Understanding the User Perception and Experience of Interactive Algorithmic Recourse Customization

SEUNGHUN KOH, KAIST, Daejeon, Republic of Korea
BYUNG HYUNG KIM, Inha University, Incheon, Republic of Korea
SUNGHO JO, KAIST, Daejeon, Republic of Korea

Generating actionable algorithmic recourse requires understanding each user's preferences. Users provide their relevant information, and the system uses it to generate recourse that can be easily followed by individual users. To gain insight into users' perceptions and experiences of this novel form of interaction, we developed a prototype that enables users to provide the required information for algorithmic recourse customization. With the prototype, we conducted a user study where participants customized the recourse. Through both quantitative and qualitative analysis, we found that: (1) repetitive user-AI interaction not only enables users to customize the recourse but also explore other possibilities, (2) users prefer recourse customization method that offers high controllability and understandability, and (3) degree of customization users want depends on various factors. With these findings, we discuss the implications for systems that aim to provide actionable algorithmic recourse in real-life situations.

CCS Concepts: • **Human-centered computing** → **User studies**; **Empirical studies in HCI**;

Additional Key Words and Phrases: Explainable artificial intelligence, algorithmic recourse, counterfactual explanation, personalization, customization

ACM Reference format:
Seunghun Koh, Byung Hyung Kim, and Sungho Jo. 2024. Understanding the User Perception and Experience of Interactive Algorithmic Recourse Customization. *ACM Trans. Comput.-Hum. Interact.* 31, 3, Article 43 (August 2024), 25 pages.
https://doi.org/10.1145/3674503

1 Introduction

With the increasing use of **machine learning (ML)** models to make high-stakes decisions that can have a significant impact on individuals' lives, there has been a growing interest in algorithmic recourse. Algorithmic recourse aims to suggest possible actions that individuals can take to make ML models produce a desired outcome. Recent approaches to generating recourse involve generating **counterfactual explanations (CE)**. CE aims to provide what-if scenarios: what minimal changes in the input could have resulted in the ML model making different decisions. CE will inform a

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (MSIT) under Grant No. RS-2023-00208052.
Authors' Contact Information: Seunghun Koh, KAIST, Daejeon, Republic of Korea; e-mail: shk0724@kaist.ac.kr; Byung Hyung Kim (Corresponding author), Inha University, Incheon, Republic of Korea; e-mail: bhyung@inha.ac.kr; Sungho Jo (Corresponding author), KAIST, Daejeon, Republic of Korea; e-mail: shjo@kaist.ac.kr.

ACM 1557-7325/2024/8-ART43
https://doi.org/10.1145/3674503

person whose loan request has been denied that "if your income were $1000 higher than it is now, your loan request could have been accepted." Such suggested changes translate into a set of actions that can be taken by the recipients (e.g., "if you can increase your income from $10,000 to $11,000, then your loan request can be accepted."), thereby becoming an algorithmic recourse.

For this approach to be useful, it is essential for the suggested actions to be actionable. That is, users should be able to easily execute those actions [28]. For example, suggesting that a person needs to increase his income by $10,000 when the maximum amount he can increase is $100 is not useful at all. The main issue with generating actionable recourse is defining what is actionable for each user, as the concept of actionability ultimately depends on individual users. The cost of executing the same actions will differ between users and may even depend on the situations they are in [5]. This makes it almost impossible to use a one-size-fits-all approach to provide actionable recourse to all users [47]. Thus, it is mandatory to elicit users' different preferences and apply them to the recourse generation process [5, 47, 54]. To support users' preference elicitation, researchers began to apply interactive **explainable AI (XAI)** to generate algorithmic recourse. Through iterative user–AI interaction, users configure their preferences and the system provides personalized recourses based on the information provided. In many cases, users express their preferences by providing two types of actionability constraints: action constraints that define a set of actions that can be executed by individual users and priority constraints that define the cost function that measures the difficulty of executing certain actions. Going back to the previous scenario of a loan request, users could provide action constraints by notifying the system that "I can't increase my income by more than $500" or "It's impossible for me to get a higher credit rating as of now." On the other hand, they could provide priority constraint by comparing the difficulty of different actions, such as "Increasing my income is slightly easier for me than improving my credit ratings." To facilitate interaction between users and AI, it is important to design a user interface that helps users accurately express their preferences.

With the increasingly important role of AI in user interfaces, the **human–computer interaction (HCI)** communities have emphasized the necessity of understanding user experience with those AI interfaces from novel viewpoints [18, 38, 53]. Recent studies have shown that understanding diverse aspects of user-AI interaction provides a new perspective on how to meet the actual needs of users when it comes to AI interfaces [7, 33, 46, 57]. In light of this, designing interfaces for interactive algorithmic recourse customization requires understanding and analyzing how users utilize various actionability constraints to elicit their preferences, as well as how users perceive and experience the interactive recourse customization.

This article serves as a preliminary exploration of interactive recourse customization, focusing on the user aspect of this new interaction. Specifically, we aim to investigate how users' perceived workload is affected by interactive algorithmic recourse customization, as well as their perception and experience of this customization. For this purpose, we designed a prototype to simulate algorithmic recourse customization within the context of a loan application. As an applicant whose loan request has been rejected, users interact with the system by providing action and priority constraints. In turn, the system generates algorithmic recourses based on those constraints. With the prototype, we conducted a user study using both qualitative and quantitative approaches. The results of this study showed that:

—Users perceive the interaction as a necessary feature for not only customizing recourse but also exploring other possible recourses.
—Users prefer and find it less demanding to provide action constraints rather than priority constraints. This is because action constraints are perceived to be more controllable and understandable in the process of customizing recourses.

—While providing diverse actionability constraints helps users find useful recourse, users might not always perceive them as useful or necessary.

Based on these findings, we discuss their implications for designing AI interfaces that aim to provide actionable guidance using algorithmic recourse.

Our main contributions are as follows:

—We proposed users' workload as a necessary criterion to be considered while evaluating interactive algorithmic recourse customization.
—Through both quantitative and qualitative analysis, we observed and analyzed the process of interactive algorithmic recourse customization and discovered intriguing aspects of this user–AI interaction.
—We discussed the potential implications of designing interactive algorithmic recourse customization interfaces, which would allow users to easily and effectively configure their preferences and receive well-customized recourses.

2 Background and Related Works

In this section, we provide background on algorithmic recourse, focusing on (1) its generation, (2) proposed approaches for generating actionable algorithmic recourse, and (3) the use of interactive factors to support the generation of actionable algorithmic recourse. Furthermore, we will introduce why interaction has been emphasized in XAI and discuss the emerging problems in interactive XAI.

2.1 Algorithmic Recourse

Algorithmic recourse, which refers to a set of actions that can reverse the decisions made by ML models, has gained attention due to its ability to assist users in determining what steps to take in response to the model's decision. Starting from Wachter et al. [56], who first formulated the problem of generating recourse as an optimization problem, different works have proposed generating recourse by finding CE. The main idea is to identify a set of counterfactual instances such that (1) are close to the input instance and (2) ML model makes a different decision for them than it did for the input instance. Change of features suggested by the CE then becomes the algorithmic recourse. Keys to generating CE are the underlying cost functions, which determine the level of difficulty for users to successfully execute certain actions and the candidate pools of counterfactual instances from which the CE are searched.

The central issue in generating a recourse is creating a useful recourse. This requires the recourse to not only propose minimal changes but also suggest a set of actions that the recipient can execute. In other words, the recourses should be actionable to users. To generate more actionable recourse, researchers have proposed several approaches, such as selecting actual data instances as CEs [40, 61], identifying likely changes in features by referring to the distribution of the data [24], choosing CEs that are close to pre-generated data prototypes [51], categorizing features based on their actionability [27, 50], and restricting how features can change [26]. It has even been suggested to generate diverse CEs [37, 42] to increase the likelihood of the generated recourses being actionable.

However, actionability is a concept that depends entirely on individual users [29, 48, 55]. An action that is executable for the majority of users might not be for some individuals, and the concept of "minimal" change will also vary among individual users. Thus, generating an actionable recourse requires capturing users' individual preferences for executing different actions and applying them to the recourse generation process. Researchers turned to interactive XAI to solve this issue. Systems allow users to customize the two most influential factors in algorithmic recourse generation: the cost function and a set of executable actions. Users provide constraints such as the actions they can execute in their current situation (i.e., action constraint) [13, 45, 59] and even the difficulty of

executing those actions (i.e., priority constraint) [37]. The system generates a CE based on the user's actions, taking into account the difficulties provided. To further support users in their interaction with XAI for personalized recourse generation, recent works also make use of interactive visual interfaces [11, 19].

Despite the proposal of various methods for customizing recourse, very few of them have investigated how end-users utilize such functionalities to express their preferences. With the widespread use of AI interfaces, HCI researchers emphasize the necessity of investigating the user experience with these interfaces, understanding users' needs, and designing user-centered AI interfaces. In this study, we investigated how users utilize and perceive various actionability constraints and how they affect users' workload and overall experience. These findings will provide a step towards designing user-centered interactive recourse customization interfaces.

