2020.

[35] Candice Schumann, Jeffrey S. Foster, Nicholas Mattei, and John P. Dickerson. We need fairness and explainability in algorithmic hiring. In *Adaptive Agents and Multi-Agent Systems*, 2020.

[36] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[37] Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. Tabdiff: a mixed-type diffusion model for tabular data generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[38] Denise Tampieri, Giovanni De Toni, and Paolo Giudici. Personalized causal recourse: A human-in-the-loop approach, 2026.

[39] Giovanni De Toni, Paolo Viappiani, Stefano Teso, Bruno Lepri, and Andrea Passerini. Personalized algorithmic recourse with preference elicitation. *Transactions on Machine Learning Research*, 2024.

[40] Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, Shuiwang Ji, Sergey Levine, and Tommaso Biancalani. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and DNA design. In *Forty-second International Conference on Machine Learning*, 2025.

[41] A. W. van der Vaart. *Asymptotic Statistics.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 1998.

[42] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[43] Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified training-free guidance for diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[44] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model, 2023.

[45] XiangCheng Zhang, Haowei Lin, Haotian Ye, James Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search. In *The Fourteenth International Conference on Learning Representations*, 2026.

Contents of the Appendix

A Instantiations of the Cost Family **13**
A.1 Continuous features: two-sided window . . . 14
A.2 Continuous features: one-sided window for monotone features . . . 14
A.3 Categorical features . . . 15

B Algorithm **16**

C Assumptions and Proofs **16**
C.1 Proof of (A): exactness . . . 16
C.2 Proof of (B): allocation . . . 17
C.3 Proof of (C): coverage floor . . . 17

D Dataset and Metrics **18**
D.1 GMM . . . 18
D.2 Real Datasets . . . 19

E Architecture of $\mathcal{P}_\theta$ **19**

F Baseline Details **21**
F.1 GMMs . . . 21
F.2 Baselines on Real-world Datasets . . . 21

G Detailed Results & Ablations **23**
G.1 Experiments on Version 1 . . . 23
G.2 Comparison with Forward Sampling . . . 23
G.3 Sensitivity: Performance across Different α_0 and β_0 . . . 23
G.4 RQ4: Prior Mis-Specification Results . . . 24
G.5 RQ5: Quality of Feasibility Estimator, $\mathcal{M}_\psi(x, U)$ . . . 24
G.6 Extended Results Along with Standard Deviation . . . 25
G.7 Societal Impact of Work . . . 25

H Extended Related Work **25**
H.1 Inference-Time Guidance in Diffusion Models . . . 25
H.2 Search-Based Recourse . . . 26
H.3 Amortized and Personalized Recourse . . . 26
H.4 Feasibility, Verification, and Consumer Protection . . . 27

A Instantiations of the Cost Family

Recall from Sec. 3.2.2 that each feature j carries base costs $c_j^{(\ell)}$, random variables $\omega_j \sim P(\cdot; \Omega_j)$, and a level mapping $m_j(x^+; x, \omega_j)$. Everything factorizes over features, so per

feature the inverted sampler needs three quantities:

$$g_j^{(\ell)}(x^+;x) := P_{\omega_j}\big(m_j(x^+;x,\omega_j)=\ell\big), \quad M_j(x^+,x)=\sum_{\ell=1}^{L_j} g_j^{(\ell)}\, M_j^{(\ell)}, \quad M_j^{(\ell)} := \Big(\tfrac{\beta_j}{\beta_j + c_j^{(\ell)}(x^+;x)}\Big)^{\alpha_j}, \tag{16}$$

together with a sampler for the marginal posterior $P_{\Omega_j}(\omega_j \mid x^+, x) \propto P(\omega_j;\Omega_j)\, M_j^{(m_j(x^+;x,\omega_j))}$, obtained by integrating out λ_j. Given ω_j, the preference weight is the conjugate draw $\lambda_j \sim \mathrm{Gamma}\big(\alpha_j,\ \beta_j + c_j^{(m_j(x^+;x,\omega_j))}(x^+;x)\big)$. We now derive $g_j^{(\ell)}$ and the posterior samplers for our three instantiations, suppressing the subscript j throughout and writing $\Delta = x_j^+ - x_j$, $M^{(1)} = \big(\beta/(\beta+|\Delta|)\big)^{\alpha}$, $M^{(2)} = \big(\beta/(\beta+B)\big)^{\alpha}$; note $M^{(1)} > M^{(2)}$ whenever $|\Delta| < B$.

A.1 Continuous features: two-sided window

This is the instantiation of Eq. 10: $c^{(1)} = |\Delta|$, $c^{(2)} = B$, level 1 iff $\omega_1 - \omega_2 \le \Delta \le \omega_1$, with $\omega_1 \sim \mathrm{Exp}(\Omega_1)$, $\omega_2 \sim \mathrm{Exp}(\Omega_2)$. The window is $[\omega_1 - \omega_2, \omega_1]$; its lower end may be negative, so both increases and decreases can be tolerated. The hyper-parameters are interpretable: $1/\Omega_1$ is the mean tolerated increase and $1/\Omega_2$ the mean width of the window.

Level probability. Level 1 requires $\omega_1 \ge \Delta$ and $\omega_2 \ge \omega_1 - \Delta$, hence

$$g^{(1)}(\Delta) = \int_{\max(\Delta,0)}^{\infty} \Omega_1 e^{-\Omega_1 a}\, e^{-\Omega_2(a-\Delta)}\, da = \begin{cases} \dfrac{\Omega_1\, e^{-\Omega_1\Delta}}{\Omega_1+\Omega_2}, & \Delta \ge 0, \\[2ex] \dfrac{\Omega_1\, e^{\Omega_2\Delta}}{\Omega_1+\Omega_2}, & \Delta < 0, \end{cases} \qquad g^{(2)} = 1 - g^{(1)}, \tag{17}$$

and $M_j = g^{(1)}M^{(1)} + (1-g^{(1)})M^{(2)}$ by (16).

Posterior over ω_1. Marginalizing λ and ω_2,

$$P_{\Omega_1}(\omega_1 \mid x^+, x) \ \propto\ \Omega_1 e^{-\Omega_1\omega_1} \begin{cases} M^{(2)}, & \omega_1 < \Delta, \\ M^{(2)} + e^{-\Omega_2(\omega_1-\Delta)}\big(M^{(1)} - M^{(2)}\big), & \omega_1 \ge \Delta, \end{cases} \tag{18}$$

whose normalizer is exactly M_j. Integrating each term shows (18) is a two-component mixture of exponentials, sampled exactly as:

- with probability $M^{(2)}/M_j$: $\quad \omega_1 \sim \mathrm{Exp}(\Omega_1)$;
- with probability $g^{(1)}\big(M^{(1)} - M^{(2)}\big)/M_j$: $\quad \omega_1 = \max(\Delta, 0) + \mathrm{Exp}(\Omega_1 + \Omega_2)$.

