Learning Reward Functions from Scale Feedback

Nils Wilde[*†] **Erdem Bıyık**[*‡] **Dorsa Sadigh**[‡§] **Stephen L. Smith**[†]
[†] Electrical and Computer Engineering, University of Waterloo
[‡] Electrical Engineering, Stanford University
[§] Computer Science, Stanford University
{nwilde,stephen.smith}@uwaterloo.ca, {ebiyik,dorsa}@stanford.edu
[*] Authors contributed equally.

Abstract: Today's robots are increasingly interacting with people and need to efficiently learn inexperienced user's preferences. A common framework is to iteratively query the user about which of two presented robot trajectories they prefer. While this minimizes the users effort, a strict choice does not yield any information on *how much* one trajectory is preferred. We propose scale feedback, where the user utilizes a slider to give more nuanced information. We introduce a probabilistic model on how users would provide feedback and derive a learning framework for the robot. We demonstrate the performance benefit of slider feedback in simulations, and validate our approach in two user studies suggesting that scale feedback enables more effective learning in practice.

Keywords: HRI, reward learning, learning from choice, active learning

1 Introduction

While autonomous robots are able to accomplish an increasing variety of tasks, a key challenge that still remains is *how* they should pursue and trade off between their goals. In recent years, there has been significant work on interactively learning user preferences of robot behaviors [1–15]. Usually, the user is provided with one or more robot trajectories, and is asked to provide feedback through pairwise comparisons [2–8], rankings of the trajectories [16, 17], or physical feedback [18, 12, 14]. The underlying reward function governing human preferences can then be learned through this implicit feedback. Specifically, one framework with minimal complexity for the user is *learning from choice feedback* [2–8], where the robot demonstrates two alternative trajectories for some task. The user then simply chooses their preferred behavior allowing the robot to infer an underlying reward function for the user preferences.

Choice feedback, although simple to collect, is limiting in a number of ways. Consider the example shown in Fig. 1, where a robot is tasked to serve a drink to a customer. The customer might have different preferences over the type of drink to have (milk, orange juice, or water), or the specifics of the trajectory the robot takes (e.g., if it goes over the stove or around it which can affect the temperature of the drink or the likelihood of the robot accidentally hitting the pan handle). A strict choice feedback between two trajectories does not really capture these intricacies of human preferences. We thus need to have a more expressive way of collecting data from humans. Our key insight is that allowing users to provide a scaled approach on a slider (as shown in Fig. 1) can provide a more expressive medium for learning from humans and capture nuances in their preferences.

In this work, we propose *scale feedback* as a new mode of interaction: Instead of a strict question on which of the two proposed trajectories the user prefers, we allow for more nuanced feedback using a slider bar. We design a Gaussian model for how users provide scale feedback, and learn a reward function capturing human preferences. Similar to prior work in robotics, we assume this reward is a linear function of a set of features [19, 11, 13, 7], where the main task of learning from scale feedback is to recover the weights of this reward function. To learn in a data-efficient manner, we actively generate our queries to the user, i.e., pairs of trajectories demonstrated to a user similar to Fig. 1, by optimizing two well-known objectives of information gain [4] and max regret [5].

We demonstrate the performance benefit of scale feedback over choice in a driving simulation. Further, we investigate its practicality in two user studies with the real robot experiment shown in Fig. 1. Our results suggest scale feedback leads to significant improvements in learning performance.

5th Conference on Robot Learning (CoRL 2021), London, UK.

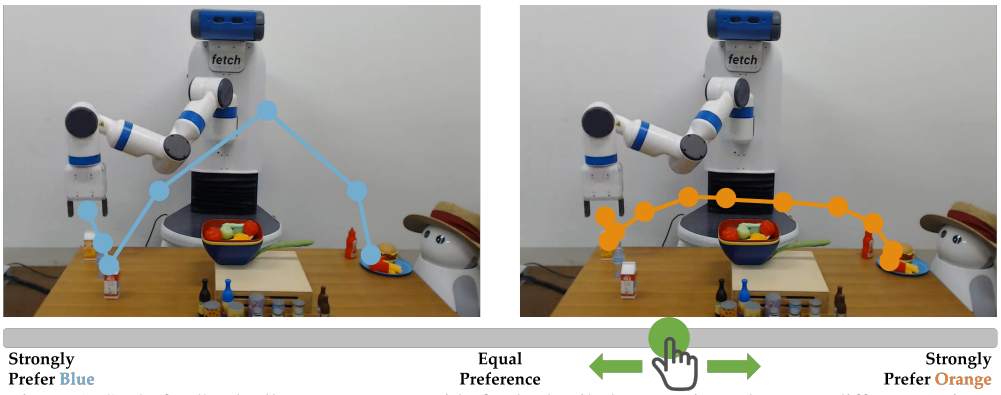


Figure 1: Scale feedback allows users to provide finely detailed comparisons between different options.

2 Related Work

Learning from human feedback is an important problem in developing interactive robots that work alongside humans. Researchers study learning from demonstrations [19–21], corrections [22, 23, 14], ordinal feedback [24, 8], rankings [16, 17, 25, 26], critiques [27, 28], and choices [2–5, 29].

While demonstrations usually are very informative, they are not always viable: Demonstrating the desired behavior might require a high level of expertise [30], or can be difficult in high-order systems [31–33]. Choice questions minimize interface complexity and mental effort for the user. However, when the user is indifferent towards both options, learning becomes difficult since users may become noisier in their responses. Thus, [7, 4, 6] investigate modifications of learning from choice where users can also answer *About Equal*. These two forms of choice feedback are usually referred to as *strict* and *soft* choice. When the user chooses the neutral answer, the robot learns to assign about equal reward to the presented trajectories. While choice feedback provides an easy medium for learning from humans, it provides at most one bit of information. Thus, to effectively learn from choice feedback, we often need to actively generate the queries made to humans. Previous work has investigated different auxiliary measures that are greedily optimized to enable efficient learning, including expected volume removal [2], information gain [4], and regret [5].

In the proposed scale feedback framework, we take the soft choice approach one step further: Instead of three discrete values for feedback (prefer A, prefer B, neutral) users give quasi-continuous feedback. This allows the user to indicate by *how much* they prefer one option over the other.

Slider bars have been used in robotics for tuning parameters [34]. More related to our work, Cabi et al. [35] proposed using them for *reward sketching*. Instead of assigning a numerical preference between presented options, users continuously indicate the robot's progress towards some goal. However, this requires users to assign scores to different parts of trajectories. Developing the scale feedback for preference-based learning, we retain the ease of comparing trajectories.

3 Problem Formulation

We now introduce the notation we use in this paper and formulate the learning problem.

Reward function. We consider the scenario where a robot needs to customize its behavior to the preferences of a user Alice. We assume Alice evaluates robot paths $P \in \mathcal{P}$ based on a vector of features $\boldsymbol{\phi}^P = [\phi_1(P), \dots, \phi_n(P)]$. Similar to prior works in robotics [19, 11, 13, 7], we define a linear reward function r that assigns a numerical value to a path P by weighting a set of features:

$$r(P, \boldsymbol{w}) = \boldsymbol{\phi}^P \cdot \boldsymbol{w}. \tag{1}$$

These features are usually provided by a domain expert incorporating the core factors that the reward needs to capture, e.g., collision with other objects, or distance to the goal. Further, the robot has access to a motion planner that finds an optimal path given a set of weights, i.e., the planner is a (deterministic) function $\rho : \mathbb{R}^n \to \mathcal{P}$ where $\rho(\boldsymbol{w}) = \arg\max_{P \in \mathcal{P}} r(P, \boldsymbol{w})$.

Regret. Similar to [5], we define the *regret* between any two weights $(\boldsymbol{w}, \boldsymbol{w}')$ as the difference in the reward $\boldsymbol{w}'$ assigns to the paths $\rho(\boldsymbol{w})$ and $\rho(\boldsymbol{w}')$:

$$R(\boldsymbol{w}, \boldsymbol{w}') = \boldsymbol{\phi}(\rho(\boldsymbol{w}')) \cdot \boldsymbol{w}' - \boldsymbol{\phi}(\rho(\boldsymbol{w})) \cdot \boldsymbol{w}', \tag{2}$$

which quantifies the suboptimality when the true weights are $\boldsymbol{w}'$, but the path is optimized using $\boldsymbol{w}$.

Learning. Let $\boldsymbol{w}^*$ denote Alice's weights for the reward function. These weights are not known to the robot; the only information initially available is a prior distribution $\mathbb{P}(\boldsymbol{w} = \boldsymbol{w}^*)$. The robot

learns $\boldsymbol{w}^*$ by iteratively presenting her with two paths P and Q for K iterations. We extend the *learning from choice* framework, where users simply indicate the path they prefer, to a setting where they instead provide a more finely detailed *scale feedback*.

Definition 1 (Scale feedback)**.** Presented with two paths P and Q, Alice returns numerical feedback $\psi \in [-1, 1]$. If $\psi = 0$, this means Alice has no preference between the paths, $\psi = 1$ equals a strong preference for path P and $\psi = -1$ a strong preference for path Q.

From an interface design and expressiveness perspective, it is undesirable to have users give a numerical value for ψ. Instead, they can express such a feedback with a slider bar with a more fine-grained set of options. An example is illustrated in Fig. 1. We let $D_K = \{(P_1, Q_1, \psi_1), \ldots, (P_K, Q_K, \psi_K)\}$ be the set of recorded user feedback.

