An Evaluation of Cost Functions for Algorithmic Recourse

Eoin M. Kenny **Allan Anzagira** **Tom Bewley** **Freddy Lecue** **Manuela Veloso**
Trustworthy AI Centre of Excellence, AI Research, J.P. Morgan Chase

Abstract

Algorithmic recourse is a field concerned with offering actionable recommendations to individuals who have received adverse outcomes from automated systems. Most recourse algorithms assume access to a cost function, which quantifies the effort involved in following these suggestions. However, to date, there has been no serious benchmarking of these functions both from a computational and human perspective. In this paper, we propose four metrics to evaluate whether currently popular cost functions in recourse satisfy the minimal requirements for meaningful distance calculations. In addition, we also propose extensions to current approaches using large-language models (LLMs) as surrogate human labellers, which are prompted with a cost-based desiderata. Experiments revealed that methods focused on the Bradley-Terry model perform best, but only when scaled up with our proposed LLM extensions, which would be the recommended choice in practice. We expect our insights to help practitioners in training and designing appropriate cost functions in the future.

1 INTRODUCTION

Algorithmic recourse has emerged as potentially one of the most useful areas of interpretable machine learning (Karimi et al., 2020), with major financial institutions working to deploy the technology to tens of millions of end users adversely affected by credit decisions. The field focuses on generating actionable counterfactual recommendations to users who were treated unfavorably by automated systems, with the canonical example being a rejected bank loan application, and what actions could be taken by a user to have it accepted in the future (Karimi et al., 2022; Keane et al., 2021). In such a scenario, a cost function is needed to quantify how much effort a recommendation would take; for example, increasing a down-payment requires considerably more effort for someone with low savings compared to someone with high savings.[1] However, to date, there has been no large-scale quantitative evaluation of the functions in the literature, nor a scalable solution to designing and training them in an effective manner (Tominaga et al., 2024), which has seriously hindered the field.

Typically in recourse, a cost function is taken as some variant of an L_p norm on the feature space (Keane et al., 2021; Karimi et al., 2022). For example, an L_0 norm assigns higher cost to recourse recommendations that change more features, but ignores other factors such as how much they are changed by. A weighted L_1 or L_2 norm can incorporate feature weighting information (Karimi et al., 2020), but not dependencies between features, or difference in cost across the feature's spectrum. This could be added manually, but it is challenging to formalize and prone to error (Kenny et al., 2024; Buchanan and Smith, 1988). Moreover, recent research by Tominaga et al. (2024) has shown these hard-coded cost functions only marginally correlate with people's willingness to act upon them. With this in mind, research has gravitated towards more theoretically grounded methods of learning these functions using the Bradley-Terry model, which can learn such functions through pairwise preferences (Rawal and Lakkaraju, 2024), but no comprehensive investigation has been conducted to date. With this in mind, our contributions are as follows:

1. We propose a desiderata based on four dimensions and prior research by which to evaluate cost functions.

2. We conduct the first large-scale evaluation of cost

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

[1] Separately, it is worth noting that the field has branched out to consider positive outcomes with gain functions and semifactual recourse (Kenny and Huang, 2024).

functions, discovering the Bradley-Terry model is best, but only when scaled up with large-language model (LLM) labeling which has been prompted to consider our desiderata at a high level (see Figure 1).

3. We propose a framework for learning cost functions based on Bradley-Terry which takes advantage of LLM labeling and naturally models feature dependencies for the first time in the literature.

4. We conduct the first ever collaboration with top financial experts to evaluate cost functions.

2 EVALUATION DESIDERATA

This section outlines our proposed desiderata for evaluating cost functions as shown in Figure 6. Importantly, these metrics are considered for discrete and real-valued ground truths, which are both subsequently used in our evaluation. Discrete labels are elicited from human experts and real-valued ground truths are utilized when assessing continuous model outputs.

2.1 Desideratum 1: Feature Cost

Cost functions should assign varying weights to different features, as some changes are inherently easier [e.g., mutating your number of credit cards is easier than increasing salary (Rawal and Lakkaraju, 2020)]. To evaluate this, we perturb each feature and compare against a ground truth. Let $\mathbf{x}$ be a feature vector with features $\mathbf{f} = \{f_1, f_2, \ldots, f_n\}$, and $\mathbf{r}$ denote the ground truth providing known rankings or values. To get feature costs, they are perturbed (one SD for numerical features, random categories for categorical ones), and the average predicted cost across test data for all features forms the vector $\mathbf{c_p}$. For discrete ground truth rankings, $\mathbf{c_p}$ is rank ordered to form integer rankings $\mathbf{r_p}$.

Definition 2.1 (Feature Cost)**.** *The feature cost score ϕ_{fc} quantifies how well a cost function captures the relative difficulty of changing different features. It is computed as the mean absolute error (MAE) with predicted feature costs $\mathbf{c_p}$, or Spearman's ρ with rankings $\mathbf{r_p}$, for a real-valued and discrete ground truth across a test sample, respectively, against the ground truth $\mathbf{r}$:*

$$\phi_{fc} = \begin{cases} \rho(\mathbf{r}, \mathbf{r_p}) & \text{if } \mathbf{r} \in \mathbb{Z} \\ MAE(\mathbf{r}, \mathbf{c_p}) & \text{if } \mathbf{r} \in \mathbb{R} \end{cases} \tag{1}$$

2.2 Desideratum 2: Relative Cost

Relative cost captures how the difficulty of changing a feature varies across its distribution [e.g., saving 0-10% of salary is usually easier than 30-40% (Ustun et al., 2019)]. This desideratum evaluates whether cost functions properly capture monotonic cost trends in numeric features. Let $\mathbf{f} = \{f_1, f_2, \ldots, f_n\}$ be numerical features in a domain, and let $\mathbf{r}$ provide a ground truth with Spearman's rho (r_i) for each f_i's cost trend. To predict r_i, each f_i is split into m evenly spaced ranges. Perturbing f_i upwards by a standard deviation at each range yields costs over these intervals, forming a cost vector $\mathbf{c}_i$. Spearman's rho, $\hat{\rho}_i$, is then calculated from $\mathbf{c}_i$, averaged across test data. For real-valued ground truths r_i, predicted cost trends $\hat{\rho}_i$ are evaluated using MAE. For discrete r_i indicating cost polarity (+1, −1, or 0), we define threshold-based accuracy. The predicted class $\hat{c}_i$ for each feature i is determined as:

$$\hat{c}_i = \begin{cases} +1 & \text{if } \hat{\rho}_i > \tau \\ -1 & \text{if } \hat{\rho}_i < -\tau \\ 0 & \text{if } -\tau \leq \hat{\rho}_i \leq \tau \end{cases} \tag{2}$$

where we set $\tau = 0.3$, signifying a weak correlation (Sedgwick, 2014).

Definition 2.2 (Relative Cost)**.** *The relative cost ϕ_{rc} measures how accurately a cost function captures cost trends within $\mathbf{f}$. For real-valued r_i, ϕ_{rc} is evaluated using the mean absolute error (MAE) between $\hat{\rho}_i$ and r_i. For discrete r_i, it is the average accuracy of $\hat{c}_i$ across all n features and test data:*

$$\phi_{rc} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[r_i = \hat{c}_i] \tag{3}$$

2.3 Desideratum 3: Dependent Cost

The cost of altering one feature can depend on the value of another feature [e.g., loan amount modification typically depends on credit history (Karimi et al., 2022)]. This desideratum evaluates whether cost functions capture such feature dependencies. Let $\mathbf{d} = \{(d_1, i_1), (d_2, i_2), \ldots, (d_k, i_k)\}$ denote k feature dependencies, where the cost of mutating dependent feature d_j is influenced by independent feature i_j. To evaluate a specific dependency (d_j, i_j), for each instance x from a dataset D, we create two versions: In x_A, i_j is set to a predefined "high-cost" value v_A, and in x_B, i_j is set to a "low-cost value" v_B (e.g., bad v. good credit history when increasing loan amount). Then, we apply the same perturbation to d_j in both versions, resulting in x'_A and x'_B. The Mean Dependency Effect (MDE) for the j-th dependency is computed as:

$$\text{MDE}_{(d_j, i_j)} = \frac{1}{|D|} \sum_{x \in D} \left(C(x_A, x'_A) - C(x_B, x'_B) \right) \tag{4}$$

where $C(\cdot;\cdot)$ represents the cost of mutating from x_i to x'_i with the associated dependency. When the ground

truth for MDE scores is real-valued, MAE with the predicted $M_j = \text{MDE}_{(d_j,i_j)}$ is used. If the ground truth specifies particular sets of dependencies, where one set $\mathbf{d}_\text{t}$ (i.e., target) is known to exhibit large MDE scores, and another set $\mathbf{d}_\text{o}$ (i.e., other) comparatively weaker ones, we use the following formulation.

Definition 2.3 (Dependent Cost)**.** *Dependent cost ϕ_{dc} measures the ability to distinguish strong feature dependencies from weaker/negligible ones. For real-valued ground truth, ϕ_{dc} is evaluated using the mean absolute error (MAE) between the predicted and true MDE values. For discrete, it is the difference between average MDE scores for known dependencies and other weaker (or negligible) feature pairs across test data:*

$$\phi_{dc} = \left(|\mathbf{d}_t|^{-1} \sum_{j \in \mathbf{d}_t} M_j\right) - \left(|\mathbf{d}_o|^{-1} \sum_{j \in \mathbf{d}_o} |M_j|\right) \quad (5)$$

Notably, M_j does not have its absolute value taken, as it should always be positive if the dependencies were captured correctly, higher scores are better.

2.4 Desideratum 4: Fair Cost

Cost functions must sometimes address relevant fairness considerations (Von Kügelgen et al., 2022), ensuring that costs do not vary in an unwarranted way on sensitive attributes. Evaluation of fair cost can mirror that of dependent cost by mutating a dependent feature (e.g., education level) while an independent feature corresponding to a sensitive attribute (e.g., age, gender, or race) is varied.

3 METHODS CONSIDERED

We consider seven popular methods for calculating cost, before discussing our approach for using them to scale-up the Bradley-Terry model.

3.1 Cost/Distance Functions

L_p Norms. The most standard approach to defining cost functions relies on L_p norms (Karimi et al., 2022), where different values of p capture different notions of distance and effort. We consider the L_0, L_1, and L_2 norms.

MAD. Wachter et al. (2017) proposed using Manhattan distance weighted by the inverse Median Absolute Deviation (MAD) to account for the natural variability of different features:

$$\text{dist}(\mathbf{x}, \mathbf{x}^F) = \sum_{k=1}^{D} \frac{|\mathbf{x}_k - \mathbf{x}_k^F|}{\text{MAD}_k},$$
$$\text{MAD}_k = \text{med}_{j=1,\ldots,N} \left(|X_{j,k} - \text{med}_{l=1,\ldots,N}(X_{l,k})|\right) \quad (6)$$

where $\mathbf{x} \in \mathbb{R}^D$ is the original instance, $\mathbf{x}^F \in \mathbb{R}^D$ is the counterfactual, D is the number of features, $X \in \mathbb{R}^{N \times D}$ is the training data matrix with N instances, and $X_{j,k}$ denotes the value of feature k for training instance j.