2.2 Interactive XAI and Cognitive Workload

Recently, there has been a surge of calls from HCI communities to take a more user-centered approach in XAI. To make explanations more understandable and accessible, researchers sought to draw insights from the field of social science on how people generate and utilize explanations. They anticipated that users would desire explanations in a similar manner from XAI. Millers' paper is one representative example of such works [36], where it is found that explanations are consumed through interactions between the explainers and explainees. According to Miller, the explanation is a two-stage interaction where the explainer diagnoses why an event has occurred and then transfers their knowledge to the explainees in a way that answers explainee's questions, rather than a one-way process where the explainer simply transfers their knowledge. The theme of interaction in XAI was further supported by the need for personalized explanations. Recent studies have emphasized the importance of generating explanations that are tailored to the recipients' specific needs and the reasons behind their desire for an explanation [2, 3, 16, 47]. As a way to cater to users' various needs for explainability, researchers have proposed interaction between the user and the system as a method for personalizing explanations. Allowing users to guide and customize the explanation through interaction enables them to generate explanations that satisfy their specific explainability needs. This makes the explanations more understandable and consequently more appealing to users [44, 47]. These findings have paved the way for interactive XAI, where users and the XAI system engage in bidirectional communication, where the XAI system provides a variety of explanations to answer any queries made by users.

While the concept of interactivity has been well applied from XAI algorithms to interfaces, there has been growing concern regarding the effect of users' increased cognitive workload on user-XAI interaction. In HCI communities, it has generally been agreed that the cognitive workload of users caused by using an interface should be kept low or managed at an acceptable level to improve usability [30]. When it comes to XAI, the cognitive workload of users becomes more than just a usability issue. Multiple studies have reported that users find popular XAI techniques and the explanations generated by these techniques to be difficult to use and cognitively demanding [33, 62]. Unfortunately, such a high cognitive workload has been associated with the problem of XAI pitfalls—the unexpected negative effects of AI explanations—such as over-reliance [4, 39, 58, 63]. Researchers have suggested that due to the high cognitive workload required to understand the explanation, people tend to superficially associate the ability to explain with trustworthiness instead of engaging analytically with the explanation [9, 17]. In turn, the necessity of considering users' cognitive workload while designing XAI techniques, explanations, and methods of user-XAI interaction is being emphasized [52].

Motivated by these prior works, our research aims to explore the process of users' recourse customization within the framework of interactive XAI. We measured not only users' experience but

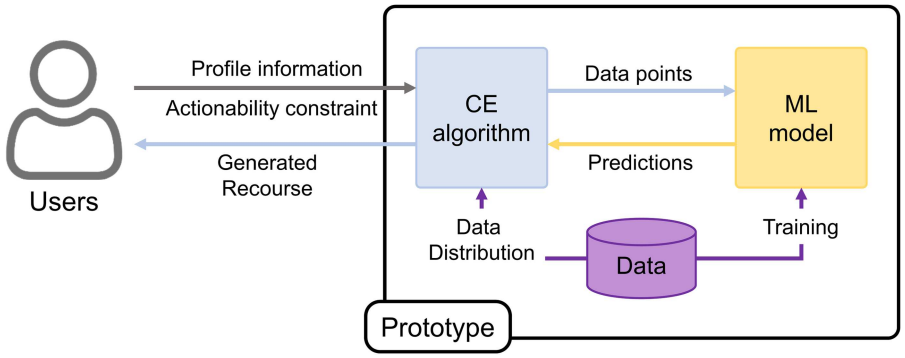

Fig. 1. Prototype overview. During the interaction, users provide actionability constraints to the system. The CE algorithm applies these constraints and utilizes the ML model's predictions to generate personalized actionable recourses for each user. In the process, the CE algorithm makes use of the data distribution (i.e., range of values for each feature) to (1) generate CEs within a plausible boundary and (2) calculate the inverse median absolute deviation (MAD) to be used for feature-wise weight, which scales the changes made in different features.

also their cognitive workload in utilizing different recourse customization techniques to understand various aspects of recourse customization. Based on those findings, we propose design rationales for user-centered and easy-to-use recourse customization interfaces.

3 Prototype Design

To understand the various aspects of algorithmic recourse customization, we developed a prototype in which the user and the AI engage in continuous interaction to create personalized recourse. When users provide their actionability constraints, the CE algorithm uses this information to provide personalized recourse, which describes the set of actions that users need to take for the ML model to make different decisions. As for actionability constraints, we chose action and priority constraints, which are the two most influential constraints in the CE generation process. These constraints have been used by many previous researchers to customize algorithmic recourse [11, 13, 19, 37, 45, 59]. An overview of CE generation process done by the prototype is shown in Figure 1.

3.1 Algorithmic Recourse Generation

Our prototype generates algorithmic recourse using **Diverse Counterfactual Explanations (DiCE)** [37]. DiCE is an optimization-based framework dedicated to generating diverse CE. For each input instance, DiCE searches for CEs that (1) are close to the input instance, (2) have different ML model predictions compared to the input instance, and (3) are as dissimilar to each other as possible. Our prototype presents algorithmic recourse to users by incorporating changes suggested by generated CEs. Just like any other CE generation algorithms, DiCE utilizes a cost function and a candidate pool of counterfactual instances to generate a set of feature-wise changes (i.e., actions). With an input instance, x, a pool of counterfactual instances, P, cost function, C and ML model, f, DiCE performs the optimization process as follows:

$$\text{find } \{p_i\} \subset P \text{ s.t. } f(p_i) \neq f(x)$$

$$\text{while minimizing } \sum C(x, p_i) \text{ and maximizing the diversity between } p_i.$$

For diversity, DiCE utilizes the concept of "Diversity via Determinantal Point Processes" [37], but the specifics will not be discussed here. As for the cost function of two data instances, DiCE

measures the cost of actions required to move from one instance to the other based on the average feature-wise difference between those instances. Specifically, for any instances x and y,

$$C(x, y) = \frac{1}{d_{cont}} \sum_{p=1}^{d_{cont}} w^p_{cont} |x^p - y^p| + \frac{1}{d_{cat}} \sum_{p=1}^{d_{cat}} w^p_{cat} I(x^p \neq y^p), \tag{1}$$

d_{cont} and d_{cat} represent the number of continuous and categorical features, respectively. x^p and y^p denote the p-th continuous or categorical feature values of x and y, while w^p represents the weight associated with each feature. Prototype sets default values of w^p to be the inverse **median absolute deviation (MAD)** of the feature in the dataset for continuous features and simply 1 for categorical features. Using inverse MAD as the weight for continuous features was first suggested by Wachter et al. [56] to control the difference of variability between feature values. Thus, with the default weight, it assumes that users have a similar difficulty in changing feature values.

For recourse customization, our prototype allows users to configure two types of constraints. Action constraints are used in our prototype to limit the search to only the actions that users can execute. Our prototype allows users to provide action constraints in two ways. First, they can specify that the values of certain features cannot change. For example, if a user states that they cannot make changes to the loan grade, the prototype will only search through a set of counterfactual instances that have the same loan grade as the user's loan grade. Alternatively, they can provide actionable ranges of values for each feature. If the user indicates that they can increase their annual income by up to $200, the prototype will search within a set of instances where the difference between the annual income of the instances and the user does not exceed $200.

Additionally, users can provide priority constraints that define the difficulty they face when executing certain actions. For example, they might say, "It is almost impossible for me to change my home ownership status" or "I could easily wait for another 2 years to increase my employment length." Prototype receives these priority constraints in the form of feature-wise weights that affect the cost function. Specifically, those weights act as feature-wise multiplicative factors in the cost function, as shown in Equation (2). For example, if the user provides a weight of two for the annual income, it doubles the cost of making changes to the annual income.

$$C(x, y) = \frac{1}{d_{cont}} \sum_{p=1}^{d_{cont}} w'^p_{cont} w^p_{cont} |x^p - y^p| + \frac{1}{d_{cat}} \sum_{p=1}^{d_{cat}} w'^p_{cat} w^p_{cat} I(x^p \neq y^p). \tag{2}$$

3.2 User Interface

While previous works have dealt with how to design the interface for algorithmic recourse customization [11, 19], the customization methods were limited to restricting certain feature values or providing actionable ranges for continuous features. As our prototype aims to enable users to provide action and priority constraints for both continuous and categorical features, we have developed our own interface design to allow users to provide a wide range of actionability constraints. Before designing our interface, we referred to works on algorithmic recourse and interactive ML to come up with two design rationales that our interface should focus on.