Performance Measures. Let $\hat{\boldsymbol{w}}$ be the robot's estimate of $\boldsymbol{w}^*$, and $\xi(\hat{\boldsymbol{w}}, \boldsymbol{w}^*)$ be a performance measure for the learning process. Previous works focused on the *alignment* of weights [2, 4], $\texttt{Alignment} = \hat{\boldsymbol{w}} \cdot \boldsymbol{w}^* / ||\hat{\boldsymbol{w}}|| \cdot ||\boldsymbol{w}^*||$, measuring the cosine similarity of vectors $\hat{\boldsymbol{w}}$ and $\boldsymbol{w}^*$, i.e., how well the parameters of Alice's reward function are learned. Alternatively, Wilde et al. [5] proposed the relative error in *cost*. We adapt this as the $\texttt{Relative\_Reward} = \boldsymbol{\phi}(\rho(\hat{\boldsymbol{w}})) \cdot \boldsymbol{w}^* / \boldsymbol{\phi}(\rho(\boldsymbol{w}^*)) \cdot \boldsymbol{w}^*$, measuring how much Alice likes the trajectory optimized for $\hat{\boldsymbol{w}}$ compared to the one optimized for $\boldsymbol{w}^*$.

Problem Statement. Let π be an adaptive policy for designing queries (P, Q), and let $D_K(\pi \mid \boldsymbol{w}^*)$ be the expected set of user feedback when a user $\boldsymbol{w}^*$ is queried by π for K iterations. Given a robot motion planner ρ, a user with preferences $\boldsymbol{w}^*$, and a budget of K rounds to query the user about their *scale feedback* on two presented paths, our goal is to find an adaptive policy π that solves

$$\max_{\pi} \xi\left(\mathbb{E}\left[\boldsymbol{w} \mid D_K(\pi \mid \boldsymbol{w}^*)\right], \boldsymbol{w}^*\right). \tag{3}$$

4 Approach

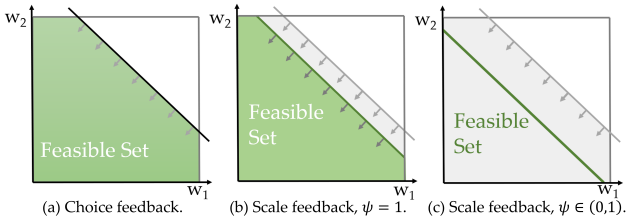
We now briefly review learning from choice, and then extend the framework to scale feedback.

4.1 Choice Feedback

When presented with two paths P and Q, a user returns an ordering $P \succeq Q$ (P is preferred) or $P \preceq Q$ (Q is preferred). In a noiseless setting, we have

$$r(P, \boldsymbol{w}^*) - r(Q, \boldsymbol{w}^*) \geq 0 \iff P \succeq Q. \tag{4}$$

That is, the path P has a reward that is at least as high as that of Q with respect to the hidden true user weights $\boldsymbol{w}^*$. Using $r(P, \boldsymbol{w}) = \boldsymbol{\phi}^P \cdot \boldsymbol{w}$, we can tighten our notation and write $(\boldsymbol{\phi}^P - \boldsymbol{\phi}^Q) \cdot \boldsymbol{w}^*$ instead of $r(P, \boldsymbol{w}^*) - r(Q, \boldsymbol{w}^*)$. Equation (4) already contains an observation model: If the user chooses path P, the robot can infer that P has a higher reward with respect to $\boldsymbol{w}^*$. This inequality defines a halfspace $\Lambda(P, Q) = \{\boldsymbol{w} \mid (\boldsymbol{\phi}^P - \boldsymbol{\phi}^Q) \cdot \boldsymbol{w}^* \geq 0\}$ containing all weights that are *feasible* given the observed user choice. Over k iterations, we can intersect the sets $\Lambda(P_1, Q_1), \ldots, \Lambda(P_k, Q_k)$ to obtain the *feasible set* $\mathcal{F}_k$ shown in Fig. 2(a). By definition, this feasible set is convex.



(a) Choice feedback. (b) Scale feedback, $\psi = 1$. (c) Scale feedback, $\psi \in (0,1)$.

Figure 2: Different feasible sets learned from choice and scale feedback. Shown is the updated weightspace (green) after observing user feedback for one (P, Q) pair. If $\psi = 1$ scale feedback learns a tighter halfspace; when $\psi \in (0, 1)$ scale feedback learns an equality, i.e., a hyperplane.

4.2 Scale Feedback

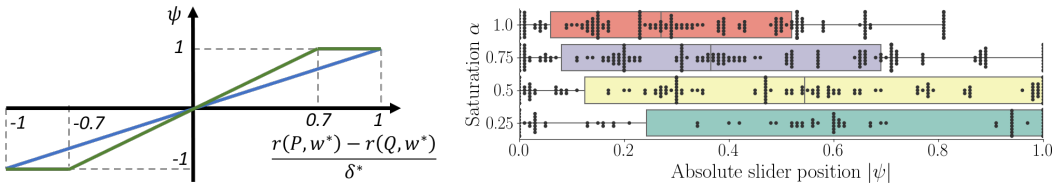
Scale feedback allows the robot to gain more information: the robot can also infer by *how much* the user prefers P, allowing for learning tighter feasible sets. We extend the model in (4) and show how a noiseless user would provide scale feedback and then study how a robot can learn from it.

Definition 2 (Maximum Reward Gap)**.** Given a user $\boldsymbol{w}^*$, the maximum reward gap is

$$\delta^* = \max_{P,Q \in \mathcal{P}} r(P, \boldsymbol{w}^*) - r(Q, \boldsymbol{w}^*) = \max_{P,Q \in \mathcal{P}} (\boldsymbol{\phi}^P - \boldsymbol{\phi}^Q) \cdot \boldsymbol{w}^*. \tag{5}$$

We notice that the maximum reward gap cannot be computed, since $\boldsymbol{w}^*$ is unknown to the robot. Nevertheless, we can formulate the user choice model and then derive an observation model.

User model. The maximum reward gap helps to define when a noiseless user would indicate a strong preference. We assume this occurs if and only if the difference in reward of P and Q with respect to $\boldsymbol{w}^*$ is at least $\alpha^* \delta^*$ for some $0 < \alpha^* \leq 1$. Here α^* is a saturation parameter which governs



(a) User model for providing scale feedback with $\alpha^* = 1$ (blue) and $\alpha^* = 0.7$ (green).

(b) Example slider feedback for different α. The boxplots indicate the four quartiles of the absolute slider values.

Figure 3: Noiseless user model.

at what reward difference (w.r.t. to the maximum gap) the user's feedback gets saturated to a strong preference. For any other (P, Q) where $|(\boldsymbol{\phi}^P - \boldsymbol{\phi}^Q) \cdot \boldsymbol{w}^*| \in [0, \alpha^*\delta^*)$ we assume the user to linearly scale ψ between -1 and 1, which leads to the following model.

Definition 3 (Noiseless User Model)**.** Presented with two paths P and Q, a noiseless user with parameter $\alpha^* \in (0, 1]$ will always provide the following feedback:

$$\psi = \begin{cases} 1 & \text{if } r(P, \boldsymbol{w}^*) - r(Q, \boldsymbol{w}^*) \geq \alpha^*\delta^*, \\ -1 & \text{if } r(Q, \boldsymbol{w}^*) - r(P, \boldsymbol{w}^*) \geq \alpha^*\delta^*, \\ {}^{(r(P,\boldsymbol{w}^*) - r(Q,\boldsymbol{w}^*))}/_{\alpha^*\delta^*} & \text{otherwise .} \end{cases} \tag{6}$$

We illustrate the noiseless user model in Fig. 3a under different saturation parameters α^*. In Fig. 3b, we show a simulated example: for a fixed $\boldsymbol{w}^*$ we simulate how users with different values for α^* would provide scale feedback to the same 20 queries. For larger α^*, they position the slider closer to the neutral position. Finally, we derive an observation model for the noiseless user:

$$\begin{aligned} \psi = -1 &\implies r(P, \boldsymbol{w}^*) - r(Q, \boldsymbol{w}^*) \leq \psi\alpha^*\delta^* \\ \psi \in (-1, 1) &\implies r(P, \boldsymbol{w}^*) - r(Q, \boldsymbol{w}^*) = \psi\alpha^*\delta^*, \\ \psi = 1 &\implies r(P, \boldsymbol{w}^*) - r(Q, \boldsymbol{w}^*) \geq \psi\alpha^*\delta^*. \end{aligned} \tag{7}$$

Figures 2(b) and 2(c) illustrate the resulting feasible sets from (7). Moreover, we notice the user-specific and unknown parameters α^* and δ^* always appear as a product. Thus, we can introduce an auxiliary parameter $\beta = \alpha^*\delta^*$ to write (7) as $\left[-\psi, \boldsymbol{\phi}^P - \boldsymbol{\phi}^Q\right] \cdot [\beta, \boldsymbol{w}^*] \lessgtr 0$. As the model remains linear, the notion of halfspaces and the feasible set $\mathcal{F}$ can be extended to the augmented vector space.

4.3 Probabilistic User Feedback

In practice, users are often noisy; they might consider additional or slightly different features than the robot, not follow the linear reward function, or simply be uncertain in some answers. Since we cannot expect users to always provide slider feedback following (6), we introduce a probabilistic model where we add uncertainty to the placement of the slider.

