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 $\texttt{Err}^{\max}(\boldsymbol{w}^*, D^C)$. $\square$

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	$\texttt{mean}[\texttt{exp}(-30\cdot\min\{d_1,d_2,d_3\})]/0.15343634$
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	$\texttt{mean}[\texttt{exp}(-7\cdot\Delta\boldsymbol{x}^2)+3\cdot\Delta\boldsymbol{y}^2]/0.15258019$
ϕ_5	Collision avoidance 2: min distance to other car	$\texttt{min}[\texttt{exp}(-7\cdot\Delta\boldsymbol{x}^2)+3\cdot\Delta\boldsymbol{y}^2]/0.10977646$
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	$\texttt{int}(y(\vert\vert t_{\texttt{final}}) - c_1\vert\vert < 0.08)$
ϕ_9	Final lane M: robot end in the center lane	$\texttt{int}(y(\vert\vert t_{\texttt{final}}) - c_2\vert\vert < 0.08)$
ϕ_{10}	Final lane R: robot end in the right lane	$\texttt{int}(y(\vert\vert t_{\texttt{final}}) - c_3\vert\vert < 0.08)$

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

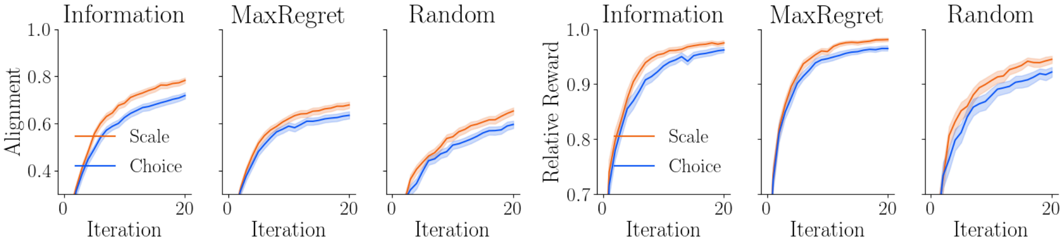
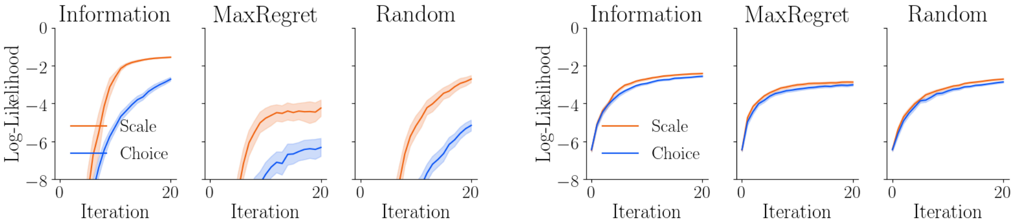


Figure 7: Alignment (left) and Relative Reward (right) for the Extended Driver with $\sigma = 0.3$.

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.

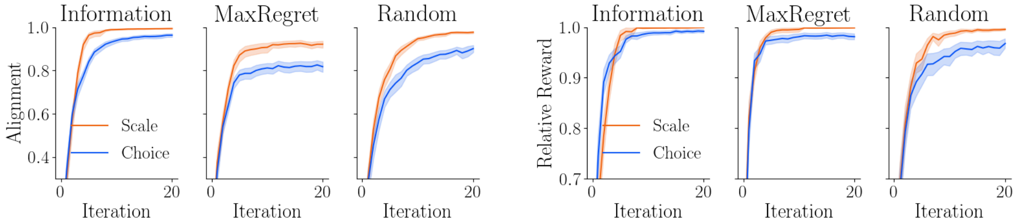


Figure 9: Alignment and Relative Reward for the Original Driver with $\sigma = 0.1$.