- *R1 Detailed input and control*. Researchers have shown that granting users control over AI increases their satisfaction and acceptance of the system [22, 23, 41]. Furthermore, Smith-Renner et al. showed that users prefer to provide detailed feedback on the decisions made by ML models [46]. We designed our interface so that users can control the recourse customization process by providing their constraints in a basic but detailed way.
- *R2 Showing diverse recourse*. Previous works on recourse have emphasized the importance of providing a diverse set of recourse to end-users [5, 37, 42]. Researchers argue that it will

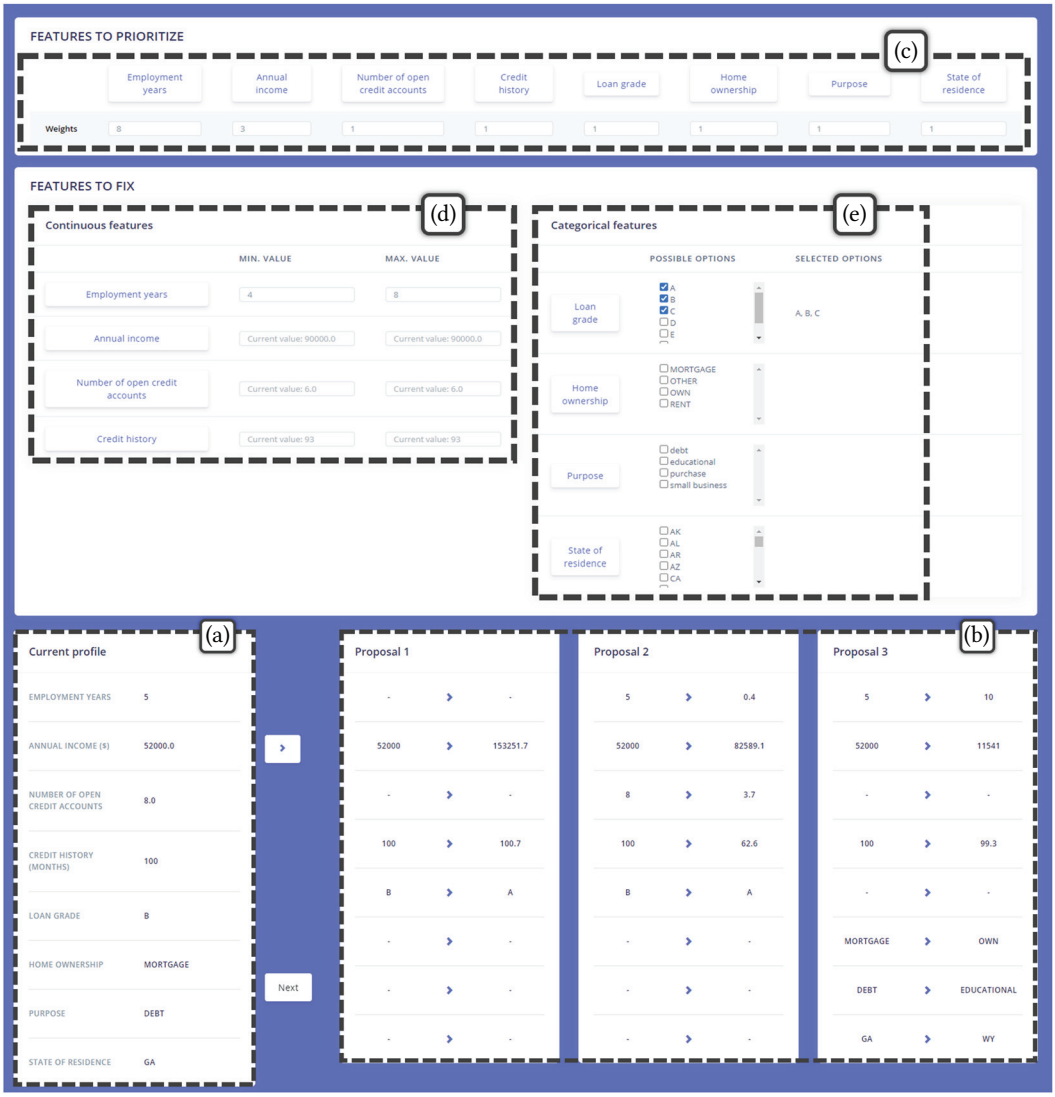

Fig. 2. Graphical user interface of the prototype. Users can provide feature-wise action (d and e) and priority (c) constraints. On the bottom, user's current status is shown (a) along with the generated recourse (b).

not only increase the chance of end-users finding more actionable recourse but also enable them to compare proposed recourses and consider possible alternatives that may be more actionable. Our interface displays multiple (up to three) recourses that match user-provided constraints.

Based on these rationales, we implemented a prototype interface that allows users to check their current status, provide appropriate constraints, and receive generated recourse. It allows users to directly control the CE generation process by providing two types of constraints. Our user interface can be divided into three parts.

—*Applicant profile and generated recourse* (Figure 2(a) and (b)): Profile information of the loan applicants is displayed on the left side of the interface under the heading *Current profile*

(Figure 2(a)). When participants provide actionability constraints, personalized recourses are displayed on the right under the name of *Proposals* (Figure 2(b)). For each participant's request, our system generates and displays up to three personalized recourses (R2). Each proposal contains a different set of feature changes that need to be made for the loan application to be accepted. The interface only displays the features for which the values change in each proposal, allowing participants to focus solely on the features that need to be changed.

—*Priority constraints* (Figure 2(c)): Users can provide priority constraints within the UI box labeled *FEATURES TO PRIORITIZE*. To support users in providing detailed inputs (R1), we have enabled users to configure the priorities of each feature. Users can provide feature-wise weights or click a button to automatically set the weight to the highest value within the available range. These feature-wise weights are directly applied to the cost function (Equation (2)).

—*Action constraints* (Figure 2(d) and (e)): Users can provide action constraints through the UI box labeled *FEATURES TO FIX*. Our prototype offers two functionalities for users to input their action constraints (R1). Users can either click the buttons to prevent changes to certain features or provide an actionable range for each feature. Users can provide minimum and/or maximum values for each continuous feature (Figure 2(d)). On the other hand, for categorical features, users can click the checkboxes to restrict the feature values that the system can use when generating CE (Figure 2(e)). For the convenience of users, a list of all values for each feature that users checked is shown under the column *SELECTED OPTIONS*.

4 User Study

4.1 Participants

We recruited 30 participants (26 males and four females) with a mean age of 27.17 and a standard deviation of 3.93 from our local institution. For the user study, we prepared a guideline containing details of the experiment and a brief description of each functionality in the prototype. Participants could refer to the guideline at any time during the user study and could also request further clarification on any content in the document. Each experiment lasted approximately 1 hour and 30 minutes, including the post-hoc interview. We gave $10 to each participant as an incentive.

4.2 Task and Dataset

User study was conducted based on a simulated task of loan application and recourse customization. Participants played the roles of loan applicants whose loan requests had been rejected by the ML model and were tasked with interacting with the system to find a useful recourse.

For this simulated task, we used the Lending Club dataset [1] as the basis for our user study pipeline. The Lending Club dataset, provided by an online peer-to-peer lending company in the United States, contains profile information of all loan applicants from 2007 to 2011 and indicates whether they have defaulted on their loan. It consists of over 100 features that characterize each applicant, and some of them require users' expertise in the financial field to understand their meaning. To simplify the interaction task, we consulted previous analyses on the LendingClub dataset [14, 21, 49] and selected 8 features that have a significant impact on whether applicants defaulted on their loans and are familiar with the general public.

—Employment year: Length of Employment in years
—Annual income: Annual income in dollars self-reported by the applicant
—Number of open credit accounts: Number of open credit lines in the applicant's credit file
—Credit history: Length of time in months since the applicant's credit line was opened

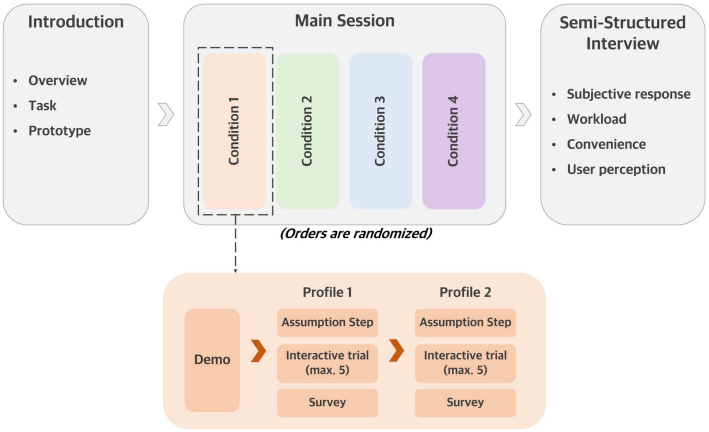

Fig. 3. Flow of the user study. User study is composed of an introduction, main session, and semi-structured interview. During the main session, participants underwent each of the four experimental conditions in a random order. In each condition, they were introduced to the functionality of an accompanying interface and tasked with finding a useful recourse as a simulated person whose loan request had been rejected. Participants repeated the task twice for each condition. After each task, participants filled out a survey asking for their assumptions and opinions toward the generated recourse and workload (measured on a Likert scale). When participants underwent all four experimental conditions, a semi-structured interview was conducted. During the interview, we asked for their perceptions and experiences of going through each condition and using actionability constraints.

—Loan grade: Loan grade issued by Lending Club
—Home ownership: Applicant's home ownership status
—Purpose: Purpose of borrowing the loan
—State of address: The state where the applicant currently resides

The ML model is responsible for predicting whether an applicant will default on their loan or not. To be specific, the system receives data instances containing eight selected features (applicant's profile information) and produces the expected status of their loan (whether it is defaulted or not). For the architecture and training process of the ML model, we referred to Mothilal et al. [37]. The ML model consists of a single hidden layer with only five neurons. Increasing the number of layers and neurons has worsened the model's generalization ability. To address the issue of imbalanced labels in the dataset, we employed the SMOTE algorithm [10] to oversample data instances in the minority group. For training, we divided the dataset into training and validation sets using an 80:20 ratio.

4.3 Procedures

The study was conducted in a 2×2 within-subject design. We generated four experimental conditions by combining factors of whether action and priority constraints were available in the user interface—(action (A)/no action (N) constraints and priority (P)/no priority (N) constraints). These four conditions are referred to as N-N, N-P, A-N, and A-P in the rest of the article. Participants experienced all the conditions in a randomized order to reduce the bias caused by the sequence of conditions.

