Adaptive Querying for Reward Learning from Human Feedback

Yashwanthi Anand, Nnamdi Nwagwu, Kevin Sabbe, Naomi T. Fitter and Sandhya Saisubramanian

Oregon State University, Corvallis, USA.

Correspondence*:
Yashwanthi Anand
anandy@oregonstate.edu

ABSTRACT

Learning from human feedback is a popular approach to train robots to adapt to user preferences and improve safety. Existing approaches typically consider a single querying (interaction) format when seeking human feedback and do not leverage multiple modes of user interaction with a robot. We examine how to learn a penalty function associated with unsafe behaviors using *multiple* forms of human feedback, by optimizing both the *query state* and *feedback format*. Our proposed *adaptive feedback selection* is an iterative, two-phase approach which first selects critical states for querying, and then uses information gain to select a feedback format for querying across the sampled critical states. The feedback format selection also accounts for the cost and probability of receiving feedback in a certain format. Our experiments in simulation demonstrate the sample efficiency of our approach in learning to avoid undesirable behaviors. The results of our user study with a physical robot highlight the practicality and effectiveness of adaptive feedback selection in seeking informative, user-aligned feedback that accelerate learning. Experiment videos, code and appendices are found on our website: https://tinyurl.com/AFS-learning

Keywords: Learning from human feedback, Information gain, Learning from multiple formats, Interactive Imitation Learning

1 INTRODUCTION

A key factor affecting an autonomous agent's behavior is its reward function. Due to the complexity of real-world environments and the practical challenges in reward design, agents often operate with incomplete reward functions corresponding to underspecified objectives, which can lead to unintended and undesirable behaviors such as negative side effects (NSEs) (Amodei et al., 2016; Saisubramanian et al., 2021a; Srivastava et al., 2023). For example, a robot optimizing the distance to transport an object to the goal, may damage items along the way if its reward function does not model the undesirability of colliding into other objects in the way (Figure 1).

Several prior works have examined learning from various forms of human feedback to improve robot performance, including avoiding side effects (Cui and Niekum, 2018; Cui et al., 2021b; Lakkaraju et al., 2017; Ng et al., 2000; Saran et al., 2021; Zhang et al., 2020). In many real-world settings, the human can provide feedback in many forms, ranging from binary signals indicating action approval to correcting robot actions, each varying in the granularity of information revealed to the robot and the human effort required to provide it. To efficiently balance the *trade-off* between seeking feedback in a format that accelerates

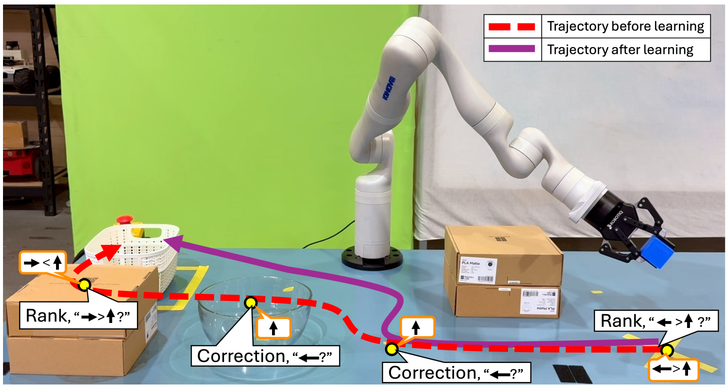


Figure 1. An illustration of adaptive feedback selection. The robot arm learns to move the blue object to the white bin, without colliding with other objects in the way, by querying the human in different format across the state space.

robot learning and reducing human effort involved, it is beneficial to seek detailed feedback sparingly in certain states and complement it with feedback types that require less human effort in other states. Such an approach could also reduce the sampling biases associated with learning from any one format, thereby improving learning performance (Saisubramanian et al., 2022). In fact, a recent study indicates that users are generally willing to engage with the robot in more than one feedback format (Saisubramanian et al., 2021b). Existing approaches, however, typically utilize a single feedback format throughout the learning process and *do not support* gathering feedback in different formats in different regions of the state space (Cui et al., 2021a; Settles, 1995).

How can a robot identify *when to query* and in *what format*, while accounting for the cost and availability of different forms of feedback? We present a framework for *adaptive feedback selection* (AFS) that enables a robot to seek feedback in multiple formats in its learning phase, such that its information gain is maximized. In the interest of clarity, the rest of this paper grounds the discussion of AFS as an approach for robots to learn to avoid negative side effects (NSEs) of their actions. The NSEs refer to unintended and undesirable outcomes that arise as the agent performs its assigned task. In object delivery example in Figure 1, the robot may inadvertently collide with other objects on the table, producing NSEs. Focusing on NSEs provides a well-defined and measurable setting–quantified by the number of NSE occurrences–to evaluate how AFS improves an agent's learning efficiency and safety. However, note that AFS is a general technique that can be applied broadly to learn about various forms of undesirable behavior.

In each querying cycle, AFS selects a feedback format that maximizes the robot's information gain, given its current knowledge of NSEs. The information gain of a feedback format is measured as the Kullback–Leibler (KL) divergence between the probability distributions over the NSE labels. Specifically, for both the true NSE labels revealed through human feedback and the robot's current knowledge of NSEs labels learned from the feedback, we calculate the frequency of each NSE category and normalize it to form a probability mass function (PMF). The KL divergence is then computed between these resulting PMFs, quantifying the divergence in the NSE label distributions (detailed in Sec. 4). When collecting feedback in every state is infeasible, the robot must prioritize querying in *critical states*—states where human feedback is crucial for learning an association of state features and NSEs, i.e., a predictive model of NSE severity. Querying in critical states maximizes information gain about NSEs, compared to other states. Prior works, however, query for feedback in states randomly sampled or along the shortest path to the goal, which may not result in a faithful NSE model (Saisubramanian et al., 2021a; Zhang et al., 2020).

Minimizing NSEs using AFS involves four iterative steps (Figure 4): (1) states are partitioned into clusters, with a cluster weight proportional to the number of NSEs discovered in it; (2) a critical states set is formed by sampling from each cluster based on its weight; (3) a feedback format that maximizes the information gain in critical states is identified, while accounting for the cost and uncertainty in receiving a feedback, using the human feedback preference model; and (4) cluster weights and information gain are updated, and a new set of critical states are sampled to learn about NSEs, until the querying budget expires. The learned NSE information is mapped to a penalty function and augmented to the robot's model to compute an NSE-minimizing policy to complete its task.

We evaluate AFS in both simulation and using a user study where participants interact with a robot arm. First, we evaluate the approach in three simulated proof-of-concept settings with simulated human feedback. Second, we conduct a pilot study where 12 human participants interact with and provide feedback to the agent in a simulated gridworld domain. Finally, we evaluate using a Kinova Gen3 7DoF arm and 30 human participants. Besides the performance and sample efficiency, our experiments also provide insights into how the querying process can influence user trust. Together, these complementary studies demonstrate both the practicality and effectiveness of AFS.

2 BACKGROUND AND RELATED WORK

2.1 Markov Decision Processes (MDPs)

The MDPs are a popular framework to model sequential decision making problems. An MDP is defined by the tuple $M = \langle S, A, T, R, \gamma \rangle$, where S is the set of states, A is the set of actions, $T(s,a,s')$ is the probability of reaching state $s' \in S$ after taking an action $a \in A$ from a state $s \in S$ and $R(s,a)$ is the reward for taking action a in state s. An optimal deterministic policy $\pi^* : S \rightarrow A$ is one that maximizes the expected reward. When the objective or reward function is incomplete, even an optimal policy can produce unsafe behaviors such as side effects. **Negative Side Effects** (NSEs) are immediate, undesired, unmodeled effects of an agent's actions on the environment (Krakovna et al., 2018; Saisubramanian and Zilberstein, 2021; Srivastava et al., 2023). We focus on NSEs arising due to incomplete reward function (Saisubramanian et al., 2021a), which we mitigate by learning a penalty function using human feedback.

2.2 Learning from Human Feedback

Learning from human feedback is a popular approach to train agents when reward functions are unavailable or incomplete (Abbeel and Ng, 2004; Ng et al., 2000; Ross et al., 2011; Najar and Chetouani, 2021), including to improve safety (Brown et al., 2020b, 2018; Hadfield Menell et al., 2017; Ramakrishnan et al., 2020; Zhang et al., 2020; Saisubramanian et al., 2021a). Feedback can take various forms such as *demonstrations* (Ramachandran and Amir, 2007; Brown and Niekum, 2018), *corrections* (Losey and O'Malley, 2018; Bobu et al., 2021; Cui et al., 2023), *critiques* (Cui and Niekum, 2018; Saisubramanian et al., 2021a), *ranking* trajectories (Brown et al., 2020a), or may be *implicit* in the form of facial expressions and gestures (Cui et al., 2021b; Xu et al., 2020; Strokina et al., 2022; Candon et al., 2023).

While the existing approaches for learning from feedback have shown success, they typically assume that a single feedback type is used to teach the agent. This assumption limits learning efficiency and adaptability. Some efforts combine demonstrations with preferences (Bıyık et al., 2022; Ibarz et al., 2018), showing that utilizing more than one format accelerates learning. Extending this idea, recent works integrate richer modalities such as language and vision with demonstrations. Yang et al. (2024) learn reward function from comparative language feedback, while Sontakke et al. (2023) show that a single demonstration or natural language description can help define a proxy reward when used along with a vision-language models (VLM) that is pretrained on a large amount of out-of-domain video demonstrations and language pairs. Kim et al. (2023) use multimodal embeddings of visual observations and natural language descriptions to compute alignment-based rewards. A recent study even emphasizes that combining multiple feedback modalities can further enhance learning outcomes (Beierling et al., 2025). Together, these works highlight that combining complementary feedback formats help advance reward learning beyond using a fixed feedback format. In contrast, our approach uses multiple forms of human feedback for learning.

Other approaches that learn from human feedback focus on modeling variations in human behavior. Huang et al. (2024) model the heterogeneous behaviors of human, capturing differences in feedback frequency, delay, strictness, and bias to improve the robustness during the learning process, as optimal behaviors vary across users. Along the same line, the reward learning approach proposed by Ghosal et al. (2023), selects a single feedback format based on the user ability to provide feedback in that format, resulting in an interaction that is tailored to a user's skill level. Collectively, these works reveal a shift towards adaptive and user-aware querying mechanisms that improves reward inference and learning efficiency, motivating our approach to dynamically select both when to query and in what feedback format.

3 PROBLEM FORMULATION

Setting: Consider a robot operating in a discrete environment modeled as a Markov Decision Process (MDP), using its acquired model $M = \langle S, A, T, R_T \rangle$. The robot optimizes the completion of its assigned task, which is its primary objective described by reward R_T. A *primary policy*, π^M, is an optimal policy for the robot's primary objective.