Combo. Karimi et al. (2020) propose a more flexible weighted combination of multiple L_p norms that allows practitioners to balance different desirable properties.

$$\text{dist}(\mathbf{x}; \mathbf{x}^F) = \alpha||\boldsymbol{\delta}||_0 + \beta||\boldsymbol{\delta}||_1 + \gamma||\boldsymbol{\delta}||_2 + \zeta||\boldsymbol{\delta}||_\infty \quad (7)$$

where $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_D]^T$ and $\delta_k : \mathcal{X}_k \times \mathcal{X}_k \to [0, 1] \, \forall \, k \in [D]$. The weights α, β, γ, and ζ allow practitioners to balance between sparsity of changes (through the L_0 norm), elastic distance measures (through L_1 and L_2 norms), and maximum normalized change constraints (through the L_∞ norm).[2]

CDF. Ustun et al. (2019) introduce a maximum percentile shift measure that automatically accounts for the distribution of data. Given features $\mathbf{x}$ and an action $\mathbf{a} = [0, a_1, \ldots, a_D]$ where a_k represents the change to feature k, the cost is defined as:

$$\text{cost}(\mathbf{a}; \mathbf{x}) = \max_{k \in \{1,\ldots,D\}} |Q_k(x_k + a_k) - Q_k(x_k)| \quad (8)$$

where $Q_k(\cdot)$ is the cumulative distribution function (CDF) of feature x_k in the target population, and D is the number of actionable features. This metric naturally accounts for the relative difficulty of moving to unlikely regions of the data distribution—for instance, improving from the 50th to 55th percentile in school grades requires less effort than moving from the 90th to 95th percentile.

LLM. It is also possible to directly query LLMs for cost judgements, we do so by prompting it to produce a cost score between 0-10, where 0 represents essentially no difference, and 10 the maximum possible cost. See Appendix N for the prompt.

3.2 Expanding to Bradley-Terry

Here we propose a novel scalable solution for training cost functions. For all the measurement approaches listed above, it is possible to use them to label pairwise comparisons and train a cost function with the

[2]The authors gave no recommendations on the weights, so we set them as equal.

Bradley-Terry model. Prior work has attempted this with human labellers (Rawal and Lakkaraju, 2024), but mostly due to scalability issues it has never reached its full potential or modelled relative/dependent cost. Here we propose a solution to this based on using LLMs (or any distance function) as surrogate labellers, which can be seen in Figure 7. When considering the prior cost/distance functions, we use them to label two paired recourses, and then assign a discrete label post-hoc. In the case of LLMs however, both recourses are shown in a prompt and the LLM is asked to provide a label in a direct comparison. We describe our novel pipeline in detail next.

Generating synthetic recourses. Let $\mathcal{D} = \{x_i\}_{i=1}^N \subset \mathbb{R}^d$ denote a given dataset, where each x_i represents a d-dimensional feature vector. We define a stochastic perturbation function $\phi : \mathbb{R}^d \times \mathcal{A} \to \Delta(\mathbb{R}^d)$, where $\mathcal{A}$ denotes a set of actionability constraints (see Appendix A for details). The number of features to be perturbed is problem-specific. Here, we randomly select this number from a truncated geometric distribution, favoring perturbations of one feature, see Appendix B. For each datum x_i and perturbed feature $f \in \{1, \ldots, d\}$, we apply the following perturbation:

$$x_i'[f] = \begin{cases} \sim \text{Uniform}(\text{categories}_f) & \text{if } f \text{ is cat.} \\ x_i[f] + \epsilon : \epsilon \sim \mathcal{E}_f & \text{if } f \text{ is num.,} \end{cases} \quad (9)$$

where $\text{Uniform}(\text{categories}_f)$ is a uniform distribution over categories and $\mathcal{E}_f$ is either a finite set of perturbations for a numerical feature (positive/negative multiples of the standard deviation across $\mathcal{D}$), or a uniform sample from this same range for diversity (chosen 50% of the time). This process generates a set of recourse examples $\mathcal{R} = \{(x_i, x_i')\}_{i=1}^N$, where each x_i represents an original instance and x_i' is the corresponding synthetic perturbation. We incorporate a finite set of perturbation magnitudes for numerical features because it allows a direct comparison between exactly the same change at different parts of a feature distribution to learn relative costs.

Selecting recourse pairs. Next, we select a set of $K \leq N^2$ pairs of recourse examples from $\mathcal{R}$ to label cost comparisons into an undirected graph structure. We first enforce a minimal-spanning tree (MST), which allows the costs for all recourses to be estimated on the same scale (Thurstone, 1994; Hunter, 2004). We then randomly add edges to enforce that each recourse has a minimum of $K_{\min}$ edges. We also prioritize edges from $\mathcal{E}_f$ between recourses which perturb the same numerical feature at two different parts of the distribution by exactly the same amount. This forces the comparisons to reason about the difference in cost between e.g. increasing salary from \$30-35k vs. \$50-55k (i.e., *relative cost*). The total additional edges from this enforcement is set to 5% of the total data.

(Beginning of prompt omitted...)

- Some features are naturally harder to change than others, use this logic.

- For numerical features, the difficulty of changing them can often depend on their starting values.

- Apart from the mutated features, consider the other features which are different between Alex and Jaden, and how this may affect difficulty.

(Ending of prompt omitted...)

Figure 1: These three points represent the high-level desiderata instructions given in the *Full Prompt* and removed in the *Ablated Prompt*. Although later generation models (i.e., `Claude3.7-Sonnet`) needed these instructions less, we found that overall it boosted performance consistently.

Labeling pairwise. When using e.g. L_p norms to label pairwise comparisons one need only to use the said function on both recourses and label the higher cost one appropriately. When using the LLM prompting approach, we begin by instructing the LLM that the task is comparing two individuals and their respective feature changes. We then enumerate the features, as well as their descriptions. Crucially, the prompt then gives instructions to consider the full desiderata in Section 2 (henceforth known as the "full prompt", see Figure 1). The LLM is then asked to reason about which of the two recourses requires more effort for the individual (i.e. cost), and finally to respond with a label of 1 (first requires more effort), 0 (second requires more effort), or 0.5, indicating equal effort, which is a useful de-biasing signal where features represent sensitive demographic attributes. In addition, the LLM is instructed to use chain-of-thought (Kamruzzaman and Kim, 2024). The full prompt is given in its entirety in Appendix N. The output of this stage is a set of K comparisons $\mathcal{Q} = \{(i, j, y)\}_{k=1}^K$, where i and $j \neq i$ are indices of a pair of recourse examples from $\mathcal{R}$ and $y \in \{0, 0.5, 1\}$ denotes the LLM's cost judgment.

Training a cost function. Finally, we use the dataset $\mathcal{Q}$ to train a cost function $C : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ using the Bradley-Terry model (Kwon et al., 2023). That is, given a cost function C and a pair of recourses (x_i, x_i') and (x_j, x_j'), we define the predicted probability that recourse i has higher cost than recourse j as:

$$\hat{y}_C(i, j) = \frac{1}{1 + \exp(C(x_j, x_j') - C(x_i, x_i'))}. \quad (10)$$

Our training objective is to minimize the binary cross-entropy between these predicted comparison probabilities and the labels across all training examples:

$$\arg\min_{C\in\mathcal{M}}\Big[-\sum_{(i,j,y)\in\mathcal{Q}} y\log(\hat{y}_C(i,j)) + (1-y)\log(1-\hat{y}_C(i,j))\Big], \tag{11}$$

where $\mathcal{M}$ is a chosen model class. Since this loss is differentiable, we defined $\mathcal{M}$ as the class of multi-layered perceptron (MLP) neural networks and trained by stochastic gradient descent. We one-hot encode categorical features and concatenate the original data point x, the perturbed recourse point x' and the feature-wise difference $x'-x$ into a single vector $[x, x', x'-x] \in \mathbb{R}^{3d}$. In practice, we find that the simple feature augmentation step of including $x'-x$ significantly improves the models' ability to learn costs. As a final post-processing step, we shift the outputs of trained models to be non-negative on all training data, see Figure 7.

4 EXPERIMENTS

Three datasets are considered, Adult Census (Becker and Kohavi, 1996), HELOC (Mstz, 2024), and German Credit (Hofmann, 1994). All were preprocessed to have a mix of numerical and categorical features, and are binary classification tasks. For details such as actionability constraints see Appendix A. For LLMs we used three models of varying performance, specifically 'claude-3-5-haiku-20240307:v1:0', 'claude-3-5-sonnet-20240620:v1:0', and 'claude-3-7-sonnet-20250219:v1:0'.

4.1 Acquiring a ground truth

A critical aspect of our study is to acquire a reliable ground truth. Hence, five senior AI financial researchers working on the deployment of cost functions in industry were recruited to label our data in two settings, firstly across three synthetic datasets, and secondly across the three benchmark ones listed above. We acknowledge a potential mismatch between financial experts and the general population, but our focus is on evaluating general cost functions deployed by institutions, prioritizing domain expertise ensures the ground truth accurately reflects the operational and financial realities of these systems.

For the synthetic tests, we designed three datasets to exclude features used in popular ones, mainly to exclude memorization as a confounder.[3] For all datasets, we asked experts to (1) rank order the difficulty of mutating features, (2) which numerical features decrease/increase in cost monotonically (or stay the same), and (3) for as many notable dependencies as they could name. Importantly, these were all discrete labels. Feature ranking ground truths were processed using Borda counts to rank order (Saari, 1985). Numerical features took the modal response as the ground truth, and dependencies used consensus-based expert elicitation (Hsu and Sandford, 2007), retaining only those mentioned by at least two annotators for the *"target"* sets in Section 2, and all others for the *"other"* sets. Post-hoc analysis of inter-rater reliability that yielded a Kendall's W of 0.86 for feature rankings and a Fleiss' Kappa of 0.63 for relative cost labels, indicating robustness. See Appendix D and E for full details.

[3]Specifically, Adult Census (Becker and Kohavi, 1996), HELOC (Mstz, 2024), German Credit (Hofmann, 1994), Taiwan Credit (Yeh, 2009), Lending Club (Club, 2023), and Give Me Some Credit (Fusion and Cukierski, 2011).

4.2 Studying convergence

Here we investigate hyperparameter choices in the LLM Bradley-Terry modeling approach. We assume a fully connected graph will learn the best cost function, and use this model's predictions as a real-valued ground truth, so our metrics reported here reflect the MAE variants in Section 2. We consider three parts to the evaluation, the learning of the desiderata, the choice of MST algorithm, and the effect of our extra connections for relative cost outlined in Section 3.2. We take $n = 150$ training instances for each of the three benchmark datasets, and form an MST with four algorithms, Kruskal (Kruskal, 1956), Prim (Prim, 1957), Boruvka (Borvka, 1926), and Random spanning trees (Broder, 1989). After this we make a fully connect graph, and have the LLM label it with temperature settings of 0.0 and 1.0, across the three LLMs. Cost functions are trained on the initial MST data, and again after adding $x\%$ extra connections towards a fully connected graph randomly. For the dependencies, we used all those identified by experts (before filtering to the *"target"* subset). For metrics, we used MAE on the dimensions in Section 2, before normalizing and averaging the results between 0-1 for visualization in Figure 2 across 3 seeds.