During the user study, each participant underwent an introduction, main session, and semi-structured interview (Figure 3). In the introduction phase, participants were introduced to their tasks and the overall functionalities of the prototype. In the main session, participants went through

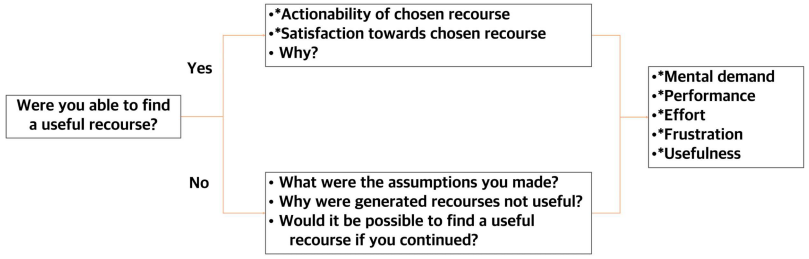

Fig. 4. Flow of the survey. Criteria that are answered with Likert scales are indicated with an asterisk. Based on whether participants were able to find a useful algorithmic recourse, they were guided to different parts of the survey. In the end, participants all responded to questions asking for their acceptance of the generated recourse, their cognitive workload during the interaction, and the usefulness of the provided interface. The full survey is included in Appendix A.

the previously introduced experimental conditions in a randomized order. For each experimental condition, user interfaces are composed differently. For all four conditions, both panel A and B are present. Then, for conditions where participants are allowed to provide action (or priority) constraints, panels D and E (C) are displayed alongside panels A and B, respectively. For example, in the N-P condition, panels A, B, and C are displayed together. In the A-N condition, panels A, B, D, and E are displayed together.

Within each condition, participants went through a series of steps, as shown in Figure 3. They were first introduced with specific functionalities for each condition and demonstrations on how to use them. At the beginning of each assumption step, participants were randomly assigned one of the applicants' profiles. We selected these profiles from the Lending Club dataset before conducting the user study. These profiles belong to individuals who have defaulted on their loans. Participants looked at the assigned profile of an applicant (Figure 2(a)) and imagined themselves as the applicant. To support participants in better immersing themselves in the assigned applicant, we allowed them to make any assumptions about their current situation as long as they were valid and did not contradict the information provided in the assigned profile.

During the interaction step, participants could provide the system with available constraints to generate useful recourse. For conditions labeled as "A," participants could provide action constraints (Figure 2(d) and (e)), while for conditions labeled as "P," they could provide priority constraints (Figure 2(c)). For the N-N condition, participants were unable to provide any constraints and could only receive recourse from the system without any interaction. The interaction ended either when participants were able to find a useful recourse or when they interacted with the prototype more than five times. We intentionally limited the maximum number of interactions because our goal was to encourage participants to actively engage in the task by imposing meaningful actionability constraints. Furthermore, the prototype took a maximum of 90 seconds to generate recourses for a single interaction. Letting participants interact with the system indefinitely may prolong the duration of the user study, potentially leading them to abandon the interaction.

After the interactions were completed, participants filled out a survey based on their interactions. The overall flow of the survey is shown in Figure 4. Depending on the outcome of their interaction, participants were directed to different sections of the survey. If they found a useful recourse, they were asked to rate its actionability and their satisfaction with the recourse based on the assumptions they made, along with their reasons. If not, they were asked to write down the assumptions they made and explain why they were not satisfied with the generated recourses. Furthermore, since participants were limited to a maximum of five interactions, we inquired about their opinions

on whether the system would eventually produce a useful recourse if the interactions were to continue. In the end, all participants were asked to provide their ratings on perceived workload and usefulness. All ratings were given using a 7-point Likert scale. For the purpose of analysis, we recorded the screen and interaction logs for each participant. Participants repeated the process of finding useful recourse twice for each experimental condition, each with a different assigned profile.

Furthermore, we conducted a qualitative study using the think-aloud method and post hoc semi-structured interviews. During the experiment, we utilized the think-aloud method [32], in which participants were encouraged to openly express their opinions regarding the experimental conditions they encountered. In the post hoc interview, we asked participants about their thoughts and impressions on each condition and actionability constraints. During the interview, we provided the recorded screens and voices from the experiment if they requested to refer to them. All interviews were audio-recorded.

4.4 Evaluation Metrics

To examine the impact of recourse customization techniques on users' ability to find actionable recourse and their overall experience, we developed a survey for participants to complete following a series of interactions. We referred to evaluation metrics in [44] that have been specifically proposed to evaluate explanation personalization methods. Out of them, we chose to primarily focus on participants' workload, inspired by previous works on interactive XAI. Within the survey, participants were asked to fill out the following:

Participants' acceptance of generated recourse. The purpose of customizing recourse customization is to provide users with actionable recourse that are acceptable to the recipients. To evaluate participants' acceptance, we asked them to assess the perceived actionability of the generated recourse (Can you follow the guide suggested by the recourse? How feasible is it to execute?) and the perceived satisfaction towards the generated recourse (Are you satisfied with the generated recourse?).

Participants' workload. User-led customization of recourse naturally is accompanied by an increased workload for them. To measure participants' workload in customizing recourse, we selected a subset of criteria from the NASA-TLX questionnaire [20]. We selected four criteria that are relevant to our study. These include mental demand (i.e., level of mental activity in interacting with the system), performance (i.e., degree of how effectively the participants were able to influence the recourse generation process?), effort (i.e., level of overall effort exerted to interact with the system, which includes not only mental demand but also cognitive energy or level of concentration) and frustration (i.e., level of irritation, stress, annoyance in interacting with the system).

Usefulness. Unlike other research on CE or algorithmic recourse, which focuses on the usefulness of the output (i.e., explanation or recourse), this study examines the usefulness of combinations of actionability constraints for customizing the recourse. Usefulness consists of a single measure to obtain a rough idea of how participants perceive the provided condition, and a more detailed survey about the usefulness is conducted in the semi-structured interview.

In this work, we did not use any other explainability-related metrics for the evaluation, such as the users' understandability of the model or users' trust towards the model. This is because our work focuses on algorithmic recourse, which aims to recommend an actionable set of actions rather than explaining the decision-making process of the ML model to users.

4.5 Analysis Method

4.5.1 Quantitative Analysis. For quantitative analysis, we utilized two analysis methods. Through **Cumulative Link Mixed Model (CLMM)** analysis and post-hoc pairwise comparison, we aimed

to investigate whether there is a significant difference in participants' responses to survey questions between each experimental condition. Furthermore, by analyzing participants' interaction logs, we aimed to investigate how participants' interactions differed between experimental conditions and how different actionability constraints affected participants' interaction with the prototype. In this section, we focus on analysis methods specifically for CLMM analysis and post hoc pairwise comparisons based on CLMM analysis.

We investigated the effect of customization techniques on each criterion in the questionnaire. We chose CLMM as our main method of analysis because of not only that the collected data did not follow normal distribution, but also due to the characteristics of our user study. Our user study requires participants to provide multiple responses within a single condition (even though the assumptions are different), which possibly violates the assumption of independent observation. Furthermore, with a simulated task and accompanied restrictions on the number of interactions, there could be other external factors affecting participants' responses, such as the fact that participants engage in the interaction as virtual entities or even simply the number of interactions they did. By adding these factors to the random variables of CLMM analysis, we sought to purely focus on the effect of diverse actionability constraints on participant's workload and experiences.

With the R package *ordinal* [12], CLMM models were constructed for each criterion in the questionnaire. Fixed effects factors included the provision of action and priority constraints, as well as their interaction, to account for the impact of individual conditions on users' perceptions of the interaction. Random factors include the unique identifier given to each participant to account for the possible effect of individual participants interacting with the system not as themselves but as simulated individuals. We also included the number of interactions conducted by each participant and whether they found useful recourses as the random factors. As previously mentioned, our prototype takes a maximum of 90 seconds to generate a set of recourse plans. It has been noted that such latencies, even on a very small scale, affect the user experience of visual analytics applications [25, 34]. Furthermore, restrictions on the number of possible interactions have led to instances where participants could not find a satisfactory recourse plan. Because the participants' task was to find a satisfactory recourse plan, their response could have been negatively affected if they were unable to find one. To mitigate the potential impact of these factors, we incorporated two previously mentioned variables as random factors to account for the potential influence of our interface on user experience.

Each CLMM model was initially constructed with a maximal random effects structure [6, 35]. If a model did not converge or the likelihood test [8] revealed that it was not statistically significant, we employed a model selection process, where we systematically removed each random effect factor based on their predictive power. After constructing the model, we conducted a Tukey's post hoc pairwise comparison test using the R package *lsmeans* [31] to compare the participants' responses for each condition. Results of the model construction and post-hoc test are shown in Tables 1 and 2.

4.5.2 Qualitative Analysis. For qualitative analysis, we used an iterative inductive analysis [43], consisting of three stages. We first transcribed the semi-structured interviews and participants' comments from the think-aloud sessions verbatim. In the second stage, we analyzed the acquired transcripts and identified approximately six to eight key observations per participant. These observations effectively summarize their responses to each experimental condition and the process of interactive recourse customization. Based on those observations, we extracted keywords that are relevant to the observations. In the final stage, all extracted keywords were compared and grouped into several themes related to participants' usage of each actionability constraint, their perception of different constraints, and their experiences with algorithmic recourse customization.