Another practical limitation is the fact that we cannot collect truly continuous feedback from the users. Instead, the slider bar has a step size $\epsilon \in (0, 1]$ such that the user provides feedback of the form $n\epsilon$ for $n \in \mathbb{Z}$ and $-\epsilon^{-1} \leq n \leq \epsilon^{-1}$. Note that $\epsilon \to 0$ retains the continuous scale feedback, whereas $\epsilon = 1$ gives the soft choice model where the feedback is always in $\{-1, 0, 1\}$.

Definition 4 (Probabilistic User Model)**.** Given a user $\boldsymbol{w}^*$ and a query (P, Q), let ψ be the user feedback defined in the noiseless user model in (6). A probabilistic user using a slider bar with a step size of ϵ then provides feedback

$$\mu = \text{round}(\psi + \nu, \epsilon) \tag{8}$$

where ν is a zero-mean Gaussian noise, i.e., $\nu \sim \mathcal{N}(0, \sigma^2)$ with standard deviation σ, and $\text{round}(x, \epsilon)$ outputs $n\epsilon$ closest to x such that $n \in \mathbb{Z} \cap [-\epsilon^{-1}, \epsilon^{-1}]$.

Probabilistic Observation Model. Given the probabilistic user model, we now show how a robot can infer about $\boldsymbol{w}^*$ from scale feedback. In the noiseless case, user feedback defines a feasible set. For the probabilistic case, we instead derive a distribution over $\boldsymbol{w}$ and α. Let $\delta(\boldsymbol{w}) = \max_{P,Q \in \mathcal{P}} (\boldsymbol{\phi}^P - \boldsymbol{\phi}^Q) \cdot \boldsymbol{w}$, similar to (5). Then for $0 < \alpha \leq 1$, the belief is defined

$$f(\boldsymbol{w}, \alpha \mid \psi, P, Q) = \begin{cases} \tilde{f}(\boldsymbol{w}, \alpha \mid \psi, P, Q) & \text{if } \psi \in (-1, 1), \\ f^+(\boldsymbol{w}, \alpha \mid \psi, P, Q) & \text{if } \psi = 1, \\ f^-(\boldsymbol{w}, \alpha \mid \psi, P, Q) & \text{if } \psi = -1, \end{cases} \tag{9}$$

where

$$\tilde{f}(\boldsymbol{w}, \alpha \mid \psi, P, Q) \propto \begin{cases} 1 \text{ if } \left[-\psi, \boldsymbol{\phi}^P - \boldsymbol{\phi}^Q\right] \cdot \left[\alpha\delta(\boldsymbol{w}), \boldsymbol{w}\right] = 0, \\ 0 \text{ otherwise .} \end{cases}$$

$$f^+(\boldsymbol{w}, \alpha \mid \psi, P, Q) \propto \begin{cases} 1 \text{ if } \left[-\psi, \boldsymbol{\phi}^P - \boldsymbol{\phi}^Q\right] \cdot \left[\alpha\delta(\boldsymbol{w}), \boldsymbol{w}\right] \geq 0, \\ 0 \text{ otherwise .} \end{cases} \tag{10}$$

$$f^-(\boldsymbol{w}, \alpha \mid \psi, P, Q) \propto \begin{cases} 1 \text{ if } \left[-\psi, \boldsymbol{\phi}^P - \boldsymbol{\phi}^Q\right] \cdot \left[\alpha\delta(\boldsymbol{w}), \boldsymbol{w}\right] \leq 0, \\ 0 \text{ otherwise .} \end{cases}$$

Given noisy user feedback μ as in (8), we can define a probabilistic density function $f(\psi \mid \mu)$. Together with (9) we derive a compound probability distribution

$$f(\boldsymbol{w}, \alpha \mid \mu, P, Q) = \int_{-1}^{1} f(\boldsymbol{w}, \alpha \mid \psi, P, Q) f(\psi \mid \mu) d\psi. \tag{11}$$

where we can write $f(\psi \mid \mu)$ for $\psi \in [-1, 1]$ as

$$f(\psi \mid \mu) \propto \begin{cases} \Phi\left(\frac{\mu - \psi + \epsilon/2}{\sigma}\right) & \text{if } \mu = -1, \\ \Phi\left(\frac{\psi - \mu + \epsilon/2}{\sigma}\right) - \Phi\left(\frac{\psi - \mu - \epsilon/2}{\sigma}\right) & \text{if } \mu \in (-1, 1), \\ \Phi\left(\frac{\psi - \mu + \epsilon/2}{\sigma}\right) & \text{if } \mu = 1, \end{cases} \tag{12}$$

and $f(\psi \mid \mu) = 0$ for $\psi \notin [-1, 1]$. Here, Φ denotes the cdf of a standard normal distribution. Finally, given a sequence $D_K = \{(P_k, Q_k, \mu_k)\}_{k=1}^K$ and some prior $f(\boldsymbol{w}, \alpha)$, the joint posterior is

$$f(\boldsymbol{w}, \alpha \mid D_K) \propto f(\boldsymbol{w}, \alpha) \prod_{k=1}^{K} f(\boldsymbol{w}, \alpha \mid \mu_k, P_k, Q_k). \tag{13}$$

Here, we can factor $f(\boldsymbol{w}, \alpha)$ as $\mathbb{P}(\boldsymbol{w})\mathbb{P}(\alpha)$ by assuming $\boldsymbol{w}$ and α are independent and we also have a prior for α^*. We then take the expectation of the posterior $f(\boldsymbol{w}, \alpha \mid D_K)$ as our learned user model.

5 Algorithm Design

We now outline the learning algorithm. Over K iterations: (i) the robot *actively* generates a query (P_k, Q_k) given previous observations D_{k-1}, (ii) the user provides feedback to the query in the form of the slider value μ_k (in the noiseless case, $\mu_k = \psi_k$), and (iii) the robot updates its dataset D_k using (13). After iteration K, the algorithm returns the expected weight $\hat{\boldsymbol{w}} = \mathbb{E}[\boldsymbol{w} \mid D_K]$.

5.1 Worst Case Error Bound

To compare scale feedback to choice feedback, we establish a worst case bound on the performance measures for both frameworks. We introduce the *worst-case error* as the maximum negative performance measure, $1 - \xi(\boldsymbol{w}, \boldsymbol{w}^*)$. The constant in front ensures a positive value, which we then discount with the posterior belief, given observations D:

$$\texttt{Err}^{\max}(\boldsymbol{w}^*, D) = \max_{\boldsymbol{w}} f(\boldsymbol{w} \mid D)(1 - \xi(\boldsymbol{w}, \boldsymbol{w}^*)). \tag{14}$$

This describes the worst $\boldsymbol{w}$ the robot could pick, discounted by the posterior distribution f learned from data D. In the noiseless setting, this simplifies to $\max_{\boldsymbol{w} \in \mathcal{F}} 1 - \xi(\boldsymbol{w}, \boldsymbol{w}^*)$.

Proposition 1 (Upper error bound)**.** Let D^S denote the observation made from scale feedback and D^C be the observation from choice feedback for the same set of queries. For any user weights $\boldsymbol{w}^*$, it holds in the noiseless setting that $\texttt{Err}^{\max}(\boldsymbol{w}^*, D^S) \leq \texttt{Err}^{\max}(\boldsymbol{w}^*, D^C)$.

The proof follows from the observation $\mathcal{F}^{\texttt{Scale}} \subseteq \mathcal{F}^{\texttt{Choice}}$, i.e., scale feedback removes more volume from the weight set. Hence, the worst choice of an estimate $\hat{\boldsymbol{w}}$ given observations is guaranteed to have a smaller worst case error when using scale feedback. The full proof is in Appendix B.

5.2 Active Query Generation

To learn $\boldsymbol{w}^*$ efficiently, the robot chooses the query (P, Q) it presents to the user. While randomly selected queries often lead to some learning progress, actively designing a query can drastically improve learning when the number of iterations is limited. Two recent approaches for learning from choice are information gain [4] and max regret [5]. Information gain seeks to reduce the robot's uncertainty over $\boldsymbol{w}$ while choosing queries that are easy to answer for the user. Max regret, on the other hand, minimizes the maximum regret by showing mutual worst case paths, which also results in easy queries. We leverage both of these methods for our active query generation in scale feedback.