Assumption 1. Similar to (Saisubramanian et al., 2021a), we assume that the agent's model M has all the necessary information for the robot to successfully complete its assigned task but lacks other superfluous details that are unrelated to the task.

Since the model is incomplete in ways unrelated to the primary objective, executing the primary policy produces negative side effects (NSEs) that are difficult to identify at design time. Following (Saisubramanian et al., 2021a), we define NSEs as immediate, undesired, unmodeled effects of a robot's actions on the environment. We focus on settings where the robot has *no prior knowledge* about the NSEs of its actions or the underlying true NSE penalty function R_N. It learns to avoid NSEs by learning a penalty function $\hat{R}_N$ from human feedback that is consistent with R_N.

We target settings where the human can provide feedback in multiple ways and the robot can seek feedback in a *specific* format such as approval or corrections. This represents a significant shift from traditional active learning methods, which typically gather feedback only in a single format (Ramakrishnan et al., 2020; Saisubramanian et al., 2021a; Saran et al., 2021). Using the learned $\hat{R}_N$, the robot computes an NSE-minimizing policy to complete its task by optimizing: $R(s,a) = \theta_1 R_T(s,a) + \theta_2 \hat{R}_N(s,a)$, where θ_1 and θ_2 are fixed, tunable weights denoting priority over objectives.

Running Example: We illustrate the problem using a simple object delivery task using a Kinova Gen3 7DoF arm shown in Figure 1. The robot optimizes delivering the blue block to the white bin, by taking the shortest path. However, passing through states with a cardboard box or a glass bowl constitutes an NSE. Since the robot has no prior knowledge about NSEs of its actions, it may inadvertently navigate through these states causing NSEs.

Human's Feedback Preference Model: The feedback format selection must account for the cost and human preferences in providing feedback in a certain format. The user's *feedback preference model* is denoted by $D = \langle \mathcal{F}, \psi, C \rangle$ where,

- $\mathcal{F}$ is a predefined set of feedback formats the human can provide, such as demonstrations and corrections;
- $\psi : \mathcal{F} \rightarrow [0, 1]$ is the probability of receiving feedback in a format f, denoted as $\psi(f)$; and
- $C : \mathcal{F} \rightarrow \mathbb{R}$ is a cost function that assigns a cost to each feedback format f, representing the human's time or cognitive effort required to provide that feedback.

This work assumes the robot has access to the user's feedback preference model D—either handcrafted by an expert or learned from user interactions prior to robot querying, as in our user study experiments. Abstracting user feedback preferences into probabilities and costs enables generalizing the preferences across similar tasks. We take the pragmatic stance that ψ is independent of time and state, denoting the user's preference about a format, such as not preferring formats that require constant supervision of robot performance. While this can be relaxed and the approach can be extended to account for state-dependent preferences, obtaining an accurate state-dependent ψ could be challenging in practice.

Assumption 2. Human feedback is immediate and accurate, when available.

Below, we describe the various feedback formats considered in this paper, and how the data from these formats are mapped to NSE severity labels.

3.1 Feedback Formats Studied

The agent learns an association between state-action pairs and NSE severity, based on the human feedback provided in response to agent queries. The NSE categories we consider in this work are $\{\text{No NSE}, \text{Mild NSE}, \text{Severe NSE}\}$. We focus on the following commonly used feedback types, each differing in the level of information conveyed to the agent and the human effort required to provide them.

Approval (App): The robot randomly selects N state-action pairs from all possible actions in critical states and queries the human for approval or disapproval. Approved actions are labeled as acceptable, while disapproved actions are labeled as unacceptable.

Annotated Approval (Ann. App): An extension of Approval, where the human specifies the *NSE severity* (or category) for each disapproved action in the critical states.

Corrections (Corr): The robot performs a trajectory of its primary policy in the critical states, under human supervision. If the robot's action is unacceptable, then the human intervenes with an acceptable action in these states. If all actions in a state lead to NSE, the human specifies an action with the least NSE. When interrupted, the robot assumes all actions except the correction are unacceptable in that state.

Annotated Corrections (Ann. Corr): An extension of Corrections, where the human specifies the severity of NSEs caused by the robot's unacceptable action in critical states.

Rank: The robot randomly selects N ranking queries of the form $\langle \textit{state}, \textit{action 1}, \textit{action 2} \rangle$, by sampling two actions for each critical state. The human selects the safer action among the two options. If both are safe or unsafe, one of them is selected at random. The selected action is marked as acceptable and the other is treated as unacceptable.

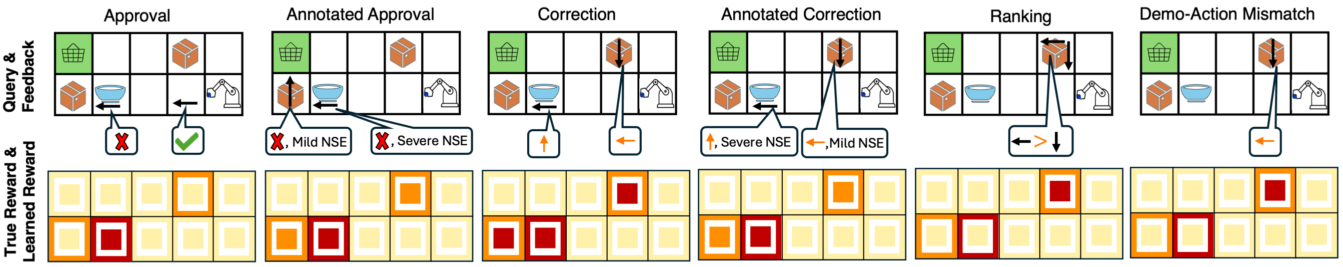


Figure 2. Visualization of reward learned using different feedback types. **(Row 1)** Black arrows indicate queries, and feedback is in speech bubbles. **(Row 2)** 🟥 denotes high, 🟧 mild, and 🟨 zero penalty. Outer box is the true reward, and inner box shows the learned reward. Mismatches between the outer and inner box colors indicate incorrect learned model.

Demo-Action Mismatch (DAM): The human demonstrates a safe action in each critical state, which the robot compares with its policy. All mismatched robot's actions are labeled as unacceptable. Matched actions are labeled as acceptable.

Mapping feedback data to NSE severity labels: We use l_a, l_m, and l_h to denote labels corresponding to no, mild and severe NSEs, respectively. An acceptable action in a state is mapped to l_a, i.e., $(s, a) \rightarrow l_a$, while an unacceptable action is mapped to l_h. When the severity of NSEs for unacceptable actions is known, actions producing mild NSEs are mapped to l_m and those producing severe NSEs to l_h. Mapping feedback to this common label set provides a consistent representation of NSE severity across diverse feedback types. The granularity of information and the sampling biases of the different feedback types affect the learned reward. Figure 2 illustrates this with the learned NSE penalty for the running example of moving an object to the bin (Fig. 1), motivating the need for an adaptive approach that can learn from more than one feedback format. In the running example, the robot arm colliding with cardboard boxes is a mild NSE, and colliding with a glass bowl is a severe NSE.

4 ADAPTIVE FEEDBACK SELECTION

Given an agent's decision making model M and the human's feedback preference model D, AFS enables the agent to query for feedback in critical states in a format that maximizes its information gain. We first formalize the NSE model learning process and then describe in detail how AFS selects critical states and the query format.

Formalizing NSE Model Learning: Let $p^* : S \times A \rightarrow \{l_a, l_m, l_h\}$ denote the *true* NSE severity label for each state-action pair, which is unknown to the agent but known to the human. The label l_a corresponds to *no NSE*, l_m denotes *mild NSE*, l_h denote the label for *severe NSE*. Let p be a sampled approximation of p^* ($p \sim p^*$), denoting the dataset of NSE labels collected via human feedback in response to the (s, a) pairs queried. That is, p^t denotes the data collected from human feedback until iteration t, where $p^t(s, a)$ represents the categorical NSE severity label assigned to the state-action pair (s, a). Let $q : S \times A \rightarrow \{l_a, l_m, l_h\}$ denote the labels predicted by the learned NSE model—learned using a supervised classifier with p as the training data. In this paper, we use a Random Forest (RF) classifier, though any classifier can be used in practice. Hyperparameters are optimized through randomized search with three-fold cross validation, and the configuration yielding the lowest mean-squared error is selected for training.

Figure 3 shows an example of p and q for the object delivery task. We encode NSE categories as $\{0, 1, 2\}$ corresponding to $\{$ no NSE, mild NSE, severe NSE$\}$ respectively. Each state has four possible actions

$A = \{a_1, a_2, a_3, a_4\}$, and the vector $p(s) = \{\cdot, \cdot, \cdot, \cdot\}$ (and similarly $q(s)$) encodes the categorical NSE labels for $(s, a_1), (s, a_2), (s, a_3), (s, a_4)$ in that order. Since the human's categorization of NSE is initially unknown, $p(s)$ is sampled from a uniform prior over the labels, and $q(s)$ is initialized to $[0, 0, 0, 0]$ (all actions are assumed to be safe) across all states.

At $t-1$, p^{t-1} reflects a single labeled state from the feedback received, while q^{t-1} reflects NSE label for the state after learning from p^{t-1}. For example, in iteration $t-1$, an action a_3 in state s is randomly selected for querying using the *Annotated Approval* feedback format. The human labels it as mild NSE, so $p^{t-1}(s, a_3) = 1$, and consequently $p^{t-1}(s) = [0, 0, 1, 0]$. After training on p^{t-1}, the classifier may sometimes incorrectly predict $q^{t-1}(s) = [0, 0, 0, 0]$, especially in early iterations when there is less data. At the next iteration t, the agent queries in a similar state using the *Approval* format, where the action a_1 is randomly selected. Because the NSE severity level (i.e., mild/severe) cannot be indicated through the Approval format, p^t is updated as $p^t(s) = [2, 0, 0, 0]$, and training now yields a prediction $q^t(s) = [2, 0, 1, 0]$ (i.e. the NSE model predicts severe NSE outcome on a_1 and a mild NSE outcome on a_3). This illustrates that q may initially disagree with p, but as feedback accumulates on related states, the generalization of q across actions begins to align with p.

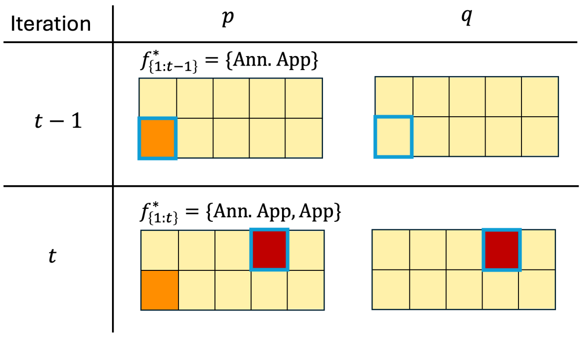

Iteration
p
q
t − 1
$f^*_{\{1:t-1\}}$ = {Ann. App}
t
$f^*_{\{1:t\}}$ = {Ann. App, App}

Figure 3. Illustration of p (accumulated feedback) and q (generalized NSE labels) for the object delivery task. $f^*_{1:t-1}$ indicates the feedback formats selected until iteration $t-1$. □ indicates no NSE; 🟧 indicates mild NSE; 🟥 indicates severe NSE. Queried states in each iteration is highlighted in blue.