Table 1. Effect of Action and Priority Constraints on Metrics

	Actionability				Satisfaction			
Predictor	Est.	Std. Err.	z	Pr(>|z|)	Est.	Std. Err.	z	Pr(>|z|)
Action(o)	1.2243	0.2434	5.030	<.001***	0.9615	0.2209	4.352	<.001***
Priority(o)	0.6495	0.2238	2.902	0.004**	0.4423	0.2096	2.110	0.035*
Action(o) : Priority(o)	−1.1236	0.3306	−3.398	<.001***	−0.6172	0.3018	−2.045	0.041*
	Mental demand				Performance			
Predictor	Est.	Std. Err.	z	Pr(>|z|)	Est.	Std. Err.	z	Pr(>|z|)
Action(o)	0.3325	0.1964	1.793	0.091	1.2268	0.2052	5.979	<.001***
Priority(o)	0.8817	0.2042	4.318	<.001***	0.8215	0.1969	4.173	<.001***
Action(o) : Priority(o)	−0.7613	0.2959	−2.573	0.010*	−0.1729	0.2690	−0.643	0.520
	Effort				Frustration			
Predictor	Est.	Std. Err.	z	Pr(>|z|)	Est.	Std. Err.	z	Pr(>|z|)
Action(o)	0.0937	0.1840	0.509	0.610	−0.7107	0.2274	−3.125	0.002**
Priority(o)	0.9644	0.1938	4.976	<.001***	0.1738	0.2129	0.816	0.414
Action(o) : Priority(o)	−0.4857	0.2693	−1.804	0.071	0.0019	0.3061	0.006	0.995
	Usefulness							
Predictor	Est.	Std. Err.	z	Pr(>|z|)				
Action(o)	1.5934	0.2136	7.460	<.001***				
Priority(o)	1.1437	0.1992	5.742	<.001***				
Action(o) : Priority(o)	−0.9034	0.2710	−3.334	<.001***				

Est., estimate; Std. Err., standard error. Statistically significant results are reported with $^{***}p < 0.001$, $^{**}p < 0.01$, and $^{*}p < 0.05$.

Each stage was performed by one author, and the analysis process has been discussed with the others to determine whether the analysis performed at each stage appears biased or not.

5 Quantitative Analysis

Through the user study, we collected participants' responses to the survey, screen recordings of their interactions with our prototype, and their interaction logs. With CLMM analysis, we aim to examine the effect of actionability constraints on participants' acceptance of generated recourse, perceived workload, and usefulness. Furthermore, we used log analysis to examine how various actionability constraints impact participants in customizing the recourse. We have categorized our findings into four themes. For the overall distribution of participants' responses, please refer to Figure 7 in Appendix B.

5.1 Effect of Recourse Customization

From post-hoc pairwise comparisons, we observed a clear benefit of providing any type of constraint. Compared to the non-interaction condition, participants were able to find more actionable and satisfactory recourse. Moreover, with the provision of actionability constraints, participants reported

Table 2. Result of *Post- Hoc* Pairwise Comparison of Criteria between Conditions

	Comparison 1 N-N : A-N			Comparison 2 N-N : N-P			Comparison 3 N-N : A-P		
Criterion	diff.	z	p	diff.	z	p	diff.	z	p
Actionability	2.855	5.164	<.001***	2.042	3.913	<.001**	2.650	4.936	<.001***
Satisfaction	1.977	4.024	<.001***	1.253	2.642	0.041*	1.985	4.079	<.001***
Mental demand	1.232	2.749	0.031*	2.008	4.403	<.001***	1.717	3.894	<.001***
Performance	1.908	4.623	<.001***	1.335	3.337	0.005**	2.897	6.505	<.001***
Effort	0.618	1.568	0.397	1.850	4.473	<.001***	1.496	3.373	0.001**
Frustration	−1.007	−2.087	0.157	0.244	0.541	0.949	−0.759	−1.621	0.367
Usefulness	3.157	7.101	<.001***	2.521	5.981	<.001***	3.871	8.229	<.001***
	Comparison 4 A-N : N-P			Comparison 5 A-N : A-P			Comparison 6 N-P : A-P		
Criterion	diff.	z	p	diff.	z	p	diff.	z	p
Actionability	−0.813	−2.105	0.152	−0.205	−0.535	0.950	0.608	1.570	0.396
Satisfaction	−0.734	−2.005	0.186	0.008	0.022	1.000	0.742	2.013	0.183
Mental demand	0.777	2.182	0.128	0.486	1.327	0.546	−0.291	−0.811	0.849
Performance	−0.573	−1.620	0.367	0.989	2.645	0.041*	1.562	4.133	<.001***
Effort	1.231	3.483	0.003**	0.878	2.491	0.061	−0.353	−0.999	0.750
Frustration	1.251	3.046	0.012*	0.248	0.609	0.929	−1.003	−2.496	0.061
Usefulness	−0.636	−1.836	0.256	0.714	1.996	0.189	1.350	3.718	0.011*

diff., difference. Statistically significant results are reported with $^{***}p < 0.001$, $^{**}p < 0.01$, and $^{*}p < 0.05$. Compared conditions are N-N, no actionability constraints are available; A-N, only action constraint is available; N-P, only priority constraint is available; and A-P, both actionability constraints are available. diff., difference.

increased mental demand. But, on the other hand, we observed an improvement in the reported level of performance and the usefulness of the interaction.

5.2 Providing Action Constraint Might Be Better Off to Users

In general, we observed that providing any type of actionability constraint led participants to generate actionable recourse better. This was indicated by the positive coefficients of both constraints for performance and usefulness criteria in the CLMM analysis. However, for other metrics, we observed that providing an action constraint was more beneficial for participants. Provision of action constraints has been accompanied by decreased frustration, while priority constraints were associated with increased mental demand and effort. Such a trend was also visible in pairwise comparisons. In comparison 5 from Table 2, we observe that the level of effort and frustration was higher for the N-P condition compared to the A-N condition with statistical significance. Furthermore, the perceived usefulness of interaction conditions increased significantly with the addition of action constraint (comparison 6) but not with the addition of priority constraint (comparison 5).

Meanwhile, it is also interesting to see that despite the benefits of action constraint over priority constraint, their effects on the quality of generated recourse were indistinguishable. Pairwise comparison showed that there are no statistically significant differences between the actionability and satisfaction of generated recourses in A-N and N-P conditions (comparison 4).

5.3 Effect of Providing Diverse Constraints

Provision of both types of constraints showed somewhat mixed effects on participants' customization of recourse. From the analysis of the interaction log, we found that participants were more likely to find an actionable recourse when they provided both types of constraints. Specifically, the success rates were 91.67% for the A-N condition, 90% for the N-P condition, and 95% for the A-P condition. Additionally, participants were able to find these solutions with fewer interactions on average, with 2.25 interactions for the A-N condition, 2.28 interactions for the N-P condition, and 2.12 interactions for the A-P condition.

Such a trend is also visible in pairwise comparisons, where participants' self-reported level of performance was higher for the A-P condition compared to conditions where they could only provide one type of constraint (comparisons 5 and 6 in Table 2). It is also notable that this increased level of performance was not accompanied by an increase in mental demand or effort, which could have possibly been caused by more active participation. Furthermore, the usefulness of the A-P condition was significantly different from the N-P condition. However, it should also be noted that the quality of the generated recourse when participants provided diverse constraints was not distinguishable with that of when participants provided one type of constraint (comparisons 5 and 6).

5.4 Change of User's Interaction Pattern

One of the interesting findings from the previous analysis was that the difference in scores for most of the criteria was not significant across the three experimental conditions where participants were allowed to provide actionability constraints. To further explore the differences in participants' perception of each actionability constraint, we analyzed how their usage of one constraint changed when the other constraint became available. From the video recordings of the experiment, we computed the average number of participants' usages of the provided functionalities (two types of buttons, range for action constraint, and weight for priority constraint) per interaction in three experimental conditions. Then, we used CLMM analysis to determine whether the inclusion of one type of actionability constraint had an impact on the average number of using functionalities of existing constraint. Again, the participant ID was included as a random effect factor to account for differences in individual usage patterns.

Figure 5 shows the average number of functionality usages for each condition. From the comparison of the A-N condition and the A-P condition, the additional provision of priority constraints had little to no effect on participants' usage patterns of both functionalities of action constraints. While there is a tendency to use less, the difference was not significant. On the other hand, the provision of an action constraint resulted in participants using the functionalities of the priority constraint much less. The average usage of button functionality per interactive trial decreased by 0.297 ($p = 0.049$), and the usage of weight functionality decreased by 1.19 ($p = 0.003$).

From further analysis, we found out that not only is their average number of priority constraint usage different but also what they provide. Figure 6 shows how participants' tendencies to provide weights changed when they were able to provide action constraints. We can see that participants began to assign smaller weights when action constraints became available. These results suggest not only that participants prefer using action constraints but also how their purpose of using priority constraints changes with the availability of action constraints.

