# Pragmatic Feature Preferences: Learning Reward-Relevant Preferences from Human Input

**Andi Peng** [1] **Yuying Sun** [2] **Tianmin Shu** [3] **David Abel** [4]

## Abstract

Humans use social context to specify preferences over behaviors, i.e. their reward functions. Yet, algorithms for inferring reward models from preference data do not take this social learning view into account. Inspired by pragmatic human communication, we study how to extract fine-grained data regarding *why* an example is preferred that is useful for learning more accurate reward models. We propose to enrich binary preference queries to ask both (1) which features of a given example are preferable in addition to (2) comparisons between examples themselves. We derive an approach for learning from these feature-level preferences, both for cases where users specify which features are reward-relevant, and when users do not. We evaluate our approach on linear bandit settings in both vision- and language-based domains. Results support the efficiency of our approach in quickly converging to accurate rewards with fewer comparisons vs. example-only labels. Finally, we validate the real-world applicability with a behavioral experiment on a mushroom foraging task. Our findings suggest that incorporating pragmatic feature preferences is a promising approach for more efficient user-aligned reward learning.

## 1. Introduction

Learning user-aligned reward functions from human data is a cornerstone of efforts in value alignment and AI safety (Fisac et al., 2020; Amodei et al., 2016; Christian, 2021; Hadfield-Menell et al., 2017). Current efforts such as reinforcement learning (RL) from human feedback (RLHF) propose to learn reward functions from pairwise comparisons provided by human users (Christiano et al., 2017; Griffith et al., 2013). Motivated by the idea that pairwise comparisons are a relatively simple and easy way for users to provide offline input for training a reward model, RLHF approaches have been used to train more efficient robotic systems (Basu et al., 2018; Hüllermeier et al., 2008; Jain et al., 2015), and safer language models (LMs) (Bai et al., 2022a;b). Unfortunately, because such feedback is provided over example pairs, valuable information regarding fine-grained components of the reward, i.e. *which* features of the examples matter and *why*, are lost (Basu et al., 2018).

As a simple example, consider taking up the task of mushroom foraging introduced by Sumers et al. (2022) (Fig. 1). How might we learn which mushrooms are good for foraging? A pairwise comparison between two examples may tell us that one mushroom is better (that is, more delicious) than the other, but not the reason why (green mushrooms tend to be zestier in flavor). Moreover, users may not hold the same preferences over which *features* of mushrooms are important—a chef may prefer mushrooms to taste delicious but a collector may instead prefer them to look exotic. In other words, there may be different *reward-relevant features* that shape each user's *preference relation* such that their underlying reward functions are different.

If we assume the user in question is not simply acting as an oracle providing labels divorced from the learning process, but rather as an engaged cooperative agent capable of providing descriptive feedback, we can treat users as active *teachers* capable of providing richer information regarding their underlying reward function. Such pedagogical models have been found to be useful for guiding RL agents from actions (Ho et al., 2016; Goyal et al., 2019) and language feedback (Bisk et al., 2016; Sumers et al., 2022; Lin et al., 2022). How might we do the same for preference learning?

In this work, we propose a pedagogical framework for modeling feature-level pairwise comparisons and design a joint loss to learn rewards from both feature and example-level comparisons. Our key insight is that humans communicate preferences *pragmatically*: when they describe which features of each example are important to their preference, they are also implicitly revealing which features *are not important*. For example, as shown in Figure 1, the fact that a user prefers a mushroom because of its color and

[1]Massachusetts Institute of Technology [2]Boston University [3]Johns Hopkins University [4]Google DeepMind. Correspondence to: Andi Peng <andipeng@mit.edu>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

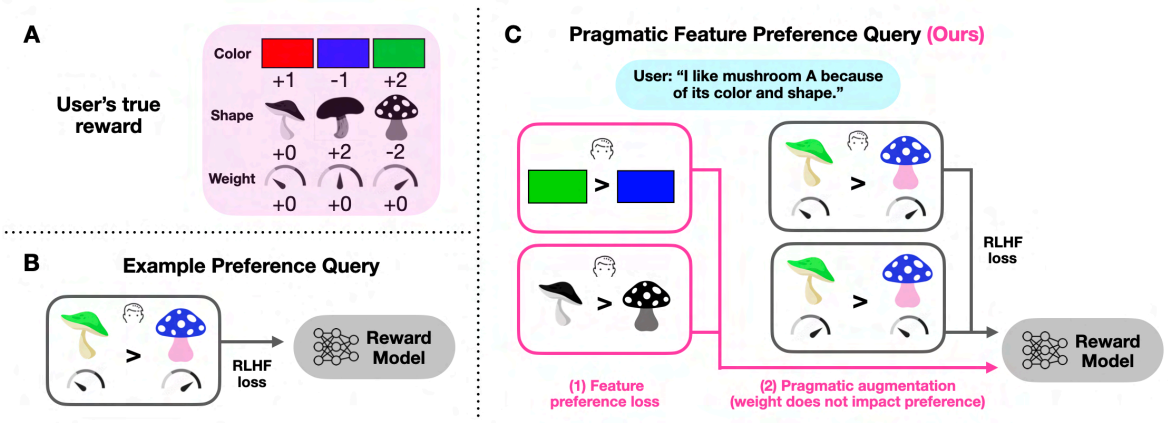


*Figure 1.* **A.** An illustrative user reward function in the mushroom foraging task. Rewards are a linear combination of color, shape, and weight features. **B.** *Example preference* queries learn a traditional RLHF loss over example-level comparisons. **C.** Our approach, *pragmatic feature preference* queries, makes use of (1) fine-grained feature-level preferences in conjunction with example-level preferences, and (2) language descriptions to infer reward-relevant features and augment preference data.

shape might implicitly reveal that they do not care about mushroom weight or that weight does not matter for their preference. This information can be used to expand the existing comparison-level data greatly, e.g., the user should hold the same preference over these two mushrooms *even if their weights were flipped*. We introduce this pedagogical approach as learning from *pragmatic feature preferences*.

First, we formalize the relationship between preferences over examples and preferences over features in a linear bandit setting. We propose a method to query for feature-level as well as example-level preferences and define a joint loss for learning from such input. Second, we contribute a pragmatic approach for making additional use of this data by performing feature-level augmentation of *non-relevant* reward features from linguistic preference descriptions.