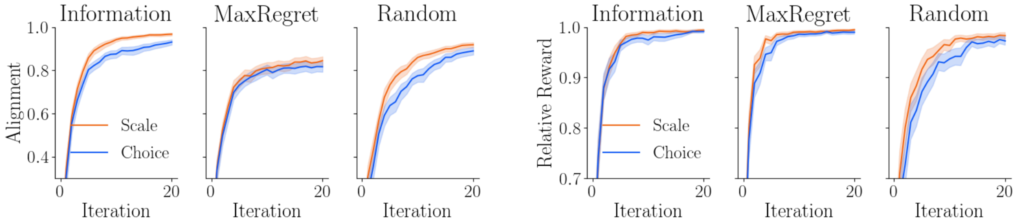
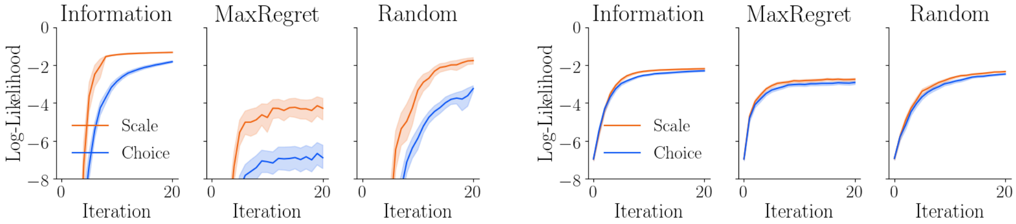


Figure 10: Alignment and Relative Reward for the Original Driver with $\sigma = 0.3$.



(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.

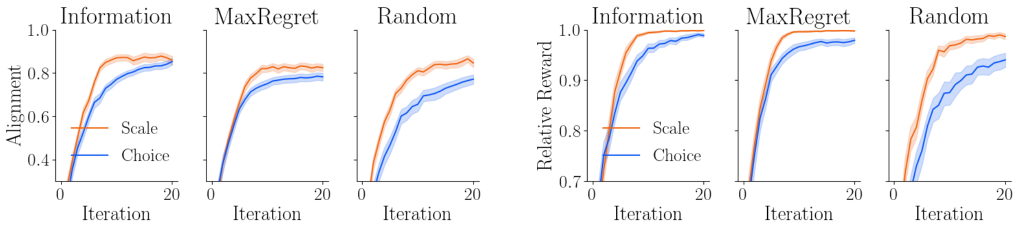


Figure 12: Fetch Experiment with $\sigma = 0.1$.

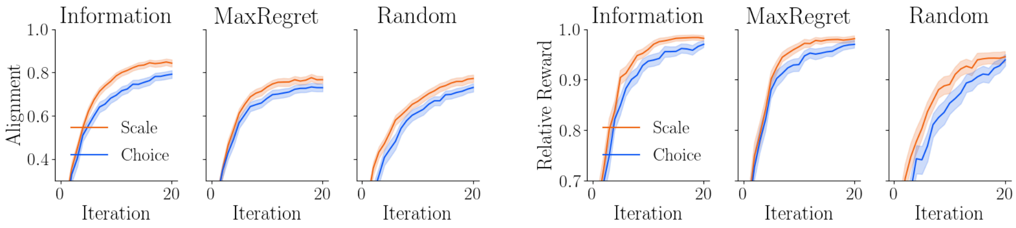
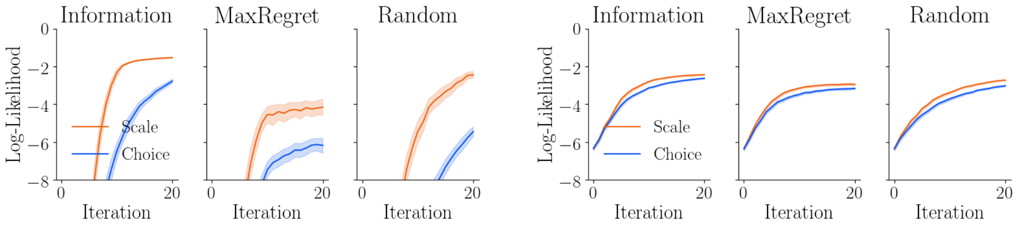


Figure 13: Fetch with $\sigma = 0.3$.



(a) $\sigma = 0.1$

(b) $\sigma = 0.3$

Figure 14: Log-Likelihood of the Fetch experiment.

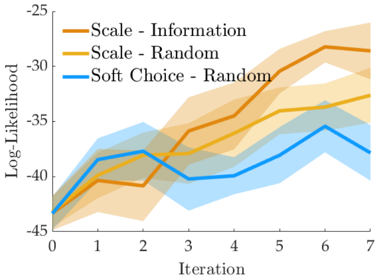


Figure 15: Additional analysis results are shown (mean±s.e. over 18 subjects).

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 $p = 0.098$.

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±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±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