6 Qualitative Analysis

From the qualitative analysis, our objective was to obtain a more comprehensive understanding of participants' thoughts and opinions in order to provide additional clarity of the quantitative

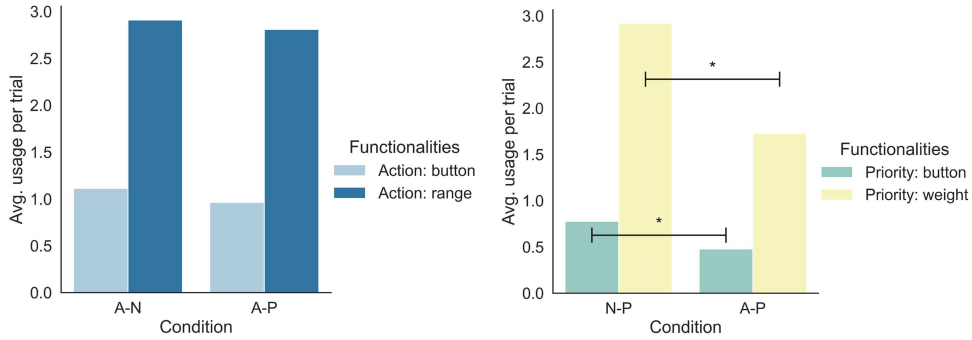

Fig. 5. Average usage of functionalities per trial. Changes with statistical significance are indicated with a horizontal line and an asterisk. The addition of priority constraints on the interface did not affect participants' frequency of using action-related functionality. On the other hand, participants used priority-related functionalities much less frequently when they could provide action constraints.

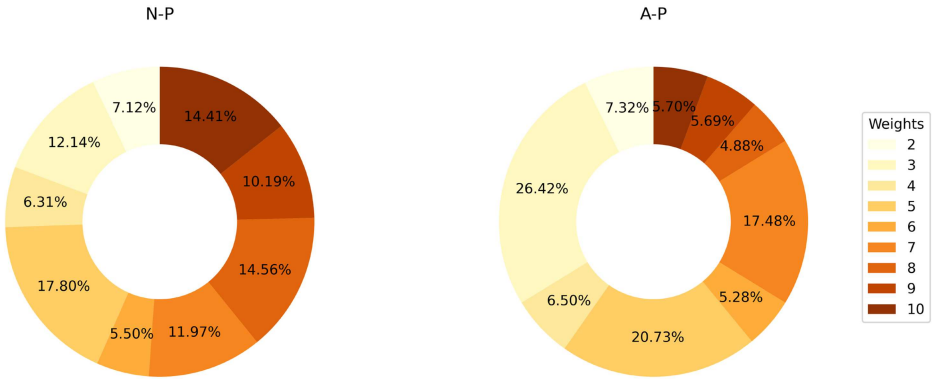

Fig. 6. Difference in frequency of weights given between N-P and A-P conditions.

analysis results. Specifically, we investigated participants' perceptions of different actionability constraints, as well as their experiences with interactive algorithmic recourse customization and providing actionability constraints.

6.1 Users Expressed the Necessity of Interactive Factor in Recourse Customization

From the interview, 25 out of 30 participants emphasized the importance of interaction in customizing the algorithmic recourse. Furthermore, 14 of the participants perceived interaction as a necessary feature, regardless of how well a system can provide personalized recourse. They pointed out that the system might not offer a perfectly personalized recourse because the system's definition of actionability might differ from their own standard of what is actionable. P16 also commented that "Interaction is necessary because even when the system provides an actionable recourse, I would still be curious and might not be confident whether it is the best option for me." They viewed interaction as a tool not only for customization but also for exploring other possibilities. Some participants expressed the need for interaction based on similar real-life scenarios. P18 stated, "When you communicate with real bankers about the loans, you not only receive the proposals but also have more interactive conversations with them. I believe that constant interaction with the system should always be available."

On the other hand, some participants mentioned that the need for interaction varies depending on different conditions. The most frequently mentioned condition was the level of personalization that the system can achieve. P8 and P13 mentioned, "If the system can provide highly personalized and actionable recourses, extra interaction won't be necessary." Some participants focused on the diversity of the generated CEs. P29 said, "If the system can show me all the possible proposals, I just need to choose one from them. In that case, the lack of interaction won't be so inconvenient."

6.2 Action Constraint Is More Intuitive, Priority Is Easier to Use

Through the qualitative study, we found that participants perceived two actionability constraints quite differently in terms of their understandability and simplicity. Within their responses, we discovered a general trend where many participants considered action constraint to be more intuitive. Out of 18 participants who discussed the understandability of actionability constraints, 12 mentioned that action constraints are easier to understand. Specifically, they found the process of users providing their capabilities and AI generating actionable recourse based on them to be straightforward. P13 further compared customizing recourse using action constraints to a user's typical interaction with others. He said, "In fact, when speaking to another person, especially a banker, you would indicate that you can adjust certain features by a specific amount. So this (A-N condition) seems more similar to the style of interaction when one tries to provide information to another person."

In terms of the priority constraint, 16 participants mentioned that it is not very intuitive. Their primary concern was how the system interprets the priorities provided by participants. P13 pointed out that "Even though users provide the weights, accurately converting users' subjective opinions into quantitative values is not possible.... Because of that, I questioned whether I could obtain the desired answer while interacting with the system." Furthermore, a small number of participants (six in total) stated that they did not understand the concept of providing priority constraints, even though they were able to provide them in other formats.

Meanwhile, 11 participants have favoured priority constraints for the simplicity it provides in the interaction. Unlike action constraints, which require participants to provide detailed input regarding their capabilities, they can easily express their preferences using simple weights and priority constraints. P3 commented that, "(When comparing the A-N and N-P conditions) While quality of the proposed recourses was similar, the N-P condition was much easier to provide input for because I didn't have to get into the details. I just had to demonstrate the challenge of modifying specific features using numerical values." P17 further mentioned that this type of simple interaction is similar to how he envisions interacting with an AI. Users would only need to provide their simplified preferences, and the AI would then offer actionable recourse.

6.3 Users Look for More Controllability in Recourse Customization

Another difference between the two constraints was the level of control that the participants had, which was noted by 25 out of 30 participants. Out of them, 13 participants commented that by providing action constraints, they can infer that the system will provide recourses using the action within the specified ranges. The property of action constraint provides participants with greater control over the customization process of algorithmic recourses, offering a significant advantage. P6 said, "(With action constraints) I could easily specify the range I considered, and I felt that the generated recourses were highly customized for my purpose. Moreover, if the generated recourses contain changes in features that I didn't consider, I can adjust the range for those features to make the recourses more personalized."

On the other hand, the majority of participants (23 out of 30) tend to question how the priority constraints affect the system's process of generating recourse. They mentioned that the issue with the priority constraint is the inability to restrict how the feature values will be changed. P29 mentioned, "There were cases when I assigned a weight of 10 to certain features, but the system still changed the feature's value in the proposal. It feels really vague for users to understand what weights actually do. It was certainly helpful to some extent, but I couldn't modify the content of the proposals as I could in the A-N condition." The limited controllability of the priority constraint seemed to place the burden on the participants. P15 said, "I felt that there was a gap between the weights I had considered and the system's concept…. I could have interacted more to bridge the gap, but it would require additional effort."

The difference in controllability was further caused by intrinsic differences in functionalities resulting from two constraints. A few participants (five in total) pointed out that using priority constraints alone does not allow the system to prevent changes to certain feature values. However, it is possible to achieve this by using action constraints. P16 commented, "I preferred using action constraints because they allowed me to restrict the potential ranges for feature values and prevent certain feature values from being altered. Of course, I could replicate these functionalities by assigning appropriate weights, but I felt that users have more degree of freedom as we could simply lock the features from changing."

6.4 Diverse Preference Configuration Might Not Always Be Needed

Participants had varying opinions on the necessity and usefulness of providing diverse actionability constraints. Many participants (15 out of 30) explicitly stated that they preferred the interaction when they could provide both types of actionability constraints. They frequently mentioned that they could better understand how the system works when they used various functionalities. P28 mentioned that a diverse preference configuration can give users a greater sense of predictability. He said, "Regardless of the generated recourse, providing weights will be important. If we only provide action constraints, we cannot be certain how the system will behave within those ranges. By assigning weights, we can instruct the system on which features to prioritize, which is quite convenient." Few have emphasized the importance of offering users with a variety of options to elicit their preferences. P10 said, "To be fair, I don't really like weight systems, but providing priority in the system would be a decent choice. Users will decide whether or not to use them."

Meanwhile, 8 out of 30 participants perceived diverse preference configuration to be unnecessary. They frequently mentioned that users experienced an increased burden due to having to provide more inputs to the system. This has also been mentioned by those who prefer to provide diverse preference configuration. P17 mentioned that while the A-P condition is most preferable, allowing users to provide all the constraints is undesirable. Furthermore, as P4 pointed out, participants had to consider more factors when configuring their preferences. They specifically needed to understand how each constraint affects the recourse generation process. Participants mentioned that using one type of actionability constraint is enough to customize the algorithmic recourse, given these burdens. Usually, participants preferred action constraints over priority constraints because of their lower intuitiveness. P20 said, "By providing only ranges, I know that the system will find a useful recourse within the specified ranges. But when I used both priority and action constraints, I couldn't figure out how the system would generate recourse based on the given input." P3 further stated that "The priority constraint feels like an additional input to the system, in addition to the action constraint. In a situation where I have already stated what I am capable of, it seems redundant to also provide my priorities."

7 Discussion

In this section, we discuss the overall findings of this study along with the possible implications for how to design user-centered interactive recourse customization. Lastly, we discuss the limitations of this study.

7.1 Help Users to Efficiently Control the Recourse Customization Process

As we have seen during the qualitative research, users preferred an intuitive and controllable method of recourse customization. The lack of intuitiveness led to users being confused about how to use each constraint, and the lack of controllability made users frustrated because they felt that they could not express their preferences as accurately as possible. In this regard, it would be beneficial to assist users in interactive customizing recourse by improving its learnability and reversibility to enhance the controllability and understandability of the recourse customization process. By designing the interface to allow users to easily understand how their inputs affect the customization process, users will be able to interactively customize and generate more actionable recourse. This will also reduce the cognitive load on users, as they will spend less time trying to figure out how to customize their preferences for more personalized results.