We evaluate our approach in experiments in both the mushroom foraging task (a vision-based domain) and a flight booking task (a language-based domain) (Lin et al., 2022). We find that learning from pragmatic feature preferences outperforms baselines that only learn from either only example-level preferences or only pragmatic-augmented features, verifying that both elements are important for making use of contextual information contained in preference descriptions. Importantly, we verify in a user study that such rich queries *do not* significantly increase user effort with providing labels compared to RLHF. Overall, our findings suggest that incorporating models of pragmatic human communication is important for efficient user-aligned reward learning.

[0]Code available at github.com/andipeng/feature-preference

## 2. Related Work

Traditional RL assumes that the reward function is given to a decision-making agent (Sutton, 1992), a practice that is subject to value misalignment and misspecification (Amodei et al., 2016). Ergo, a growing body of work proposes to instead infer the reward function from human data.

**Learning from demonstrations (IRL).** Inverse reinforcement learning (IRL) methods propose to learn the reward function from observed actions in the environment, e.g. human demonstrations (Ng & Russell, 2000; Abbeel & Ng, 2004; Ziebart et al., 2008). Unfortunately, such methods suffer from identifiability issues (Ziebart et al., 2008; Sumers et al., 2022). That is, multiple reward functions can explain the same observed behavior. Moreover, IRL suffers from strong assumptions regarding the optimality of the demonstrator, or in other words, that the observed actions are always optimal under the user's true reward.

**Learning from pairwise preferences (RLHF).** With the rise of language models (LMs), there is renewed interest in learning rewards from pairwise preferences, colloquially referred to as reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Griffith et al., 2013). Motivated by the idea that binary preference labels are less burdensome for human users to provide, RLHF has emerged as a popular method for fine-tuning LMs (Kaufmann et al., 2023; Wu et al., 2023), although there are open questions regarding its efficiency and accuracy of reward modeling to true human preferences (Casper et al., 2023).

**Learning from teachers (pedagogy).** Unlike the above approaches which assume data is generated by a user that is merely *showing* what the correct thing to do is, pragmatic approaches instead incorporate models of users that

are *teaching* (Ho et al., 2016; Sumers et al., 2022; Lin et al., 2022) why this is the correct thing to do. This subtle distinction manifests in algorithms that explicitly incorporate pedagogical models, i.e. models where human-generated data is intentionally intended to be informative about the user's underlying reward function (Hadfield-Menell et al., 2017; Fisac et al., 2020).

**Learning with state abstraction.** There is substantial evidence to suggest much of the generalizability of human learning and planning can be attributed to abstraction, i.e. the selective filtering of task-relevant information (Ho et al., 2019; 2022). This suggests that flexibly creating abstractions containing task-relevant features is important to downstream generalizable learning, particularly with limited examples (Peng et al., 2024; Bobu et al., 2023).

In this paper, we unify different streams of work in pedagogical reward learning and human abstraction to develop a model of learning from pairwise preferences that takes into account human input that is explicitly informative of task-relevant features. Such a framework offers two benefits: first, targeting preference data to learn rewards at the feature-level enables more efficient learning given limited comparisons; second, humans can provide descriptive feedback on important features in language, offering a more natural teaching process. In the next section, we explore how both can be utilized to learn better reward models.

## 3. Preliminaries

Our primary focus is on the *reward modeling problem* in which we seek to learn a reward function that aligns with a user's unknown preference relation while observing only finitely many comparisons from that preference relation.

We study reward modeling in contextual bandit problems (Langford & Zhang, 2007; Lattimore & Szepesvári, 2020), which are a middle point between $k$-armed bandits and full sequential decision-making problems. A contextual bandit presents a challenging decision-making problem due to both the explore-exploit dilemma and generalization but does not introduce the complexities of credit assignment and long-term planning. For this reason, it is a compelling choice for studying preference-based reward modeling.

**Contextual Bandits.** A contextual bandit in its general form is a model of a decision-making problem defined by the tuple $(\mathcal{C}, \mathcal{A}, \mu, R)$, where $\mathcal{C}$ is a set of contexts, $\mathcal{A}$ is a set of actions, $\mu \in \Delta(\mathcal{C})$ is a probability distribution over $\mathcal{C}$, and $R : \mathcal{C} \times \mathcal{A} \to \mathbb{R}$ is a reward function. We note that while the reward function is typically stochastic in most bandit problems, in the setting we study, the reward function is deterministic. At each time step, a context $c \in \mathcal{C}$ is sampled $c \sim \mu$ and presented to the decision-maker. The decision-maker then chooses an action $a \in \mathcal{A}$ and observes $R(c, a)$ with the goal of maximizing some measure of long term reward. We follow the conventions of Sumers et al. (2022) and study a special case of linear contextual bandits (Li et al., 2010) in which each context is a subset of the action space that the agent is allowed to choose from in that context. For instance, in the mushroom foraging task, each context is a collection of mushrooms the agent must choose from. More formally, the action space is the set of $n$-dimensional vectors, $\mathcal{A} = \mathbb{R}^n$, and each context is simply a subset of this space, $c \subseteq \mathcal{A}$. The agent is then only allowed to choose an action contained in the current context, and the reward function is only well-defined for cases where $a \in c$. In such cases, it is sufficient to express the reward function as only a function of $a$, $R : \mathcal{A} \to \mathbb{R}$.

**Reward Modeling.** In a contextual bandit of the kind described above, the reward modeling problem is defined as follows. We are given as input the context set $\mathcal{C}$, the action space $\mathcal{A}$, and a finite set of preference data over actions $\mathcal{D} = \{(a_i, a_i', f(a_i, a_i'))\}_{i=1}^m$, where $a_i, a_i' \in \mathcal{A}$ are each actions, and $f : \mathcal{A} \times \mathcal{A} \to \{\succ, \prec, \sim\}$ is a function mapping each action pair to a preference relation. We suppose the preference relation is unknown, and wish to learn a reward function, $\hat{R} : \mathcal{A} \to \mathbb{R}$ that aligns with the underlying preference relation that generated the preference data. Notice that since each context is simply a subset of the action space, the preference relation of interest is over pairs of actions, and the reward function we wish to learn is also a function of action, rather than a context-action pair.