First, examining Figure 2(A), it can be observed that convergence was slower with temperature settings of 1.0. Moreover, there is a highly significant difference between Claude Haiku and the later two generations of Sonnet 3.5 and 3.7. This shows two things, sparser graphs should be labelled with lower temperatures to have a better converged cost function, and later generation models are more consistent. Figure 2(B) shows there is little difference between MST algorithms. Figure 2(C) illustrates that adding the extra links for relative cost we proposed in Section 3.2 speeds up

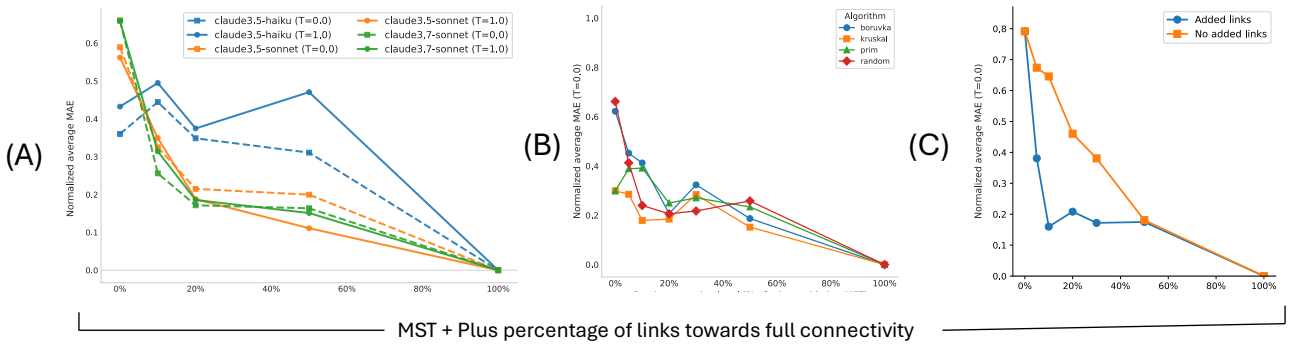

Figure 2: Convergence of cost functions as a function of connectivity: (A) Lower connectivity in the graph benefits from later generation models, as well as lower temperature settings, although higher connectivity seems to benefit slightly from noisier labels. (B) All graph algorithms perform comparably, although random spanning tree and Boruvka appears worse in low connectivity. (C) The addition of our extra links for relative cost dramatically speeds up convergence. T=Temperature of the LLM.

convergence.[4]

Remark 1 (Temperature and Sparsity)**.** *While noise typically aids preference learning (Laidlaw and Russell, 2021), our results show that low temperature is preferable when learning from sparsely connected graphs with $K = |E| \approx O(N)$ comparisons, where K denotes the number of pairwise comparisons, N the number of recourses (nodes), and $O(N)$ linear growth. Graph partitions connected by few edges create critical dependencies where noisy labels propagate errors across the graph, distorting the learned function. Label accuracy on these "bottleneck" edges is seemingly critical, see Appendix F for further discussion.*

4.3 Synthetic experiment

Here we examine the ability of the LLM to correctly identify higher costs in pairwise comparisons synthetically designed as described in Section 4.1. As the ground truth and LLM labels are both discrete, we use bootstrapped accuracy (1000 samples per N) over 300 instances as our metric. For each, the LLM was queried 8 times with temperature 1.0[5], and the most common answer taken. We compare our standard prompting scheme (i.e., the *'Full prompt'*) versus a simple L_1 norm (on normalized data), and the same prompt with the instructions to consider different dimensions mentioned in Figure 1 ablated (i.e., the *'Ablated prompt'*). This allows us to understand how different LLM models need to be prompted across generations, and their natural reasoning ability before being explicit instructed to consider relevant aspects for cost calculations. For the different prompts see Appendix N.

The results are shown in Figure 3, where the full prompt achieved significantly higher accuracy in all dimensions across all bootstrapped values of N. Moreover, there are significant improvements in the latest generation model `Claude3.7-Sonnet` compared to earlier iterations, indicating improved cost alignment with humans. Perhaps most striking was the ability of 3.7-Sonnet to label dependencies accurately when instructed with the desiderata compared to 3.5 Sonnet and 3.5-Haiku (Haiku=63% v. 3.5-Sonnet=88% v. 3.7-Sonnet=100%). In contrast to all these results, the L_1 baseline performed poorly. Similar trends were seen across desideratum and prompt types.

4.4 Benchmark datasets

Here we compare all the distance/cost functions discussed in Section 3 in both their basic form where they are directly queried for cost, and in their Bradley-Terry variant where we train a cost function using pairwise comparisons. We set our data size to $n = 666$ and perturbed each datum two times. For non-LLM variants we labelled the entire 100% connected graph, but for LLM-based methods we used a k_{min} such that the connectivity of the graphs was 10%, giving a dataset of size 93,076 after adding the additional edges for relative cost (see Section 3.2), which was seen to have a nice balance between API cost and performance in Section 4.2. Building further on the prior tests, Prim's algorithm was used for the MST, and a temperature of 0.0 in LLMs. For evaluation of the cost function dimen-

[4]We also conducted convergence tests evaluating hyperparameters against the human-labeled discrete ground truth. The tests again showed more advanced LLMs and lower temperature (T=0.0) yielded better convergence. Feature cost (ϕ_{fc}) converged the quickest, with `Claude3.7-Sonnet` achieving the best score ($\phi_{fc} = 0.65$). Due to the small size of the dataset however ($n = 150$), relative and dependent costs did not fully converge.

[5]We also tested a temperature of 0.0 with similar results.

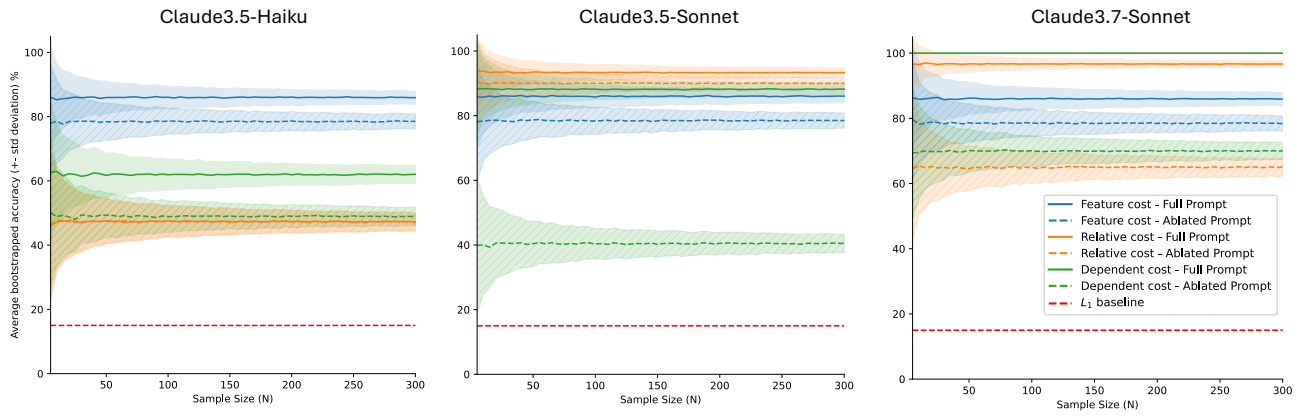

Figure 3: Testing LLM ability to label synthetic data: Overall, the LLMs were capable of correctly labeling the data across all dimensions (feature, relative, and dependent cost), and especially if prompted to consider them at a high-level only (i.e., the *'Full Prompt'*). Furthermore, there is a significantly improved difference in the later generation model `Claude3.7-Sonnet` compared to its earlier iterations.

sions, we used the ground truth defined by experts in Section 4.1, and 333 holdout instances. In LLM variations, we compare the full and ablated prompts. MLPs were trained over 3 seeds with standard error reported in Table 1. Across datasets, prompt types, and LLMs, the experiment required 1,675,368 API calls.

Table 1 shows the results. Overall, the Bradley-Terry approach performed better than querying the methods directly for cost, and using LLMs on average did better than using L_p norms. Going into greater detail, our proposed method to scale up the Bradley-Terry model worked better compared to L_p norms, in particular `Claude3.7-Sonnet` prompted with the full prompt (i.e., `Claude3.7-Sonnet`*) was the only method to achieve positive results in all three metrics. In fact, aside from one discrepancy in relative cost ϕ_{rc}, it achieved the best results in all metrics. In direct usage, the L_p norms were all incapable of capturing feature dependencies by definition and scored 0.0.

4.5 User testing

To conclude our evaluation, we consider two human studies. First, to gather the necessary expertise, we recruited 54 professionals working in a top financial institution to make pairwise judgments of cost on 18 questions (designed using summary statistics and expert help), and three raters within this group to make a further judgment on 120 recourses generated using DiCE by Mothilal et al. (2020). Both studies had an equal number of questions from the benchmark datasets. Both studies obtained IRB approval.

For the first study, users were shown two individuals, with proposed mutations, and asked to select which of these was a higher cost, and also judge on a Likert scale of 1-7 how close they were (1=equal cost). The same three LLMs reported in the paper thus far were used. The LLMs also made 54 judgments on the same materials to mimic the users with a temperature of 1.0., which has been discovered as optimal [see (Cui et al., 2024)] in larger studies.[6] These questions were split into three sets representing the metrics in Section 2, but we did not evaluate *Fair cost* due to ethical concerns. The participants were not compensated; all volunteered to participate. In total, 30 participants were male, 24 were female, all were aged between 18-65, with a mix of native/non-native English speakers. The metric of interest was the accuracy of the agreed modal responses on each question. See supplement and Figure 9 for the materials. Figure 4 shows the bootstrapped results (1000 samples per user N). Converging with all our prior results, `Claude3.7-Sonnet` achieved the best alignment when prompted using the full prompt, with its results largely converged at $N = 30$ onward with an accuracy of 82%, whilst the ablated versions plateaued with consistently lower accuracy (and L_1 at 11%).

In our raters study, the average expert rating for each recourse was taken as the ground truth to evaluate the LLM's most common label against (after taking 30 samples per question). Fleiss' kappa on the raters yielded 0.64, indicating substantial inter-rater agreement. The bootstrapped accuracy and Kappa results are shown in Figure 8(B/C). Overall, there was no significant difference between LLMs and prompting schemes ($\sim 85\%$ accuracy), indicating that for simpler labeling tasks the choice of LLM is less critical, however there was a significant difference in the L_1 baseline which scored comparatively less with 61% accuracy.

[6]We also evaluated temp=0.0 and found the same accuracy but more probability mass on the modal responses.

Table 1: We compared all methods by either querying them directly for costs, or using them to label pairwise comparisons for Bradley-Terry. All methods were evaluated across our three main desideratum metrics. For the 'Direct Usage', all L_p norm methods scored 0 in their dependency tests, illustrating they capture no dependency information. The best results overall were seen in the Bradley-Terry Model, using the most up-to-date LLM `Claude3.7-Sonnet`, and prompting it to directly consider the desiderata. LLMs prompted using the full prompt include an asterisk*, the others use the ablated prompt. Standard error is shown across 3 seeds.