Posterior over ω_2 given ω_1. If $\omega_1 < \Delta$ the pair is at level 2 for every ω_2, so $\omega_2 \sim \mathrm{Exp}(\Omega_2)$ (its prior). Otherwise set $h^* := \omega_1 - \Delta \ge 0$; then

$$P_{\Omega_2}(\omega_2 \mid \omega_1, x^+, x) \ \propto\ \Omega_2 e^{-\Omega_2\omega_2} \begin{cases} M^{(2)}, & \omega_2 < h^*, \\ M^{(1)}, & \omega_2 \ge h^*, \end{cases} \tag{19}$$

with normalizer $Z_2 = M^{(2)}\big(1 - e^{-\Omega_2 h^*}\big) + M^{(1)} e^{-\Omega_2 h^*}$. Sample the component, then the value:

- with probability $M^{(2)}\big(1 - e^{-\Omega_2 h^*}\big)/Z_2$: $\quad \omega_2 = -\frac{1}{\Omega_2}\log\big(1 - u\,(1 - e^{-\Omega_2 h^*})\big)$, $u \sim \mathrm{Unif}(0,1)$ (truncated $\mathrm{Exp}(\Omega_2)$ on $[0, h^*)$);
- with probability $M^{(1)} e^{-\Omega_2 h^*}/Z_2$: $\quad \omega_2 = h^* + \mathrm{Exp}(\Omega_2)$.

Finally $\lambda \sim \mathrm{Gamma}\big(\alpha,\ \beta + c^{(m(x^+;x,\omega))}\big)$. Multiplying the three stages and cancelling normalizers reproduces the joint posterior $\tilde{P}(U \mid x^+, x)$ of Eq. 4, so the resulting $(\omega_1, \omega_2, \lambda)$ is an exact draw.

A.2 Continuous features: one-sided window for monotone features

Features that realistically move in one direction (education, prior offences) use a one-sided window: $c^{(1)} = |\Delta|$ if $0 \le \Delta \le \omega_1$, $c^{(2)} = B$ otherwise, with the single variable

$\omega_1 \sim \mathrm{Exp}(\Omega_1)$. Then $g^{(1)}(\Delta) = e^{-\Omega_1 \Delta}\,\mathbf{1}[\Delta \geq 0]$, and for $\Delta \geq 0$ the posterior $P_{\Omega_1}(\omega_1 \mid x^+, x) \propto \Omega_1 e^{-\Omega_1 \omega_1}\big(M^{(2)}\mathbf{1}[\omega_1 < \Delta] + M^{(1)}\mathbf{1}[\omega_1 \geq \Delta]\big)$ is sampled exactly as in (19) with $h^* = \Delta$ and rate Ω_1; for $\Delta < 0$ the level is 2 regardless and ω_1 keeps its prior. A decrease-only window is obtained by negating Δ.

A.3 Categorical features

For a categorical feature with domain V and source category x_j, the random variable is the *preferred set* $\omega = A \subseteq V$ of acceptable destinations, with independent inclusion prior

$$\Pr[v \in A] = \pi_v := \frac{1}{1 + \eta_{x_j \to v}}, \qquad \eta_{x_j \to x_j} = 0, \tag{20}$$

where $\eta_{x_j \to v} \geq 0$ is an exclusion weight (large when few users tolerate the transition), and the source category is always included. The base costs are $c^{(1)} = c_{x_j \to x_j^+}$ (with $c_{x \to x} = 0$; by default $c_{x_j \to x_j^+} = \mathbf{1}[x_j^+ \neq x_j]$) and $c^{(2)} = B$, with level mapping $m = 1$ iff $x_j^+ \in A$. Hence

$$g^{(1)}(x^+; x) = \pi_{x_j^+}, \qquad M_j(x_j^+, x_j) = \begin{cases} 1, & x_j^+ = x_j, \\ \pi_{x_j^+} M^{(1)} + (1 - \pi_{x_j^+}) M^{(2)}, & x_j^+ \neq x_j, \end{cases} \tag{21}$$

with $M^{(1)} = \big(\beta/(\beta + c_{x_j \to x_j^+})\big)^{\alpha}$.

Posterior over A. The likelihood depends on A only through whether it contains x_j^+, so every other element keeps its prior inclusion probability and the draw reduces to $|V|$ independent Bernoulli trials:

$$\Pr\big[v \in A \mid x_j^+, x_j\big] = \begin{cases} \dfrac{\pi_v\, M^{(1)}}{\pi_v\, M^{(1)} + (1 - \pi_v)\, M^{(2)}}, & v = x_j^+, \\ \pi_v, & v \neq x_j^+. \end{cases} \tag{22}$$

Since $c_{x_j \to x_j^+} \ll B$ gives $M^{(1)} \gg M^{(2)}$, the observed destination is included with boosted probability, while $\eta_{x_j \to x_j} = 0$ keeps the source always allowed. Finally $\lambda \sim \mathrm{Gamma}\big(\alpha,\, \beta + c^{(m)}\big)$ as before.

B Algorithm

Algorithm 1 ACCORD: inverted-sampling training with feasibility scoring

Require: Datasets D_h^-, D_h^+; prior hyper-parameters $\{(\alpha_j, \beta_j, \Omega_j)\}_{j=1}^d$; candidate-set size v; batch size B; learning rate η; feasibility threshold τ

// **Precomputation (once, before training)**
for each $x \in D_h^-$ **do**
  $N(x) \leftarrow$ top-v candidates $x' \in D_h^+$ (nearest under the prior-mean cost)
  For each $x' \in N(x)$: compute and cache $M(x', x) = \prod_j \sum_\ell g_j^{(\ell)}(x'; x)\, M_j^{(\ell)}(x', x)$ (Eq. 3, closed forms in App. A)
end for

// **Training**
for each minibatch of $x \sim D_h^-$, repeated B times per x **do**
  Sample $x^+ \sim \tilde{Q}(\cdot \mid x) \propto M(\cdot, x)$ over $N(x)$ ▷ Step 2, cached weights
  For each j: sample $\omega_j \sim P_{\Omega_j}(\cdot \mid x^+, x)$, then $\lambda_j \sim \mathrm{Gamma}\big(\alpha_j,\, \beta_j + c_j^{(m_j(x^+;x,\omega_j))}(x^+; x)\big)$ ▷ Step 3, App. A
  $\kappa \leftarrow$ cost map of $U = (\lambda, \omega)$ over each feature's bins/categories (Sec. 4)
  Accumulate $\nabla_\theta \big[-\log \mathcal{P}_\theta(x^+ \mid x, \kappa)\big]$
  $\widehat{\mathrm{ESS}}(x, U) \leftarrow \big(\sum_{x' \in N(x)} w_{x'}\big)^2 / \sum_{x' \in N(x)} w_{x'}^2, \quad w_{x'} = e^{-C_U(x', x)}$
  Accumulate $\nabla_\psi \,\mathrm{loss}\big(\mathcal{M}_\psi(x, \kappa),\, \log \widehat{\mathrm{ESS}}(x, U)\big)$
  Update $\theta \leftarrow \theta - \eta \nabla_\theta, \quad \psi \leftarrow \psi - \eta \nabla_\psi$
end for

// **Inference** (user supplies x and $U = (\lambda, \omega)$)
$\kappa \leftarrow$ cost map of U; if $\mathcal{M}_\psi(x, \kappa) < \tau$, flag the request as low-support (generation defaults toward P_h^+, Sec. 3.3.2)
Return $x^+ \sim \mathcal{P}_\theta(\cdot \mid x, \kappa)$

C Assumptions and Proofs

Conventions. Throughout, (x, U) is a fixed query and $w := e^{-C_U(\cdot, x)}$. The ideal and plug-in targets are

$$P_{x,U} \;\propto\; P_h^+\, w, \qquad \widehat{P}_{x,U} \;\propto\; \hat{P}_h^+\, w. \tag{23}$$

Parts (A) and (B) hold conditionally on D_h^+, D_h^-; part (C) is a statement about the draw of $D_h^+ \sim P_h^+$.