Each predicted label is then mapped to a penalty value to form the learned penalty function, $\hat{R}_N$, with penalties for l_a, l_m and l_h set to $0, -5$ and -10 respectively, in our experiments. This penalty function is integrated into the agent's reward model to compute an updated policy that minimizes NSEs while completing the primary task.

In this learning setup, minimizing NSEs using AFS involves four iterative steps (Figure 4). In each learning iteration, AFS identifies (1) which states are most critical for querying (Sec. 4.1), and (2) which feedback format maximizes the expected information gain at the critical states, while accounting for user feedback preferences and effort involved (Sec. 4.2). The information gain associated with a feedback quantifies the effect of a feedback in improving the agent's understanding of the underlying reward function, and is measured using Kullback-Leibler (KL) Divergence (Ghosal et al., 2023; Tien et al., 2023). At the end of each iteration, the cluster weights and information gain are updated, and a new set of critical states are sampled to learn about NSEs, until the querying budget expires or the KL-divergence is below a problem-specific, pre-defined threshold.

4.1 Critical States Selection

When the budget for querying a human is limited, it is useful to query in states with a high *learning gap* measured as the KL-divergence between the agent's knowledge of NSE severity and the true NSE severity given the feedback data collected so far. States with a high learning gap are called *critical states* (Ω) and querying in these states can reduce the learning gap.

Since p^t and q^t contain categorical values rather than probabilities, their corresponding empirical probability mass functions (PMFs) are computed over the three NSE categories (no NSE, mild NSE, and

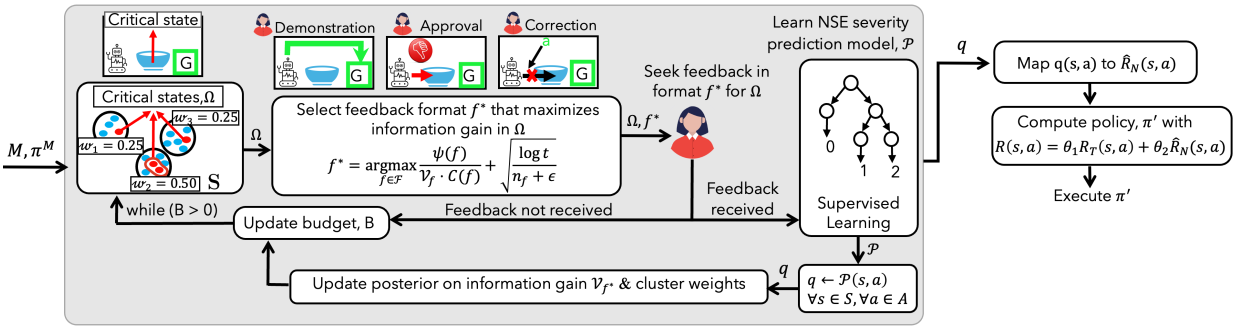

Critical state
G
Demonstration
Approval
Correction
a
Learn NSE severity prediction model, $\mathcal{P}$
Critical states, Ω
$w_3 = 0.25$
$w_1 = 0.25$
$w_2 = 0.50$
S
M, π^M
Ω
Select feedback format f^* that maximizes information gain in Ω
$f^* = \operatorname{argmax}_{f \in \mathcal{F}} \frac{\psi(f)}{\mathcal{V}_f \cdot C(f)} + \sqrt{\frac{\log t}{n_f + \epsilon}}$
Ω, f^*
Seek feedback in format f^* for Ω
0
1
2
Supervised Learning
q
Map q(s,a) to $\hat{R}_N(s,a)$
Compute policy, π' with $R(s,a) = \theta_1 R_T(s,a) + \theta_2 \hat{R}_N(s,a)$
Execute π'
while (B > 0)
Update budget, B
Feedback not received
Feedback received
$\mathcal{P}$
q
Update posterior on information gain $\mathcal{V}_{f^*}$ & cluster weights
$q \leftarrow \mathcal{P}(s,a)$ $\forall s \in S, \forall a \in A$

Figure 4. Solution approach overview. The critical states Ω for querying are selected by clustering the states. A feedback format f^* that maximizes information gain is selected for querying the user across Ω. The NSE model is iteratively refined based on feedback. An updated policy is calculated using a penalty function $\hat{R}_N$, derived from the learned NSE model.

severe NSE), yielding $\hat{p}^t$ and $\hat{q}^t$, respectively. In this case, $\hat{p}^t$ and $\hat{q}^t$ will be vectors of length three, since we consider three NSE categories.

In order to select critical states for querying, we compute the KL divergence between $\hat{q}^{t-1}$ and $\hat{p}^t$, $D_{KL}(\hat{p}^t \| \hat{q}^{t-1})$. Although $D_{KL}(\hat{p}^t \| \hat{q}^t)$ may appear as a reasonable criterion to guide critical states selection, it only measures how well the agent learns from the feedback at t. It does not reveal states where the agent's predictions were incorrect. For the example shown in Figure 3 with $q^{t-1}(s) = [0,0,0,0]$ and $p^t(s) = [2,0,0,0]$, $\hat{p}^t$ and $\hat{q}^{t-1}$ are calculated as the average occurrence of each NSE category (no NSE, mild NSE, severe NSE) across the four actions. That is, for $q^{t-1}(s) = [0,0,0,0]$, the frequency is $[\frac{4}{4}, \frac{0}{4}, \frac{0}{4}]$, resulting in $\hat{q}^{t-1}(s) = [1.0, 0.0, 0.0]$. For $p^t(s) = [2,0,0,0]$, the frequency is $[\frac{3}{4}, \frac{0}{4}, \frac{1}{4}]$, yielding $\hat{p}^t(s) = [0.75, 0.0, 0.25]$. Calculating the divergence between $\hat{p}^t(s)$ and $\hat{q}^{t-1}(s)$ reveals that the prediction was incorrect at s and therefore more data is required to align the learned model, and hence s or similar states should be selected for querying. Therefore, the sampling weight of the cluster containing s is increased (the region where the NSE model is still uncertain). In the following iteration, critical states are drawn from the reweighted clusters. Algorithm 1 outlines our approach for selecting critical states at each learning iteration, with the following three key steps.

1. Clustering states: Since NSEs are typically correlated with specific state features and do not occur at random, we cluster the states S into $\mathcal{K}$ number of clusters so as to group states with similar NSE severity (Lakkaraju et al., 2017). In our experiments, we use KMeans clustering algorithm with Jaccard distance to measure the distance between states based on their features. In practice, any clustering algorithm can be used, including manual clustering. The goal is to create meaningful partitions of the state space to guide critical states selection for querying the user.

2. Estimating information gain: We define the information gain of sampling from a cluster $k \in K$, based on the learning gap, as follows:

$$IG(k)^t = \frac{1}{|\Omega_k^{t-1}|} \sum_{s \in \Omega_k^{t-1}} D_{KL}(\hat{p}^t \| \hat{q}^{t-1}) \tag{1}$$

$$= \frac{1}{|\Omega_k^{t-1}|} \sum_{s \in \Omega_k^{t-1}} \sum_{l \in \{l_a, l_m, l_h\}} \hat{p}^t(l|s) \cdot \log\left(\frac{\hat{p}^t(l|s)}{\hat{q}^{t-1}(l|s)}\right), \tag{2}$$

Algorithm 1 Critical States Selection

```
Require: $N$: #critical states; $\mathcal{K}$:#clusters;
1:  $\Omega \leftarrow \emptyset$
2:  Cluster states into $\mathcal{K}$ clusters, $K = \{k_1, \ldots, k_{\mathcal{K}}\}$
3:  for each cluster $k \in K$ do
4:      $W_k \leftarrow \begin{cases} \frac{1}{\mathcal{K}}, & \text{if no feedback received in any iteration} \\ \frac{IG(k)}{\sum_{k \in K} IG(k)}, & \text{if feedback received} \end{cases}$
5:      $n_k \leftarrow \max(1, \lfloor W_k \cdot N \rfloor)$
6:      Sample $n_k$ states at random, $\Omega_k \leftarrow \text{Sample}(k, n_k)$
7:      $\Omega \leftarrow \Omega \cup \Omega_k$
8:  end for
9:  $N_r \leftarrow N - |\Omega|$
10: if $N_r > 0$ then
11:     $k' \leftarrow \arg\max_{k \in K} W_k$
12:     $\Omega \leftarrow \Omega \cup \text{Sample}(k', N_r)$
13: end if
14: return Set of selected critical states $\Omega$
```

where Ω_k^{t-1} denotes the set of states sampled for querying from cluster k at iteration $t-1$. $\hat{p}^t(l|s)$ and $\hat{q}^{t-1}(l|s)$ denote the probability of observing NSE category $l \in \{l_a, l_m, l_h\}$ in state s, derived from p^t and q^t, respectively. This formulation quantifies how much the predicted NSE distribution diverges from the feedback received for each state, providing a principled measure of the expected information gain from querying in a cluster, k.

3. Sampling critical states: At each learning iteration t, the agent assigns a weight w_k to each cluster $k \in K$, proportional to the new information on NSEs revealed by the most informative feedback format identified at $t-1$, using Eqn. 2. Clusters are given equal weights when there is no prior feedback (Line 4). Let N denote the number of critical states to be sampled in every iteration. We sample critical states in batches but they can also be sampled sequentially. When sampling in batches of N states, the number of states n_k to be sampled from each cluster is determined by its assigned weight. At least one state is sampled from each cluster to ensure sufficient information for calculating the information gain for every cluster (Line 5). The agent randomly samples n_k states from corresponding cluster and adds them to a set of critical states Ω (Lines 6, 7). If the total number of critical states sampled is less than N due to rounding, then the remaining N_r states are sampled from the cluster with the highest weight and added to Ω (Lines 9-12).