We start with the information gain. Let H denote Shannon's information entropy [36]. As the outcome of the query is yet unknown, a greedy step takes the expectation over μ:

$$\max_{P,Q} H(\boldsymbol{w}, \alpha \mid P, Q) - \mathbb{E}_{\mu|P,Q}\big[H(\boldsymbol{w}, \alpha \mid \mu, P, Q)\big]. \tag{15}$$

We approximate the computation of entropy by summing over a set Ω of M samples of $(\boldsymbol{w}, \alpha) \sim f$. Thus, following the derivation in Biyik et al. [4], the new query (P, Q) solves

$$\max_{P,Q} \sum_{\mu} \sum_{(\boldsymbol{w},\alpha)\in\Omega} \frac{\mathbb{P}(\mu \mid P, Q, \boldsymbol{w}, \alpha)}{M} \log_2 \left(\frac{M \cdot \mathbb{P}(\mu \mid P, Q, \boldsymbol{w}, \alpha)}{\sum_{(\boldsymbol{w}',\alpha')\in\Omega} \mathbb{P}(\mu \mid P, Q, \boldsymbol{w}', \alpha')} \right). \tag{16}$$

The max regret policy generates queries (P, Q) such that if the robot learned P but the user optimal solution would be Q is a worst case. With a symmetric perspective over P and Q, we have

$$\max_{\boldsymbol{w}^P,\alpha^P,\boldsymbol{w}^Q,\alpha^Q} \mathbb{P}(\boldsymbol{w}^P, \alpha^P \mid D_k)\mathbb{P}(\boldsymbol{w}^Q, \alpha^Q \mid D_k)\Big(R(\boldsymbol{w}^P, \boldsymbol{w}^Q) + R(\boldsymbol{w}^Q, \boldsymbol{w}^P)\Big), \tag{17}$$

where $R(\cdot, \cdot)$ is the reward difference defined in (2). By observing feedback to such queries it greedily improves the probabilistic worst case error. In contrast to the information gain approach, maximum regret requires P and Q to be optimal trajectories for some users $(\boldsymbol{w}^P, \alpha^P)$ and $(\boldsymbol{w}^Q, \alpha^Q)$. On the other hand, maximum regret does not require a one-step look-ahead and thus no summation over potential feedback values μ, making it computationally lighter.

Equations (16) and (17) now give us two different policies for solving the initial problem (3). In the simulations, we compare how the performance of both benefits from scale feedback.

6 Simulation Results

We now present our main simulation results. Additional results can be found in the Appendix.

Experiment Setup. We simulate the presented framework using the Driver experiment used in [2, 4–6]. We modify the setup by adding 6 new features, obtaining a more challenging 10-dimensional problem (details on the features, as well as results for the original driver can be found in the Appendix). 71 distinct user preferences $\boldsymbol{w}^*$ are drawn uniformly at random, and each user is simulated with $\alpha^* \in \{.25, .5, .75, 1\}$, making it 284 runs for each method. We set $\sigma = 0.1$ for the noise level. We generate a set of 200 distinct sample trajectories by drawing random weights $\boldsymbol{w}$ and then computing their optimal trajectories. The active query generation methods then optimize over this set. We evaluate learning using the alignment metric and the relative reward.

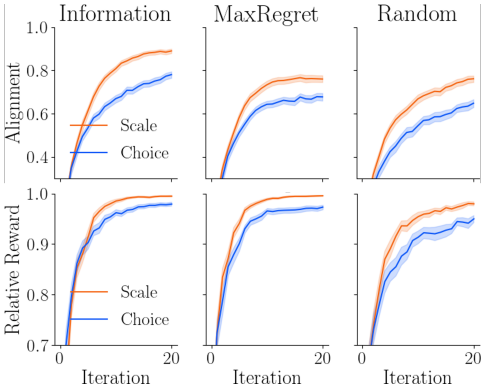


Figure 4: Comparison of scale feedback and soft choice for different active query methods.

As a baseline we use soft choice (strict choice showed a slightly poorer performance). To ensure a fair comparison, we emulate soft choice by setting the step size to $\epsilon = 1$ and use the same noise model for both forms of feedback.

Results. Fig. 4 shows the alignment and relative reward for the driver experiment for information gain, max regret and random query generation. We observe that in all cases scale feedback significantly improves the performance over soft choice in both metrics ($p < .001$ in all cases with two-sample t-test). When using the proposed scale feedback, the alignment after 20 iterations improves from .77 to .86 for information gain, from .67 to .76 for max regret, and from .64 to .75 for random queries. The relative reward improves for information gain and max regret similarly from .97 to 1, i.e., the learned solution is optimal. Both methods make most progress during the first 10 iterations. Random queries improve the final relative reward from .94 to .97. Overall, the simulation showcases that scale feedback improves learning, independent of the query selection method. For information gain and max regret, scale feedback allows for finding optimal solution, i.e., collecting 100% reward, within a small budget of iterations. In Appendix D, we show additional simulation results for higher noise.

7 User Study

Finally, we analyze the scale feedback in comparison with choice feedback and under different active querying methods with two user studies.[1] In both studies, we used $\epsilon = 0.1$ for scale queries.

Experiment Setup. We designed a serving task with a Fetch robot [37] as shown in Fig. 1, and generated a dataset of 120 distinct trajectories. Human subjects were told they should train the robot to bring the drink to the customer in the manner they prefer, paying attention to the following five factors: the drink (out of 3 options) to be served, the orientation of the pan in front of the robot, moving the drink behind or over the pan, the maximum height of the path, and the speed. The subjects were also informed about the types of queries they will respond to.

Independent Variables. In the first experiment, we wanted to compare scale and soft choice under random querying, and scale under random and information gain querying. Hence, we varied the query type and the querying algorithm among: (i) soft choice with random querying, (ii) scale with random querying, and (iii) scale with information gain querying. In the second experiment, we wanted to compare scale and soft choice under information gain querying. Hence, we employed: (i) soft choice with information gain querying, and (ii) scale with information gain querying. For all, we took $\sigma = 0.35$ based on pilot trials with different users (see Appendix E).

Procedure. We recruited 18 participants (5 female, 13 male, ages 20 – 55) for the first, and 14 participants (5 female, 9 male, ages 20 – 56) for the second experiment. Due to the pandemic conditions, the subjects participated in the study remotely with an online interface as in Fig. 1. The study started with an instructions page with a two-question quiz to make sure the participants understood how to use the interface. After reading the instructions, we had the subjects fill a form where they indicated their preferences for each of the five individual factors described above, to encourage them to be consistent in their responses during the data collection.

In the experiments, each participant responded to 10 queries generated with each of the algorithms. After each of these 10-query sets, they were shown the optimal trajectory from the dataset with respect to their learned reward function. The participants responded to a 5-point Likert scale survey (1-Strongly Disagree, 5-Strongly Agree) for this trajectory: "The displayed trajectory fits my preferences on the task." We also collected scale feedback for 10 more randomly-generated queries for validation in each experiment. We randomized the order of these sets (of 10 queries) to prevent any bias. The interface provided a "Sync Videos" button to restart both videos for easier comparison.

Dependent Measures. As an objective measure of the learning performance, we calculated the log-likelihood of the validation set (of 10 scale queries[2]) under the posterior $f(\boldsymbol{w}, \alpha \mid D)$ learned using the 10 queries generated via each algorithm, i.e., we calculated:

$$\texttt{Log}-\texttt{Likelihood} = \log \mathbb{P}(D_{\text{validation}} \mid D) = \log \mathbb{E}_{\boldsymbol{w}|D}\left[\mathbb{P}(D_{\text{validation}} \mid \boldsymbol{w})\right] \tag{18}$$

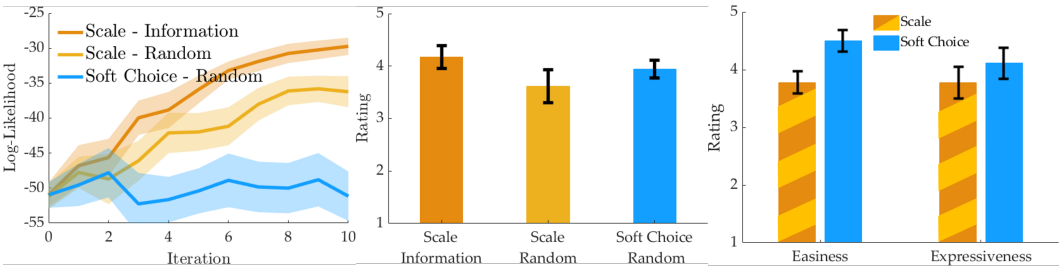
We also used the responses to the 5-point Likert scale survey questions to measure how well the learned rewards achieve the task. Finally, the users took a post-experiment survey where they rated (from 1 to 5) the easiness and expressiveness of soft choice and scale questions.

Hypotheses. We test the following hypotheses.
H1. *Scale feedback leads to faster learning than soft choice feedback.*
H2. *Querying based on information gain accelerates learning compared to random querying.*
H3. *Users will prefer information gain over random querying in terms of the optimized trajectories.*
H4. *Users will prefer scale feedback over soft choice feedback in terms of the optimized trajectories.*
H5. *Users will rate the scale feedback as easy as soft choice feedback.*
H6. *Users will rate the scale feedback as expressive as soft choice feedback.*

Results. We present results of the first and the second experiments in Figs. 5 and 6, respectively. It can be seen that the log-likelihood of the validation set after learning the reward function via scale feedback is higher than learning via soft choice feedback, under both random and information querying. Besides, information gain based query generation accelerates the learning and leads to higher log-likelihood values compared to random querying. All of these comparisons are statistically significant with $p < .001$ (paired-sample t-test), so they strongly support **H1** and **H2**.

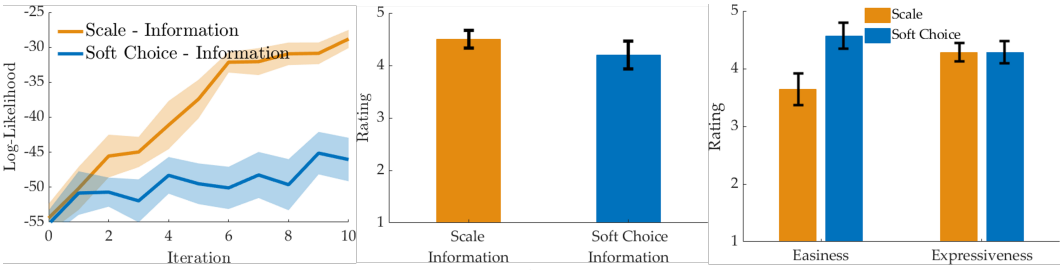
[1] We have IRB approval from a research compliance office under the protocol number IRB-52441. A summary video is at https://sites.google.com/view/reward-learning-scale-feedback, and the code at https://github.com/Stanford-ILIAD/reward-learning-scale-feedback.