Assumption 1 (Costs)**.** *For every query (x, U) the cost $C_U(\cdot, x)$ is finite and non-negative.*

Hence $w \in (0, 1]$, so $Z(x, U) := \mathbb{E}_{P_h^+}[w]$ and $\widehat{Z}(x, U) := \mathbb{E}_{\hat{P}_h^+}[w]$ lie in $(0, 1]$, and the targets in (23) and the weight $\rho(x, U) = \widehat{Z}(x, U)/\widehat{\bar{Z}}(x) > 0$ are well defined.

C.1 Proof of (A): exactness

Forward. Step 2 draws $U \sim P(C_U)$; step 3 draws x^+ from $\widehat{P}_{x,U}(x^+) = \hat{P}_h^+(x^+)\, w(x^+)/\widehat{Z}(x, U)$. Multiplying the two steps gives (12).

Inverted. Step 2 draws x^+ from $\tilde{Q}(x^+ \mid x) \propto \hat{P}_h^+(x^+)\, M(x^+, x)$ with $M(x^+, x) = \mathbb{E}_{U \sim P(C_U)}[e^{-C_U(x^+, x)}]$; exchanging the two expectations (Tonelli, the integrand being non-negative) its normalizer is $\mathbb{E}_U\, \mathbb{E}_{\hat{P}_h^+}[e^{-C_U}] = \mathbb{E}_U[\widehat{Z}(x, U)] = \widehat{\bar{Z}}(x)$. Step 3 draws U from

$\tilde{P}(U \mid x^+, x) = P(C_U)\, e^{-C_U(x^+,x)}/M(x^+,x)$. Multiplying, $M(x^+,x)$ cancels:

$$P_{\text{inv}}(x^+, U \mid x) = \frac{\hat{P}_h^+(x^+)\, M(x^+,x)}{\hat{\widehat{Z}}(x)} \cdot \frac{P(C_U)\, e^{-C_U(x^+,x)}}{M(x^+,x)} = \frac{\hat{P}_h^+(x^+)\, P(C_U)\, e^{-C_U(x^+,x)}}{\hat{\widehat{Z}}(x)}, \quad (24)$$

which is (13). The normalizer does not depend on U, so one pass over D_h^+ serves every cost drawn at x.

Comparison. Dividing (24) by (12),

$$\frac{P_{\text{inv}}(x^+, U \mid x)}{P_{\text{fwd}}(x^+, U \mid x)} = \frac{\widehat{Z}(x,U)}{\hat{\widehat{Z}}(x)} = \rho(x,U), \quad (25)$$

which is free of x^+. Two joint densities whose ratio is constant in one variable share the conditional in that variable, giving the first claim of (14); summing both joints over x^+ gives the second.

Reweighting and common minimizer. Write $\widehat{F}_{x,U}(\theta) := \mathbb{E}_{\widehat{P}_{x,U}}[-\log \mathcal{P}_\theta(\cdot \mid x, U)]$ for the per-query plug-in cross-entropy. By (14),

$$\mathcal{L}_{\text{fwd}}(\theta) = \mathbb{E}_x\, \mathbb{E}_{U\sim P(C_U)}\left[\widehat{F}_{x,U}(\theta)\right], \qquad \mathcal{L}_{\text{inv}}(\theta) = \mathbb{E}_x\, \mathbb{E}_{U\sim P(C_U)}\left[\rho(x,U)\, \widehat{F}_{x,U}(\theta)\right], \quad (26)$$

so $\mathcal{L}_{\text{inv}}$ is $\mathcal{L}_{\text{fwd}}$ with each query reweighted by $\rho(x,U) > 0$. Let $G_{x,U}(\theta) := \widehat{F}_{x,U}(\theta) - \min_{\theta'} \widehat{F}_{x,U}(\theta') \geq 0$. Each loss is a θ-independent constant plus a positive-weight average of $G_{x,U}(\theta)$. If some θ^0 has $G_{x,U}(\theta^0) = 0$ on the support of the training joint, both averages vanish there, so θ^0 minimizes both losses; conversely, a minimizer of either loss has zero weighted average of the non-negative G, hence $G_{x,U} = 0$ almost everywhere, and so minimizes the other. The minimizer sets coincide. As $T \to \infty$ with D_h^+, D_h^- fixed, each empirical loss converges to its population counterpart and both fitted models converge to this common set. □

C.2 Proof of (B): allocation

Define allocation $T_\bullet(x,U) = T\, P_\bullet(U \mid x)$, so the second identity in (14) gives $T_{\text{inv}}(x,U) = \rho(x,U)\, T_{\text{fwd}}(x,U)$. If r is non-increasing, then $\rho \geq 1$ implies $r(T_{\text{inv}}) = r(\rho\, T_{\text{fwd}}) \leq r(T_{\text{fwd}})$, and $\rho \leq 1$ implies the reverse; $\rho \geq 1$ is $\widehat{Z}(x,U) \geq \hat{\widehat{Z}}(x)$ by (14). For $r(n) = d_\Theta/n$, $r(T_{\text{inv}}) - r(T_{\text{fwd}}) = d_\Theta\big(1/(\rho\, T_{\text{fwd}}) - 1/T_{\text{fwd}}\big)$. □

The rate $r(n) = d_\Theta/n$ is the standard expected excess KL of a maximum likelihood fit with n observations from a well-specified model (41, Ch. 16), $2d_\Theta$ being the effective parameter count; the direction of (B) does not depend on it.

C.3 Proof of (C): coverage floor

Effective sample size. To estimate an expectation under a target P from n draws of a proposal Q one reweights by $w \propto dP/dQ$; the effective sample size

$$\text{ESS}_Q := \frac{n}{1 + \chi^2(P \,\|\, Q)} = \frac{n\, \mathbb{E}_Q[w]^2}{\mathbb{E}_Q[w^2]} \quad (27)$$

is the number of direct draws from P carrying the same information. Here $w = e^{-C_U}$ and P is the tilt of Q. With $n = |D_h^+|$ we use $\text{ESS}(x,U)$ ($Q = P_h^+$, population) and $\widehat{\text{ESS}}(x,U) = \big(\sum_i w_i\big)^2 / \sum_i w_i^2$ ($Q = \hat{P}_h^+$, plug-in), with $w_i = e^{-C_U(x_i^+,x)}$; the latter is computed in one pass over D_h^+.

Upper bound on $\widehat{\text{ESS}}$ (second claim of Eq. 14). Apply (27) with $Q = \hat{P}_h^+$, so $\widehat{\text{ESS}} = |D_h^+|\, \widehat{Z}^2 / \mathbb{E}_{\hat{P}_h^+}[w^2]$. Since $\widehat{P}_{x,U} = \hat{P}_h^+\, w/\widehat{Z}$, changing measure to the plug-in tilt gives $\mathbb{E}_{\hat{P}_h^+}[w^2] = \widehat{Z}\, \mathbb{E}_{\widehat{P}_{x,U}}[e^{-C_U}] \geq \widehat{Z}\, e^{-\bar{C}(x,U)}$ by Jensen's inequality for $t \mapsto e^{-t}$, where

$\bar{C}(x,U) := \mathbb{E}_{\widehat{P}_{x,U}}[C_U(\cdot,x)]$ is the mean cost under the plug-in target. Hence $\widehat{\mathrm{ESS}}(x,U) \le |D_h^+|\,\widehat{Z}(x,U)\,e^{\bar{C}(x,U)}$: low coverage forces a small effective sample size. (Both sides are unchanged when a constant is added to C_U. The same computation with $Q = P_h^+$ gives the population analogue $\mathrm{ESS}(x,U) \le |D_h^+|\,Z(x,U)\,e^{\mathbb{E}_{P_{x,U}}[C_U]}$.)