4.2 Feedback Format Selection

To query in the critical states, Ω, it is important to select a feedback format that not only maximizes the expected information gain about NSEs but also accounts for likelihood and cost of the feedback. The *information gain* of a feedback format f at iteration t, for $N = |\Omega|$ critical states, is computed as the KL divergence between the observed and predicted NSE severity distributions, $\hat{p}^t$ and $\hat{q}^t$:

$$\mathcal{V}_f = \frac{1}{N} \sum_{s \in \Omega} D_{KL}(\hat{p}^t \| \hat{q}^t) \cdot \mathbb{I}[f = f_H^t] + \mathcal{V}_f \cdot (1 - \mathbb{I}[f = f_H^t]), \tag{3}$$

where, $\mathbb{I}[f = f_H^t]$ is an indicator function that checks whether the format provided by the human, f_H^t, matches the requested format f. If no feedback is received, the information gain for that format remains unchanged. The following equation is used to select the feedback format f^*, accounting for feedback cost

Algorithm 2 Feedback Selection for NSE Learning

```
Require: B, Querying budget; $D$, Human preference model; $\delta$: KL divergence threshold
 1: $t \leftarrow 1$; $\mathcal{V}_f \leftarrow 0$ and $n_f \leftarrow 0,\ \forall f \in \mathcal{F}$
 2: Initialize $p$ and $q$:        // $p$: random initialization, $q$: all safe
    $\forall s \in S, \forall a \in A$, $p(s,a) \leftarrow$ RandomNSELabel($\{l_a, l_m, l_h\}$); $q(s,a) \leftarrow l_a$
 3: while $B > 0$ or $\forall s \in S, D_{KL}(\hat{p}^t \| \hat{q}^t) \leq \delta$ do
 4:     Sample critical states using Algorithm 1
 5:     Query user with feedback format $f^*$, selected using using Eqn. 4, across sampled $\Omega$
 6:     if feedback received in format $f^*$ then
 7:         $p^t \leftarrow$ Update distribution based on the feedback received in format $f^*$
 8:         $\mathcal{P} \leftarrow$ TrainClassifier($p^t$)
 9:         $q^t \leftarrow \{\mathcal{P}(s,a), \forall a \in A, \forall s \in \Omega\}$
10:         Update $\mathcal{V}_{f^*}$, using Eqn. 3
11:         $n_{f^*} \leftarrow n_{f^*} + 1$
12:     end if
13:     $B \leftarrow B - C(f^*)$; $t \leftarrow t + 1$
14: end while
15: return NSE classifier model, $\mathcal{P}$
```

and user preferences:

$$f^* = \operatorname*{argmax}_{f \in \mathcal{F}} \underbrace{\frac{\psi(f)}{\mathcal{V}_f \cdot \mathrm{C}(f)} + \sqrt{\frac{\log t}{n_f + \epsilon}}}_{\text{Feedback utility of } f}, \tag{4}$$

where $\psi(f)$ is the probability of receiving a feedback in format f and $C(f)$ is the feedback cost, determined using the human preference model D. t is the current learning iteration, n_f is the number of times feedback in format f was received, and ϵ is a small constant for numeric stability. The selected format f^* represents the most informative feedback format given the agent's current knowledge, balancing exploration (less frequently used formats) and exploitation (formats known to provide high information gain).

Algorithm 2 outlines our feedback format selection approach. Since the agent has no prior knowledge of how the human categorizes NSE for each state-action pairs, the labeling function p is instantiated by sampling from a uniform prior over the three NSE labels (l_a, l_m, l_h) for every (s, a), while q is initialized assuming all actions are safe (l_a) (Line 2). These initial labels are progressively refined as human feedback is received. At each iteration, the agent samples $|\Omega|$ critical states using Algorithm 1 (Line 4), and selects a feedback format f^* is selected using Eqn. 4. The agent queries the human for feedback in f^* (Line 5). If the feedback is received (with probability $\psi(f^*)$), the observed NSE labels p^t are updated and an NSE prediction model $\mathcal{P}$ is trained (Lines 6-8). The classifier $\mathcal{P}$ predicts the labels for the sampled critical states Ω, yielding q^t. We restrict the prediction to Ω since these states indicate regions of high uncertainty and contribute to reducing the divergence between the true and learned NSE distributions. Further, restricting predictions to Ω also reduces computational overhead during iterative querying. $\mathcal{V}_{f^*}$ recomputed using Eqn. 3, and n_{f^*} is incremented (Lines 9-11). This repeats until either the querying budget is exhausted or the KL divergence between $\hat{p}^t$ and $\hat{q}^t$ over all states is within a problem-specific threshold δ.

Figure 5 illustrates the critical states and the most informative feedback formats selected at each iteration in the object delivery task using AFS, demonstrating that feedback utility changes over time, based on the robot's current knowledge.

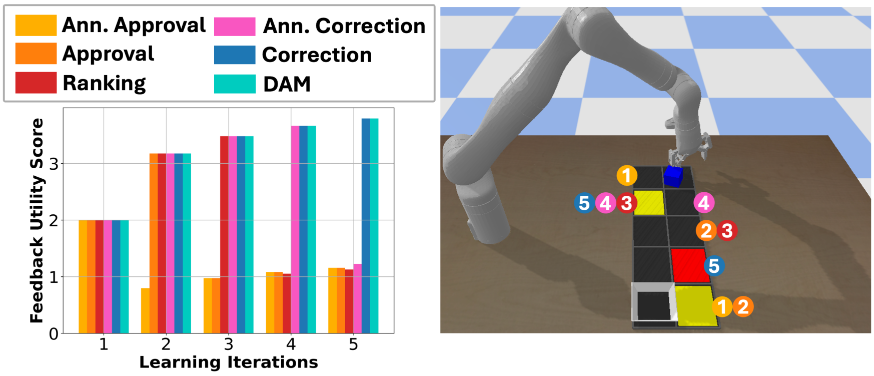
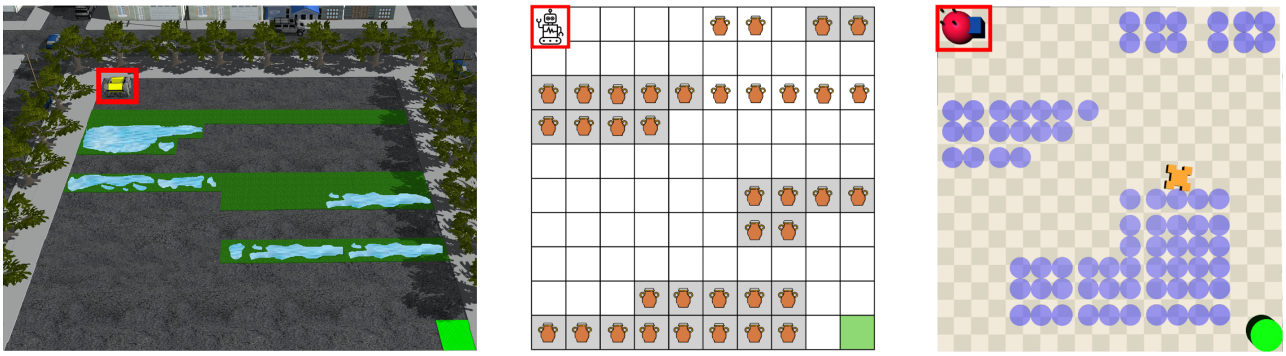


Figure 5. Feedback utility of each format across iterations. Numbers mark when a state was identified as critical, and circle colors denote the chosen feedback format.



(a) Navigation: Unavoidable NSE (b) Vase: Unavoidable NSE (c) Safety-gym Push

Figure 6. Illustrations of evaluation domains. Red box denotes the agent and the goal location is in green.

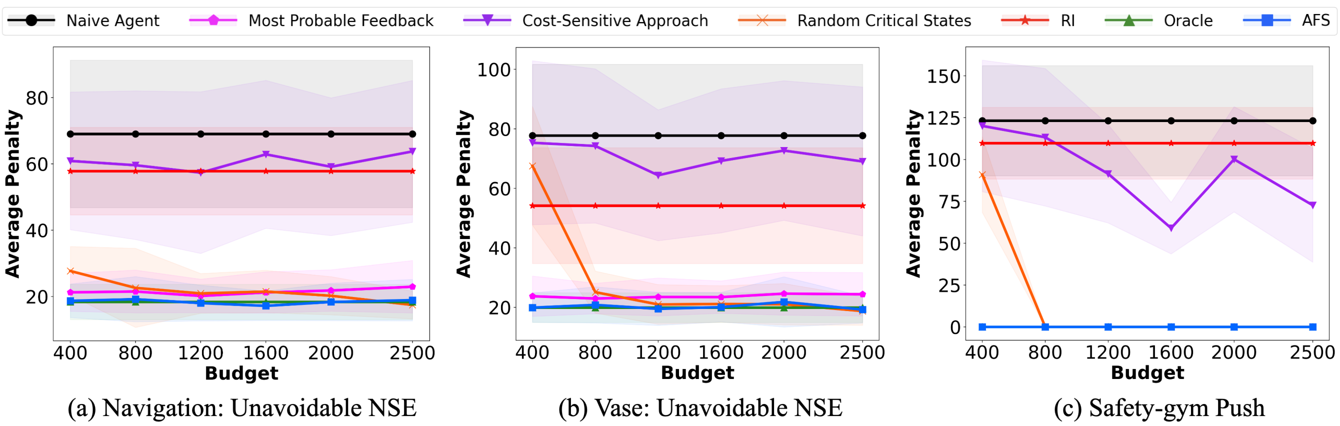
4.3 Stopping Criteria

Besides guiding the selection of critical states and feedback format, the KL-divergence also serves as an indicator of when to stop querying. The querying phase can be terminated when $D_{KL}(\hat{p}^t \| \hat{q}^t) \leq \delta$, where δ is a problem-specific threshold. When $D_{KL}(\hat{p}^t \| \hat{q}^t) \leq \delta$, it indicates that the learned model is a reasonable approximation of the underlying NSE distribution, and therefore the querying can be terminated even if the allotted budget B has not been exhausted. The choice of δ provides a trade-off between thorough learning and human effort, and can be tuned based on domain-specific safety requirements.

5 EXPERIMENTS IN SIMULATION

We first evaluate AFS on three simulated domains (Fig. 6). Human feedback is simulated by modeling an oracle that selects safer actions with higher probability using a softmax action selection (Ghosal et al., 2023; Jeon et al., 2020): the probability of choosing an action a' from a set of all safe actions A^* in state s is, $\Pr(a'|s) = \frac{e^{Q(s,a')}}{\sum\limits_{a \in A^*} e^{Q(s,a)}}$.

Baselines (i) *Naive Agent*: The agent naively executes its primary policy without learning about NSEs, providing an upper bound on the NSE penalty incurred. (ii) *Oracle*: The agent has complete knowledge about R_T and R_N, providing a lower bound on the NSE penalty incurred. (iii) *Reward Inference with β Modeling (RI)* (Ghosal et al., 2023): The agent selects a feedback format that maximizes information gain



(a) Navigation: Unavoidable NSE (b) Vase: Unavoidable NSE (c) Safety-gym Push

Figure 7. Average penalty incurred when querying with different feedback selection techniques.

according to the human's inferred rationality, β. (iv) *Cost-Sensitive Approach*: The agent selects a feedback method with the least cost, according to the preference model D. (v) *Most-Probable Feedback*: The agent selects a feedback format that the human is most likely to provide, based on D. (vi) *Random Critical States*: The agent uses our AFS framework to learn about NSEs, except the critical states are sampled randomly from the entire state space. We use $\theta_1 = 1$ and $\theta_2 = 1$ for all our experiments. AFS uses learned $\hat{R}_N$.