Following previous work in IRL (Sumers et al., 2022), we assume that the reward $\hat{R}$ (e.g. tastiness of a mushroom) is a linear combination of feature rewards $\hat{R}^j$ (e.g. tastiness of a green mushroom), so that: $\hat{R}_\theta(a) = \sum_{j=1}^n \theta^j \hat{R}^j(a^j)$, where $a^j$ is the value associated with a specific feature (e.g. green), and $\theta^i$ is the $i$-th element of a linear weight vector on feature rewards. When clear from context, we abbreviate $\hat{R}^j(a^j)$ to simply $\hat{R}(a^j)$. The traditional goal is then to learn a $\theta$ such that $\hat{R}_\theta(a_1) > \hat{R}_\theta(a_2)$ if and only if $f(a_1, a_2) = \succ$.

We propose to consider pairwise *feature preferences* over different settings of an individual feature of each action. For instance, consider two actions comprised of three features, $a_1 = \langle 0, -1, 2\rangle$ and $a_2 = \langle 1, 0, 0\rangle$. We refer to the $j$-th feature of action $a_2$ as $a_2^j$. We then let $\phi : \mathbb{R} \times \mathbb{R} \to \{\succ, \prec\}$ express a *feature preference* relation, indicating whether the value of the $j$-th feature of one action is preferred to another.

For example, consider two actions $a_1 = \langle 1, 0, 20\rangle$, and $a_2 = \langle 5, 2, 12\rangle$ each describing a mushroom. Suppose the first feature ($a_1^1 = 1, a_2^1 = 5$) captures the zestiness of the mushroom. A user that dislikes zesty might be thought of as maintaining the feature preference relation $\phi(a_1^1, a_1^2) = \succ$. As a shorthand, we denote such outcomes as $a_1^1 \succ_\phi a_1^2$.

Our assumption is that an individual's preference about the

*features* of an object will inform their overall preferences regarding that object. Our primary hypothesis is that decomposing a preference relation about a pair of objects into preferences about the *features* of those objects allows for more effective reward modeling. It is worth noting that there are situations where the assumptions introduced thus far don't hold, such as when it is impossible to decompose an example-based reward into its constituent feature-based rewards. Such situations may arise in scenarios where humans do not hold preferences over features of an object independent of the object itself (for example, a human may prefer a football to a basketball, and otherwise not care about individual features of balls such as bounciness, color, size, etc.). In our experiments, we study some deviations from these assumptions and acknowledge that a full analysis of how our method accommodates these more general settings is an important direction for future work.

## 4. Approach: Pragmatic Feature Preferences

To address the reward modeling problem, our primary assumption is that any individual's preference relation about elements of a given domain is tightly coupled with *how* that individual represents elements from that domain. For example, suppose an individual were to prefer a zesty mushroom to a mild mushroom—if zestiness is a primary determining factor in a person's preference about mushrooms, it is likely that zestiness is directly represented by that person, too. This assumption unlocks two key elements.

**Element 1: Feature-level comparisons.** First, we can solicit extra preference information from users as *feature-level comparisons*, rather than solely at the example level. In the mushroom case, this means we can simply ask whether someone prefers spicy to non-spicy foods, rather than ask which of the two mushrooms they prefer. We formalize this below by forming a joint loss term that balances between feature-level comparisons and example-level comparisons.

**Element 2: Pragmatic data augmentation.** Second, we can infer which features are *unimportant* to the user's preference in order to significantly expand the available labeled preference data. For instance, if we ask a user to point out which features are most significant for deciding between two mushrooms and they respond with "spice level" and "color", we suggest it is natural to infer that the other mushroom features are *unimportant* for the given comparison, and consequently we can synthesize new training data where the unimportant feature values are swapped while preserving the object-level preference relation. We provide more concrete details below.

### 4.1. Feature-level queries: Enriched Loss

First, we enrich the preference data collected by not only capturing example-level comparisons but also feature-level comparisons. For example, in the mushroom domain introduced by Sumers et al. (2022), each mushroom is associated with some features such as its size and color. In such a case, we can ask users: (1) Do you prefer mushroom A or mushroom B?, and (2) Do you prefer the size of mushroom A or mushroom B? Do you prefer the color of mushroom A or mushroom B? These fine-grained queries are intended to extract additional information per example pair that can be used to train a reward model.

**RLHF Loss.** More formally, we adopt the standard conventions of RLHF in which the learned reward model, $\hat{R}$, is chosen to minimize the cross-entropy between the reward model's predicted preference labels and the actual labels provided by the user, following the Bradley-Terry model which states humans are noisily rational in identifying the correct example (Bradley, 1976) :

$$\texttt{rlhf-loss}(\hat{R}, \mathcal{D}) = -\sum_{(a_1,a_2,f)\in\mathcal{D}} \left(\mathbb{1}^f_{a_1,a_2} \log \hat{P}(a_1 \succ a_2) \; + \mathbb{1}^f_{a_2,a_1} \log \hat{P}(a_1 \prec a_2)\right). \tag{1}$$

where $\mathbb{1}^f_{a_i,a_j}$ expresses the indicator function on whether $a_i \succ_f a_j$, and $\hat{P}(a_1 \succ a_2)$ is the learned reward function's inferred preference over $(a_1, a_2)$ as defined by the ratio:

$$\hat{P}(a_1 \succ a_2) = \frac{\exp(\hat{R}(a_1))}{\exp(\hat{R}(a_1)) + \exp(\hat{R}(a_2))}. \tag{2}$$

**Feature-Pairwise Loss.** We propose to enrich this loss with a feature-level loss following the same convention. That is, given two actions, $a_1$ and $a_2$, where the features of the first action are all preferred to the second,

$$\texttt{feat-loss}(\hat{R}, \mathcal{D}) = -\sum_{(a_1,a_2,\phi\in\mathcal{D})} \sum_{j=1}^{n} \left(\mathbb{1}^\phi_{a_1^j,a_2^j} \log \hat{P}(a_1^j \succ a_2^j) + \mathbb{1}^\phi_{a_2^j,a_1^j} \log \hat{P}(a_1^j \prec a_2^j)\right). \tag{3}$$

Again, $\mathbb{1}^\phi_{a_1^j,a_2^k}$ denotes the indicator function on whether $a_1^j \succ_\phi a_2^k$, and $\hat{P}(a_1^j \succ a_2^j)$ is the ratio for the learned reward function's output of feature $j$:

$$\hat{P}(a_1^j \succ a_2^j) = \frac{\exp(\hat{R}(a_1^j))}{\exp(\hat{R}(a_1^j)) + \exp(\hat{R}(a_2^j))}. \tag{4}$$

Again, we note that this is where we exploit the linearity assumption—we assume that all reward models of interest $\hat{R}$ can compute a per-feature reward, $\hat{R}(a_i^j)$, for any action $a_i$ and feature $j$.