Deterministic floor (first claim of Eq. 14). The plug-in target places mass $\big(\sum_{i:x_i^+=x} w_i\big)/\sum_i w_i$ on each point x. Since a squared group sum dominates the group's sum of squares,

$$\big\|\widehat{P}_{x,U}\big\|_2^2 = \frac{\sum_x \big(\sum_{i:x_i^+=x} w_i\big)^2}{\big(\sum_i w_i\big)^2} \;\ge\; \frac{\sum_i w_i^2}{\big(\sum_i w_i\big)^2} \;=\; \frac{1}{\widehat{\mathrm{ESS}}(x,U)}, \tag{28}$$

and the reverse triangle inequality in ℓ_2 gives $\|\widehat{P}_{x,U} - P_{x,U}\|_2 \ge \|\widehat{P}_{x,U}\|_2 - \|P_{x,U}\|_2 \ge \widehat{\mathrm{ESS}}(x,U)^{-1/2} - \|P_{x,U}\|_2$, which holds for every D_h^+ with no expectation. Reading: $1/\|P_{x,U}\|_2^2$ is the effective support size of the ideal target, so the bound is positive exactly when the target is spread over more points than the reweighted data effectively contain.

Accuracy in expectation. The floor above is deterministic; in expectation over $D_h^+ \sim P_h^+$ the plug-in target is also accurate to the same order. $\mathbb{E}_{\widehat{P}_{x,U}}[f] = \sum_i w_i f(x_i^+)/\sum_i w_i$ is a self-normalized importance-sampling estimator of $\mu := \mathbb{E}_{P_{x,U}}[f]$, whose delta-method asymptotic variance is (31, f9.2)

$$\frac{1}{|D_h^+|} \cdot \frac{\mathbb{E}_{P_h^+}\big[w^2(f-\mu)^2\big]}{\mathbb{E}_{P_h^+}[w]^2} = \frac{\mathbb{E}_{Q_2}\big[(f-\mu)^2\big]}{\mathrm{ESS}(x,U)}, \qquad Q_2 \propto P_h^+\, w^2, \tag{29}$$

using $\mathbb{E}_{P_h^+}[w^2]/\,\mathbb{E}_{P_h^+}[w]^2 = 1 + \chi^2\big(P_{x,U}\,\|\,P_h^+\big)$. Applied to $f \equiv 1$ before normalization this gives $\mathrm{Var}[\widehat{Z}(x,U)]/Z(x,U)^2 = \mathrm{ESS}(x,U)^{-1} - |D_h^+|^{-1}$, so the plug-in diagnostic $\widehat{\mathrm{ESS}}$ has relative error $O(\mathrm{ESS}^{-1/2})$: it is accurate precisely when it is large and unreliable precisely when it is small, where a query is declined in any case.

Remark 1 (Combining (B) and (C))**.** *Under the parametric rate, the risk of sampler $\bullet$ at a query is approximately the sum of an amortization term $d_\Theta/T_\bullet(x,U)$, the cost of fitting the plug-in target from $T_\bullet$ triplets, and a representation term of order $d_\Theta/\mathrm{ESS}(x,U)$, the cost of using the plug-in target in place of the ideal one. Only the first differs between samplers, by (B); the second is shared and does not shrink with T, by (C). A risk tolerance ε is therefore attainable only when $\mathrm{ESS}(x,U) \gtrsim d_\Theta/\varepsilon$, whatever the sampler. On costs with below-average coverage, (B) charges inverted sampling $d_\Theta(\rho^{-1}-1)/T_{\mathrm{fwd}}$, while by the upper bound in Eq. 15 low coverage forces $\widehat{\mathrm{ESS}}$ small and hence the shared floor large as well; the penalty is therefore incurred exactly where neither sampler can serve the query. We state this as a reading of (B) and (C) rather than as a theorem, since it depends on the heuristic rate $r(n) = d_\Theta/n$.*

D Dataset and Metrics

D.1 GMM

We create the GMM dataset by using 10 modes for both x and x^+. The mean of the modes have been taken on a noisy circle of different radii. The covariance matrix for each mode has been sampled from a Wishart distribution with degree of freedom as 2 and scaled by a constant factor. We take the cost inside the user preferred region to be L_2 squared, as most baseline for GMM are derivative based, also owing to the simplicity of this setup, we replace our primary parameterization of $\mathcal{P}_\theta$ with a standard DDPM (16).

Distributional Metrics.
Wasserstein Distance (WD): Optimal transport distance between distributions based on Euclidean cost; lower is better.

NOTE: 500 test instances are sampled iid from the GMM of x. λs are also sampled iid from the prior.

GMM classifier. A simple classifier of polynomial degree 2(because of circular layout) is trained for evaluating validity score.

Metric	TFG	SVDD	BFS	TTS	Accord (ours)
Constraints	0.46 ± 0.35	0.66 ± 0.39	0.77 ± 0.35	0.76 ± 0.35	0.92 ± 0.15
Cost	55.45 ± 34.29	36.39 ± 38.23	24.77 ± 34.66	25.36 ± 34.88	13.17 ± 14.21
RScore	0.16 ± 0.14	0.23 ± 0.21	0.18 ± 0.16	0.18 ± 0.15	0.44 ± 0.25
Val	0.76 ± 0.18	0.86 ± 0.14	0.63 ± 0.36	0.57 ± 0.40	0.99 ± 0.03

Table 2: Full comparison of performance across different guidance methods. Results are reported as mean ± standard deviation.

D.2 Real Datasets

Table 3: Training Data Statistics

Dataset	#Features	#Categoricals	Immutable Features	#Positives	#Negatives
ADULT	13	7	`age`, `race`, `sex`	7,508	22,654
COMPAS	7	4	`race`, `sex`	3,764	865
HELOC	21	0	—	3,548	3,855

We train a RandomForestClassifier calibrated with CalibratedClassifierCV using isotonic calibration. Binary labels generated from RF Classifier are used for downstream tasks such as training ANN and $\mathcal{P}_\theta$.

D.2.1 Hyperparameters of ANN

We train fully connected binary classifiers with ReLU activations and a sigmoid output layer. The ANN is trained on labels sampled from the calibrated RF probabilities.

- Architecture: Fully connected network: $[d, 10, 10, 10, 1]$ where d is the input dimensionality
- Output activation: Sigmoid
- Optimizer: Adam
- Learning rate: 0.001
- Batch size: 64
- Epochs: 100
- Loss function: Weighted binary cross-entropy (balanced sample weights)

D.2.2 Test Dataset

We use a 90:10 train-test split and report the results for each dataset accordingly. Among the test samples, those classified by the ANN as negative instances are treated as x^-. The final reported numbers are computed over these selected samples for each dataset.