[2] We present results with a validation set that consists of both scale and soft choice feedback in Appendix F.



(a) (b) (c)

Figure 5: All results are shown for the first experiment (mean±s.e. over 18 subjects).



(a) (b) (c)

Figure 6: All results are shown for the second experiment (mean±s.e. over 14 subjects).

In Fig. 5(b), it can be seen active querying led to learning reward functions that better optimize trajectories compared to random querying – this comparison was somewhat significant with $p \approx .05$, supporting **H3**. In fact, when we fit a Gaussian distribution to the ratings, we observe that it is 1.95 times as likely to get a better rating with information gain querying than random querying. Surprisingly, learning via soft choice achieved slightly higher reward than learning via scale when queries were randomly selected, and slightly lower reward when queries were generated based on information gain. However, these comparisons are not statistically significant. This is indeed analogous to the relative reward comparisons in Fig. 4: more complex tasks might be needed to better analyze the difference between the two methods. Thus, we neither reject nor accept **H4**.

Finally, the subjective results in Fig. 5(c) and 6(c) suggest that users find the soft choice feedback slightly, but consistently, easier than the scale feedback ($p < .01$), rejecting **H5**. This is not surprising, as it is often easier to make a pairwise comparison and the "About Equal" option in the soft choice questions makes them even easier [4]. On the other hand, there was no statistically significant difference in terms of expressiveness of scale and soft choice feedback, partially supporting **H6**. In summary, it is interesting that our users perceived the soft choice as easier and even more expressive at times; even though quantitatively, the scale feedback significantly outperforms the soft choice.

8 Discussion

Summary. We proposed scale feedback for reward learning where users provide more nuanced feedback than choice. We introduced a user model and showed how a robot can infer reward from noisy scale feedback. We adapted state-of-the-art query generation methods to accelerate learning. In simulations and a user study, scale feedback significantly improved learning. Users rank choice feedback as slightly easier, but both forms of feedback as equally expressive. However, the minor decrease in ease of use is out-weighted by a strong improvement in learning performance.

Future Work. We proposed scale queries as a way to give nuanced feedback between two trajectories. It is possible to extend them to $n + 1$ trajectories, with specialized user interfaces that allow users to select a point from an n-simplex instead of a slider bar. Future work should investigate this and if users can still give reliable feedback to these more complex queries.

In our experiments, we used a pre-computed trajectory set. Alternatives, e.g., optimizing queries over action sets as in [2], or using planners as in [5], should be studied for real-time online learning systems. The high estimate of σ in the user studies suggests the proposed probabilistic model may be inaccurate. Future work should refine the user model, including interactively learning σ; or fit a new user model that does not necessarily adopt a Gaussian noise. Surprisingly, users did not perceive scale feedback as more expressive. This could be addressed with improving interface design as well as designing a query generation method that actively exploits the slider's expressiveness.

Acknowledgments

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would also like to acknowledge funding by NSF grants #1849952 and #1941722, FLI grant RFP2-000, and DARPA.

References

[1] H. J. Jeon, S. Milli, and A. D. Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. In *Advances in Neural Information Processing Systems (NIPS)*, Dec. 2020.

[2] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2017.

[3] E. Biyik, N. Huynh, M. J. Kochenderfer, and D. Sadigh. Active preference-based gaussian process regression for reward learning. In *Proceedings of Robotics: Science and Systems (RSS)*, July 2020.

[4] E. Biyik, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh. Asking easy questions: A user-friendly approach to active reward learning. In *Proceedings of the 3rd Conference on Robot Learning (CoRL)*, 2019.

[5] N. Wilde, D. Kulić, and S. L. Smith. Active preference learning using maximum regret. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10952–10959, 2020.

[6] C. Basu, M. Singhal, and A. D. Dragan. Learning from richer human guidance: Augmenting comparison-based learning with feature queries. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 132–140, 2018.

[7] R. Holladay, S. Javdani, A. Dragan, and S. Srinivasa. Active comparison based learning incorporating user uncertainty and noise. In *RSS Workshop on Model Learning for Human-Robot Communication*, 2016.

[8] K. Li, M. Tucker, E. Biyik, E. Novoseller, J. W. Burdick, Y. Sui, D. Sadigh, Y. Yue, and A. D. Ames. Roial: Region of interest active learning for characterizing exoskeleton gait preference landscapes. In *International Conference on Robotics and Automation (ICRA)*, May 2021.

[9] A. Shah, S. Wadhwania, and J. Shah. Interactive robot training for non-markov tasks. *arXiv preprint arXiv:2003.02232*, 2020.

[10] N. Wilde, D. Kulić, and S. L. Smith. Bayesian active learning for collaborative task specification using equivalence regions. *IEEE RA-L*, 4(2):1691–1698, Apr. 2019. ISSN 2377-3766.

[11] N. Wilde, A. Blidaru, S. L. Smith, and D. Kulić. Improving user specifications for robot behavior through active preference learning: Framework and evaluation. *IJRR*, 39(6):651–667, 2020.

[12] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan. Learning robot objectives from physical human interaction. In *Conference on Robot Learning*, pages 217–226. PMLR, 2017.

[13] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh. Learning reward functions by integrating human demonstrations and preferences. In *Proceedings of Robotics: Science and Systems (RSS)*, June 2019.

[14] M. Li, A. Canberk, D. P. Losey, and D. Sadigh. Learning human objectives from sequences of physical corrections. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[15] M. Cakmak, S. S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler. Human preferences for robot-human hand-over configurations. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1986–1993. IEEE, 2011.

[16] V. Myers, E. Biyik, N. Anari, and D. Sadigh. Learning multimodal rewards from rankings. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, Nov. 2021.

[17] D. Brown, W. Goo, P. Nagarajan, and S. Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*, pages 783–792. PMLR, 2019.

[18] M. Kollmitz, T. Koller, J. Boedecker, and W. Burgard. Learning human-aware robot navigation from physical interaction via inverse reinforcement learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11025–11031. IEEE, 2020.

[19] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[20] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[21] D. S. González, O. Erkent, V. Romero-Cano, J. Dibangoye, and C. Laugier. Modeling driver behavior from demonstrations in dynamic environments using spatiotemporal lattices. In *International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018.

[22] D. P. Losey and M. K. O'Malley. Including uncertainty when learning from human corrections. In *Proceedings of the 2nd Conference on Robot Learning (CoRL)*, pages 123–132. PMLR, 2018.

[23] J. Y. Zhang and A. D. Dragan. Learning from extrapolated corrections. In *International Conference on Robotics and Automation (ICRA)*, pages 7034–7040, 2019.

[24] W. Chu, Z. Ghahramani, and C. K. Williams. Gaussian processes for ordinal regression. *Journal of machine learning research*, 6(7), 2005.

[25] D. S. Brown, W. Goo, and S. Niekum. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conference on Robot Learning*, pages 330–359. PMLR, 2020.

[26] L. Chen, R. Paleja, and M. Gombolay. Learning from suboptimal demonstration via self-supervised reward regression. In *Proceedings of the 4th Conference on Robot Learning (CoRL)*, 2020.

[27] B. Argall, B. Browning, and M. Veloso. Learning by demonstration with critique from a human teacher. In *2007 2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 57–64. IEEE, 2007.

[28] Y. Cui and S. Niekum. Active reward learning from critiques. In *International Conference on Robotics and Automation (ICRA)*, pages 6907–6914. IEEE, 2018.

[29] C. Wirth, R. Akrour, G. Neumann, J. Fürnkranz, et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

[30] V. Villani, F. Pini, F. Leali, and C. Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55:248–266, 2018.

[31] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *ACM/IEEE international conference on Human-Robot Interaction*, pages 391–398. ACM, 2012.

[32] D. P. Losey, K. Srinivasan, A. Mandlekar, A. Garg, and D. Sadigh. Controlling assistive robots with learned latent actions. In *International Conference on Robotics and Automation (ICRA)*, pages 378–384. IEEE, 2020.

[33] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz. Keyframe-based learning from demonstration. *International Journal of Social Robotics*, 4(4):343–355, 2012.

[34] M. Racca, V. Kyrki, and M. Cakmak. Interactive tuning of robot program parameters via expected divergence maximization. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 629–638, 2020.

[35] S. Cabi, S. G. Colmenarejo, A. Novikov, K. Konyushkova, S. Reed, R. Jeong, K. Zolna, Y. Aytar, D. Budden, M. Vecerik, et al. Scaling data-driven robotics with reward sketching and batch reinforcement learning. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[36] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Publishing Company, Incorporated, 2010. ISBN 1441923225, 9781441923226.

[37] M. Wise, M. Ferguson, D. King, E. Diehr, and D. Dymesich. Fetch and freight: Standard platforms for service robot applications. In *Workshop on autonomous mobile service robots*, 2016.