Our overall loss is then simply a weighted sum of the two,

$$\texttt{loss}(\hat{R}, \mathcal{D}) = (1-\beta)\texttt{rlhf-loss}(\hat{R}, \mathcal{D}) + \beta\texttt{feat-loss}(\hat{R}, \mathcal{D}), \tag{5}$$

with $\beta \in [0, 1]$ a hyperparameter that trades off between the strength of the feature pairwise loss (`feat-loss`) and the example pairwise loss (`rlhf-loss`).

### 4.2. Pragmatic Data Augmentation

The second consequence of asking a user for their feature preferences is that we can also ask them to describe features that are *important* for determining their overall preference. By doing so, we can isolate which features contribute to their preference between the two examples, and thus also infer which features are irrelevant for determining their preference.[1] One immediate benefit of knowing which features are irrelevant to a user in forming a specific preference is that we can expand the available labeled preference data by synthesizing new data points where the preference label remains the same, but the irrelevant features are modified.

Concretely, when we query a user for their choice between $a_1$ and $a_2$, we further ask a user to describe, in language, what features are important for making this decision. We then infer that any feature not mentioned is irrelevant for determining preference, and synthesize a new data point for each possible swap of the irrelevant features' values.

For example, consider a simple case where each action is characterized by only two features, and we query a user on the pair $a_1 = \langle 0, 1\rangle$ and $a_2 = \langle 10, 2\rangle$. Suppose the user prefers $a_1$ to $a_2$, and indicates that the first feature was most important for determining their preference (perhaps this captures the potential poison content of a mushroom). Then, we synthesize a single new data point by keeping the preference label but swapping the irrelevant features. So, we construct $a_1^\circ = \langle 0, 2\rangle$ and $a_2^\circ = \langle 10, 1\rangle$, and assert that $a_1^\circ \succ_f a_2^\circ$. Naturally, with only two features the amount of synthesized data is minimal, but as the number of total features increases, the opportunity for this approach to improve training speed increases as well.

The pseudocode for carrying out this pragmatics-inspired preference augmentation is given in Algorithm 1. The `mask` function is assumed to set the inferred irrelevant features to a special character, "$\emptyset$", and `feat-combos` constructs the set of all combinations of indices of the features set to $\emptyset$. We then use the notation $\vec{j}$ to refer to a vector of indices, and $a_1^\circ[\,\vec{j}\,] = a_2[\,\vec{j}\,]$ as shorthand to refer to assigning each feature with an index contained in $\vec{j}$ to its value in $a_2$. We provide an expanded pseudocode that adds additional clarity in the Appendix as Algorithm 2. It is worth noting that there is an important subtlety to how we approach this data augmentation that depends on how we implement `feat-combos`. In the first method of implementation, we construct all possible combinations of new data where the irrelevant features take on *any possible value*. In the second, we only construct combinations that can result from swapping the feature values that are seen in the specific data point. These two methods each make different assumptions about the underlying pragmatic inference: the first method assumes that *the features inferred to be irrelevant are irrelevant in general*, whereas the second method assumes that *the features are inferred to be irrelevant in this specific context, for these specific values*. We make this second assumption as it is a more cautious approach to data augmentation, but note that exploring the general difference between the two methods could be a useful direction.

**Algorithm 1** Pragmatic Feature Preference Augmentation

INPUT: $\mathcal{D}$ the preference dataset.
OUTPUT: $\mathcal{D}'$ the augmented preference dataset.

**Init:** $\mathcal{D}' = \mathcal{D}$
**for** $(a_1, a_2, f(a_1, a_2), \texttt{mask}) \in \mathcal{D}$ **do**
  $a_1^\circ, a_2^\circ = \texttt{mask}(a_1, a_2, f(a_1, a_2))$
  **for** $\vec{j} \in \texttt{feat-combos}(a_1^\circ, a_2^\circ)$ **do**
    $a_1^\circ[\,\vec{j}\,] = a_2[\,\vec{j}\,]$
    $a_2^\circ[\,\vec{j}\,] = a_1[\,\vec{j}\,]$
    $f^\circ = f(a_1, a_2)$
    $\mathcal{D}' = \mathcal{D}' \cup \{(a_1^\circ, a_2^\circ, f^\circ)\}$
**return** $\mathcal{D}'$

## 5. Experiments

To validate our approach empirically, we conduct experiments in two domains: a vision-based mushroom foraging task and a language-based flight booking task. We begin with experiments that simulate user preference responses based on some known ground truth reward functions to study the learning efficiency of feature preferences in the mushroom task, which is a domain that allows for direct control over the reward functions and their feature densities. Second, we conduct experiments with real language descriptions collected in the flight booking task to explore the benefits of our pragmatic framework with linguistic data.

**Evaluation.** Our goal is to learn accurate reward models that assign high rewards to actions that better satisfy a user's true reward function. To measure the success of learned models, we evaluate the probability of the ground truth best examples (Basu et al., 2018). The higher the probability assigned to the ground truth (i.e. the example that maximizes the true reward), the more accurate the learned reward parameters are. We report results on five independently trained seeds.

**Implementation details.** We implement all reward models as linear networks (single layer, no activations). Each

[1] We refrain from also asking users about which features are irrelevant, both due to the redundancy of the query and the potential for a high number of irrelevant features. However, non-pragmatic feature preference queries can learn from a full set of feature preference labels if available.