E Architecture of $\mathcal{P}_\theta$

$\mathcal{P}_\theta$ is a mixed-type diffusion model. Continuous columns are diffused with Gaussian noise following EDM (21), and categorical columns with an absorbing (masked) discrete diffusion following MDLM (36), similar to TabDiff (37). Both are driven by a shared time $t \in [0, 1]$. Before training, continuous columns are mapped to a standard normal with a quantile transformation fit on the positive instances, and mapped back after sampling. One reason to choose diffusion models is that inference-time guidance is easy to apply, which lets us compare against steering baselines on the same backbone.

Conditioning. $\mathcal{P}_\theta$ is conditioned on the cost map, which already encodes x, λ and the user's constraints. For every continuous column we discretise the range into 50 bins and store the cost of moving x to each bin, with B outside the user's window. For every categorical column we store the cost of each category, with B for the disallowed ones. The binning is only used for the condition; x^+ itself is generated in continuous space.

Denoiser. The denoiser consists of a Tokenizer, two Transformer encoders with an MLP between them, and a per-column decoder head. The Tokenizer projects each column of the noisy x^+ into a $d = 32$ dimensional token with a column-specific weight, and adds a learned column-identity embedding. The cost map of each column is projected to a d dimensional token in the same way and concatenated with the corresponding x^+ token along the channel dimension. A timestep token is prepended, so the Tokenizer outputs a tensor of shape (batch size, $1 +$ number of columns, $2d$). The first encoder models the correlations across columns. Its output is flattened and passed through the MLP, where the time embedding is added again, and reshaped back into tokens for the second encoder. Finally, the timestep token is dropped and each column is decoded by its own linear head: a scalar for continuous columns and logits over the categories for categorical columns.

Training objective. For continuous columns, we use the EDM denoising loss with standard preconditioning, and for categorical columns, we adopt the MDLM objective, which parameterizes the variational bound (negative ELBO):

$$\mathcal{L} = \underbrace{\mathbb{E}\Big[\tfrac{\sigma^2+\sigma_{\text{data}}^2}{(\sigma\,\sigma_{\text{data}})^2}\,\big\|D_\theta(x_t;\sigma,c) - x_0^{\text{num}}\big\|^2\Big]}_{\mathcal{L}_{\text{num}}} + w_{\text{cat}}\,\underbrace{\mathbb{E}\Big[\tfrac{\sigma'_{\text{cat}}(t)}{e^{\sigma_{\text{cat}}(t)}-1}\sum_j \mathbf{1}[x_t^j = \texttt{MASK}]\,\big(-\log p_\theta(x_0^j \mid x_t, c)\big)\Big]}_{\mathcal{L}_{\text{cat}}} \tag{30}$$

where c is the cost map, D_θ is the preconditioned denoiser and $w_{\text{cat}} = 2.5$.

- **Continuous Columns Noise Schedule (EDM):** Continuous variables follow the Elucidated Diffusion Model (EDM) noise schedule:

$$\sigma(t) = \left(\sigma_{\min}^{1/\rho} + t\left(\sigma_{\max}^{1/\rho} - \sigma_{\min}^{1/\rho}\right)\right)^\rho$$

  where $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$, $\rho = 7$, and data variance scale $\sigma_{\text{data}} = 1$.

- **Training Noise Sampling:** During training, noise levels σ are sampled from a log-normal distribution, $\ln\sigma \sim \mathcal{N}(-1.2, 1.2^2)$, and mapped back to the corresponding timestep t.

- **Categorical Columns Noise Schedule:** Categorical variables utilize a log-linear schedule evaluated at the same timestep t:

$$\sigma_{\text{cat}}(t) = -\log\left(1 - ((1 - \epsilon_{\max} - \epsilon_{\min})t + \epsilon_{\min})\right)$$

  with boundary bounds $\epsilon_{\max} = 10^{-3}$ and $\epsilon_{\min} = 10^{-5}$.

Sampling. We use 50 steps. Continuous columns follow the stochastic EDM sampler with a second-order (Heun) correction, and categorical columns follow the MDLM reverse process, starting from all columns masked.

Training.

- Optimizer: AdamW (learning rate 10^{-3}, no weight decay)
- LR scheduler: `ReduceLROnPlateau` on the training loss (factor = 0.9, patience = 50)
- EMA decay: 0.997, updated once per epoch
- Epochs: 4000
- Batch size: 4096 (Adult), 512 (COMPAS, HELOC)
- Model selection: from epoch 1000, every 200 epochs, we keep the checkpoint with the lowest EMA loss

Denoiser network.

- Transformer encoders: 2, each with 2 layers and 2 attention heads
- Token dimension: $d = 32$ per stream, $2d = 64$ after concatenating the condition
- Feed-forward dimension: 256
- MLP between encoders: hidden width 512 (1024 in the middle layers), SiLU activations
- Dropout: 0.1
- Architecture: Pre-Norm (LayerNorm before attention and feed-forward)

F Baseline Details

F.1 GMMs

All the methods share the same DDPM implementation with the following hyper-parameters for denoising network:

- Hidden layers: 3
- Activtion layers: SiLU
- Hidden dim: 128
- Time embedding: 128
- Denoising steps: 1000
- Noise Schedule: Linear
- Training Epochs: 200
- Batch Size: 512
- Optimizer: Adam
- Learning rate: 10^{-4}
- Dataset size: 3K negative samples are linked with 20 λs and 20 positive samples, resulting in a dataset size of 1200K triplets.
- Test instances: 1K instances are sampled iid from the GMM of x. λs are also sampled iid from the prior.
- $\alpha = 4$ and $\beta = 10$ for the results reported in 2
- $\bar{\lambda} = \frac{\alpha}{\beta} = 0.4$: the value for which RefM model is trained and used across baselines.
- Normalization: Standard
- p = 0.25, prior 1
- q = 0.5, prior 2

Following hyper-parameters have been used for the baseline after intensive selection for best results on baselines:

- TFG: rho=5×10^{-4}, mu=10^{-4}, iterSteps=2, recurSteps=1
- TTS: rho=4×10^{-5}, mu=10^{-5}, iterSteps=2, recurSteps=1, # particles=2
  NOTE: BFS(Increasing, Current, Multinomial) variant has been used in this paper.
- SVDD: # particles = 2, with increasing temperature for resampling.
  NOTE: Temperature is multiplied with scores.

F.2 Baselines on Real-world Datasets

RERD Performs S outer iterations of partial re-masking and reward-guided denoising. At each inner step, L candidate transitions are generated and selected via importance sampling (greedy when $\alpha_{\text{local}} = 0$). After all inner steps, a global resampling selects among N particles with temperature α_{global}.

- Particles per query (N): 4
- Outer iterations (S): 10
- IS candidates per step (L): 5
- Total timesteps (T): 100
- Noise level: 0.1 (re-masks $K = \lfloor 0.1 \times T \rfloor = 10$ steps)
- Local IS temperature (α_{local}): 0.0 (greedy argmax)
- Global selection temperature (α_{global}): 0.2

BFS Maintains a population of N particles and periodically resamples them based on reward scores (taken max till current timestep) using SSP (Srinivasan Sampling Process) resampling. Temperature increases geometrically with denoising progress. This is the best configuration reported in this paper.

- Particles (N): 4
- Timesteps (T): 16
- Eval start step: 4
- Eval step size: 4 (resampling at steps $\{4, 8, 12\}$)
- Temperature schedule: $\tau_t = (1 + \gamma)^{T-t-1} \cdot \tau_{\text{base}}$ (increasing)
- γ: 0.008
- τ_{base}: 1.0
- Scoring type: MAX (running maximum of scores)
- Resampling: SSP (Srinivasan Sampling Process)

DFS Follows a single trajectory with backtracking. If the reward score falls below a threshold, the trajectory backtracks by re-noising ΔT steps and re-denoising, up to a fixed budget.