Domains, Metrics and Feedback Formats We evaluate the performance of various techniques on three domains in simulation (Figure 6): outdoor navigation, vase and safety-gym's push. We optimize costs (negations of rewards) and compare techniques using average NSE penalty and average cost to goal, averaged over 100 trials. For navigation, vase and push, we simulate human feedback. The cost for l_a, l_m, and l_h are 0, $+5$, and $+10$ respectively.

Navigation: In this ROS-based city environment, the robot optimizes the shortest path to the goal location. A state is represented as $\langle x, y, f, p\rangle$, where, x and y are robot coordinates, f is the surface type (concrete or grass), and p indicates the presence of a puddle. The robot can move in all four directions and each costs $+1$. Actions succeed with probability 0.8. Navigating over grass damages the grass and is a mild NSE. Navigating over grass with puddles is a severe NSE. Features used for training are $\langle f, p\rangle$. Here, NSEs are unavoidable.

Vase: In this domain, the robot must quickly reach the goal, while minimizing breaking a vase as a side effect (Krakovna et al., 2020). A state is represented as $\langle x, y, v, c\rangle$ where, x and y are robot's coordinates. v indicates the presence of a vase and c indicates if the floor is carpeted. The robot moves in all four directions and each costs $+1$. Actions succeed with probability 0.8. Breaking a vase placed on a carpet is a mild NSE and breaking a vase on the hard surface is a severe NSE. $\langle v, c\rangle$ are used for training. All instances have unavoidable NSEs.

Push: In this `safety-gymnasium` domain, the robot aims to push a box quickly to a goal state (Ji et al., 2023). Pushing a box on a hazard zone (blue circles) produces NSEs. We modify the domain such that in addition to the existing actions, the agent can also *wrap* the box that costs $+1$. Every move action succeeds with probability 0.8, and the wrap action succeeds with probability 1.0. The NSEs can be avoided by pushing a wrapped box. A state is represented as $\langle x, y, b, w, h\rangle$ where, x, y are the robot's coordinates, b indicates carrying a box, w indicates if box is wrapped and h denotes if it is a hazard area. $\langle b, w, h\rangle$ are used for training.

Table 1. Avg. cost and standard error at task completion.

Method	Navigation: unavoidable NSE	Vase: unavoidable NSE	Safety-gym Push: avoidable NSE
Oracle	51.37 ± 2.69	54.46 ± 6.70	44.62 ± 9.97
Naive	36.11 ± 1.39	36.0 ± 2.89	39.82 ± 5.44
RI	40.10 ± 0.69	37.42 ± 1.01	42.15 ± 2.44
AFS	64.8 ± 2.3	52.68 ± 7.87	48.32 ± 4.42

5.1 Results and Discussion

Effect of learning using AFS: We first examine the benefit of querying using AFS, by comparing the resulting average NSE penalties and the cost for task completion, across domains and query budget. Figure 7 shows the average NSE penalties when operating based on an NSE model learned using different querying approaches. Clusters for critical state selection were generated using KMeans clustering algorithm with $K = 3$ for navigation, vase and safety-gym's push domains (Figure 7 (a-c)). The results show that our approach consistently performs similar to or better than the baselines.

There is a trade-off between optimizing task completion and mitigating NSEs, especially when NSEs are unavoidable. While some techniques are better at mitigating NSEs, they significantly impact task performance. Table 1 shows the average cost for task completion at $B = 400$. *Lower* values are better for both NSEs and task completion cost. While the Naive Agent has a lower cost for task completion, it incurs the highest NSE penalty as it has no knowledge of R_N. RI causes more NSEs, especially when they are unavoidable, as its reward function does not fully model the penalties for mild and severe NSEs. Overall, the results show that our approach consistently mitigates avoidable and unavoidable NSEs, without affecting the task performance substantially.

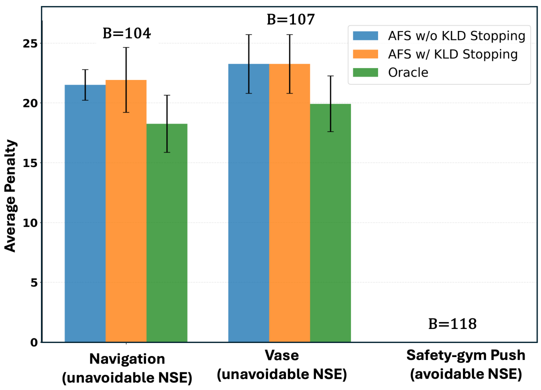
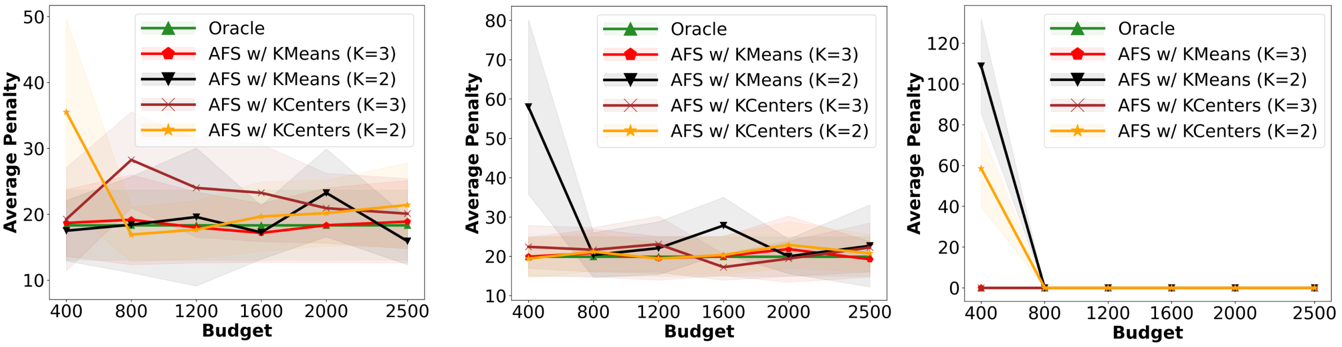


Figure 8. Average penalty incurred when learning with AFS using querying budget $B = 400$, and KL divergence (KLD) as the stopping criterion. The budget utilized by AFS with KLD stopping is annotated in the plot.

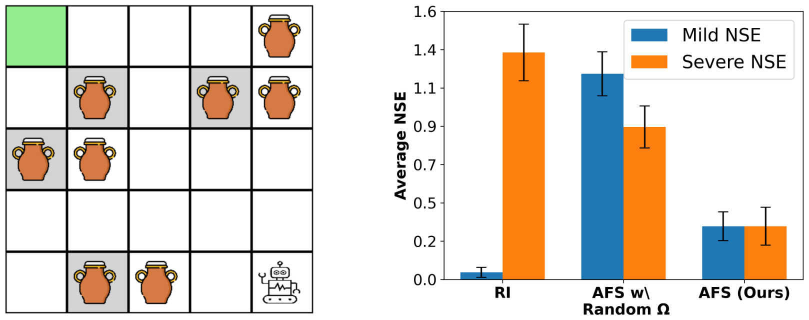
Figure 8 shows the average penalty when AFS uses KL-divergence (KLD) as the stopping criteria, compared to querying with budget $B = 400$. For comparison, we also annotate in the plot the querying budget used by AFS with KLD stopping at the time of termination. The results show that despite terminating earlier and using few queries, AFS with the KLD stopping achieves comparable performance to that of AFS with query budget $B = 400$, demonstrating the usefulness of KLD as a stopping criterion.

Clustering Figure 9 shows the average penalty incurred using our approach (AFS) with the KMeans and KCenters clustering algorithms for varying numbers of clusters ($K = \{2, 3\}$ in the navigation, vase and push domains). We restrict our evaluation to these K values since the maximum number of distinct clusters in each domain is determined by number of unique combinations of state features. In the navigation domain, features used for clustering states are $\langle f, p \rangle$. The valid unique combinations are $\langle f =$ concrete, $p =$ no puddle$\rangle$, $\langle f =$ grass, $p =$ no puddle$\rangle$, and $\langle f =$ grass, $p =$ puddle$\rangle$. Hence, having $K > 3$ will not produce unique clusters. Similarly, in the vase domain, features used for clustering are $\langle v, c \rangle$, where the unique, valid



(a) Navigation: Unavoidable NSE (b) Vase: Unavoidable NSE (c) Safety-gym Push

Figure 9. Average penalty incurred using our approach (AFS) with KMeans and KCenters clustering algorithm, evaluated across varying number of clusters (K).



(a) Vase domain setup (b) Average number of NSEs

Figure 10. Results from the user study on a simulated domain.

combinations are ⟨no vase, no carpet⟩, ⟨vase, no carpet⟩, ⟨vase, carpet⟩. For the push domain, the features used for clustering are $\langle b, w, h \rangle$, with valid unique combinations including ⟨no box, not wrapped, hazard⟩, ⟨box, not wrapped, hazard⟩, ⟨no box, not wrapped, no hazard⟩, and ⟨box, wrapped, no hazard⟩. The results in Figure 9 demonstrate that increasing K generally improves the performance of our approach, with both clustering methods. A higher number of clusters allows for a more refined grouping of states based on distinct state features, enabling the agent to query the human for feedback across a more diverse set of states. This diversity enhances the agent's ability to accurately learn and mitigate NSEs.

6 HUMAN SUBJECTS PILOT STUDY IN SIMULATION

We conducted a within-subjects pilot study on a 5×5 Vase domain in simulation as shown in Fig. 10(a), with 12 human participants who had completed at least one course in Reinforcement Learning. The objective of this study is to evaluate whether: (1) AFS outperforms the baselines when a feedback preference model is learned from user interactions; (2) the selected feedback formats and critical states enhance agent's learning, and align with user preferences. The study was conducted with approval from Oregon State University IRB, and the participants were compensated with a $\$10$ Amazon gift card for their time.

6.1 Study Design

After introducing the domain and the agent's objective, users completed a tutorial where they interacted with the system by providing feedback in each of the six formats. The study interface included feedback

Table 2. Participants' qualitative assessment from the pilot study on a simulated domain.