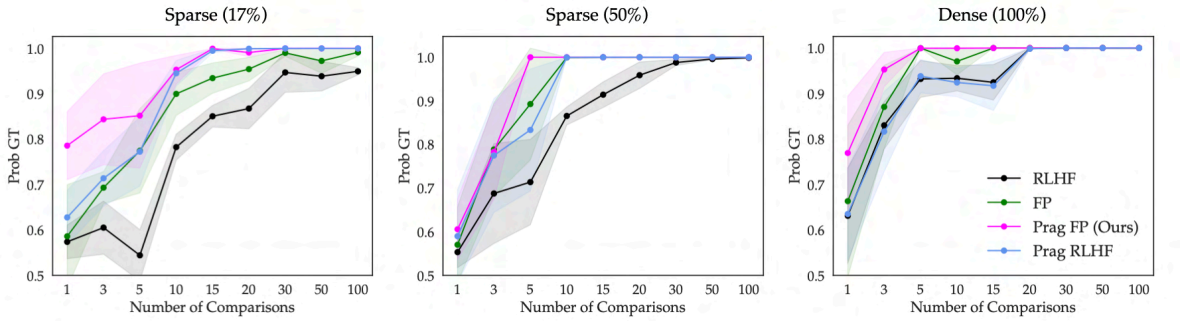


*Figure 2.* Results with simulated preference labels on the mushroom foraging task. **Prag FP** outperforms other methods, converging to a more accurate learned reward given fewer seen examples. This effect is especially prominent as the reward-relevant features become more sparse. Confidence bounds depict standard error across five independent seeds.

feature predictor in the joint model is trained independently without sharing parameters, and their resulting outputs are concatenated and fed through a final layer for reward prediction. We swept possible $\beta$ values and found $0.5$ consistently achieved the best performance.

### 5.1. Understanding reward sparsity's impact on learning efficiency

We begin by testing the hypothesis that given perfect user labels, i.e. an oracle user that answers both example and feature preference queries along with providing reward-relevant features according to the ground truth, feature preference queries will learn more accurate rewards from less examples compared to baselines. In particular, we study how the two distinct elements of our approach—feature preferences and our pragmatic augmentation framework—are impacted by the sparsity of reward features. That is, we explore how the percentage of task-relevant reward features that characterize the ground truth preference relation will impact the quality of the learned reward given a fixed budget of example pairs.

**Task 1: mushroom foraging.** To disentangle these two factors, we make use of a highly controlled task where we can design different types of ground truth preference relations in terms of the types of reward functions used to represent these preferences. Inspired by Sumers et al. (2022), we create a vision-based task where users play the role of a mushroom forager in charge of teaching which mushrooms are preferred. Mushrooms are parameterized by six possible discrete features: *texture, color, shape, height, weight* and *smell* with three possible values for each feature (e.g. *stinky, pleasant,* and *neutral* for smell). We generate reward functions of three different kinds, each characterized by a parameter vector $\theta_{\text{GT}} \in \{-2, -1, 0, 1, 2\}^6$: (1) **dense (100%)** (all six features are reward-relevant), (2) **sparse (17%)** (only a single feature is reward-relevant), and (3) **sparse (50%)** (three features are reward-relevant). For each reward type, we generate two reward functions by randomly sampling the subset of features that are task-relevant, and then randomly sampling the value of each feature.

User queries consist of a task, a reward function, and a randomly sampled comparison (see Figure 1.) For each query, we change the type of labels collected for learning: (1) comparison (**RLHF**, baseline) queries use example-level comparisons only, (2) feature preference (**FP**, ablation) queries use feature-level in addition to example-level comparisons, (3) pragmatic comparison queries (**Prag RLHF**, ablation) use linguistic utterances describing reward-relevant features in addition to example-level comparisons, and (4) pragmatic feature preference (**Prag FP**, our approach) queries combine the pragmatics augmentation framework in conjunction with feature- and example-level preference comparisons. Queries return 1 if A is preferred to B, and $-1$ otherwise.

**Results.** As shown in Figure 2, our results indicate that **Prag FP** converge to a more accurate learned reward with fewer examples required compared to other approaches. Across all three types of reward functions, we find that **FP** as well as **Prag RLHF** contribute meaningfully to learning efficiency, particularly in low-example regimes. When we remove either method, we see performance slightly falter when compared to combining both, highlighting their combined value. **RLHF** performs the worst, indicating that valuable information is lost by modeling the problem solely over example-level comparisons without context.

We further evaluate the quality of the learned reward model based on the sparsity of the reward function generated. This is motivated by the belief that real-world rewards are generally feature sparse (Bajcsy et al., 2018)—that is, users hold preferences based on a few, not many, task-relevant features, As seen across each plot in Figure 2, the magnitude of improvement is especially apparent in the more sparse reward features, confirming the hypothesis that pragmatics-motivated fine-grained feedback is most advantageous when few features impact the final preference relation.

### 5.2. Analyzing the impact of linguistic descriptions

In the previous experiment, we simulated perfect knowledge of reward-relevant features. Now that we have established

the value of including both the pragmatics framework in conjunction with feature-level preferences, we next explore how real linguistic descriptions impact learning.

**Task 2: flight booking.** To study more natural linguistic input, we require a domain where large amounts of real human linguistic descriptions are collected. With this in mind, we use a language-based task from Lin et al. (2022) where users play the role of a flight booker in charge of teaching which flights are preferred. Flights are parameterized by eight possible features: *arrival time before meeting, american, delta, jetblue, southwest, longest stop, number of stops* and *price* (airline features are discrete whereas the rest are continuous). Reward functions are randomly generated, where $\theta_{GT} \in \{-1, -0.5, 0, 0.5, 1\}^8$. Importantly, Lin et al. (2022) collected a human dataset where these reward functions are paired with human linguistic utterances generated by real users describing their preferences over those rewards in language.[2] Some example descriptions include "american or delta prefered. more stops good, but long length of stops bad" and "i want the longest stop and the fewest number of stops". We randomly sample 20 reward functions and their corresponding descriptions from the full dataset.

To make use of linguistic descriptions, we must convert unstructured linguistic utterances into structured feature maps specifying reward-relevant features. In the simplest case, we could require the user to specify the reward-relevant features from the full list, but doing so requires additional human data collection and is subject to misspecification (Peng et al., 2023). Therefore, we deployed a language model (LM) to parse descriptions into structured feature representations. Specifically, we prompt GPT-4 (Achiam et al., 2023) with the linguistic description and full feature space to generate a feature map $\in \{0, 1\}^8$ specifying reward-relevant features.

**Results.** As shown in Figure 3, even with messy real linguistic data, **Prag FP** outperforms traditional **RLHF**, converging to a more accurate reward with fewer examples required. We report results evaluated over 20 randomly sampled reward functions, each with five independent seeds.

We note that we did not attempt to perform high robustness prompt engineering (Chen et al., 2023), nor explicitly study question-answering mechanisms to elicit more accurate linguistic utterances from human users, although improvements across both axes would certainly further improve the accuracy of the pragmatic feature preference modeling.

[2]The original dataset can be found at github.com/jlin816/rewards-from-language. Note, Lin et al. (2022) presented a set of three options to users and asked for a description of the option that was most optimal under the reward function, which we converted into two pairwise comparisons. The linguistic data was also collected over iterative feedback rounds to study the effect of recursive reasoning, which we disregard.