- Timesteps (T): 16
- Backtrack budget: 4
- Backtrack depth (ΔT): $T/4 = 4$ steps
- Quality threshold (δ): 0.5

SVDD The implementation is same as that of GMMs.

- Number of Particles: 10

DiCE

- Search: genetic (`DiceGenetic`)
- Number of counterfactuals: 1
- Proximity / sparsity weight: 0.2 / 0.2, with both losses replaced by the user's cost $C_U(x, x^+)$ (continuous / categorical part)
- Post-hoc sparsity: Disabled
- Target: classifier probability > 0.5

FACE

- Graph: k-NN
- Fraction of data per graph: 0.1
- Immutable features: Edges restricted to instances with equal immutable values
- Candidate selection: Lowest user cost $C_U(x, x^+)$ among positive candidates

F.2.1 Compute Resources

For the synthetic GMM dataset, the training time is approximately 1 hour, while evaluation takes only a few seconds.

For the real dataset, both the baselines and our method require approximately 1.5 hours for training, and evaluation takes a few minutes. In contrast, the comparison method based on the forward mechanism requires around 4 hours of training, which we argue is computationally inefficient.

All experiments were conducted on a B200 GPU with 16 GB VRAM and 8 GB memory.

G Detailed Results & Ablations

G.1 Experiments on Version 1

	ADULT			COMPAS			HELOC		
Method	**InDist$^+$↑**	**Val↑**	**Cost ↓**	**InDist$^+$↑**	**Val↑**	**Cost ↓**	**InDist$^+$↑**	**Val↑**	**Cost ↓**
RERD	0.81 ± 0.39	0.91 ± 0.29	3.96 ± 1.88	0.72 ± 0.45	0.97 ± 0.18	1.41 ± 0.82	1.00 ± 0.03	0.95 ± 0.22	3.09 ± 0.99
BFS	0.79 ± 0.41	0.91 ± 0.28	4.39 ± 2.02	0.76 ± 0.42	0.96 ± 0.20	1.30 ± 0.75	1.00 ± 0.04	0.96 ± 0.19	3.15 ± 0.89
DFS	0.79 ± 0.41	0.91 ± 0.28	3.88 ± 1.91	0.71 ± 0.45	0.98 ± 0.15	1.21 ± 0.76	1.00 ± 0.04	0.96 ± 0.20	3.06 ± 0.99
SVDD	0.69 ± 0.46	0.92 ± 0.27	4.86 ± 2.38	0.73 ± 0.44	0.88 ± 0.32	1.29 ± 1.28	1.00 ± 0.05	0.97 ± 0.18	3.31 ± 0.93
ACCORDV1	0.80 ± 0.40	0.89 ± 0.31	3.23 ± 1.66	0.74 ± 0.44	0.99 ± 0.12	0.95 ± 0.62	1.00 ± 0.00	0.96 ± 0.20	2.39 ± 0.80

Table 4: Performance of ACCORD Version 1 against different baselines. Score = Val + InDist$^+$- Cost

G.2 Comparison with Forward Sampling

Sampling	**Setup | Query Time (s/ms) ↓**			**Training Time (hrs) ↓**		
	ADULT	**COMPAS**	**HELOC**	**ADULT**	**COMPAS**	**HELOC**
Forward	1.61 | 1.81	0.09 | 0.70	0.36 | 1.49	16.21	2.03	4.12
Inverted	14.53 | 0.79	0.31 | 0.56	1.63 | 0.86	8.21	1.47	2.84

Table 5: Setup, per-query and total training time for FOW and INV sampling (ACCORD). Setup time is pre-computation time; per-query is time taken to draw one sample.

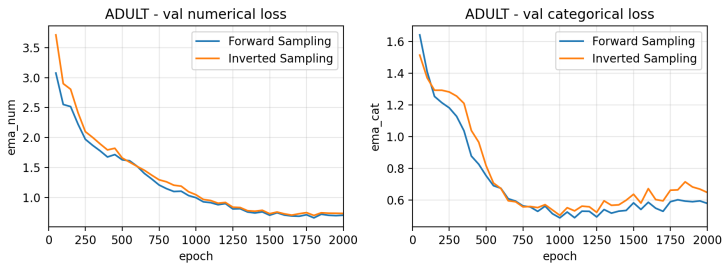


Figure 3: Convergence Plots of FOWand INVSampling on ADULT Dataset.

G.3 Sensitivity: Performance across Different α_0 and β_0

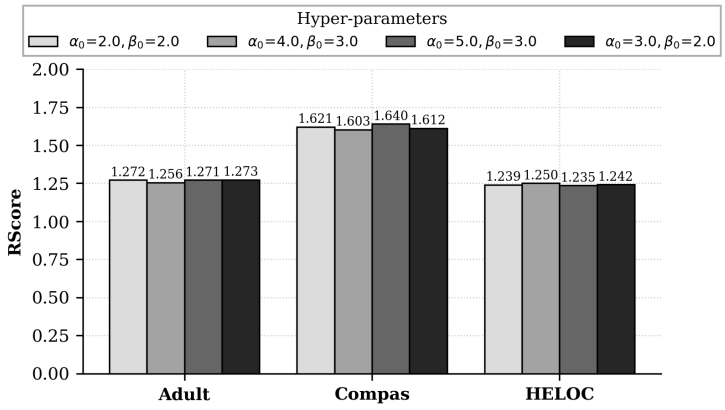


Figure 4: ACCORD performance on different α_0 and β_0.

Dataset	Parameters	RScore ↑	PRF ↑	Val ↑	Cost ↓
Adult	$\alpha_0 = 2.0, \beta_0 = 2.0$	1.27	0.57	0.90 ± 0.29	7.28 ± 12.08
	$\alpha_0 = 4.0, \beta_0 = 3.0$	1.26	0.56	0.90 ± 0.30	7.27 ± 12.10
	$\alpha_0 = 5.0, \beta_0 = 3.0$	1.27	0.56	0.90 ± 0.31	7.15 ± 12.00
	$\alpha_0 = 3.0, \beta_0 = 2.0$	1.27	0.57	0.90 ± 0.30	7.01 ± 11.93
COMPAS	$\alpha_0 = 2.0, \beta_0 = 2.0$	1.62	0.90	0.98 ± 0.13	0.21 ± 0.73
	$\alpha_0 = 4.0, \beta_0 = 3.0$	1.60	0.90	0.99 ± 0.12	0.19 ± 0.66
	$\alpha_0 = 5.0, \beta_0 = 3.0$	1.64	0.91	0.99 ± 0.09	0.16 ± 0.54
	$\alpha_0 = 3.0, \beta_0 = 2.0$	1.61	0.91	0.99 ± 0.10	0.17 ± 0.61
HELOC	$\alpha_0 = 2.0, \beta_0 = 2.0$	1.24	0.30	0.96 ± 0.18	4.76 ± 5.07
	$\alpha_0 = 4.0, \beta_0 = 3.0$	1.25	0.31	0.96 ± 0.21	4.71 ± 5.09
	$\alpha_0 = 5.0, \beta_0 = 3.0$	1.24	0.29	0.97 ± 0.17	4.83 ± 5.03
	$\alpha_0 = 3.0, \beta_0 = 2.0$	1.24	0.30	0.96 ± 0.20	4.81 ± 5.14

Table 6: Sensitivity of ACCORD to the Gamma prior hyper-parameters (α_0, β_0) used during training. All models are evaluated on the same test users, whose costs are drawn with the default $\alpha_0 = 3.0, \beta_0 = 2.0$. Metrics are computed over all test points and RScore is defined as in Table 9.