	Direct Usage			Bradley-Terry Model		
	ϕ_{fc}	ϕ_{rc}	ϕ_{dc}	ϕ_{fc}	ϕ_{rc}	ϕ_{dc}
l_0	-0.25 ± 0.22	0.22 ± 0.14	**0.00 ± 0.00**	-0.25 ± 0.20	0.44 ± 0.17	-0.17 ± 0.53
l_1	-0.17 ± 0.28	0.22 ± 0.14	**0.00 ± 0.00**	0.01 ± 0.24	0.22 ± 0.14	-0.17 ± 0.07
l_2	-0.17 ± 0.28	0.22 ± 0.14	**0.00 ± 0.00**	0.04 ± 0.24	0.22 ± 0.14	-0.15 ± 0.07
MAD. Wachter et al. (2017)	0.43 ± 0.06	0.22 ± 0.14	**0.00 ± 0.00**	-0.53 ± 0.09	0.22 ± 0.14	-0.12 ± 0.04
Combo. Karimi et al. (2020)	**0.67 ± 0.04**	0.22 ± 0.14	**0.00 ± 0.00**	-0.73 ± 0.07	0.22 ± 0.14	-0.14 ± 0.07
CDF. Ustun et al. (2019)	0.45 ± 0.16	0.56 ± 0.17	**0.00 ± 0.00**	-0.42 ± 0.17	0.22 ± 0.14	-0.12 ± 0.02
Claude 3.5.Haiku	0.36 ± 0.11	0.44 ± 0.17	-0.10 ± 0.05	0.33 ± 0.08	0.78 ± 0.14	-0.01 ± 0.15
Claude 3.5.Haiku*	0.29 ± 0.12	0.56 ± 0.17	-0.08 ± 0.09	0.60 ± 0.08	0.67 ± 0.16	-0.13 ± 0.13
Claude 3.5.Sonnet	0.29 ± 0.13	**0.67 ± 0.16**	-0.31 ± 0.09	0.44 ± 0.16	**0.89 ± 0.10**	-0.16 ± 0.06
Claude 3.5.Sonnet*	0.24 ± 0.13	0.56 ± 0.17	-0.39 ± 0.13	0.68 ± 0.07	0.67 ± 0.16	-0.12 ± 0.08
Claude 3.7.Sonnet	0.40 ± 0.09	0.44 ± 0.17	-0.05 ± 0.07	0.37 ± 0.16	0.78 ± 0.14	-0.10 ± 0.07
Claude 3.7.Sonnet*	0.48 ± 0.06	0.44 ± 0.17	-0.04 ± 0.09	**0.71 ± 0.07**	0.67 ± 0.16	**0.15 ± 0.07**

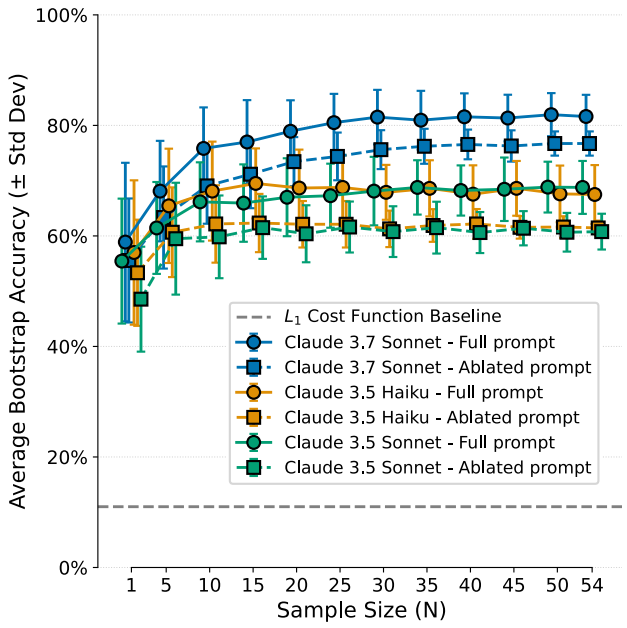

Figure 4: User testing: Bootstrapping users in our study we found the overall accuracy better with the full prompt using `Claude3.7-Sonnet`, which begun to plateau after N=30, with a mean accuracy of ∼82%.

4.6 Fairness Evaluation

We tested the same models using both the ablated prompt and the full prompt to see if their cost assessments exhibited bias against underprivileged groups, specifically bias towards gender or race when mutating education with Equation 5 in the Adult Census dataset.

Table 2 shows the models were mostly unbiased, but the more advanced model, `Claude3.7-Sonnet`, displayed significantly greater bias with the full prompt (Mean Bias = +0.5552). To mitigate this, we inserted the explicit instruction *"Do not use demographic information in your reasoning."* as a fourth rule to the desiderata. This single additional rule successfully de-biased the model, reducing the mean bias in `Claude3.7-Sonnet` to +0.0761, and employed the needed characteristics to the final cost function. This mitigation resulted in minimal difference to the cost function's overall predictive performance. Overall, while the full prompt serves as useful general guidance, practitioners should be mindful of potential biases and utilize explicit fairness constraints during prompt design.

Table 2: **Fairness Evaluation**. Higher values indicate greater bias against underprivileged groups. `Claude3.7-Sonnet` exhibited the greatest bias, but was successfully mitigated with the addition of an additional fairness rule. (*) indicates usage of the Full Prompt in Figure 1, (†) indicates the usage of the additional fairness instruction.

LLM Model	Mean Bias
Claude 3.5-Haiku	$+0.17 \pm 0.01$
Claude 3.5-Haiku*	$+0.05 \pm 0.02$
Claude 3.5-Sonnet	-0.10 ± 0.01
Claude 3.5-Sonnet*	$+0.24 \pm 0.01$
Claude 3.7-Sonnet	$+0.21 \pm 0.01$
Claude 3.7-Sonnet*	$\mathbf{+0.56 \pm 0.01}$
Claude 3.7-Sonnet*†	$\mathbf{+0.08 \pm 0.01}$

4.7 Deployment Study

To validate the practical utility of our approach, we integrated our trained cost functions into two established algorithmic recourse algorithms by Keane and Smyth (2020) and Wachter et al. (2017). The objective was to observe whether our learned cost functions yield more realistic and actionable recommendations compared to standard distance metrics (L_1 and MAD).

We evaluated the frequency of recommended feature mutations on the Adult Census dataset, grouping the features by mutability. "Immutable" features include Male, Age, Native-US, and Caucasian, while "Mutable" features include Married, Education, Hours-Work, and Private Work. As shown in Figure 5, our learned cost function consistently shifts recommendations away from immutable attributes toward highly actionable ones. Similar improvements were observed on the HELOC dataset, the full results of which can be found in Appendix I. Lastly, we also conducted robustness tests in our prompting scheme, demonstrating slight variations make no significant impact, see Appendix M

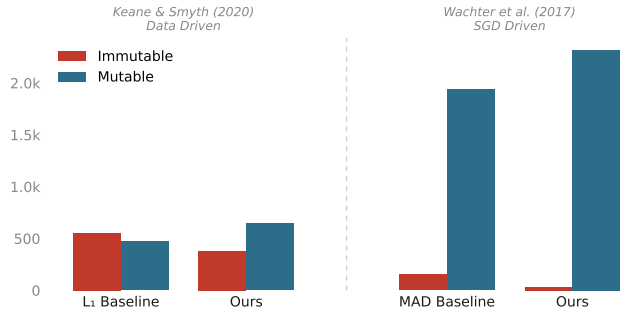

Figure 5: Deployment test: Plugging our cost functions directly into two established and varied algorithms illustrates how more actionable features are chosen with our method than heuristic approaches using e.g. L_1 norms or median-absolute deviation (MAD). The plot shows the total number of feature modifications across all instances in our tests.

5 RELATED WORK

There is an extensive body of literature focused on personalized algorithmic recourse. Approaches such as PEAR (De Toni et al., 2022) emphasize tailoring cost evaluations to the individual through preference elicitation. Similarly, other recent works have explored interactive, user-guided customization to improve the actionability and user perception of algorithmic recourse (Yetukuri et al., 2023; Wang et al., 2023; Koh et al., 2025; Esfahani et al., 2024).

There has been no systematic cost function evaluation in the literature, but some authors have done some comparative testing. Poyiadzi et al. (2020) compared standard L_1 and L_2 distance costs against a density-weighted path cost, showing path-based costs produced more feasible recourses. Nguyen et al. (2024) compared recourses under their adaptive Mahalanobis model to those under PEAR's linear one (De Toni et al., 2022), showing their method ReAP yielded significantly lower recourse costs. Tominaga et al. (2024) conducted a large user study to see if L_0 and L_1 distances correlated with user acceptance and willingness to act on recourses, finding minor correlation. Lastly, Chen et al. (2025) evaluated recourse under cost functions, demonstrating trade-offs between cost criteria.

To move beyond heuristic L_p norms, researchers have relied on preference elicitation to model decision-making (Pigozzi et al., 2016). In the context of recourse, Rawal and Lakkaraju (2024) recently proposed learning costs from pairwise feature comparisons. Our work builds on this, but we extend it to capture more cost dimensions such as dependencies and automation with LLMs.

Applying LLMs as labelers shares similarities with reward modeling in Reinforcement Learning from Human Feedback. However, applying this to high-stakes requires caution. The literature on LLM alignment demonstrates that models can harbor demographic biases and that their judgments can be sensitive to the format of the information passed, such as table-to-text serialization (Azime et al., 2025). To mitigate these risks, our framework explicitly incorporates instructions to neutralize these demographic biases.

6 DISCUSSION

The problem of recourse has grown in importance with the widespread use of ML (Gajcin and Dusparic, 2024; Kothari et al., 2024). However, a core unresolved issue in the field has been a lack of appropriate cost functions, which has limited the practical value of research. In this paper, we conducted the first thorough evaluation of cost functions in order to give a concrete understanding of progress and the state of research. In doing so, we proposed an evaluation desiderata and the usage of LLMs as surrogate labellers to scale up the Bradley-Terry model, showing this performed best, particularly when prompted to consider the points in Figure 1. A limitation of our method is the need for a large number of LLM labels, but building this dataset is only required once, and it can then train many cost functions in limited time. In future work it would be interesting to investigate similar methods in individualized recourse (Nguyen et al., 2024), medicine (Lacerda et al., 2023) or semifactual recourse (Kenny and Huang, 2024), all of which likely involve other considerations.

DISCLAIMER

This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co., and its affiliates ("JP Morgan") and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation, and warranty whatsoever, and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

ETHICS STATEMENT

Algorithmic recourse operates in high-stakes domains, such as financial lending, where automated decisions directly impact human lives and livelihoods. While our proposed framework leverages Large Language Models (LLMs) to effectively scale the learning of cost functions, this approach introduces specific ethical considerations that practitioners must carefully navigate. LLMs are known to reflect and occasionally amplify societal biases present in their training data. In our extended fairness evaluations, we observed that while earlier models were relatively unbiased, more advanced models (e.g., `Claude3.7-Sonnet`) could exhibit greater demographic biases when making cost judgments under complex prompts. Although we demonstrated that this can be heavily mitigated by injecting explicit fairness constraints into the prompt (e.g., explicitly instructing the model to ignore protected attributes), prompt-based debiasing is not a foolproof guarantee of fairness. Therefore, we strongly advise that LLM-derived cost functions should not be deployed autonomously in production environments without rigorous, domain-specific fairness auditing. Our work provides a scalable, empirically tested foundation for modeling complex cost dynamics, but it remains imperative that institutions couple these systems with continuous human oversight to ensure equitable outcomes for all users.