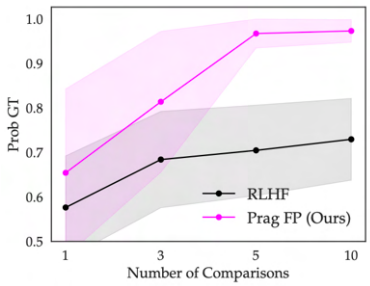


*Figure 3.* Results with real user descriptions on the flight booking task across 20 randomly sampled reward functions. Confidence bounds depict standard error across five independent seeds.

## 6. User Study

In Section 5, we evaluated our framework with simulated human preference labels. We now expand on these results with a behavioral study conducted with real users on the same mushroom foraging task. We are interested in addressing two questions. First, do real user labels for **Prag FP** queries validate our simulated results? Second, do users exert significantly more effort when giving these queries compared to **RLHF** pairwise comparisons?

**Study Overview.** We conducted a between-subjects user study where participants were asked to play the role of a mushroom forager tasked with selecting tasty mushrooms. Users were trained to read reward functions and calculate tastiness scores (rewards), and then given mushroom pairs and asked to give preferences about these pairs according to provided reward functions. We asked participants to answer three types of preference queries: **RLHF** queries (preferences over mushroom examples), **FP** queries (preferences over mushroom features in addition to examples), and **Prag FP** queries (language descriptions in addition to preferences over examples and features). Each user answered 30 queries for the same six reward functions as in Section 5 (five per reward). We used the responses from each query type to train a reward model. We additionally asked participants brief survey questions (Table 1) regarding their perceived effort, performance, and frustration after queries. Responses were assessed on a Likert scale (Likert, 1932), with 1 being the lowest ("strongly disagree") and 7 the highest ("strongly agree"). We also recorded total time spent on the task.

**Participants.** We recruited 36 participants, 12 for each query type, from Prolific, an online crowdsourcing site. Participants were screened according to the following characteristics: hold above a 95% approval rating, speak English as a primary language, and reside primarily in the United Kingdom or United States. We paid participants at a rate of $16 per hour and rejected responses that were low-effort (e.g. left responses blank, repeated the same answer for all questions, etc.). Our study passed institutional IRB review.

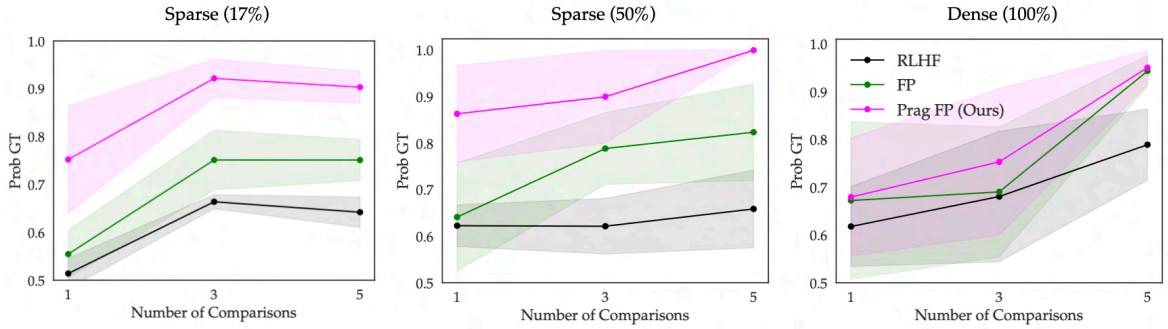


*Figure 4.* Results with real user responses on the mushroom foraging task. These results corroborate the simulated results from Figure 2. Confidence bounds depict standard error across five independent seeds.

### 6.1. Learning with real human responses.

We begin by analyzing the impact of learning reward models from real human preferences, i.e. labels that may be noisily generated from users. We conduct the same training protocol as in Section 5 using randomly sampled user responses to train reward models for each reward function. For each reward, we report five independent seeds.

**Results.** As shown in Figure 4, **Prag FP** outperforms baselines, especially as the reward is more sparse. These results corroborate the simulated results from above and provide meaningful evidence that real users are able to generate both linguistic descriptions and preference responses that can be used to accurately train reward models.

| **Survey Questions** |
|---|
| Q1. Choosing the best mushroom was challenging. |
| Q2. I could accurately communicate the best mushroom. |
| Q3. Describing my preferences was relatively easy. |
| Q4. I was frustrated with providing labels. |

*Table 1.* Post-user study survey questions. User responses are assessed on a 1 to 7 Likert scale (with 7 being "strongly agree").

### 6.2. Understanding impact on user effort.

To ensure that **Prag FP** does not significantly negatively impact the user data collection process, we assessed the survey responses collected from participants at the conclusion of the study. Questions are shown in Table 1. We analyzed Likert ratings using one-tailed independent $t$-tests, where both **FP** and **Prag FP** queries are compared to **RLHF** queries.

**Results.** First, there was no significant difference in whether participants felt they could communicate preferences accurately (Q2) (no significant difference for either query type ($t(11) = -1.81, 0.29, p = 0.08, 0.77$)). Participants who answered **Prag FP** queries *did not* find it more challenging to describe their preferences (Q2, $t(11) = 0.16, p = 0.43$), while participants who answered **FP** queries *did* find it more challenging ($t(11) = 2.31, p = 0.02$)). This supports the hypothesis that allowing users to provide descriptions of important features pragmatically is more natural than providing feedback on all features. Importantly, providing **Prag FP** queries *did not* cause users to experience more frustration with providing labels (Q4, $t(11) = -0.87, p = 0.19$) compared to **FP** ($t(11) = -3.07, p = 0.01$). Lastly, users who provided **Prag FP** queries *did not* take significantly more time on the task ($t(11) = -0.24, p = 0.41$) compared to **FP** queries, who did take longer ($t(11) = -1.86, p = 0.03$).

## 7. Discussion

We study a new form of user query, pragmatic feature preferences, for use in learning reward models from fine-grained human input. Our method relies on two key elements: first, that human preferences at the feature level are valuable for learning accurate reward functions from fewer provided examples, and second, that what humans choose to describe in language tells us important information regarding which features are reward-relevant in their preference relation.