G.4 RQ4: Prior Mis-Specification Results

The RScore remains relatively stable across different cost priors, indicating that ACCORD is robust to prior mis-pecification. Despite changes in the assumed prior distribution, after training on Gamma distribution, its performance exhibits only minor variation, suggesting that the method does not strongly depend on accurately matching the prior to the underlying cost distribution. The changed prior were matched in expectation and variance to the training Gamma prior.

Gamma					Beta					LogNormal				
TFG	**SVDD**	**BFS**	**TTS**	**ACCORD**	**TFG**	**SVDD**	**BFS**	**TTS**	**ACCORD**	**TFG**	**SVDD**	**BFS**	**TTS**	**ACCORD**
0.20	0.28	0.22	0.23	0.43	0.19	0.28	0.22	0.22	0.41	0.20	0.28	0.22	0.22	0.45

Table 7: RScore comparison across different cost priors. Results are reported as mean values rounded to two decimal places.

G.5 RQ5: Quality of Feasibility Estimator, $\mathcal{M}_\psi(x, U)$

Coverage estimator M_ϕ. M_ϕ predicts, for a given cost map, how many training positives satisfy the user's constraints. It is a residual MLP: an input projection to 256 dimensions, three residual blocks (each two linear layers with LayerNorm and GELU), and a two-layer output head. To train it, we sample 50,000 queries by pairing training negatives with constraints, half drawn from the posterior (inverted sampling) and half from the prior (forward sampling), so that both well-supported and poorly-supported preferences are seen. The target is $\log(1 + n)$, where n is the number of training positives inside the constraints, computed exactly. Inputs and targets are standardised, and the model is trained with an MSE loss using AdamW (learning rate 10^{-3}, weight decay 10^{-4}, batch size 1024) for 50 epochs on a 90:10 train/validation split, keeping the checkpoint with the lowest validation loss. At test time, a query is marked as covered when the predicted count is at least 50.

Table 8 shows that $\mathcal{M}_\psi$ recovers the coverage regime of a query without access to D_h^+, reaching $F1 = 0.99$ on GMM, where coverage is determined by a known density, and 0.72–0.89 on the real datasets. Precision exceeds recall throughout (1.00 on COMPAS, 0.94 on Adult), so a raised flag is almost always correct and the errors are predominantly missed low-support queries rather than feasible requests wrongly declined, since a missed flag still returns a recourse near P_h^+ (Sec. 3.3.2). Recall is weakest on Adult (0.60) and HELOC (0.65), the two datasets with the largest feature spaces, where preferred regions are defined by more interacting windows and coverage is correspondingly harder to predict from (x, U) alone.

Dataset	# Tests	Prec.	Rec.	F1
GMM	5000	1.00	0.99	0.99
Adult	8834	0.94	0.60	0.73
COMPAS	659	1.00	0.81	0.89
HELOC	1364	0.81	0.65	0.72

Table 8: Quality of the ESS estimator on test datasets.

G.6 Extended Results Along with Standard Deviation

	ADULT					COMPAS					HELOC				
Method	**RScore↑**	**PRF↑**	**InDist⁺↑**	**Val↑**	**Cost ↓**	**RScore↑**	**PRF↑**	**InDist⁺↑**	**Val↑**	**Cost ↓**	**RScore↑**	**PRF↑**	**InDist⁺↑**	**Val↑**	**Cost ↓**
RERD	0.91	0.18	0.81 ± 0.39	0.91 ± 0.29	16.61 ± 13.91	1.32	0.62	0.74 ± 0.44	0.97 ± 0.17	0.87 ± 1.45	1.03	0.08	1.00 ± 0.04	0.95 ± 0.21	9.30 ± 6.83
BFS	1.11	0.39	0.81 ± 0.39	0.90 ± 0.30	12.75 ± 14.08	1.57	0.88	0.72 ± 0.45	0.98 ± 0.15	0.30 ± 0.92	1.17	0.22	1.00 ± 0.00	0.96 ± 0.20	7.88 ± 7.08
DFS	0.96	0.24	0.80 ± 0.40	0.91 ± 0.29	16.56 ± 14.14	1.47	0.77	0.73 ± 0.44	0.98 ± 0.13	0.51 ± 1.09	1.07	0.13	1.00 ± 0.03	0.95 ± 0.22	8.88 ± 6.97
SVDD	0.93	0.24	0.77 ± 0.42	0.91 ± 0.28	14.99 ± 14.10	1.19	0.51	0.76 ± 0.43	0.90 ± 0.30	1.03 ± 1.41	1.03	0.08	1.00 ± 0.00	0.96 ± 0.20	8.75 ± 6.56
FACE	0.74	0.53	0.76 ± 0.43	0.29 ± 0.45	7.41 ± 11.88	**1.73**	0.98	0.88 ± 0.32	0.85 ± 0.36	0.03 ± 0.20	**1.45**	0.85	1.00 ± 0.04	0.65 ± 0.48	0.44 ± 1.08
DICE	0.76	0.53	0.70 ± 0.46	0.34 ± 0.48	8.02 ± 12.63	1.11	0.75	0.74 ± 0.44	0.49 ± 0.50	0.79 ± 3.81	1.08	0.53	0.98 ± 0.15	0.58 ± 0.49	2.17 ± 3.11
FOW	1.22	0.52	0.79 ± 0.41	0.91 ± 0.28	7.73 ± 12.45	1.62	0.93	0.73 ± 0.45	0.98 ± 0.12	0.14 ± 0.50	1.07	0.13	1.00 ± 0.07	0.95 ± 0.22	6.99 ± 5.91
ACCORD	**1.27**	0.57	0.81 ± 0.39	0.90 ± 0.30	7.01 ± 11.93	1.61	0.91	0.72 ± 0.45	0.99 ± 0.10	0.17 ± 0.61	1.24	0.30	1.00 ± 0.00	0.96 ± 0.20	4.81 ± 5.14

Table 9: Comparing ACCORD against baselines across datasets.

G.7 Societal Impact of Work

Our work has potential positive societal impacts by enabling more personalized and user-aligned algorithmic recourse. Existing recourse methods typically assume a single fixed notion of cost, which may not reflect the practical constraints, resources, or preferences of different individuals. By conditioning recourse generation on user-specific cost preferences, our approach allows individuals to receive actionable recommendations that better align with their circumstances, thereby increasing user agency and the practical usefulness of recourse systems. This flexibility may improve accessibility and fairness in high-stakes decision-making settings such as lending, hiring, or education, where different users may face different barriers to changing specific features.

At the same time, our work also carries potential negative societal impacts and limitations. As with prior recourse methods, generated recourse does not guarantee that an individual who follows the suggested actions will ultimately receive a favorable outcome, since the underlying classifier, institutional policies, or real-world environment may change over time. In addition, recourse systems may inadvertently encourage users to optimize for model predictions rather than meaningful long-term outcomes, and poorly calibrated user preferences could lead to unrealistic or undesirable recommendations. Finally, because our method relies on historical data distributions, biases present in the data may propagate into generated recourse suggestions, potentially leading to unequal quality or feasibility of recommendations across different populations.