Approach	Intelligent Feedback	Critical Points (%)			Improved Performance (%)	
		Yes	No	Overlap	Yes	No
RI	3.33 ± 1.23	83.30 ± 0.37	16.70 ± 0.37	73.47 ± 5.49	91.70 ± 0.28	8.30 ± 0.28
AFS w/ Random Ω	2.82 ± 0.94	66.70 ± 0.47	33.30 ± 0.94	75.51 ± 5.27	41.70 ± 0.49	58.30 ± 0.49
AFS (Ours)	3.25 ± 0.83	100.00 ± 0.00	0.00 ± 0.00	81.63 ± 4.94	100.00 ± 0.00	0.00 ± 0.00

buttons that varied based on the format[1]. This was followed by a calibration phase, during which the users' preference model was learned. Each user was prompted five times per format to provide feedback, with the option to respond or ignore, allowing them to express their interaction preferences. The probability of receiving feedback in a given format was determined by the fraction of prompts the user responded to, while the cost was based on their self-reported effort.

The study comprised three phases, each evaluating a different baseline approaches to select feedback queries: (1) RI, (2) AFS with Random Ω, and (3) AFS with our proposed method for critical state selection. To prevent bias, users were unaware of the approach used in each phase. After completing a phase, they were shown a trajectory of the agent's learned policy and asked to evaluate the approach used in that phase.

6.2 Results and Takeaways

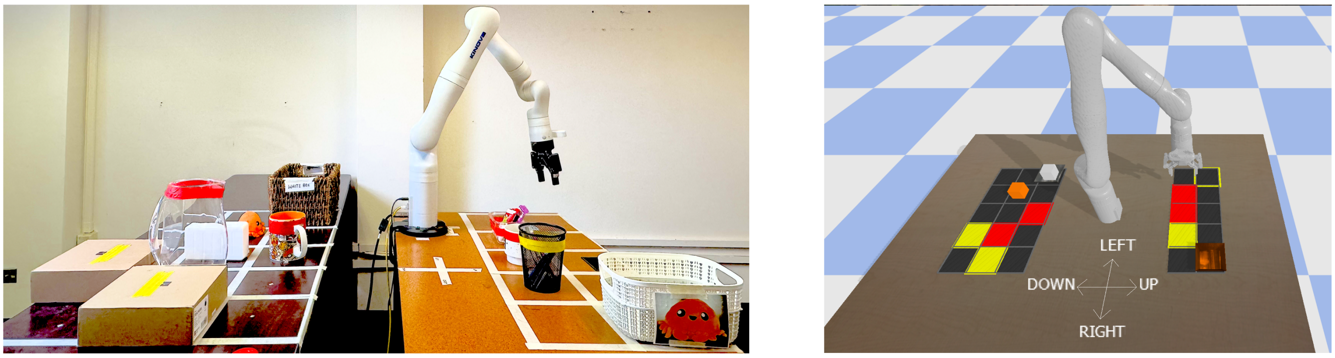
Fig. 10(b) shows that our approach tends to result in fewer NSEs, compared to the baselines. Since the NSE penalty is an aggregate measure that obscures severity distribution, we report exact NSE encounters by category for this study. Table 2 reports average over responses to our questions: "On a scale of 1 to 5, how intelligent do you think the agent's choice of feedback formats are, given your preferences?", "Were the states in which the agent requested for feedback critical to its learning?", and "Did the agent's performance improve at the end of the learning phase?". In addition, we also report the overlap between user-identified important query points and query points chosen by each approach.

Overall, the results of this pilot study indicate that (1) AFS tends to effectively select query points and lead to improved learning outcomes, when operating under a learned feedback model; and (2) AFS's performance in this pilot study where users interact with a simulated agent is comparable to that of our results in simulation (Sec. 5). Building on these results and the insights gained from this pilot study, we next conduct a user study where human participants interact with a physical robot. Such a setting will enable us to evaluate how well the observed trends extend to human-robot physical interactions, and how that affects the usability, trust and the users' perceived workload when interacting with a system that learns using AFS.

7 IN-PERSON USER STUDY WITH A PHYSICAL ROBOT ARM

We conducted an in-person study with a Kinova Gen3 7DoF arm (Kinova, 2025) tasked with delivering two objects—an orange toy and a white box—across a workspace containing items of varying fragility (Figure 11). This setup involves users providing both interface-based and kinesthetic feedback to the

[1] See Appendix Sec.2.1 for the interface corresponding to each feedback format.



(a) User Study physical setup

(b) Setup in simulation

Figure 11. Task setup for the human subject study. **(a)** Physical setup of the task for human subjects study; **(b)** Replication of the physical setup using PyBullet. A dialog box corresponding to the current feedback format is shown for every query.

robot. The study was approved by Oregon State University IRB. Participants were compensated with a $15 Amazon gift card for their participation in the study.

This user study had three goals: (1) to measure our approach's effectiveness in reducing NSEs for a real-world task, (2) to understand how users perceive the adaptivity, workload and competence of the robot operating in the AFS framework, and (3) to evaluate the extent to which AFS captures user preferences in practice, while ensuring maximum information gain during the learning process.

7.1 Methods

7.1.1 Participants

Besides the pilot study in Sec. 6, we conducted another pilot study with $N=10$ participants to evaluate the study setup with the Kinova arm. In particular, this pilot study assessed the clarity of instructions, survey wording, and feasibility of the task design in the object delivery task of the Kinova arm. Based on the participant feedback, we simplified the survey questions and included example trajectories that demonstrated safe and NSE-causing behaviors. For the main study, we recruited $N = 30$ participants with basic computer literacy from the *general population* through university mailing lists and public forums. Participants were aged 18–72 years ($M=32.10, SD=13.11$), with 53.3% men and 46.7% women. Participants reported varied prior experience with robots: 73.3% had general awareness of similar robot products, 6.7% had researched or investigated robots, 3.3% had interacted through product demos, and 13.3% had no prior awareness of similar products.

7.1.2 Robotic System Setup

The Kinova Gen3 arm was equipped with a joint space compliant controller which allowed participants to physically move the joints of the arm through space with gravity compensation when needed. Additionally, a task-space planner allowed for navigation to discrete grid positions for both feedback queries and policy execution (Kinova, 2025). Figure 11(a) shows the physical workspace and the two delivery objects, while Figure 11(b) shows the corresponding PyBullet simulation used for visualization during GUI-based feedback. A dialog box was displayed to prompt the participant whenever feedback was queried[2].

[2] See Appendix Sec.3.1 for details on the dialog box and examples for each feedback format.

7.1.3 Interaction Premise

The interaction simulated an assistive robot delivering objects to their designated bins. Specifically, the task required the Kinova arm to deliver an orange plush toy and a rigid white box to their respective bins while avoiding collision with surrounding obstacles of different fragility. Collisions with fragile obstacles (e.g. a glass vase) during delivery of the plush toy were considered a mild NSE. Collisions involving the white rigid box were severe NSEs if with a fragile object and were mild NSEs if with a non-fragile object. All other scenarios were considered safe. The workspace was discretized into a grid of cells marked with tape on the tabletop and mirrored in the GUI. Each cell represented a state corresponding to possible end-effector position.

7.1.4 Study Design

The robot's state space was discretized and represented as $\langle x, y, i_1, i_2, o, f, g_1, g_2 \rangle$, where (x, y) denote the end-effector position, i_1 and i_2 indicate the presence of either orange plush toy or white rigid box in the end effector, o indicates the presence of an obstacle, and f indicates obstacle fragility, and g_1 and g_2 indicate whether either of the objects were delivered in their corresponding goal locations (i.e., orange plush toy in white bin and the white box in the wicker bin).

Participants interacted with the robot through *four* feedback formats, $\mathcal{F} = \{\text{App, Corr, Rank, DAM}\}$, during both the training and main experience phases. Depending on the feedback format, the Kinova arm executed the queried action in the physical workspace or displayed a simulation of the action in the graphical user interface (GUI). Interaction across the four feedback formats are described below.

1. **Approval:** The robot executed a single action in simulation, and participants indicated whether it was safe by selecting "yes" or "no" in the GUI.
2. **Correction:** The robot first executes action prescribed by its policy in simulation. If the action in simulation is deemed unsafe by the participant, the robot in the physical setup moves to the queried location. Participants then correct the robot by physically moving the robot arm to demonstrate a safe alternative action.
3. **Demo-Action Mismatch:** The robot first physically moved its arm to a specific end-effector position in the workspace. Participants then provided feedback by guiding the arm to a safe position, thereby demonstrating the safe action. The robot compares the action given by its policy to the demonstrated action. If the robot's action and the demonstrated actions do not match, then the robot's action is considered unsafe.
4. **Ranking:** Simulation clips of two actions selected at random in a given state were presented in GUI. Participants compared the two candidate actions and selected which was safer. If both actions were judged equally safe or unsafe, either option could be chosen.

Each participant experienced four learning conditions in a within-subjects, counterbalanced design:

1. The baseline RI approach proposed in Ghosal et al. (2023),
2. AFS with random Ω, where critical states are randomly selected,
3. AFS with a fixed feedback format (DAM) for querying, consistent with prior works that rely primarily on demonstrations, and
4. The proposed AFS approach, where both the feedback format and the critical states are selected to maximize information gain.

Each condition is a distinct feedback query selection strategy controlling how the robot queried participants during learning. These conditions are the independent variables. The dependent measures include NSE occurrences, their severity, perceived workload, trust, competence and user alignment.

7.1.5 Hypotheses

We test the following hypotheses in the in-person study. These hypotheses were derived from trends observed in the experiments and human subjects study in simulation (Sections 5 and 6).

H1: *Robots learning using AFS will have fewer NSEs in comparison to the baselines.*
This hypothesis is derived from the results of our experiments on simulated domains (Figure 7) where AFS consistently reduced NSEs while completing the assigned task. We hypothesize that this trend extends to physical human-robot interactions.

H2: *AFS will achieve comparable or better performance compared to the baselines, with a lower perceived workload for the users.*
The results on simulated domains (Figure 8) show that AFS achieved better or comparable performance to the baselines, using fewer feedback queries. While the in-person user study requires relatively greater physical and cognitive effort, we expect the advantage of the sample efficiency to persist and investigate whether it translates to reduced perceived workload.

H3: *Participants will report AFS as more trustworthy, competent, and aligned with user expectations, in comparison to the baselines.*
In the human subjects simulation study (Table 2), participants reported that AFS selected intelligent queries, targeted critical states, and improved the agent's performance, reflecting indicators of trust, competence and user alignment. We hypothesize that this trend extends to physical settings as well.

Hypotheses **H1** and **H2** explore trends identified in simulation and are therefore confirmatory. Hypothesis **H3** builds on the perception measures used in the human subjects study in simulation, and is hence treated as an extended confirmatory hypothesis.

7.1.6 Procedure

Each study session lasted approximately one hour and followed three phases.