Conceptually, our model builds on a rich history of work in pragmatic reasoning by explicitly modeling humans as teaching when giving feedback. While we studied our learning in an entirely offline setting, there are exciting directions for incorporating recursive reasoning in developing models that learn to ask the right questions for further clarifying inference of feature preferences in uncertain settings (Li et al., 2023). Moreover, we made the assumption that given language descriptions, we can ground the identified features from those utterances to the correct features in the state representation, an assumption that is challenging in practice due to ambiguity in grounding ambiguous descriptions to an agent's perceptual state (Harnad, 1990). Lastly, the data augmentation aspect of our pragmatic framework relied on the ability to easily swap non-reward-relevant features in the comparison examples, which may be challenging with text-based models (e.g. swapping the *toxicity* in outputs). Nonetheless, we are excited about the promise of incorporating pragmatics-inspired models of human abstract reasoning to learning more user-aligned reward models.

## Impact Statement

This paper presents work on how to better learn individualized reward functions from human input. While this work is intended to help learn more accurate user objectives, we did not discuss possible misuse associated with malicious actors teaching models dangerous features. We will leave such discussion open to the broader socio-technical community.

## Acknowledgements

We thank Jacob Andreas and Alexis Ross for their formative insights and discussions, as well as reviewing earlier drafts of the paper. We are grateful to Ted Sumers and Jessy Lin for helpful advice on the mushroom foraging and flight tasks, respectively. We additionally thank Mark Rowland and the rest of the Google DeepMind team who reviewed an early draft of the paper. Andi Peng is supported by a NSF Graduate Research Fellowship and Open Philanthropy.

## References


Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. *Proceedings of the International Conference on Machine Learning*, 2004.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

Bajcsy, A., Losey, D. P., O'Malley, M. K., and Dragan, A. D. Learning from physical human corrections, one feature at a time. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 141–149, 2018.

Basu, C., Singhal, M., and Dragan, A. D. 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*, pp. 132–140, 2018.

Bisk, Y., Yuret, D., and Marcu, D. Natural language communication with robots. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 751–761, 2016.

Bobu, A., Peng, A., Agrawal, P., Shah, J., and Dragan, A. D. Aligning robot and human representations. *arXiv preprint arXiv:2302.01928*, 2023.

Bradley, R. A. Science, statistics, and paired comparisons. *Biometrics*, 32(2):213–239, 1976.

Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

Chen, B., Zhang, Z., Langrené, N., and Zhu, S. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*, 2023.

Christian, B. *The alignment problem: How can machines learn human values?* Atlantic Books, 2021.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Fisac, J. F., Gates, M. A., Hamrick, J. B., Liu, C., Hadfield-Menell, D., Palaniappan, M., Malik, D., Sastry, S. S., Griffiths, T. L., and Dragan, A. D. Pragmatic-pedagogic value alignment. In *Robotics Research: The 18th International Symposium ISRR*, pp. 49–57. Springer, 2020.

Goyal, A., Islam, R., Strouse, D., Ahmed, Z., Botvinick, M., Larochelle, H., Bengio, Y., and Levine, S. InfoBot: Transfer and exploration via the information bottleneck. 2019.

Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., and Thomaz, A. L. Policy shaping: Integrating human feedback with reinforcement learning. *Advances in neural information processing systems*, 26, 2013.

Hadfield-Menell, D., Milli, S., Abbeel, P., Russell, S. J., and Dragan, A. Inverse reward design. *Advances in neural information processing systems*, 30, 2017.

Harnad, S. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990.

Ho, M. K., Littman, M., MacGlashan, J., Cushman, F., and Austerweil, J. L. Showing versus doing: Teaching by demonstration. *Advances in neural information processing systems*, 29, 2016.

Ho, M. K., Abel, D., Griffiths, T. L., and Littman, M. L. The value of abstraction. *Current Opinion in Behavioral Sciences*, 2019.

Ho, M. K., Abel, D., Correa, C. G., Littman, M. L., Cohen, J. D., and Griffiths, T. L. People construct simplified mental representations to plan. *Nature*, 606(7912):129–136, 2022.

Hüllermeier, E., Fürnkranz, J., Cheng, W., and Brinker, K. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008.

Jain, A., Sharma, S., Joachims, T., and Saxena, A. Learning preferences for manipulation tasks from online coactive feedback. *The International Journal of Robotics Research*, 34(10):1296–1313, 2015.

Kaufmann, T., Weng, P., Bengs, V., and Hüllermeier, E. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 2023.

Langford, J. and Zhang, T. The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in Neural Information Processing Systems*, 2007.

Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

Li, B. Z., Tamkin, A., Goodman, N., and Andreas, J. Eliciting human preferences with language models. *arXiv preprint arXiv:2310.11589*, 2023.

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

Likert, R. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

Lin, J., Fried, D., Klein, D., and Dragan, A. Inferring rewards from language in context. *ACL*, 2022.

Ng, A. Y. and Russell, S. Algorithms for inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2000.

Peng, A., Netanyahu, A., Ho, M. K., Shu, T., Bobu, A., Shah, J., and Agrawal, P. Diagnosis, feedback, adaptation: A human-in-the-loop framework for test-time policy adaptation. In *International Conference on Machine Learning*, pp. 27630–27641. PMLR, 2023.

Peng, A., Sucholutsky, I., Li, B., Sumers, T., Griffiths, T., Andreas, J., and Shah, J. Learning with language-guided state abstractions. In *International Conference on Learning Representations*, 2024.

Sumers, T., Hawkins, R., Ho, M. K., Griffiths, T., and Hadfield-Menell, D. How to talk so ai will learn: Instructions, descriptions, and autonomy. *Advances in Neural Information Processing Systems*, 35:34762–34775, 2022.

Sutton, R. S. Introduction: The challenge of reinforcement learning. In *Reinforcement Learning*, pp. 1–3. Springer, 1992.

Wu, Z., Hu, Y., Shi, W., Dziri, N., Suhr, A., Ammanabrolu, P., Smith, N. A., Ostendorf, M., and Hajishirzi, H. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*, 2023.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2008.

## A. User Study

Figure 5 depicts one of the six reward functions presented in the user study. Users were trained to read reward functions in the familiarization stage of the study and then presented six unique reward functions to reference for queries.