H Extended Related Work

H.1 Inference-Time Guidance in Diffusion Models

Diffusion models support sampling from a tilted target $p(x \mid c) \propto p(x)\exp(f(x,c))$, where f encodes a task objective and, in our setting, $c = (x, U)$. Existing methods fall into three families.

Score-derivative-based guidance. These methods perturb the denoising direction with gradients of an auxiliary objective (43; 3), covering classifier, classifier-free, and energy-based variants (44). They are effective in continuous domains but require the cost to be differentiable, incur a gradient evaluation per step, are sensitive to the guidance scale (3), and do not extend naturally to the mixed discrete-continuous profiles typical of recourse. We compare against TFG (43) on our synthetic benchmark, where costs are smooth.

Particle and search-based guidance. These maintain several candidate trajectories and select among them. SMC-style methods propagate and reweight particles under an arbitrary reward (40); others rank partial samples with a learned value function (26) or search the trajectory tree (15), with hybrids combining particle updates and gradients (45). They accommodate non-differentiable costs, which makes them the natural baselines on our real datasets, but they pay repeated reward evaluations per query and suffer particle degeneracy when the reward is sharply peaked — precisely the regime a strongly expressed user preference induces.

Variational methods. These cast conditioning as minimizing a KL divergence to the target posterior, either per instance or amortized, and have been applied to inverse problems (28; 29) and to reward alignment via variational EM (25). ACCORD is amortized in the same spirit, but amortizes over a *family* of rewards indexed by U rather than fitting one approximation per reward, which is what lets a single model serve heterogeneous users.

Common to all three is that they adapt a model trained under one objective at inference time. When the user's cost departs substantially from the training cost, the model's support and the tilted target become misaligned, importance weights degenerate, and quality collapses. This is the failure mode ACCORD avoids by conditioning on the cost.

H.2 Search-Based Recourse

Search-based methods solve a constrained optimization per query and therefore accept a user-specified cost at inference without retraining. Wachter (42) minimizes a differentiable objective trading off classifier score against an ℓ_p cost; GS (24) performs a growing sphere search; DICE (30) returns a diverse set of counterfactuals subject to proximity and sparsity; and FACE (34) constructs a k-NN graph over the data and returns the lowest-cost accepted point reachable from x. The recurring weakness is that none of them models P_h^+: Wachter and DICE readily return instances off the data manifold, and both tend to stop just past the decision boundary, making the result fragile to input perturbations. (33) addresses the latter by maximizing the margin under an invalidation target, but still does not account for density. FACE is density-aware only through graph connectivity, requires access to D_h^+ at inference, and returns nothing when x is disconnected from every accepted point. (14) document this plausibility gap even when the cost is fixed and all hyper-parameters are tuned for it, which motivates the generative approach we build on. FARE (9) sits between these and learned methods: it trains RL policies that act as priors for a Monte Carlo tree search over a hand-specified action set, and obtains realism from action preconditions and discretized argument sets rather than from any distributional constraint.

H.3 Amortized and Personalized Recourse

Generative recourse. (14) replace test-time search with a conditional generative model $\mathcal{P}_\theta(x^+ \mid x)$ trained on pairs that jointly encode validity, plausibility and proximity, so recourse reduces to one forward sample. The cost is fixed when the training pairs are built, so a deployment serving users with heterogeneous costs would need one model per cost. Removing that restriction is the starting point of our work.

Preference elicitation. PEAR (39) makes the user's cost the object of inference: it maintains a Bayesian posterior over per-feature cost weights, queries the user with small choice sets chosen to maximize the Expected Utility of Selection, and generates interventions with W-FARE, a cost-aware extension of FARE whose reward rewards short *and* cheap interventions under the current posterior. Its cost model is notable for allowing dependence between action costs through a DAG, so that acting on one feature changes the cost of acting on another. Follow-up work (38) studies the interaction design of such elicitation. This line is complementary to ours: elicitation supplies a cost, whereas ACCORD takes one as given and asks how to generate from it efficiently and in-distribution. The two compose — a posterior over U could be fed directly to our conditional model — and because ACCORD samples in one forward pass, it is well suited to producing the candidate sets that choice-based elicitation

consumes. We also note that PEAR obtains plausibility only from its action space and preconditions, and inherits the density blindness discussed above.

Amortized causal recourse. CARMA (27) amortizes causal recourse by training two networks over a pre-trained causal normalizing flow: a mask network learning a Bernoulli distribution over which features to intervene on, and an action network predicting intervention magnitudes, jointly optimized for cost with a validity hinge loss. iCARMA (7) extends CARMA to *individual actionability*, which it splits into hard constraints (which features a user can act on) and soft constraints (per-feature difficulty weights elicited as ranks or Likert scores), enforcing the former with an individualized mask and the latter through weighted losses and priors, and adding a plausibility regularizer. It is the closest prior work to ACCORD: both train a single model amortized over a space of user costs, and its hard/soft split corresponds to the two levels of our cost curves — the large constant B outside a preferred range, and the Gamma-distributed weight λ_j inside it. Three differences matter. First, iCARMA samples actionability scores uniformly and independently of the data during training, which is exactly the *forward* sampler we compare against; it has no prior over costs to marginalize and hence no closed-form pairing kernel. Second, its central empirical finding — that tighter individual constraints degrade validity and plausibility for both an amortized solver and a non-amortized causal oracle — is precisely the phenomenon our Proposition 1(C) explains, through the coverage of the cost by D_h^+ and the effective sample size it induces; we further use that quantity to reallocate the training budget rather than merely to diagnose. Third, iCARMA operates within a causal framework and can account for downstream effects of an intervention, which ACCORD does not; our independent per-feature priors are the price of conjugacy, and lifting them is the main direction for future work.

H.4 Feasibility, Verification, and Consumer Protection

A parallel literature asks not how to generate recourse but whether recourse exists. (23) verify recourse by enumerating reachable sets under actionability constraints, certifying when no feasible action changes a prediction. (8) build on this to define a feature's *responsiveness score*, the fraction of feasible single-feature actions that attain the target prediction, and show that the SHAP- and LIME-based explanations used to satisfy adverse-action requirements routinely highlight features that cannot deliver recourse; on HELOC, over a fifth of denied applicants have predictions no feasible action can change, and they argue such individuals should be flagged rather than sent an explanation. Their analysis also documents that responsiveness is frequently non-monotonic and counterintuitive — one applicant is approved only by reducing credit utilization into a disjoint interval, another only by *decreasing* monthly income — which is direct evidence from lending data that a cost family must express directions and ranges, not merely per-feature prices. This motivates our move beyond linearly weighted fixed bases (Sec. 3.2). Their constraint taxonomy (immutability, monotonicity, integrality, encoding validity, logical and causal implication) also delineates what our family covers: immutability through a large λ_j, monotonicity through a one-sided window, encoding validity through categorical preferred sets, and integrality through binning; logical and causal implications span features and are outside the independent-prior family. Finally, this line establishes the infeasibility they diagnose as *model-side* — no reachable point flips h — in contrast to our *data-side* coverage floor, where the user's preferred region is attainable and valid but unpopulated in D_h^+. Both argue that a deployed system should recognize when it cannot serve a request; our $\widehat{\mathrm{ESS}}$ estimator (Sec. 3.3.2) is the analogue of their triage rule for the data-side case.