A Error Function

We used three different measures for learning performance: alignment, relative reward and log-likelihood. We offer a brief discussion about the advantages and limitations of these measures. First, we note that alignment and relative reward require knowing the ground truth $\boldsymbol{w}^*$. Hence, they are only applicable in simulations where $\boldsymbol{w}^*$ is synthetically generated, but not applicable in user studies. Nevertheless, they allow for in-depth analysis of the learning progress in simulations.

The alignment directly describes how well the reward function of a user is learned. An advantage is that it is global, i.e., there are no different test and training alignments. However, unless a perfect alignment of 1 is obtained for some $\boldsymbol{w}$, it does not give a direct indication how *good* the behavior of a robot is (that is how much reward is collected) when optimizing for $\boldsymbol{w}$.

The relative reward directly addresses this issue. It expresses how much reward is collected when optimizing for learned weights $\boldsymbol{w}$, compared to optimizing for $\boldsymbol{w}^*$. This exploits the fact that the underlying problem of finding robot trajectories that maximize reward is sensitive towards the objective, i.e,. the weights. Thus, even for some weight $\boldsymbol{w} \neq \boldsymbol{w}^*$ the motion planner ρ might return the optimal trajectory: $\rho(\boldsymbol{w}) = \rho(\boldsymbol{w}^*)$. In that case, the $\boldsymbol{w}$ has an alignment of less than 1, i.e., not accurately describe the users reward function, but still leads to the optimal solution, which is captured with the relative reward measure. However, the main limitation of relative reward is that it is not global. Instead the measure is grounded in specific scenarios for which roll-outs are computed. Considering test scenarios in addition to the training can mitigate this limitation.

The log-likelihood measure has a key advantage over alignment and relative reward: It does not require $\boldsymbol{w}^*$. The log-likelihood measures how well the learned probability density function over $\boldsymbol{w}$ predicts a user's answer to a randomly generated set of validation queries. Unfortunately, this measure is indirect: the log-likelihood does not have a direct interpretation similar to the relative reward, and thus it is more suitable when comparing different methods. Furthermore, noise has a large impact on the log-likelihood: When the noise in the user responses is high, the user has a high-enough probability for moving the slider to anywhere on the bar. Thus, inaccurate predictions are not penalized heavily, leading to higher log-likelihood values.

B Proof of Proposition 1

We provide a proof for Proposition 1 in the paper.

Proposition 1 (Upper error bound)**.** Let D^S denote the observation made from scale feedback and D^C be the observation from choice feedback for the same set of queries. For any user weights $\boldsymbol{w}^*$, it holds in the noiseless setting that $\texttt{Err}^{\max}(\boldsymbol{w}^*, D^S) \leq \texttt{Err}^{\max}(\boldsymbol{w}^*, D^C)$.

Proof. To prove the statement, we show the feasible set obtained from scale feedback is a subset of the feasible set from choice feedback. We note $\delta^* > 0$ for any non-trivial problem instance, as otherwise every path would be equally optimal for any $\boldsymbol{w}^*$. For one of the queries that form D^S and D^C, say query k, we assume the user prefers P over Q without loss of generality, implying $\psi \geq 0$. For this query, choice feedback defines a feasible set $\mathcal{F}_k^{\texttt{Choice}} = \{\boldsymbol{w} \mid (\boldsymbol{\phi}^P - \boldsymbol{\phi}^Q) \cdot \boldsymbol{w} \geq 0\}$. First, we consider $\psi = 1$. This yields $\mathcal{F}_k^{\texttt{Scale}} = \{\boldsymbol{w} \mid (\boldsymbol{\phi}^P - \boldsymbol{\phi}^Q) \cdot \boldsymbol{w} \geq \alpha\delta(\boldsymbol{w})\}$. Since both $\alpha > 0$ and $\delta(\boldsymbol{w}) \geq 0$, we obtain $\mathcal{F}_k^{\texttt{Scale}} \subseteq \mathcal{F}_k^{\texttt{Choice}}$. For the case $\psi \in [0, 1)$, we have $\mathcal{F}_k^{\texttt{Scale}} = \{\boldsymbol{w} \mid (\boldsymbol{\phi}^P - \boldsymbol{\phi}^Q) \cdot \boldsymbol{w} = \psi\alpha\delta(\boldsymbol{w})\}$; the right hand side is non-negative and thus any $\boldsymbol{w}$ satisfying the equality must satisfy $(\boldsymbol{\phi}^P - \boldsymbol{\phi}^Q) \cdot \boldsymbol{w} \geq 0$. This also implies $\mathcal{F}_k^{\texttt{Scale}} \subseteq \mathcal{F}_k^{\texttt{Choice}}$. As $\texttt{Err}^{\max}(\boldsymbol{w}^*, D^S)$ maximizes over $\mathcal{F}^{\texttt{Scale}}$, which is the intersection of $\mathcal{F}_k^{\texttt{Scale}}$'s over queries, while $\texttt{Err}^{\max}(\boldsymbol{w}^*, D^C)$ maximizes over $\mathcal{F}^{\texttt{Choice}}$, $\texttt{Err}^{\max}(\boldsymbol{w}^*, D^S)$ cannot attain a larger value than $\texttt{Err}^{\max}(\boldsymbol{w}^*, D^C)$. □

C Environment Features

Before we present additional simulation results, we now describe the features of the simulation and user study environments we used. These environments are: Extended Driver, which we used for the simulations in the main paper, Original Driver, which was used in [4] and we present the results in Appendix C.2, and finally Fetch Robot, which we used for the user studies again in the main paper.

C.1 Extended Driver

In Table 1 we detail the features of the extended driver scenarios. Notation: d_1, d_2, d_3 are the squared distances of the robot car to the center of the left, middle and right lane; $\boldsymbol{v}$ is the speed profile of the

Table 1: Features of the Extended Driver Environment

	Description	Definition
ϕ_1	Lane keeping: mean distance to closest lane center	$\texttt{mean}[\exp(-30 \cdot \min\{d_1, d_2, d_3\})] / 0.15343634$
ϕ_2	Keep speed: mean difference to speed 1	$\texttt{mean}[(1-\boldsymbol{v})^2] / 0.42202643$
ϕ_3	Driving straight: mean heading θ	$\texttt{mean}[\theta] / 0.06112367$
ϕ_4	Collision avoidance 1: mean distance to other car	$\texttt{mean}[\exp(-7 \cdot \Delta\boldsymbol{x}^2) + 3 \cdot \Delta\boldsymbol{y}^2] / 0.15258019$
ϕ_5	Collision avoidance 2: min distance to other car	$\texttt{min}[\exp(-7 \cdot \Delta\boldsymbol{x}^2) + 3 \cdot \Delta\boldsymbol{y}^2] / 0.10977646$
ϕ_6	Smoothness: mean jerk	$\texttt{mean}[\Delta\dot{\boldsymbol{v}}] / 0.00317041$
ϕ_7	Distance travelled: progress along the road	$x(t_{\texttt{final}}) - x(0) / 1.01818467$
ϕ_8	Final lane L: robot end in the left lane	$\texttt{int}(y(\lvert\lvert t_{\texttt{final}}) - c_1 \rvert\rvert < 0.08)$
ϕ_9	Final lane M: robot end in the center lane	$\texttt{int}(y(\lvert\lvert t_{\texttt{final}}) - c_2 \rvert\rvert < 0.08)$
ϕ_{10}	Final lane R: robot end in the right lane	$\texttt{int}(y(\lvert\lvert t_{\texttt{final}}) - c_3 \rvert\rvert < 0.08)$

robot trajectory; $\dot{\boldsymbol{v}}$ the acceleration profile; θ is the heading of the car, $x(t)$ and $y(t)$ are the robots x and y position at a given time $t \in [0, t_{\texttt{final}}]$ (x is orthogonal to the road, y is along the road); $\Delta\boldsymbol{x}$ and $\Delta\boldsymbol{y}$ are the ordinal distance between the robot car and the other car; and c_1, c_2, c_3 are the y-coordinates of the lane centers.

C.2 Original Driver

We refer to the Section 9.4 of [4] for the features of the original driver environment.

C.3 Fetch Robot

In the user studies presented in the main paper and the simulations presented in Appendix D.3, we used the following eight features for the Fetch robot experiment:

- Speed of the end-effector $\in \{0, 0.33, 0.67, 1\}$
- Maximum height of the end-effector $\in \{0, 0.33, 0.67, 1\}$
- Selected drink being the orange juice $\in \{0, 1\}$
- Selected drink being the water $\in \{0, 1\}$
- Selected drink being the milk $\in \{0, 1\}$
- Orientation of the pan $\in \{0, 1\}$
- Moving the drink behind or over the pan $\in \{0, 1\}$
- Robot hitting the pan while moving the drink $\in \{0, 1\}$

D Simulation results

We present additional simulation results to compare the proposed scale feedback with soft choice. For the extended driver model from the main paper, we additionally show data with higher noise, and show results with the log-likelihood measure used in the user study. Further, we show the same analysis for the original driver experiment, and for the simulated version of the fetch robot experiment from the user study.

For all the simulation results in this Appendix, we simulated 40 different $\boldsymbol{w}^{\text{user}}$ vectors, each with four different $\alpha^{\text{user}} \in \{.25, .5, .75, 1\}$, making 160 runs in total.