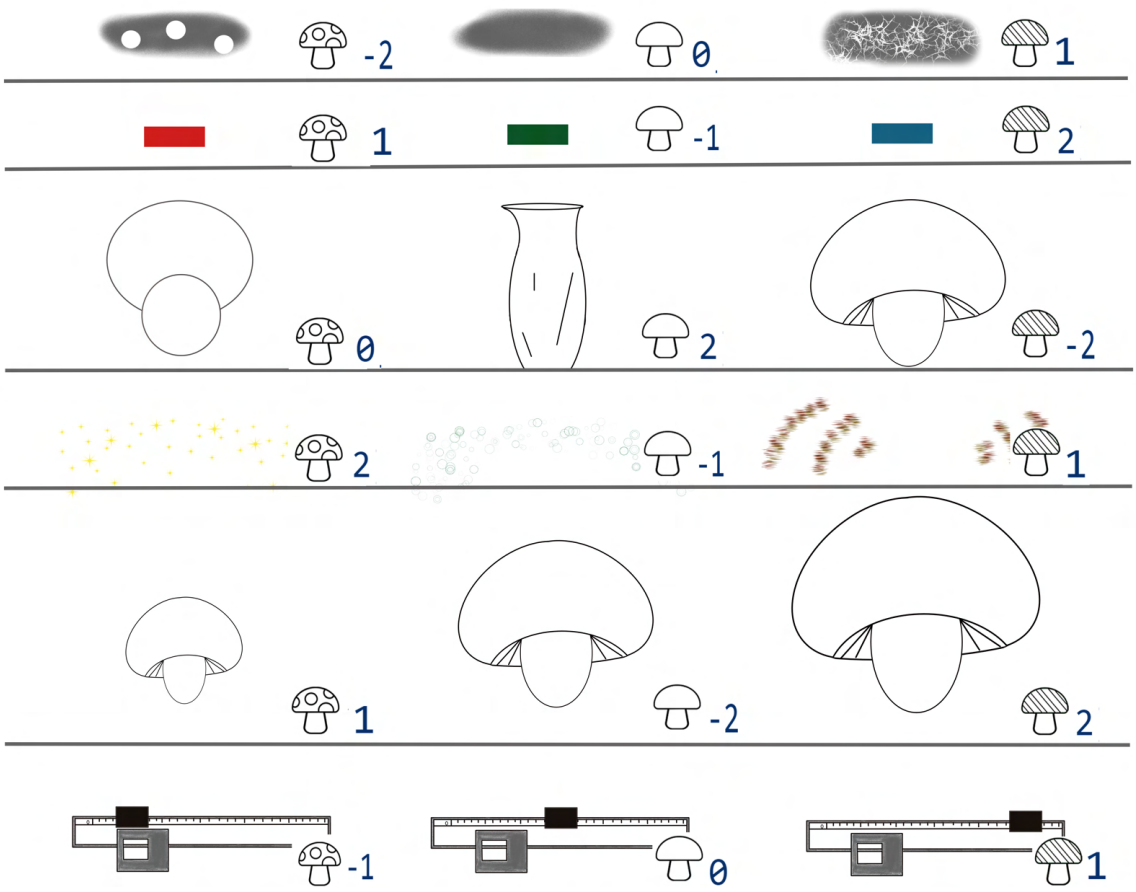


*Figure 5.* Example reward function provided in the user study.

Figure 5 illustrate four possible mushrooms.

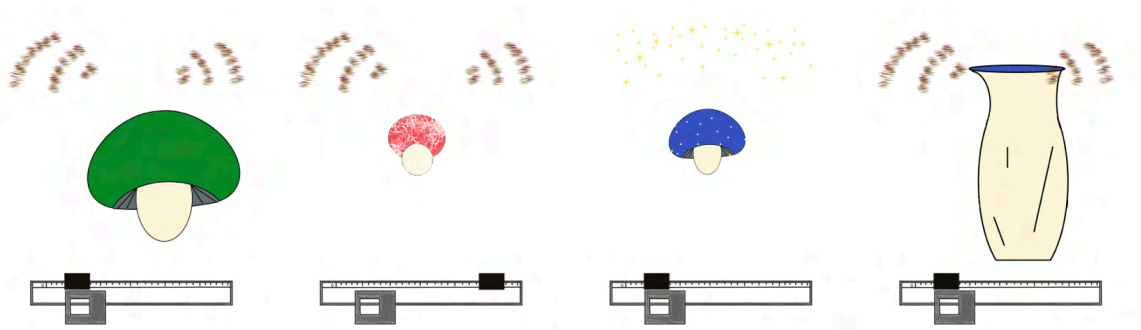

*Figure 6.* Four example mushrooms used in the user study.

## B. Expanded Pseudocode

To remove any ambiguity about Algorithm 1, we here present a more detailed version of the pseudocode that makes the specifics of the algorithm more clear. We present this expanded version as Algorithm 2. Here, $\texttt{feat-combos}(a_1^\circ, a_2^\circ)$ constructs a set of ordered pairs of feature indices. This set, active-feature-pairs, contains pairs of features $(j_x, j_y)$ whose values should be swapped. The first element ($j_x$) is an index into $a_1$, and the second element ($j_y$) is an index into $a_2$. Lines 8 and 9 perform the swapping on copies of $a_1$ and $a_2$, while line 10 computes the new preference relation.

**Algorithm 2** Expanded version of Pragmatic Feature Preference Augmentation

INPUT: $\mathcal{D}$ the preference dataset.
OUTPUT: $\mathcal{D}'$ the augmented preference dataset.

1: **Init:** $\mathcal{D}' = \mathcal{D}$
2: **for** $(a_1, a_2, f(a_1, a_2), \texttt{mask}) \in \mathcal{D}$ **do**
3: $\quad a_1^\circ, a_2^\circ = \texttt{mask}(a_1, a_2, f(a_1, a_2))$
4: $\quad$ **for** active-feature-pairs in $\texttt{feat-combos}(a_1^\circ, a_2^\circ)$ **do**
5: $\quad\quad a_1' = \texttt{copy}(a_1)$
6: $\quad\quad a_2' = \texttt{copy}(a_2)$
7: $\quad\quad$ **for** $j_x, j_y$ in active-feature-pairs **do**
8: $\quad\quad\quad a_1'[j_x] = a_2[j_y]$
9: $\quad\quad\quad a_2'[j_y] = a_1[j_x]$
10: $\quad\quad f' = f(a_1, a_2)$
11: $\quad\quad \mathcal{D}' = \mathcal{D}' \cup \{(a_1', a_2', f')\}$
**return** $\mathcal{D}'$