Specifically, we believe that significant improvements should come from priority constraints. Users found the priority constraint to be non-intuitive and lacking in controllability. Priority constraint could be designed in a more user-centered way to enhance usability in this aspect. For example, rather than delving into the detailed handling of the cost function, users could simply provide priorities in a much simpler manner. They could offer binary options for selecting features to be changed or multi-level options where each level is labeled "very easy," "easy," and so on, as demonstrated in Wang's work [60]. It could even be an option for users not to provide priority constraints and instead elicit them implicitly through repetitive interaction [15] if the controllability of priority constraint cannot be guaranteed.

7.2 Encourage Users in Exploring and Comparing Diverse Algorithmic Recourse

While customizing the algorithmic recourse, we noticed that users were curious about whether the generated recourse is the best option for them. They also expressed frustration when they were not allowed to explore other options. Interestingly, they showed this tendency regardless of the system's ability to provide personalized recourse. In such a context, it would be advisable to provide users with a diverse set of algorithmic recourses. This could be achieved by providing users with either the entire set of available recourse, as demonstrated in [59] or by offering different recourses upon their request. Providing users with a variety of algorithmic recourses will increase the likelihood of generating actionable recourse [37] while also addressing users' questions about "What other actions can I take?." Furthermore, it should be emphasized that the ultimate goal of exploration is for users to compare different recourses to find the optimal one for their needs. Thus, the interface should be designed to assist users in comparing diverse recourses, for example, by allowing users to bookmark their favourite recourses [60]. We believe these principles could be a step towards providing a good experience for users in customizing algorithmic recourses.

7.3 Consider the Difference of Various End Users

From the user study, we observed that users enjoy controlling various aspects of recourse customization. Users felt that they performed better in customizing the recourses despite the fact that there were no significant differences in the resulting recourse. It also enhanced users' understanding of how the system will generate personalized recourse. In this respect, it would be recommended to allow users to configure their preferences using various methods. Granting users more control over

recourse customization will improve their understanding of the system, thereby increasing the likelihood of obtaining more personalized recourse. Furthermore, it will provide users with greater freedom in customizing recourse, allowing them to choose their preferred configuration methods.

Meanwhile, it should be emphasized that users had varying perceptions of providing diverse preference configurations. Although such responses mostly relate to the intuitiveness and usability of customization methods, we believe that it brings another factor to consideration: end-users. Some participants might seek more customized recourse even if it means going through the arduous task of interactive customization with various inputs. Some participants might not understand the concept of priority constraints and may only provide action constraints. Various users have different cognitive and perceptual needs when it comes to customizing recourses. By prioritizing users and gaining a deep understanding of their needs, motivations, and external influences, we can take a significant step towards designing a customizable interface that allows users to efficiently and effortlessly configure their preferences.

7.4 Limitations and Future Works

There are a few important limitations that should be discussed. While our prototype ensures to find counterfactual example within plausible boundary, the recourse suggested based on the counterfactual example might not be sensible in the real world. Example of such a strange recourse can be either suggesting a user to increase his annual income while decreasing the loan grade or change his purpose to get his loan request accepted. We acknowledge that there has been a lot of research on generating plausible algorithmic recourse, and it's difficult to consistently generate perfectly plausible recourse. Although we have noticed the participants about this issue of our prototype, the existence of counter-intuitive algorithmic recourse could have affected participants' level of acceptance and their response to the surveys.

Another limitation is that user study was conducted in a simulated setting, where participants engaged with a given task not as themselves but as a simulated subject. This raises two concerns regarding the user study. First is that how users responded to the survey could have been affected by not only the participants' own intuition but also the imaginary roles they played in. Second is that how participants used functionalities of the prototype and how they responded can possibly be different from how actual users might have used or responded. To mitigate this issue, we implemented several measures. Firstly, we encouraged participants to make any necessary assumptions to enhance their engagement with the simulated subject. Additionally, we utilized participant IDs as random variables to account for the effects mentioned earlier. However, despite measures taken, we acknowledge that recorded responses and their usage of each constraints could have been affected and possibly be different from actual users. Thus, further research needs to be conducted to analyze how the target end users of the system will respond and how their responses may differ based on their characteristics, such as level of background knowledge, gender, age, and so on. We leave this to our future work.

8 Conclusion

This study examined users' perceptions and experiences of interactive algorithmic recourse customization. We designed a web-based prototype where users can interact with the system that provides algorithmic recourse. We conducted a user experiment using both quantitative and qualitative approaches. The analysis results revealed that (1) continuous interaction between the user and the recourse-providing system can help users find diverse options of actionable recourses, (2) controllability and understandability affect how users engage in recourse customization, and (3) the degree of customization sought by users depends on various factors. Based on these findings, we have discussed the design implications of a user-centered algorithmic recourse customization interface. We

hope that these findings will serve as a step towards a more inclusive understanding of the user-led recourse customization process and interfaces that support this new type of user-AI interaction.

Appendices

A Survey Details

During the user study, participants had to fill out surveys after each time they tried to find a useful algorithmic recourse. We included the full survey, including the questions and how participants were directed to different sections of the survey depending on their responses. After each question, the format of participants' responses is indicated within the parenthesis. For Section 4, some questions are accompanied with sub-questions that help participants understand what is sought in each of the metric.

Section 1:

(1) Were you able to find a useful algorithmic recourse after interacting with the given interface? (Yes or No).
    —If yes, go to section 2 of the survey and otherwise, go to section 3 of the survey.

Section 2 (After you are done with section 2, please move on to section 4 of the survey):

(1) How actionable do you think the action suggested by the chosen algorithmic recourse is? (7-point Likert scale)
(2) How satisfied are you with the chosen algorithmic recourse? (7-point Likert scale)
(3) Please explain the reason behind your response to the above two questions regarding perceived actionability and satisfaction towards the chosen algorithmic recourse. (Free-form answer box)

Section 3 (After you are done with section 3, please move on to section 4 of the survey):

(1) What were the assumptions you made after receiving new profile? (Free-format answer box)
    —e.g., What kind of person did you imagine to be? What kind of situation is he in?
(2) Please describe reasons why you found the generated algorithmic recourse not useful.
(3) If you can interact with the given interface more (more than the allocated 5 interactions), do you think you can eventually find a useful algorithmic recourse? (Free-format answer box)

Section 4:

(1) How mentally demanding was the task of finding a useful algorithmic recourse? (7-point Likert scale)
    —e.g., How difficult was it to provide adequate constraints and understand the output of it? How difficult was it to provide the next round of constraints based on previous interaction?
(2) How well did you interact with the system to generate a useful algorithmic recourse? (7-point Likert scale)
    —e.g., How effectively did you use the provided interface to generate a useful recourse?
(3) How much effort did you have to put in order to find a useful algorithmic recourse? (7-point Likert scale)
    —e.g., Including the mental demand, how hard did you have to try to generate a useful recourse? How hard did you have to concentrate on understanding the interaction and the generated results?
(4) How frustrated were you while you interacted with the system? (7-point Likert scale)
(5) How useful was the provided system in finding a useful algorithmic recourse? (7-point Likert scale)

B Analysis Results

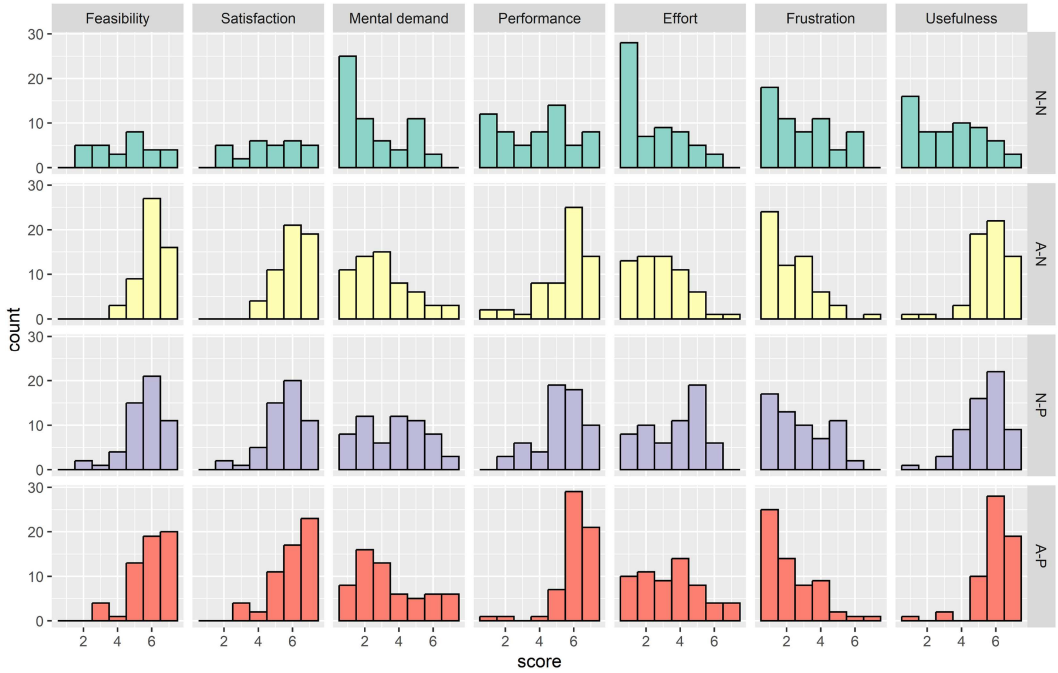

Fig. 7. Participants' responses to each criterion for each experimental condition. The number of responses is grouped based on the experimental conditions and the evaluation criteria used.