References

Azime, I. A., Kanubala, D. D., Afonja, T., Fritz, M., Valera, I., Klakow, D., and Slusallek, P. (2025). Accept or deny? evaluating llm fairness and performance in loan approval across table-to-text serialization approaches. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17478–17503.

Becker, B. and Kohavi, R. (1996). Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

Borvka, O. (1926). O jistém problému minimálním.

Broder, A. Z. (1989). Generating random spanning trees. In *FOCS*, volume 89, pages 442–447.

Buchanan, B. G. and Smith, R. G. (1988). Fundamentals of expert systems. *Annual review of computer science*, 3(1):23–58.

Chen, W.-L., Huang, H.-C., Lin, K.-H., Hwang, S.-W., and Yang, H.-T. (2025). Pareto optimal algorithmic recourse in multi-cost function. *arXiv e-prints*, pages arXiv–2502.

Club, L. (2023). Lending club loan dataset. Kaggle Dataset. This dataset contains detailed information about loans issued through the Lending Club platform, including borrower credit scores, loan amounts, employment information, and loan status.

Cui, Z., Li, N., and Zhou, H. (2024). Can ai replace human subjects? a large-scale replication of psychological experiments with llms. *A Large-Scale Replication of Psychological Experiments with LLMs (August 25, 2024)*.

De Toni, G., Viappiani, P., Teso, S., Lepri, B., and Passerini, A. (2022). Personalized algorithmic recourse with preference elicitation. *arXiv preprint arXiv:2205.13743*.

Esfahani, S., De Toni, G., Lepri, B., Passerini, A., Tentori, K., and Zancanaro, M. (2024). Preference elicitation in interactive and user-centered algorithmic recourse: an initial exploration. In *Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pages 249–254.

Fusion, C. and Cukierski, W. (2011). Give me some credit. `https://kaggle.com/competitions/GiveMeSomeCredit`. Kaggle.

Gajcin, J. and Dusparic, I. (2024). Redefining counterfactual explanations for reinforcement learning: Overview, challenges and opportunities. *ACM Computing Surveys*, 56(9):1–33.

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

Hsu, C.-C. and Sandford, B. A. (2007). The delphi technique: making sense of consensus. *Practical assessment, research, and evaluation*, 12(1).

Hunter, D. R. (2004). Mm algorithms for generalized bradley-terry models. *The annals of statistics*, 32(1):384–406.

Kamruzzaman, M. and Kim, G. L. (2024). Prompting techniques for reducing social bias in llms through system 1 and system 2 cognitive processes. *arXiv preprint arXiv:2404.17218*.

Karimi, A.-H., Barthe, G., Balle, B., and Valera, I. (2020). Model-agnostic counterfactual explanations for consequential decisions. In *International conference on artificial intelligence and statistics*, pages 895–905. PMLR.

Karimi, A.-H., Barthe, G., Schölkopf, B., and Valera, I. (2022). A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 55(5):1–29.

Keane, M. T., Kenny, E. M., Delaney, E., and Smyth, B. (2021). If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques. *arXiv preprint arXiv:2103.01035*.

Keane, M. T. and Smyth, B. (2020). Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai). In *Case-Based Reasoning Research and Development: 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8–12, 2020, Proceedings 28*, pages 163–178. Springer.

Kenny, E. and Huang, W. (2024). The utility of "even if" semifactual explanation to optimise positive outcomes. *Advances in Neural Information Processing Systems*, 36.

Kenny, E. M., Ruelle, E., Keane, M. T., and Shalloo, L. (2024). A hybrid model that combines machine learning and mechanistic models for useful grass growth prediction. *Computers and Electronics in Agriculture*, 219:108805.

Koh, S., Kim, B. H., and Jo, S. (2025). Understanding the user perception and experience of interactive algorithmic recourse customization. *ACM Transactions on Computer-Human Interaction*, 31(3):1–25.

Kothari, A., Kulynych, B., Weng, T.-W., and Ustun, B. (2024). Prediction without preclusion: Recourse verification with reachable sets. In *The Twelfth International Conference on Learning Representations*.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50.

Kwon, M., Xie, S. M., Bullard, K., and Sadigh, D. (2023). Reward design with language models. *arXiv preprint arXiv:2303.00001*.

Lacerda, A., Almeida, C., Ferreira, L., Pereira, A., Pappa, G. L., Meira, W., Miranda, D., Romano-Silva, M. A., and Diniz, L. M. (2023). Algorithmic recourse in mental healthcare. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Laidlaw, C. and Russell, S. (2021). Uncertain decisions facilitate better preference learning. *Advances in Neural Information Processing Systems*, 34:15070–15083.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35.

Mothilal, R. K., Sharma, A., and Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617.

Mstz (2024). Heloc dataset. `https://huggingface.co/datasets/mstz/heloc`. Accessed: 2024-09-13.

Nguyen, D., Nguyen, B., and Nguyen, V. A. (2024). Cost-adaptive recourse recommendation by adaptive preference elicitation. *arXiv preprint arXiv:2402.15073*.

Pigozzi, G., Tsoukias, A., and Viappiani, P. (2016). Preferences in artificial intelligence. *Annals of Mathematics and Artificial Intelligence*, 77(3):361–401.

Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., and Flach, P. (2020). Feasible and actionable counterfactual explanations. *New York: Association for Computing Machinery*.

Prim, R. C. (1957). Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401.

Rawal, K. and Lakkaraju, H. (2020). Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. *Advances in Neural Information Processing Systems*, 33:12187–12198.

Rawal, K. and Lakkaraju, H. (2024). Learning recourse costs from pairwise feature comparisons. *arXiv preprint arXiv:2409.13940*.

Saari, D. G. (1985). The optimal ranking method is the borda count. Technical report, Discussion paper.

Sedgwick, P. (2014). Spearman's rank correlation coefficient. *Bmj*, 349.

Thurstone, L. L. (1994). A law of comparative judgment. *Psychological review*, 101(2):266.

Tominaga, T., Yamashita, N., and Kurashima, T. (2024). Reassessing evaluation functions in algorithmic recourse: An empirical study from a human-centered perspective. *arXiv preprint arXiv:2405.14264*.

Ustun, B., Spangher, A., and Liu, Y. (2019). Actionable recourse in linear classification. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 10–19.

Von Kügelgen, J., Karimi, A.-H., Bhatt, U., Valera, I., Weller, A., and Schölkopf, B. (2022). On the fairness of causal algorithmic recourse. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 9584–9594.

Wachter, S., Mittelstadt, B., and Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841.

Wang, Z. J., Wortman Vaughan, J., Caruana, R., and Chau, D. H. (2023). Gam coach: Towards interactive and user-centered algorithmic recourse. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–20.

Yeh, I.-C. (2009). Default of Credit Card Clients. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C55S3H.

Yetukuri, J., Hardy, I., and Liu, Y. (2023). Towards user guided actionable recourse. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 742–751.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] Please see Section 3 and Appendix C.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] See attached code with submission.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] See supplement.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] See Section 4.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] See Section 4.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] See Appendix J.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes] See Section 4.
   (b) The license information of the assets, if applicable. [Yes] See Appendix K.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes] See supplement and Figure 9.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes] See Section 4.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes] See Section 4.

An Evaluation of Cost Function for Algorithmic Recourse: Supplementary Materials

A Actionability constraints and features used

The following are the datasets and features used, alongside any actionability constraints employed throughout the paper:

HELOC dataset: Here, the actionability constraints were to clamp feature mutations at the highest and lowest values observed in the dataset.

- *MSinceMostRecentInqexcl7days*: Number of months passed since the last credit inquiry on the individual.
- *NumRevolvingTradesWBalance*: The number of the individual's current credit accounts (e.g. credit cards) that have balances on them.
- *NumTradesOpeninLast12M*: The number of new credit accounts opened in the last 12 months.
- *NumInqLast6M*: The number of credit inquiries carried out on the individual in the last 6 months.

Adult census dataset: Here, the actionability constraints were to clamp feature mutations at the highest and lowest values observed in the dataset. Also, age and education number were only allowed to move upwards.

- *isMale*: If the person is male, or female, represented as 1 or 0, respectively.
- age: The person's age, represented as a floating point number.
- *native-country-United-States*: If the person's birthplace is the United States, or not, represented as 1 or 0, respectively.
- *marital-status-Married*: If the person is married, or not, represented as 1 or 0, respectively.
- *education-num*: The person's level of education, represented by a positive integer, where higher numbers are higher levels of education.
- *hours-per-week*: The number of hours the person works per week, represented by a positive integer.
- *workclass-Private*: If the person works for a private company, or is self-employed, represented as 1 or 0, respectively.
- *isCaucasian*: Is the person white or not, represented as 1 or 0, respectively.

German credit dataset: Here, the actionability constraints were to clamp numeric feature mutations at the highest and lowest values observed in the dataset.

- *status*: Status of existing checking account.
- *duration*: The proposed duration of the loan in months, expressed as an integer.
- *credit history*: The person's credit history with the options.
- *purpose*: The purpose of the loan.
- *amount*: The size of the loan asked for.

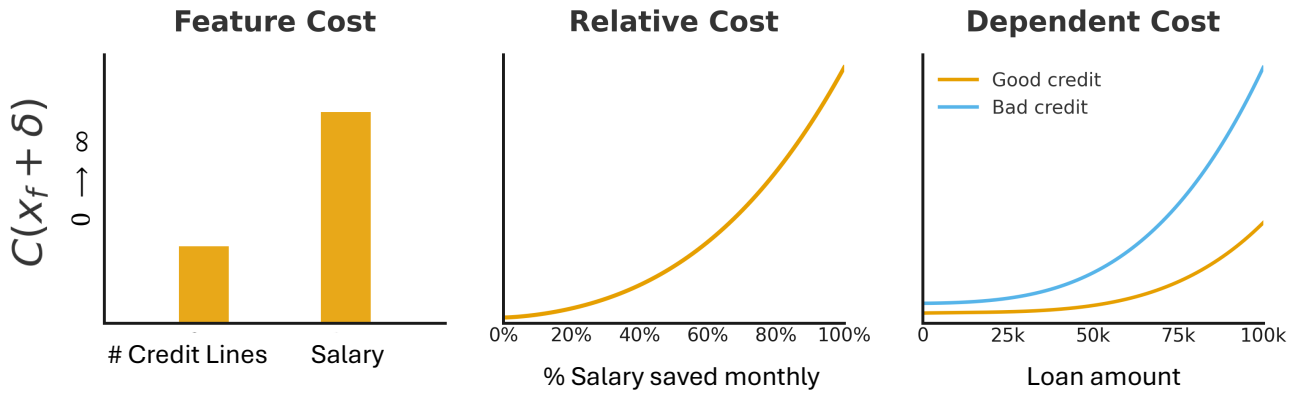

Figure 6: Understanding the need for the desiderata dimensions: The y-axis represents the cost associated with taking a particular action. (left) The cost of increasing salary is typically higher than increasing the number of credit lines a person has. (middle) The amount of cost associated with increasing the percentage of salary saved each month increases the higher the person's starting point. (right) The size of the loan asked for becomes more difficult if the person applying has bad credit compared to good credit.