D.1 Extended Driver

High Noise. In the main paper we showed results for user noise $\sigma = 0.1$ in Fig. 4. In addition, we repeat the same experiment but with $\sigma = 0.3$; shown in Fig. 7. Overall, we observe a poorer performance for all approaches compared to $\sigma = 0.1$ – higher noise in the user feedback makes learning more difficult. Nevertheless, scale feedback still leads to an improvement on both measures, alignment and relative reward.

Log-Likelihood. Fig. 8 shows the log-likelihood for the extended driver simulations. When the noise is small, scale feedback significantly outperforms soft choice under all three active querying methods. Further, information gain performs best overall, followed by random. It might be surprising that max regret achieves a lower log-likelihood than random. Max regret greedily tries to find solutions that are close to optimal. Thus, this approach does not gather information about comparably good or bad trajectories (with respect to collected reward). Since the set of validation queries is generated randomly, it might contain numerous queries about which the max regret approach is still uncertain since it only focused on finding close to optimal solutions. Information gain on the

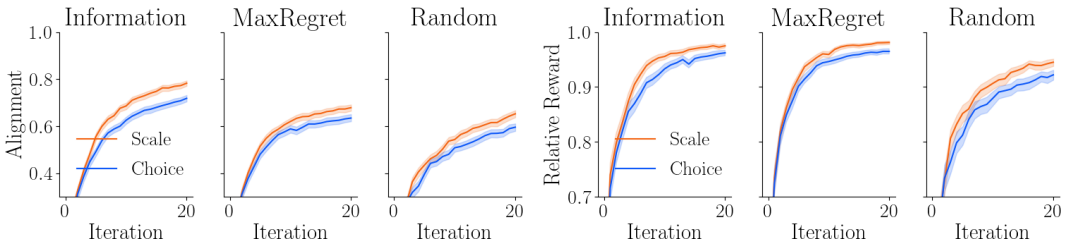
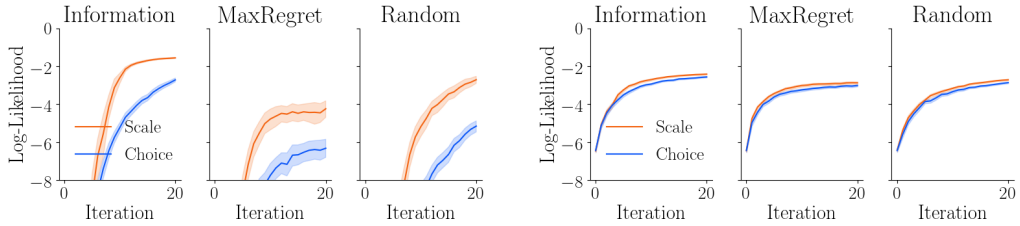


Figure 7: Alignment (left) and Relative Reward (right) for the Extended Driver with $\sigma = 0.3$.

other hand minimizes the uncertainty about weights, regardless of how different the resulting trajectories are. Similarly, random querying is completely unbiased and thus does not focus on a subset of queries as the max regret approach does.

In Fig. 8 (b) we show the log-likelihood for high noise. Here all three active querying methods perform nearly identical, and the difference between scale and soft choice feedback is very small. This is because, when the noise is high, i.e., when the Gaussian over the feedback value has high variance, the log-likelihood measure does not heavily penalize bad predictions, which causes all methods to acquire high log-likelihood values.



(a) $\sigma = 0.1$ (b) $\sigma = 0.3$

Figure 8: Log-Likelihood for the Extended Driver.

D.2 Original Driver

Alignment and Relative Reward. Next, we show results for the original driver experiment. Fig. 9 shows the alignment and relative reward for low noise ($\sigma = 0.1$), Fig.10 shows the same measures for high noise ($\sigma = 0.3$). While scale feedback still improves alignment and relative reward for all querying methods, the gap to soft choice feedback is smaller than for the extended driver. However, we observe that all querying methods achieve a substantially stronger performance than in the extended driver model with 10 features, indicating that the original driver model poses a less difficult learning problem with only 4 features. We notice that the result for soft choice using information gain achieves a higher alignment after 20 iterations than reported in [4]. There are two reasons for this: First, we use a Gaussian noise instead of the Boltzmann model. Second, by emulating soft choice using a slider with step size 1, we change the model for when users give a neutral ("About Equal") feedback. Nonetheless, the stronger performance compared to [4] suggests that these differences do not negatively impact the performance of soft choice with information gain, and thus that the shown comparisons of scale feedback and soft choice feedback are fair.

Log-Likelihood. We also report the results in the log-likelihood measure Fig. 11. The results are very similar to the results of the extended driver environment, except the log-likelihood values increase faster. This is again because the reward is easier to learn in the original driver environment with the fewer number of features.

D.3 Fetch Robot

Finally, we also show simulation results for the experimental setup from the user study, using the fetch robot. Fig. 12 shows the alignment and relative reward for low noise ($\sigma = 0.1$), Fig. 13 shows the same measures for high noise ($\sigma = 0.3$), and Fig. 14 shows the log-likelihood. In terms of the comparisons between different feedback types and different active querying methods, the results have the same trend as the extended driver and the original driver environments.

E Choice of σ in the User Studies

In the paper, we stated we took $\sigma = 0.35$ in the user studies based on pilot trials with different users. We now describe the procedure that yielded this selection of σ.

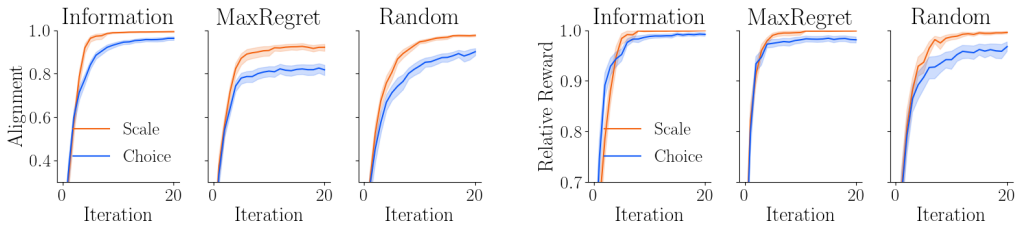


Figure 9: Alignment and Relative Reward for the Original Driver with $\sigma = 0.1$.

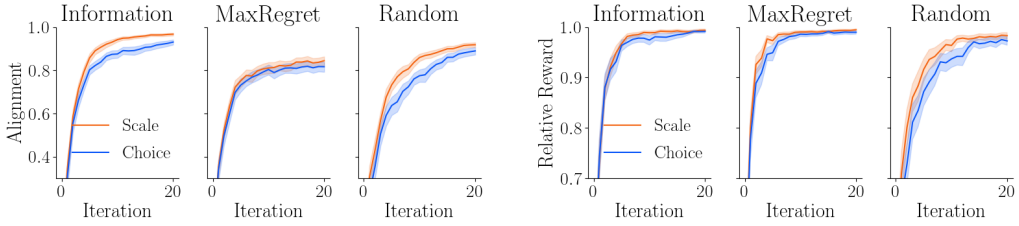
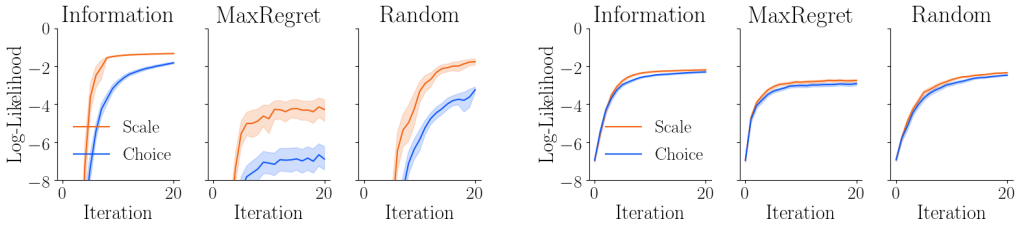


Figure 10: Alignment and Relative Reward for the Original Driver with $\sigma = 0.3$.



(a) $\sigma = 0.1$ (b) $\sigma = 0.3$

Figure 11: Log-Likelihood for the Original Driver.

Before all the actual experiments, we recruited 3 participants (3 male, ages 27–40) for a pilot study. In this study, the participants followed the same procedure as in our actual experiments, but responded to only 30 randomly generated queries. These 30 queries were formed by three sets: 10 scale queries, 10 soft-choice queries and another 10 scale queries. We randomized the order of these three sets to avoid any bias.

After we collected these data, we repeated the following procedure for $\sigma = 0.05, 0.10, \ldots, 1.00$. We learned a single posterior for each user by using 10 scale and 10 soft choice query responses under σ noise, i.e., the posteriors included both scale and soft choice feedback. We then checked the validation loglikelihood (with the remaining 10 queries) under the learned posterior and the same σ.

The σ value that yielded the highest validation loglikelihood, $\sigma = 0.35$, was then used for all of the actual experiments with real users.

F Validation Set with Mixture Data

In both of our user studies, we used a validation set that consists of randomly generated scale questions. Given the fact that the subjective user ratings did not point out a significant difference between learning from scale and soft choice feedback, one might argue that the superiority of learning from scale feedback in terms of the log-likelihood metric is simply because the validation set also consists of scale feedback. Mathematically, this should not happen, because a good posterior should be able to correctly predict any form of user feedback. However, humans have cognitive biases, which makes it possible that the posterior learned with the scale questions captures the bias caused by the scale questions, whereas the posterior learned with the soft choice questions cannot do this.