References

[1] JFdarre. 2015. Lending Club Statistics. Retrieved Mar 5, 2021 from https://rpubs.com/jfdarre/119147

[2] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160.

[3] Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. 2019. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. arXiv:1909.03012. Retrieved from https://arxiv.org/abs/1909.03012

[4] Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. 2021. Does the whole exceed its parts? The effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–16.

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

[6] Dale J. Barr, Roger Levy, Christoph Scheepers, and Harry J. Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language* 68, 3 (2013), 255–278.

[7] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. 'It's reducing a human being to a percentage' perceptions of justice in algorithmic decisions. In *Proceedings of the 2018 Chi Conference on Human Factors in Computing Systems*. 1–14.

[8] Benjamin M. Bolker, Mollie E. Brooks, Connie J. Clark, Shane W. Geange, John R. Poulsen, M. Henry H. Stevens, and Jada-Simone S. White. 2009. Generalized linear mixed models: A practical guide for ecology and evolution. *Trends in Ecology & Evolution* 24, 3 (2009), 127–135.

[9] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z. Gajos. 2021. To trust or to think: Cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–21.

[10] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W Philip Kegelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.

[11] Furui Cheng, Yao Ming, and Huamin Qu. 2020. DECE: Decision explorer with counterfactual explanations for machine learning models. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2020), 1438–1447.

[12] R. H. B. Christensen. 2022. Ordinal—Regression Models for Ordinal Data. R package version 2022.11-16. Retrieved from https://CRAN.R-project.org/package=ordinal

[13] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. 2020. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature*. Springer, 448–469.

[14] Kevin Davenport. 2015. Lending Club Data Analysis with Python. Retrieved April 5, 2021 from https://kldavenport.com/lending-club-python/

[15] Giovanni De Toni, Paolo Viappiani, Bruno Lepri, and Andrea Passerini. 2022. Generating personalized counterfactual interventions for algorithmic recourse by eliciting user preferences. arXiv:2205.13743. Retrieved from https://arxiv.org/abs/2205.13743

[16] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. arXiv:1702.08608. Retrieved from https://arxiv.org/abs/1702.08608

[17] Upol Ehsan, Samir Passi, Q. Vera Liao, Larry Chan, I. Lee, Michael Muller, Mark O. Riedl, et al. 2021. The who in explainable AI: How AI background shapes perceptions of AI explanations. arXiv:2107.13509. Retrieved from https://arxiv.org/abs/2107.13509

[18] Umer Farooq, Jonathan Grudin, Ben Shneiderman, Pattie Maes, and Xiangshi Ren. 2017. Human computer integration versus powerful tools. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 1277–1282.

[19] Oscar Gomez, Steffen Holter, Jun Yuan, and Enrico Bertini. 2020. Vice: Visual counterfactual explanations for machine learning models. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 531–535.

[20] Sandra G. Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 50. Sage Publications Sage CA, 904–908.

[21] JFdarre. 2015. Project 1: Lending Club's Data. Retrieved April 5, 2021 from https://rpubs.com/jfdarre/119147

[22] Yucheng Jin, Bruno Cardoso, and Katrien Verbert. 2017. How do different levels of user control affect cognitive load and acceptance of recommendations? In *Proceedings of the 4th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys '17)*. Y. Jin, B. Cardoso, and K. Verbert (Eds.), Vol. 1884. 35–42.

[23] Yucheng Jin, Nava Tintarev, and Katrien Verbert. 2018. Effects of personal characteristics on music recommender systems with different levels of controllability. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 13–21.

[24] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. 2020. DACE: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI-20)*. 2855–2862.

[25] Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer. 2012. Enterprise data analysis and visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2917–2926.

[26] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2020. Model-agnostic counterfactual explanations for consequential decisions. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*. PMLR, 895–905.

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

[28] Mark T. Keane and Barry Smyth. 2020. Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI). In *Proceedings of the 28th International Conference on Case-Based Reasoning Research and Development*. Springer, 163–178.

[29] Lara Kirfel and Alice Liefgreen. 2021. What if (and how…)? - Actionability shapes people's perceptions of counterfactual explanations in automated decision-making. In *Proceedings of the International Conference on Machine Learning Workshop on Algorithmic Recourse*.

[30] Thomas Kosch, Jakob Karolus, Johannes Zagermann, Harald Reiterer, Albrecht Schmidt, and Pawe Å. W. Woźniak. 2023. A survey on measuring cognitive workload in human-computer interaction. *ACM Computing Surveys* 55, 13s (2023), 1–39.

[31] Russell V. Lenth. 2016. Least-squares means: The R package lsmeans. *Journal of Statistical Software* 69, 1 (2016), 1–33.

[32] Clayton Lewis and John Rieman. 1993. *Task-Centered User Interface Design: A Practical Introduction.* Clayton Lewis and John Rieman.
[33] Q. Vera Liao, Daniel Gruen, and Sarah Miller. 2020. Questioning the AI: Informing design practices for explainable AI user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–15.
[34] Zhicheng Liu and Jeffrey Heer. 2014. The effects of interactive latency on exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2122–2131.
[35] Hannes Matuschek, Reinhold Kliegl, Shravan Vasishth, Harald Baayen, and Douglas Bates. 2017. Balancing Type I error and power in linear mixed models. *Journal of Memory and Language* 94 (2017), 305–315.
[36] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1–38.
[37] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 607–617.
[38] Changhoon Oh, Taeyoung Lee, Yoojung Kim, SoHyun Park, Saebom Kwon, and Bongwon Suh. 2017. Us vs. them: Understanding artificial intelligence technophobia over the google deepmind challenge match. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2523–2534.
[39] Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach. 2021. Manipulating and measuring model interpretability. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–52.
[40] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. FACE: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 344–350.
[41] Quentin Roy, Futian Zhang, and Daniel Vogel. 2019. Automation accuracy is good, but high controllability may be better. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–8.
[42] Chris Russell. 2019. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 20–28.
[43] Johnny Saldana. 2011. *Fundamentals of Qualitative Research*. OUP USA.
[44] Johanes Schneider and Joshua Handali. 2019. Personalized explanation in machine learning: A conceptualization. arXiv:1901.00770. Retrieved from https://arxiv.org/abs/1901.00770
[45] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. 2020. Certifai: A common framework to provide explanations and analyse the fairness and robustness of black-box models. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 166–172.
[46] Alison Smith-Renner, Ron Fan, Melissa Birchfield, Tongshuang Wu, Jordan Boyd-Graber, Daniel S. Weld, and Leah Findlater. 2020. No explainability without accountability: An empirical study of explanations and feedback in interactive ml. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
[47] Kacper Sokol and Peter Flach. 2020. One explanation does not fit all. *KI-Künstliche Intelligenz* (2020), 1–16.
[48] Emily Sullivan and Philippe Verreault-Julien. 2022. From explanation to recommendation: Ethical standards for algorithmic recourse. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*. 712–722.
[49] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. 2018. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 303–310.
[50] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 10–19.
[51] Arnaud Van Looveren and Janis Klaise. 2019. Interpretable counterfactual explanations guided by prototypes. arXiv:1907.02584. Retrieved from https://arxiv.org/abs/1907.02584
[52] Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael S. Bernstein, and Ranjay Krishna. 2023. Explanations can reduce overreliance on AI systems during decision-making. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (2023), 1–38.
[53] Michael Veale, Max Van Kleek, and Reuben Binns. 2018. Fairness and accountability design needs for algorithmic support in high-stakes public sector decision-making. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–14.
[54] Suresh Venkatasubramanian and Mark Alfano. 2020. The philosophical basis of algorithmic recourse. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 284–293.
[55] Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E. Hines, John P. Dickerson, and Chirag Shah. 2020. Counterfactual explanations and algorithmic recourses for machine learning: A review. arXiv:2010.10596. Retrieved from https://arxiv.org/abs/2010.10596
[56] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology* 31 (2017), 841.

[57] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y. Lim. 2019. Designing theory-driven user-centric explainable AI. In *Proceedings of the 2019 CHI conference on Human Factors in Computing Systems*. 1–15.

[58] Xinru Wang and Ming Yin. 2021. Are explanations helpful? A comparative study of the effects of explanations in AI-assisted decision-making. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*. 318–328.

[59] Yongjie Wang, Qinxu Ding, Ke Wang, Yue Liu, Xingyu Wu, Jinglong Wang, Yong Liu, and Chunyan Miao. 2021. The skyline of counterfactual explanations for machine learning decision models. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2030–2039.

[60] Zijie J. Wang, Jennifer Wortman Vaughan, Rich Caruana, and Duen Horng Chau. 2023. GAM Coach: Towards Interactive and User-Centered Algorithmic Recourse. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–20.

[61] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. 2019. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2019), 56–65.

[62] Yao Xie, Melody Chen, David Kao, Ge Gao, and Xiang 'Anthony' Chen. 2020. CheXplain: Enabling physicians to explore and understand data-driven, AI-enabled medical imaging analysis. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[63] Yunfeng Zhang, Q. Vera Liao, and Rachel K. E. Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 295–305.

Received 29 September 2022; revised 31 January 2024; accepted 19 April 2024