B Perturbation function

In the context of feature vector perturbation, we employ a probabilistic approach to introduce controlled mutations to the feature set. Specifically, we perturb a feature vector by altering a random subset of its components. The number of features to be perturbed, denoted as k, is selected from the discrete set $\{1, 2, 3, 4\}$ with a predefined probability distribution. The probability mass function (PMF) for k is given by:

$$P(K = k) = \begin{cases} 0.8 & \text{if } k = 1 \\ 0.1 & \text{if } k = 2 \\ 0.05 & \text{if } k = 3 \\ 0.05 & \text{if } k = 4 \end{cases}$$

This distribution ensures that perturbing a single feature is the most probable event, while perturbing four features is the least probable. The purpose was to focus on sparsity for the cost function training, but also have some robustness.

C LLM-Based Bradley-Terry Pipeline

Figure 7 shows the pipeline for training our cost functions using LLM labeling and the Bradley-Terry Model for pairwise preference learning.

D Expert defined ground truths

In Section 4.4 we used expert defined ground truths for evaluating the trained cost function obtained from five raters. To average their ratings for the rank ordering of feature mutation difficulty we used the Borda count method. For HELOC the hardest features to change from hardest to easiest were MSinceMostRecentInqexcl7days, NumRevolvingTradesWBalance, NumTradesOpeninLast12M, NumInqLast6M. For Adult they were native-country-United-States, isWhite, isMale, age, marital-status-Married, education-num, workclass-Private, hours-per-week. For German Credit they were credit history, status, purpose, duration, amount.

To identify which features were most relevant for evaluating relative cost and dependent cost, we applied a consensus threshold. For relative cost the most common labelled identified by experts was taken as the ground truth. For dependent cost we collected a list of all dependencies identified by experts, and divided it into two sets. The first set was all dependencies which were identified by more than one labeler, the second set was all remaining dependencies. With this we were able to form our "target" and "other" sets from Equation 4, respectively. This

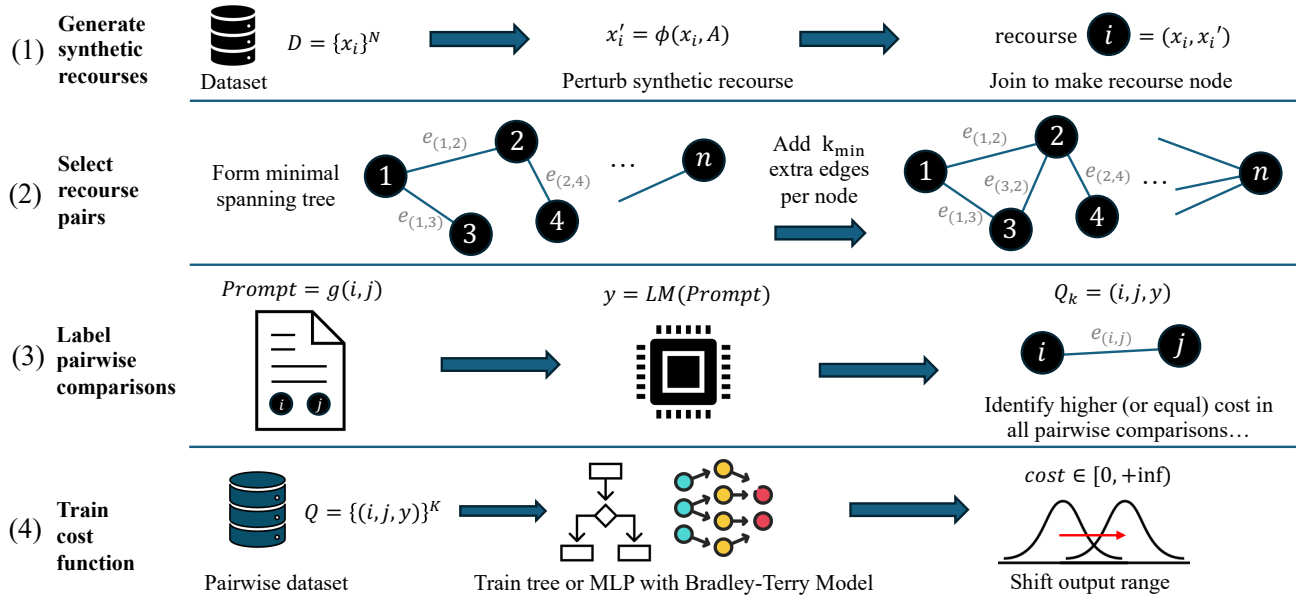

Figure 7: Method Schematic: (1) Each instance x_i in a dataset D is perturbed with $\phi(\cdot,\cdot)$ within actionability constraints A to simulate a recourse action taking x_i to x_i', which forms one node in the graph. (2) The nodes are connected with a minimal-spanning tree (MST), before adding additional random edges subject to a K_{min} constraint which dictates the minimum number of edges per node. (3) Each edge in the graph is then labeled by an LLM which judges which of the two corresponding recourses takes a higher cost to achieve (or optionally an additional equal cost option). (4) The dataset of comparisons is used to train the cost function (e.g., a decision tree or multi-layered perceptron).

approach draws on principles from the Delphi method and consensus thresholding commonly used in expert judgment research (Hsu and Sandford, 2007), balancing the need to capture shared expert insights while filtering out idiosyncratic cases.

D.1 Relative cost ground truth labels

For relative cost modeling, each feature was classified as having a monotonically increasing cost (+1), decreasing cost (–1), or neutral cost (0) as its value changes. The full set of features and their labels is shown below:

- HELOC: **MSinceMostRecentInqexcl7days**: 0 (neutral)
- HELOC: **NumRevolvingTradesWBalance**: –1 (monotonically decreasing)
- HELOC: **NumTradesOpeninLast12M**: –1 (monotonically decreasing)
- HELOC: **NumInqLast6M**: 0 (neutral)
- Adult Census: **age**: –1 (monotonically decreasing)
- Adult Census:**education-num**: +1 (monotonically increasing)
- Adult Census:**hours-per-week**: +1 (monotonically increasing)
- German Credit:**duration**: –1 (monotonically decreasing)
- German Credit: **amount**: –1 (monotonically decreasing)

D.2 Full list of dependencies (strongest in bold)

For dependent cost, the full set of expert-elicited dependencies is listed below. The set labelled as "target" from Equation 4 is highlighted in **bold**.

Dataset: HELOC

- **Increasing NumRevolvingTradesWBalance (harder if NumTradesOpeninLast12M is low)**
- **Increasing NumTradesOpeninLast12M (harder if NumRevolvingTradesWBalance is low)**
- Increasing MSinceMostRecentInqexcl7days (harder if NumInqLast6M is high)
- **Increasing NumTradesOpeninLast12M (harder if NumInqLast6M is higher)**
- Increasing NumTradesOpeninLast12M (harder if MSinceMostRecentInqexcl7days is high)
- Increasing NumRevolvingTradesWBalance (harder if NumInqLast6M is high)

Dataset: Adult Census

- **Decreasing working hours (harder if working for private company)**
- **Getting married (harder if older)**
- **Increasing education (harder if married)**
- Working longer hours (easier if older/more experienced)
- **Increasing education (harder if work hours is high)**
- Increasing work hours (easier if working for private company)

Dataset: German Credit

- **Increasing loan amount (harder with bad credit history)**
- **Increasing duration (harder with bad credit history)**
- Decreasing duration (harder for larger loan amounts)

E Synthetic datasets

Here are the features used for the three synthetic datasets. Their generation scripts are given also to foster reproducibility more easily. In all comparisons, accuracy of the pairwise comparison was taken as our evaluation metric against the known pre-defined ground truths.

The process worked as follows:

1. We took all popular recourse datasets and aggregated a full list of their features (113 in total) to exclude from this process.
2. Three datasets were designed (one for each cost function dimension excluding fair cost).
3. For 'Feature cost', experts were asked to simply rank order the difficulty of mutating the features.
4. For 'Relative cost', they were asked to specify if each numerical feature monotonically increased or decreased (or neither) in cost as the value goes higher.
5. For 'Dependent cost', they were asked to label at maximum the three strongest dependencies present, as well as appropriate values for these while testing and mutating.

Feature ranking ground truths were processed using Borda counts (Saari, 1985). Numerical features took the most common label, and dependencies used consensus-based expert elicitation (Hsu and Sandford, 2007), retaining only those mentioned by at least two annotators, which totalled three.

Dataset 1: feature cost

- Inheritance Wealth: integer in dollars.
- DSCR: float representing the ratio of income to debt payments.
- Liquid Asset Value: integer in dollars.
- Monthly Savings Rate (%): integer between 0-100.

Listing 1: Python code for generating synthetic dataset for 'Feature Cost'

```python
import numpy as np

np.random.seed(42)
n_samples = 1000

inheritance_wealth = np.random.randint(0, 1000001, n_samples)
dscr = np.random.uniform(0.5, 3.0, n_samples)
liquid_asset_value = np.random.randint(0, 500001, n_samples)
monthly_savings_rate = np.random.randint(0, 101, n_samples)
```

All these features were mutated upwards 5% when comparing cost. The ground truth was that the features should be hardest to mutate from the first (inheritance wealth) to the last (monthly saving rate) in order. During the tests we randomly compared two features mutations on each test instance and took the accuracy of the comparisons.

Dataset 2: relative cost

- On-Time Payment Streak (Months): integer representing consecutive months of on-time payments.
- Savings Rate Each Month (%): integer between 0-100.
- Business Revenue Growth Rate (%): integer between 0-100.

All these features where labeled as monotonically increasing in cost and mutated upwards a value of 5 when comparing cost. Initially, the datum is duplicated, one has a feature increased by 5, and then they both mutated this same feature upwards by 5 to do the pairwise comparison. The ground truth was specified to be that higher starting points would be higher cost.

Listing 2: Python code for generating synthetic dataset for 'Relative Cost'

```python
import numpy as np

np.random.seed(42)
n_samples = 1000

on_time_payment_streak = np.random.randint(0, 116, n_samples)
savings_rate_monthly = np.random.randint(0, 96, n_samples)
business_revenue_growth = np.random.randint(0, 96, n_samples)
```

Dataset 3: dependent cost

- Liquid Assets: the amount of liquid assets in dollars.
- DSCR: Debt Service Coverage Ratio, a float value.
- On-Time Payment Streak (Months): integer representing consecutive months of on-time payments.
- Business Revenue Growth Rate (%): integer between 0-100.
- Monthly Savings Rate (%): integer between 0-100.

The "target" dependencies tested for the latter are:

- Increasing liquid assets is hard with a low DSCR.
- Extending your on time pay streak is harder with low business revenue growth rate.
- Increasing monthly savings rate is harder with low liquid assets.

For all dependent features, they were mutated upwards a standard deviation. For the independent features, DSCR had either 0.5 or 2.0, Business revenue growth rate had either 10 or 50%, and liquid assets had either \$10,000 or \$80,000, all for the harder and easier mutations, respectively. Again, with the ground truth specified, we tested these dependencies on 300 test instances using accuracy as our metric.