To show this is not the case, we present an additional analysis on the same human data as in our first user study. For this analysis, we take the reward posteriors that have been learned with the first 7 queries (of "Scale - Information Gain", "Scale - Random", and "Soft Choice - Random"). Next, we alter the validation set as follows. We take (i) the first 3 scale queries from the original validation set, and (ii) the last 3 soft choice queries from the original training set of randomly generated soft choice queries (and this is why we only take the first 7 posteriors – we do not mix the training and validation data). Finally, we perform the log-likelihood analysis on this modified validation set.

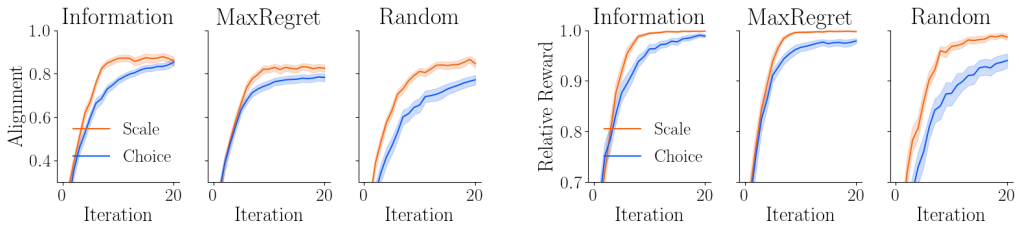


Figure 12: Fetch Experiment with $\sigma = 0.1$.

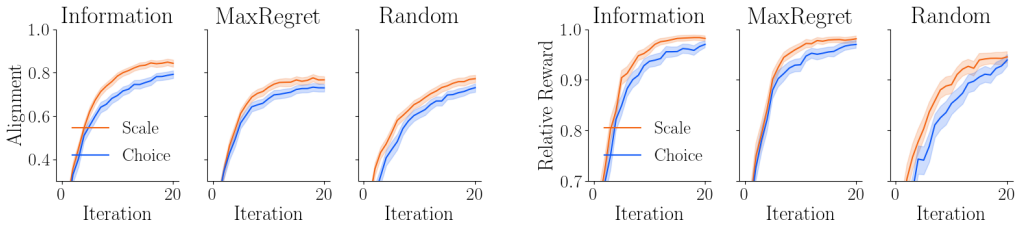
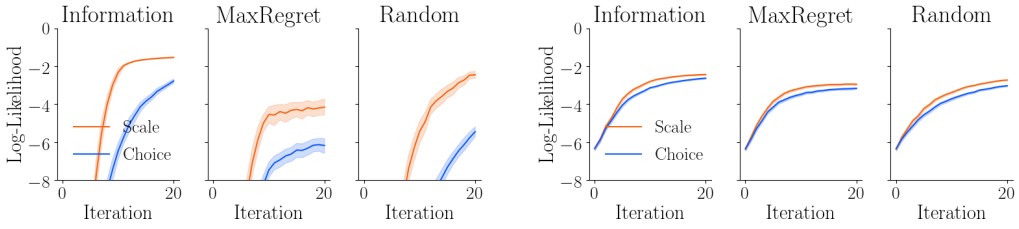


Figure 13: Fetch with $\sigma = 0.3$.



(a) $\sigma = 0.1$ (b) $\sigma = 0.3$

Figure 14: Log-Likelihood of the Fetch experiment.

Results are shown in Fig. 15. It can be seen that even with a validation set that consists of mixture data, the results have the same trend as in the original study results. While having smaller validation set (6 instead of the 10 in the original study) causes larger standard errors, "Scale - Information" and "Scale - Random" both outperform "Soft Choice - Random" with statistical significance ($p < 0.05$ in both comparisons). On the other hand, the comparison between "Scale - Information" and "Scale - Random" gives $p = 0.098$.

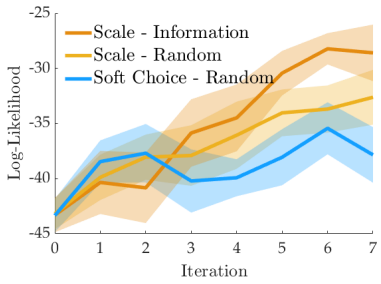


Figure 15: Additional analysis results are shown (mean±s.e. over 18 subjects).

This analysis shows the fact that scale feedback outperforms soft choice feedback in terms of log-likelihood is not because of the data in the validation set. Even with a validation set that consists of both scale and soft choice questions, we see the benefits of learning from scale queries.

However, this analysis does not answer the question why user ratings did not have a significant difference between the two feedback types. While the answer to this question requires more analysis and possibly more data collection, we speculate the following reason: the mean user ratings are always around 4, and even higher than 4 when queries are actively generated with information gain. This means the users are happy with the optimized trajectories, so we can say that 10 queries are enough in this task to find the optimal trajectory. However, while user ratings measure how close the optimal trajectory with respect to the robot's posterior is to the optimal trajectory the user has in mind; log-likelihood measures the predictive performance of the posterior. Therefore, having a high user rating does not necessarily mean the robot can accurately compare two suboptimal trajectories. On the other hand, a high log-likelihood value indicates good predictive performance, which is crucial in many robotics applications, such as behavior modeling. Hence, we claim: (i) learning from scale feedback improves the predictive performance over learning from soft choice feedback, and (ii) a more complex task might be needed to show scale feedback leads to more efficient learning than soft choice feedback, which is also suggested by our simulation studies.

G Numerical Results

Finally, we present Table 2 where we report the numerical results of the simulations in the main paper at iterations $0, 5, 10, 20$; and Table 3 where we report the final numerical results of the user studies. Consistent with the paper, the numbers are presented as mean $\pm$ standard deviation (simulations) and standard error (user study).

Table 2: Numerical results of the simulations at selected iterations k

	Mean$\pm$Standard Deviation			
Plot	$k = 0$	$k = 5$	$k = 10$	$k = 20$
Fig. 4 Scale - Information (Alignment)	$-.01 \pm .33$	$\mathbf{.62 \pm .19}$	$\mathbf{.81 \pm .16}$	$\mathbf{.9 \pm .08}$
Fig. 4 Choice - Information (Alignment)	$-.02 \pm .31$	$.52 \pm .18$	$.67 \pm .16$	$.79 \pm .15$
Fig. 4 Scale - MaxRegret (Alignment)	$.01 \pm .31$	$.57 \pm .19$	$.71 \pm .16$	$.75 \pm .16$
Fig. 4 Choice - MaxRegret (Alignment)	$-.03 \pm .3$	$.47 \pm .23$	$.59 \pm .17$	$.67 \pm .18$
Fig. 4 Scale - Random (Alignment)	$.01 \pm .33$	$.52 \pm .2$	$.67 \pm .17$	$.77 \pm .17$
Fig. 4 Choice - Random (Alignment)	$.02 \pm .32$	$.4 \pm .21$	$.52 \pm .2$	$.63 \pm .21$
Fig. 4 Scale - Information (Rel. Reward)	$.51 \pm .32$	$.92 \pm .12$	$.98 \pm .04$	$\mathbf{1.0 \pm .01}$
Fig. 4 Choice - Information (Rel. Reward)	$.5 \pm .3$	$.89 \pm .12$	$.95 \pm .07$	$.98 \pm .04$
Fig. 4 Scale - MaxRegret (Rel. Reward)	$.52 \pm .31$	$\mathbf{.96 \pm .07}$	$\mathbf{.99 \pm .02}$	$\mathbf{1.0 \pm .01}$
Fig. 4 Choice - MaxRegret (Rel. Reward)	$.51 \pm .3$	$.91 \pm .12$	$.95 \pm .06$	$.96 \pm .06$
Fig. 4 Scale - Random (Rel. Reward)	$.52 \pm .32$	$.89 \pm .14$	$.96 \pm .07$	$.99 \pm .03$
Fig. 4 Choice - Random (Rel. Reward)	$.52 \pm .32$	$.85 \pm .15$	$.89 \pm .12$	$.93 \pm .12$

Table 3: Final numerical results of the user study

Plot	Mean$\pm$Standard Error
Fig. 5(a) Scale - Information	-29.7 ± 1.2
Fig. 5(a) Scale - Random	-36.2 ± 2.2
Fig. 5(a) Soft Choice - Random	-51.2 ± 3.5
Fig. 5(b) Scale - Information	4.2 ± 0.2
Fig. 5(b) Scale - Random	3.6 ± 0.3
Fig. 5(b) Soft Choice - Random	3.9 ± 0.2
Fig. 5(c) Scale (Easiness)	3.8 ± 0.2
Fig. 5(c) Soft Choice (Easiness)	4.5 ± 0.2
Fig. 5(c) Scale (Expressiveness)	3.8 ± 0.3
Fig. 5(c) Soft Choice (Expressiveness)	4.1 ± 0.2
Fig. 6(a) Scale - Information	-28.8 ± 1.3
Fig. 6(a) Soft Choice - Information	-46.0 ± 3.1
Fig. 6(b) Scale - Information	4.5 ± 0.2
Fig. 6(b) Soft Choice - Information	4.2 ± 0.3
Fig. 6(c) Scale (Easiness)	3.6 ± 0.3
Fig. 6(c) Soft Choice (Easiness)	4.6 ± 0.2
Fig. 6(c) Scale (Expressiveness)	4.3 ± 0.2
Fig. 6(c) Soft Choice (Expressiveness)	4.3 ± 0.2