7.1.6.1 Training

Participants were first introduced to the task objective, workspace, and the four feedback formats. For each format, they provided feedback on four sample queries to practice both GUI-based and kinesthetic interactions. After the completing each format, the participants rated the following: (i) probability of responding to a query in that format, $\psi(f)$, (ii) perceived cost or effort required to provide feedback, $C(f)$, and (iii) the overall task workload. This phase helped establish measures like feedback likelihood, perceived effort, and workload.

7.1.6.2 Main Experience

Following training, participants completed the four learning conditions corresponding to different approaches under evaluation. In each condition, the participants provided feedback to train the robot to avoid collision while performing the object-delivery task. Depending on the feedback format selected by the querying strategy, participants either evaluated short simulation clips on the GUI or physically guided the robotic arm. At the end of each condition, the robot executed its learned policy based on its learning under that condition. The participants then observed its performance and completed a brief post-condition questionnaire assessing workload, trust, perceived competence, and user-alignment.

7.1.6.3 Closing

At the end of the study, participants compared the four learning approaches in terms of trade-offs between learning speed and safety. Participants reported their preferences on providing feedback through multiple formats versus relying on a single feedback format. These responses offered qualitative insight into AFS's practicality and user acceptance.

7.1.7 Measures

We collected both quantitative and qualitative measures. The quantitative measure captured task-level performance through the frequency and the severity of NSEs (mild and severe). Qualitative measures captured participants' perceptions of the following.

1. **Workload:** Participants' perceived workload across the feedback formats and learning conditions were measured using the NASA Task Load Index (NASA TLX) (Hart and Staveland, 1988). The questionnaire scales were transformed to 7-point subscales ranging from "Very Low" (1) to "Very High" (7). Responses were collected during the training phase and after each condition in the main experience phase.
2. **Robot Attributes:** Perceived robot attributes, like competence, warmth and discomfort, were measured using the 9-point Robotic Social Attributes Scale (RoSAS) (Carpinella et al., 2017), ranging from "Strongly Disagree" (1) to "Strongly Agree" (9). Participants completed this questionnaire after each learning condition.
3. **Trust:** A custom 10-point trust scale $(0\% - 100\%)$ was used to measure participants' confidence in the robot's ability to act safely under each learning condition. Participants rated their trust both before and after the robot's training phase to capture changes in its learning performance.
4. **User Alignment:** Participants' perception of user alignment was assessed using a custom 7-point Likert scale ranging from "Strongly Disagree" (1) to "Strong Agree" (7). Participants rated (i) how well the critical states queried by the robot aligned with their own assessment of which states were important for learning, and (ii) how well the feedback formats chosen across conditions matched their personal feedback preferences. Higher rating indicated stronger perceived alignment between the robot's querying strategy and the participants' expectations.

7.1.8 Analysis

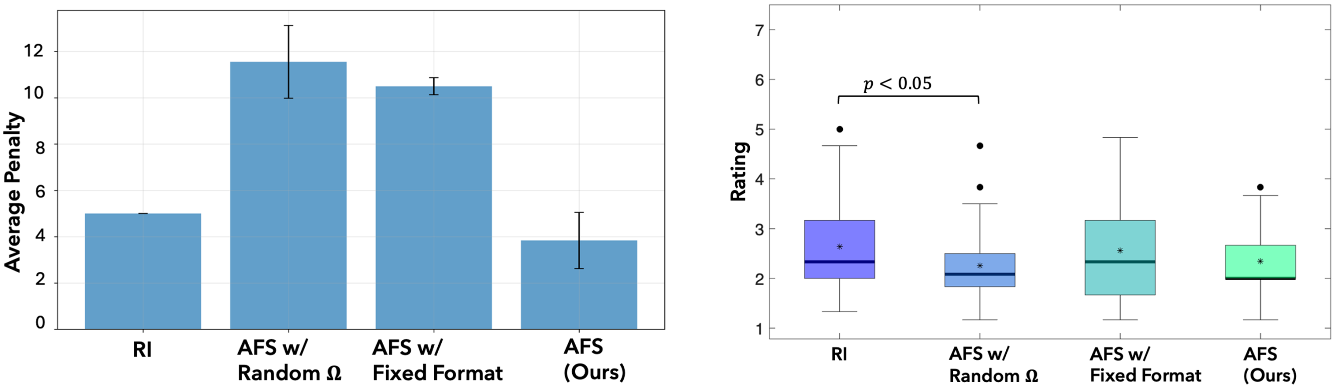
Survey responses were compiled into cumulative RoSAS (competence, warmth, discomfort) and NASA-TLX workload scores. A repeated-measures ANOVA (rANOVA) tested for significant differences across learning conditions; we report the F-statistic, p-value and effect size as generalized eta-squared (η_G^2). When effects were significant, Tukey's post-hoc tests identified pairwise differences. All results are reported with means (M), standard errors (SE), and p-values.

7.2 Results

We evaluate hypotheses **H1-H3** using both objective and subjective measures. Data from all 30 participants were included in the analysis, as all sessions were completed successfully.

7.2.1 Effectiveness of AFS in Mitigating NSEs (Hypothesis **H1**)

Figure 12(a) shows the average penalty incurred under each condition. AFS approach incurred the least NSE penalty ($M = 3.83, SE = 1.21$), substantially lower than AFS with random Ω ($M = 11.55, SE = 1.57$) and AFS with a fixed feedback format ($M = 10.50, SE = 0.37$). The RI baseline incurred higher penalties ($M = 5.00, SE = 0.00$) compared to AFS. These results confirm hypothesis **H1** and demonstrate that adaptively selecting both critical states and feedback formats reduced unsafe behaviors more effectively than random or fixed querying strategies.



(a) Average penalty incurred across methods in the human subjects study.

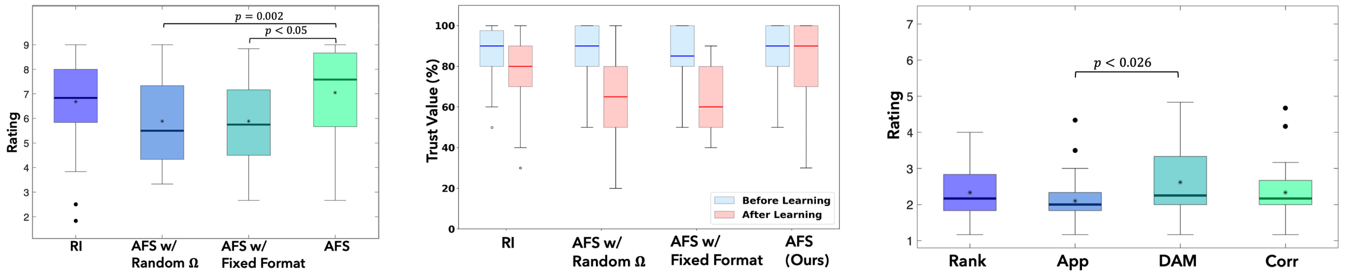
(b) NASA-TLX workload across the four conditions.

Figure 12. Results from the user study on the Kinova 7DoF arm.

7.2.2 Learning Efficiency and Workload (Hypothesis **H2**)

We first compare the perceived workload across different feedback formats, followed by the results across learning conditions. Demonstration is the most widely used feedback format in existing works but was perceived as the most demanding (Figure 13(c)). While corrections offer corrective action in addition to disapproving agent's action, it also imposed substantial effort on the users. Approval required the least workload but conveyed limited information. A repeated-measures ANOVA revealed a significant effect of feedback format on perceived workload, ($F(3, 87) = 3.33, p = 0.023, \eta^2_G = 0.046$). Post hoc comparisons indicated that Approval ($M = 2.11, SE = 0.12$) imposed significantly lower workload ($p = 0.026$) than Demo-Action Mismatch ($M = 2.62, SE = 0.19$), while no other pairwise differences reached significance. This trade-off underscores the need for an adaptive selection strategy to balance informativeness with user effort.

The rANOVA analysis across the four learning conditions further revealed a significant effect in the NASA-TLX workload ratings ($F(3, 87) = 3.73, p = 0.014, \eta^2_G = 0.030$). Among the four conditions, AFS achieved one of the lowest perceived workload ratings ($M = 2.34, SE = 0.12$), comparable to AFS with random Ω ($M = 2.26, SE = 0.15$) and lower than both AFS with fixed format ($M = 2.56, SE = 0.19$) and RI ($M = 2.64, SE = 0.19$). Tukey post-hoc tests showed that workload in AFS with random Ω imposed a significantly lower workload than RI ($p = 0.033$). Overall, these results support **H2**, indicating that adaptively selecting queries helps reduce perceived workload relative to the baselines (Figure 12(b)).



(a) RoSAS competence across the four conditions.

(b) Participants' perception of trust before and after learning.

(c) NASA-TLX scores across feedback formats

Figure 13. User study results. **(a)-(b)** RoSAS competence and NASA Task-Load across the four conditions in the main study; **(c)** NASA Task-Load across feedback formats.

7.2.3 Trust, Competence, and Preference Alignment (Hypothesis **H3**)

Participants' rating on the robot's ability to act safely increased after learning with AFS, as shown in Figure 13(b). A significant effect was also found for perceived robot competence ($F(3, 87) = 10.6, p < 0.001, \eta^2_G = 0.082$) (Figure 13(a)). AFS was rated highest ($M = 7.04, SE = 0.32$), significantly greater than AFS with random Ω ($M = 5.88, SE = 0.32, p = 0.002$) and AFS with fixed format ($M = 5.88, SE = 0.30, p < 0.001$), while comparable to RI ($M = 6.68, SE = 0.32$). These results support **H3**—AFS was perceived as more competent and trustworthy compared to the baselines.

Descriptive analyses of user alignment on state criticality and feedback alignment ratings, indicated consistent trends across participants. While differences between conditions were not statistically significant ($p > 0.05$), AFS consistently received higher ratings for feedback alignment ($M = 3.79, SE = 0.42$) relative to state criticality ($M = 3.14, SE = 0.40$), suggesting that participants found AFS's query selections relevant and aligned with their preferences. Participants (both those aware and unaware of similar robotic systems) perceived AFS's queries as critical for learning and well-aligned with their feedback preferences. Participants with prior research experience rated state criticality and format alignment comparable, indicating confidence in adaptivity of AFS's querying process.

8 DISCUSSION

Our experiments followed an increasingly realistic progression in design. In the experiments in simulation with both avoidable and unavoidable NSEs, AFS incurred lower penalties and overall costs compared to the baselines, demonstrating its ability to balance task performance with safety. The results of our pilot study, where users interacted with a simulated agent, showed that AFS effectively learns the participant's feedback preference model and uses them to select formats aligned with user expectations. Finally, the in-person user study with the Kinova arm, showed the practicality of using AFS in real-world settings, achieving favorable ratings on trust, workload, and user-preference alignment. These finding support our three hypotheses regarding the performance of AFS: (H1) it reduces unsafe behaviors more effectively than the baselines, (H2) it improves learning efficiency while reducing user workload, and (H3) it is perceived as more trustworthy and competent. The results collectively highlight that adaptively selecting both the query format and the states to pose the queries to the user enhances learning efficiency and reduces user effort.