Listing 3: Python code for generating synthetic dataset for 'Dependent Cost'

```
import numpy as np

np.random.seed(42)
n_samples = 1000

liquid_assets = np.random.randint(1000, 100001, n_samples)
dscr = np.random.uniform(0.5, 3.0, n_samples)
on_time_payment_streak = np.random.randint(0, 121, n_samples)
business_revenue_growth = np.random.randint(0, 96, n_samples)
monthly_savings_rate = np.random.randint(0, 96, n_samples)
```

F Temperature, Sparsity, and Information Bottlenecks

The role of noise in preference learning. While noise and temperature typically help prevent overfitting in preference learning (Laidlaw and Russell, 2021), we observe that this benefit diminishes when working with sparsely connected comparison graphs. In this section, we provide a detailed explanation of why low temperature is preferable in sparse settings.

Sparse comparison graphs. Our method learns a cost function $C : \mathcal{R} \to \mathbb{R}$ from a dataset of pairwise comparisons $\mathcal{Q} = \{(i, j, y)\}_{k=1}^{K}$, where each comparison corresponds to an edge in a comparison graph $G = (V, E)$ with nodes V representing recourses and edges E representing comparisons. For scalability, we aim to use sparse graphs where $K = |E| \approx O(N)$ rather than the dense case where $K \approx O(N^2)$. This sparsity regime makes each comparison edge $(i, j) \in E$ significantly more influential in determining the relative costs learned by $\hat{C}$, as each edge must convey more information about the global cost ordering.

Information bottlenecks in sparse graphs. The critical issue arises when the sparse graph G contains *information bottlenecks*—structural vulnerabilities where the graph can be partitioned into two large sets of nodes connected by only a small number of edges. Formally, consider a partition of V into two sets S and $V \setminus S$, each of substantial size, connected by a small cut set $E_{cut} = \{(i, j) \in E \mid i \in S, j \in V \setminus S\}$ with $|E_{cut}| \ll |S|, |V \setminus S|$.

In this scenario, *all* information about the relative costs between recourses in S and recourses in $V \setminus S$ must flow through the labels $\{y_{ij}\}_{(i,j)\in E_{cut}}$ associated with the edges in E_{cut}. If these labels are corrupted by noise (e.g., due to high temperature sampling from the language model), the learned cost function $\hat{C}$ will have systematically biased estimates of cost differences across the partition. This error cannot be corrected by comparisons within S or within $V \setminus S$, as these only provide relative information within each partition.

Error propagation through bottlenecks. To illustrate the severity of this issue, suppose $|E_{cut}| = m$ edges connect $|S| = n_1$ and $|V \setminus S| = n_2$ nodes, where $m \ll n_1, n_2$. Under a Bradley-Terry or similar preference learning model, the learned cost differences between S and $V \setminus S$ are determined by aggregating the m noisy labels. If these labels have noise with standard deviation σ, the uncertainty in the relative cost scale between the two partitions is approximately $\sigma/\sqrt{m}$. As m becomes small relative to n_1 and n_2, this uncertainty can dominate the

learned cost relationships, effectively "decoupling" the cost scales between the two partitions and leading to poor generalization.

Implications for temperature selection. Given these information bottlenecks, reducing temperature (and thus noise) in the language model's preference labels becomes critical for sparse graphs. Lower temperature increases the signal-to-noise ratio of the crucial bottleneck edges, reducing error propagation across graph partitions. This stands in contrast to dense graphs where $K \approx O(N^2)$: with many redundant paths between any two nodes, moderate noise can be averaged out, and higher temperature may even provide beneficial regularization. For sparse graphs, however, preserving label accuracy (particularly on structurally important edges) takes precedence over noise-induced regularization benefits.

G User study extra details

An example question from the human study for the HELOC dataset is shown in Figure 9. The full survey can be seen in the attached paper materials. We also present the full details of the rating study in Figure 8.

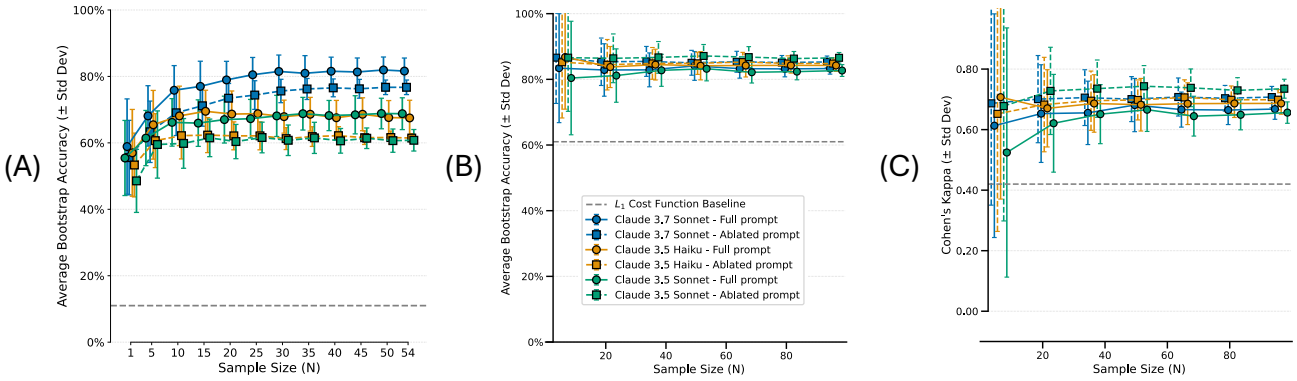

Figure 8: User testing: (A) Bootstrapping users in our study we found the overall accuracy better with the full prompt, which begun to plateau after N=30, with a mean accuracy of ∼82% in `Claude3.7-Sonnet`. (B) Taking the average of our ratings as the ground truth, and bootstrapping questions, the results were highly stable for all N, averaging ∼87% for the full prompt, and ∼80% for the ablated version. (C) Similar results were seen in the rater's study when comparing Cohen's Kappa.

H Finetuning fairness

As fairness is domain dependent (Mehrabi et al., 2021), we did not evaluate this aspect of cost functions in our quantitative tests, but here we investigate GPT-4o's built-in biases and the ability of prompt finetuning to modify them if needed. Hence, we added an extra rule to the full prompt that the LLM should *'never use demographic information when considering the cost of mutations'*, and tested against the same prompt with this rule ablated. For both we mutated education upwards a standard deviation to observe how cost differed between demographics and prompts. Specifically, we considered male/female, white/not-white, and age 25/65, while taking their MDE scores with *education-level* as the dependent feature. Figure 10 shows the results were the full prompt with these constraints was significantly less biased than the ablated version. Specifically, Cohen's d was -0.59, -0.35, and 1.43 for age, gender, and race, respectively, illustrating significant effect sizes.

I Deployment study on real recourse algorithms

This section serves to give full details about the implementation of our deployment tests mentioned in the final paragraph of Section 4.5. We implemented the methods of Keane and Smyth (2020) and Wachter et al. (2017). The data used was Adult Census with 30,000 for training, and 6,000 for testing the recourse generation.

I.1 Keane and Smyth (2020)

This method is data driven and works by defining a case-base of recourse options for training data (Keane and Smyth, 2020). In practice, each training data has its nearest unlike neighbor found in the case-base and the difference between the two is taken as one recourse option. Recourses of 2 or less feature changes are preferred by the authors, we focus on single feature changes. At test time, a query has its nearest neighbor found in the case base and its recourse is applied to the query, this is repeated for all nearest neighbors to find the best recourse option adhering to some constraints. For us, these constraints are a single feature mutation, and that the result must be a valid counterfactual. Finally, we also considered the 100 nearest neighbors as possible recourses.

I.2 Wachter et al. (2017)

A heavily implemented framework in research (Wachter et al., 2017), the method works by generating a set of random recourses which optimize to be closer to the query, while optimizing to also be the counterfactual class. The second constraint is gradually up-weighted with a lambda term to be more important throughout several optimization steps. We implement the method as normal with 300 possible counterfactuals during optimization, categorical features are snapped to the closest real value, the results are filtered to those which are valid counterfactuals, and the closest chosen as the answer. Because we are interested in sparse explanations, we also clamp each possible counterfactual to have one possible feature mutation, which in practice is done allowing the largest currently mutated feature to be the recommended recourse action.

I.3 Deployment study results

Here we display the full results. The frequency of change for each feature is displayed on each table.

Table 3: Case Study Results: Each number represents the number of times each method recommended mutating that feature for recourse. In Keane and Smyth (2020), our method recommended mutating age less and hours-worked more as the main trade-off. In Wachter et al. (2017), our method recommended mutating education and hours-work in comparison to MAD which favored features such as male, native-US, and race.

	Male	Age	Native-US	Married	Education	Hours-Work	Private Work	Caucasian
Keane and Smyth (2020) - Data Driven								
L_1	0	554	1	29	266	177	0	0
Ours	0	380	1	30	275	341	0	0
Wachter et al. (2017) - SGD Driven								
MAD	20	53	78	332	1594	4	8	1
Ours	1	28	3	313	1750	174	84	0

Table 4: Heloc Results: On average the baselines favored Months Since Most Recent Inquiry Excluding 17 days, in contrast to our cost function which favored Number of inquiries in the last 6 months and number of revolving trades with balance as a trade-off. Considering the first feature has no time span (i.e., it could be greater than 6 months), and NumInqLast6M is bounded, it is typically faster to take action upon.

	MSinceMostRecentInqexcl7days	NumRevolvingTradesWBalance	NumTradesOpeninLast12M	NumInqLast6M
Keane and Smyth (2020) - Data Driven				
L_1	336	9	1	0
Ours	270	43	8	25
Wachter et al. (2017) - SGD Driven				
MAD	167	1	0	0
Ours	5	10	5	148

J Compute resources

To reproduce our results, all that is needed is a CPU and access to OpenAI API and Anthropic API with model gpt-4o-2024-05-13 or the Claude models listed in the main paper. To run all tests will take 1-2 weeks minimum for labeling with the API and training of the cost functions across all experiments (including the finetuning tests). We were able to query 50x in parallel for our tests, if your system allowances are lower, reproducing the results will take longer, however we will include all the LLM labelled datasets to make reproducing the work trivial, requiring only a CPU.

K Dataset License

- **HELOC**: `https://github.com/samanthajmichael/heloc_credit_modeling` (MIT License)
- **Adult Census**: `https://archive.ics.uci.edu/dataset/2/adult` (CC BY 4.0)
- **German Credit**: `https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data` (CC BY 4.0)

L Training hyperparameters

To train the cost functions we used the libraries

`https://github.com/tombewley/hyperrectangles`

`https://github.com/tombewley/Reward-Trees`

With all the default hyperparameters for training the MLPs and decision trees.

M Prompt robustness tests

We ran 500 randomly sampled pairwise comparisons from each of the three datasets using our standard prompt and a perturbed version. The perturbation involved (i) replacing 10 random words with synonyms and (ii) prompting the LLM to reword one sentence (without modifying the feature values). We then ran both 30 times per question and checked if they produced the same modal response. The agreement rates were: HELOC: 95%, Adult Census: 94%, and German Credit: 94%. This illustrates that our results are quite robust to prompt perturbations.

N Prompts

In this section we detail the prompts used in this paper. In total, there were 30+ different prompts used, to avoid cluttering the appendix, and in the interest of saving paper usage, we only give the variants for the HELOC dataset, however, all other prompts are easily found in our source code. Go to `src/prompts.py` to see the prompts for the direct usage of LLMs, the full prompts, and the ablated versions, as well as other examples of finetuning the prompts to e.g. elicit fairness. Go to `src/synthetic_des1.py`, `src/synthetic_des2.py`, or `src/synthetic_des3.py` to see the prompts for the synthetic tests.