Beyond confirming these hypotheses, the findings provide important design implications for human-in-the-loop learning systems. By modeling the trade-off between informativeness and effort, AFS offers a framework to balance user workload with the need for high-quality feedback. The learned feedback preference model allows the agent to adaptively select querying formats while minimizing human effort. Using KL-divergence as stopping criterion further enables adaptive termination of the querying process. This overcomes the problem of determining the “right” querying budget for a problem, and shows that AFS enables efficient learning while minimizing redundant human feedback. These design principles can inform the development of interactive systems that adapt query format and frequency based on agent’s current knowledge and user feedback preferences. Overall the results show that AFS (1) consistently outperforms the baselines across different evaluation settings, and (2) can be effectively deployed in real-world human-robot interaction scenarios.

A key strength of this work lies in its extensive evaluation, from simulation to real robot studies, supporting AFS’s robustness and practicality. One limitation, however, is that the current evaluation focuses on discrete environments. Extending AFS to continuous domains introduces challenges such as identifying critical states and estimating divergence-based information gain in high-dimensional spaces. While gathering feedback at the trajectory-level is relatively easier in continuous settings, gathering state-level feedback, which is the focus of this work, is challenging. These challenges stem from the need for scalable state representations and efficient sampling strategies, which will be a focus for future work.

9 CONCLUSION AND FUTURE WORK

The proposed Adaptive Feedback Selection (AFS) facilitates querying a human in different formats in different regions of the state space, to effectively learn a reward function. Our approach uses information gain to identify critical states for querying, and the most informative feedback format to query in these states, while accounting for the cost and uncertainty of receiving feedback in each format. Our empirical evaluations using four domains in simulation and a human subjects study in simulation demonstrate the effectiveness and sample efficiency of our approach in mitigating avoidable and unavoidable negative side effects (NSEs). The subsequent in-person user study with a Kinova Gen3 7DoF arm further validates these finding, showing that AFS not only improves NSE avoidance but also enhances user trust, competence perception, and user-alignment. Future work will focus on extending AFS to continuous state and action spaces, strengthening AFS’s applicability to complex, safety-critical domains where user-aware interaction is essential.

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

YA: Writing – Original Draft and Editing, Methodology, Experiment Design, Data Curation, Visualization, User Study Design, User Study Execution; NN: Writing – Review and Editing, User Study Design, Experiment Setup (Kinova Arm); KS: User Study Execution, Data Curation, Data Analysis; NF: Supervision, User Study Oversight, Resources, Writing – Review; SS: Supervision, Writing – Original Draft, Review and Editing, Funding Acquisition, Resources, Experiment Design, User Study Design.

FUNDING

This work was supported in part by National Science Foundation grant number 2416459.

DATA AVAILABILITY STATEMENT

The raw data supporting these conclusions will be made available by the corresponding author upon request.

REFERENCES

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-first International Conference on Machin Learning, (ICML)*

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

Beierling, H., Beierling, R., and Vollmer, A. (2025). The power of combined modalities in interactive robot learning. *Frontiers in Robotics and AI*

Bıyık, E., Losey, D. P., Palan, M., Landolfi, N. C., Shevchuk, G., and Sadigh, D. (2022). Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *The International Journal of Robotics Research (IJRR)*

Bobu, A., Wiggert, M., Tomlin, C., and Dragan, A. D. (2021). Feature expansive reward learning: Rethinking human input. In *Proceedings of ACM/IEEE International Conference on Human Robot Interaction (HRI)*

Brown, D., Coleman, R., Srinivasan, R., and Niekum, S. (2020a). Safe imitation learning via fast bayesian reward inference from preferences. In *International Conference on Machine Learning (ICML)* (PMLR)

Brown, D. and Niekum, S. (2018). Efficient probabilistic performance bounds for inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*

Brown, D., Niekum, S., and Petrik, M. (2020b). Bayesian robust optimization for imitation learning. *Advances in Neural Information Processing Systems (NeurIPS)*

Brown, D. S., Cui, Y., and Niekum, S. (2018). Riskaware active inverse reinforcement learning. In *Proceedings of The 2nd Conference on Robot Learning (CoRL)* (PMLR), vol. 87, 362372

Candon, K., Chen, J., Kim, Y., Hsu, Z., Tsoi, N., and Vázquez, M. (2023). Nonverbal human signals can help autonomous agents infer human preferences for their behavior. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, (AAMAS)*

Carpinella, C. M., Wyman, A. B., Perez, M. A., and Stroessner, S. J. (2017). The robotic social attributes scale (rosas): Development and validation. In *12th ACM/IEEE International Conference on Human Robot Interaction (HRI)*

Cui, Y., Karamcheti, S., Palleti, R., Shivakumar, N., Liang, P., and Sadigh, D. (2023). No, to the right online language corrections for robotic manipulation via shared autonomy. In *Proceedings of ACM/IEEE Conference on Human Robot Interaction (HRI)*

Cui, Y., Koppol, P., Admoni, H., Niekum, S., Simmons, R., Steinfeld, A., et al. (2021a). Understanding the relationship between interactions and outcomes in humanintheloop machine learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*

Cui, Y. and Niekum, S. (2018). Active reward learning from critiques. In *IEEE international conference on robotics and automation (ICRA)*

Cui, Y., Zhang, Q., Knox, B., Allievi, A., Stone, P., and Niekum, S. (2021b). The empathic framework for task learning from implicit human feedback. In *Conference on Robot Learning (CoRL)*

Ghosal, G. R., Zurek, M., Brown, D. S., and Dragan, A. D. (2023). The effect of modeling human rationality level on learning rewards from multiple feedback types. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*

Hadfield Menell, D., Milli, S., Abbeel, P., Russell, S. J., and Dragan, A. (2017). Inverse reward design. *Advances in Neural Information Processing Systems (NeurIPS)*

Hart, S. G. and Staveland, L. E. (1988). Development of nasatlx (task load index): Results of empirical and theoretical research. *Advances in psychology*

Huang, J., Aronson, R. M., and Short, E. S. (2024). Modeling variation in human feedback with user inputs: An exploratory methodology. In *Proceedings of ACM/IEEE International Conference on Human Robot Interaction (HRI)*

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. (2018). Reward learning from human preferences and demonstrations in atari. *Advances in Neural Information Processing Systems (NeurIPS)*

Jeon, H. J., Milli, S., and Dragan, A. (2020). Reward rational (implicit) choice: A unifying formalism for reward learning. *Advances in Neural Information Processing Systems (NeurIPS)*

Ji, J., Zhang, B., Zhou, J., Pan, X., Huang, W., Sun, R., et al. (2023). Safety gymnasium: A unified safe reinforcement learning benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*

Kim, C., Seo, Y., Liu, H., Lee, L., Shin, J., Lee, H., et al. (2023). Guide your agent with adaptive multimodal rewards. In *Thirty-seventh Conference on Neural Information Processing Systems*

[Dataset] Kinova (2025). Kinova gen3 ultra lightweight robot. Accessed: 20251013

Krakovna, V., Orseau, L., Martic, M., and Legg, S. (2018). Measuring and avoiding side effects using relative reachability. *arXiv preprint arXiv:1806.01186*

Krakovna, V., Orseau, L., Ngo, R., Martic, M., and Legg, S. (2020). Avoiding side effects by considering future tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*

Lakkaraju, H., Kamar, E., Caruana, R., and Horvitz, E. (2017). Identifying unknown unknowns in the open world: Representations and policies for guided exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*

Losey, D. P. and O'Malley, M. K. (2018). Including uncertainty when learning from human corrections. In *Conference on Robot Learning (CoRL)*

Najar, A. and Chetouani, M. (2021). Reinforcement learning with human advice: A survey. *Frontiers in Robotics and AI* Volume 8 - 2021. doi:10.3389/frobt.2021.584075

Ng, A. Y., Russell, S., et al. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*

Ramachandran, D. and Amir, E. (2007). Bayesian inverse reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI)*

Ramakrishnan, R., Kamar, E., Dey, D., Horvitz, E., and Shah, J. (2020). Blind spot detection for safe sim-to-real transfer. *Journal of Artificial Intelligence Research (JAIR)*

Ross, S., Gordon, G., and Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, (AISTATS)*

Saisubramanian, S., Kamar, E., and Zilberstein, S. (2021a). A multiobjective approach to mitigate negative side effects. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI)*

Saisubramanian, S., Kamar, E., and Zilberstein, S. (2022). Avoiding negative side effects of autonomous systems in the open world. *Journal of Artificial Intelligence Research (JAIR)*

Saisubramanian, S., Roberts, S. C., and Zilberstein, S. (2021b). Understanding user attitudes towards negative side effects of AI systems. In *Extended Abstracts of the 2021 Conference on Human Factors in Computing Systems (CHI)*

Saisubramanian, S. and Zilberstein, S. (2021). Mitigating negative side effects via environment shaping. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*

Saran, A., Zhang, R., Short, E. S., and Niekum, S. (2021). Efficiently guiding imitation learning agents with human gaze. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*

Settles, B. (1995). Active learning literature survey. *Science*

Sontakke, S. A., Zhang, J., Arnold, S., Pertsch, K., Biyik, E., Sadigh, D., et al. (2023). RoboCLIP: One demonstration is enough to learn robot policies. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*

Srivastava, A., Saisubramanian, S., Paruchuri, P., Kumar, A., and Zilberstein, S. (2023). Planning and learning for non-markovian negative side effects using finite state controllers. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*

Strokina, N., Yang, W., Pajarinen, J., Serbenyuk, N., Kämäräinen, J., and Ghabcheloo, R. (2022). Visual rewards from observation for sequential tasks: Autonomous pile loading. *Frontiers in Robotics and AI* Volume 9 - 2022. doi:10.3389/frobt.2022.838059

Tien, J., He, J. Z., Erickson, Z., Dragan, A., and Brown, D. S. (2023). Causal confusion and reward misidentification in preferencebased reward learning. In *The Eleventh International Conference on Learning Representations (ICLR)*

Xu, D., Agarwal, M., Fekri, F., and Sivakumar, R. (2020). Playing games with implicit human feedback. In *Workshop on Reinforcement Learning in Games, (AAAI)*

Yang, Z., Jun, M., Tien, J., Russell, S., Dragan, A., and Biyik, E. (2024). Trajectory improvement and reward learning from comparative language feedback. In *Conference on Robot Learning (CoRL)*

Zhang, S., Durfee, E., and Singh, S. (2020). Querying to find a safe policy under uncertain safety constraints in markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*