Ablated prompt for HELOC

You are a helpful assistant to a data scientist to help them label data.

You will be shown a datapoint representing a person Alex, and a mutation of it,

You will also be shown a datapoint representing a person Jaden, and a mutation of it,

your task is to label which of the two mutations would take more effort to achieve.

The data will be the HELOC Dataset which uses these features:

MSinceMostRecentInqexcl7days: Number of months passed since the last credit inquiry on the individual.

NumRevolvingTradesWBalance: The number of the individual's current credit accounts (e.g. credit cards) that have balances on them.

NumTradesOpeninLast12M: The number of new credit accounts opened in the last 12 months.

NumInqLast6M: The number of credit inquiries carried out on the individual in the last 6 months.

The data is represented in array form like ['MSinceMostRecentInqexcl7days', 'NumRevolving-TradesWBalance', 'NumTradesOpeninLast12M', 'NumInqLast6M']

Now consider the following individual Alex: {`x1`}

Now consider this mutation of Alex: {`x1 mutation`}

Now consider the following individual Jaden: {`x2`}

Now consider this mutation of Jaden: {`x2 mutation`}

Which of these two mutations would take more effort? You must provide an answer.

Outline your reasoning process step by step, before giving your answer as 1, 2, or 0 in the tags <answer>...</answer>, where 1 means you think the first mutation requires more effort, 2 means you think the second mutation requires more effort, and 0 means you think there is almost no difference.

Perceived Effort Required in Dataset Feature Mutations

HELOC Dataset

Data Description:

The FICO HELOC dataset contains anonymized information about home equity line of credit (HELOC) applications made by real homeowners. The customers in this dataset have requested a credit line in the range of USD 5,000 - 150,000.

Selected Features:

1. Months Since Recent Inquiries: Number of months passed since the last credit inquiry on the individual.

2. Number of Credit Accounts with Balances: The number of the individual's current credit accounts (e.g. credit cards) that have balances on them

3. Number of New Credit Accounts: The number of new credit accounts opened in the last 12 months

4. Number of Inquiries: The number of credit inquiries carried out on the individual in the last 6 months

* 2.

Alex	*Months since Recent Inquiry*	*Number of Credit Accounts with Balances*	*Number of New Credit Accounts*	*Number of Inquiries*
Features	6	3	2	4
Change(s) to Make			4	

Jaden	*Months since Recent Inquiry*	*Number of Credit Accounts with Balances*	*Number of New Credit Accounts*	*Number of Inquiries*
Features	6	3	2	4
Change(s) to Make		5		

Which individual's proposed change would require more effort?

- ◯ Alex
- ◯ Jaden

* 3. In your opinion, how much difference in effort do you perceive between the two changes (for Alex & Jaden) in the scenario above?

1 (Almost No Difference) — 7 (Really Large Difference)

Figure 9: Human study question example.

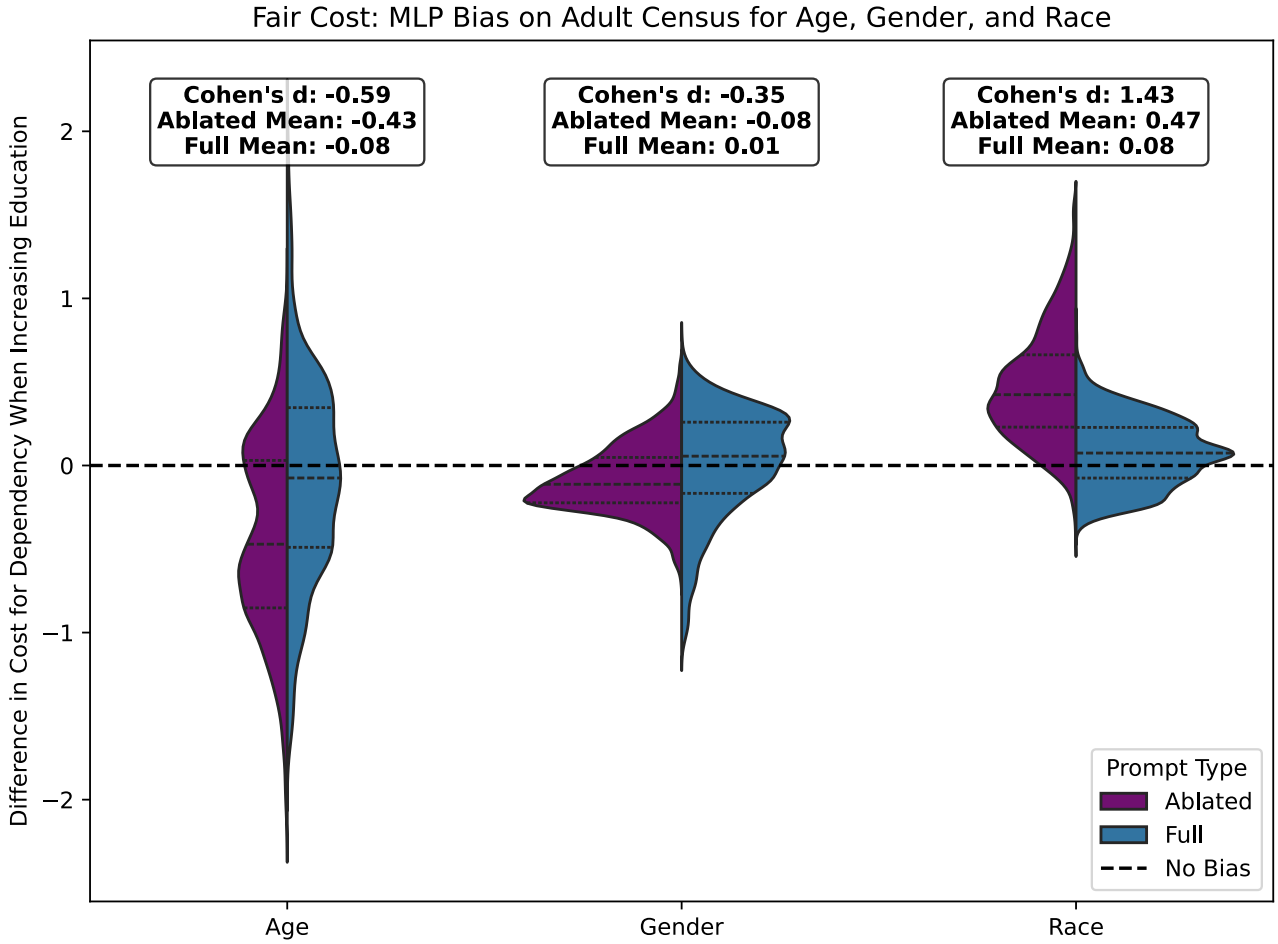

Figure 10: *Fair* Cost Fine-Tuning Results: Instructing GPT-4o to not discriminate between demographic information decreases its bias when suggesting recourse. These results show how it is possible to fine-tune the cost function on the dimension of *fair* cost. The "Full Prompt" here included an extra rule to not discriminate stating *'never use demographic information when considering the cost of mutations'*.

Full prompt for HELOC Dataset

You are a helpful assistant to a data scientist to help them label data.

You will be shown a datapoint representing a person Alex, and a mutation of it,

You will also be shown a datapoint representing a person Jaden, and a mutation of it,

your task is to label which of the two mutations would be more difficult to achieve.

The data will be the HELOC Dataset which uses these features:

MSinceMostRecentInqexcl7days: Number of months passed since the last credit inquiry on the individual.

NumRevolvingTradesWBalance: The number of the individual's current credit accounts (e.g. credit cards) that have balances on them.

NumTradesOpeninLast12M: The number of new credit card accounts opened in the last 12 months.

NumInqLast6M: The number of credit checks carried out on the individual in the last 6 months.

The data is represented in array form like ['MSinceMostRecentInqexcl7days', 'NumRevolving-TradesWBalance', 'NumTradesOpeninLast12M', 'NumInqLast6M']

Now consider the following individual Alex: {`x1`}

Now consider this mutation of Alex: {`x1 mutation`}

Now consider the following individual Jaden: {`x2`}

Now consider this mutation of Jaden: {`x2 mutation`}

Which of these two mutations would take more effort?

Remember the following rules and use them in your decision:

1. Some features are naturally harder to change than others, use this logic.

2. For numerical features, the difficulty of changing them can often depend on their starting values.

3. Apart from the mutated features, consider the other features which are different between Alex and Jaden, and how this may affect difficulty.

Outline your reasoning process step by step, before giving your answer as 1, 2, or 0 in the tags <answer>...</answer>, where 1 means you think the first mutation requires more effort, 2 means you think the second mutation requires more effort, and 0 means you think there is almost no difference.

Elicit Direct Cost: Full prompt for HELOC Dataset

You are a helpful assistant to a data scientist to help them label data.

You will be shown a datapoint representing a person Alex, and a mutation of it,

Your task is to label the effort to achieve this mutation.

The data will be the HELOC Dataset which uses these features:

MSinceMostRecentInqexcl7days: Number of months passed since the last credit inquiry on the individual.

NumRevolvingTradesWBalance: The number of the individual's current credit accounts (e.g. credit cards) that have balances on them.

NumTradesOpeninLast12M: The number of new credit card accounts opened in the last 12 months.

NumInqLast6M: The number of credit checks carried out on the individual in the last 6 months.

The data is represented in array form like ['MSinceMostRecentInqexcl7days', 'NumRevolving-TradesWBalance', 'NumTradesOpeninLast12M', 'NumInqLast6M']

Now consider the following individual Alex: {`x1`}

Now consider this mutation of Alex: {`x1 mutation`}

On a scale from 0-10, how much effort would this take Alex? You must provide an answer.

Remember the following rules and use them in your decision:

1. Some features are naturally harder to change than others, use this logic.

2. For numerical features, the difficulty of changing them can often depend on their starting values.

3. Apart from the mutated features, consider the other features, and how this may affect difficulty.

Outline your reasoning process step by step, before giving your answer in the tags <answer>...</answer>, where 0 means you think this is almost no effort, and 10 means you think it is almost impossible.

Elicit Direct Cost: Ablated prompt for HELOC Dataset

You are a helpful assistant to a data scientist to help them label data.

You will be shown a datapoint representing a person Alex, and a mutation of it,

Your task is to label the effort to achieve this mutation.

The data will be the HELOC Dataset which uses these features:

MSinceMostRecentInqexcl7days: Number of months passed since the last credit inquiry on the individual.

NumRevolvingTradesWBalance: The number of the individual's current credit accounts (e.g. credit cards) that have balances on them.

NumTradesOpeninLast12M: The number of new credit card accounts opened in the last 12 months.

NumInqLast6M: The number of credit checks carried out on the individual in the last 6 months.

The data is represented in array form like ['MSinceMostRecentInqexcl7days', 'NumRevolving-TradesWBalance', 'NumTradesOpeninLast12M', 'NumInqLast6M']

Now consider the following individual Alex: {`x1`}

Now consider this mutation of Alex: {`x1 mutation`}

On a scale from 0-10, how much effort would this take Alex? You must provide an answer.

Outline your reasoning process step by step, before giving your answer in the tags <answer>...</answer>, where 0 means you think this is almost no effort, and 10 means you think it is almost impossible.