The EMPATHIC Framework for Task Learning from Implicit Human Feedback

Yuchen Cui[1*], **Qiping Zhang**[1*], **Alessandro Allievi**[1,2],
Peter Stone[1], **Scott Niekum**[1], **W. Bradley Knox**[1,2]
[1]The University of Texas at Austin, Austin, TX
[2]Robert Bosch LLC, Austin, TX
{yuchencui, qpzhang}@utexas.edu, {pstone, sniekum}@cs.utexas.edu
{brad.knox, alessandro.allievi}@us.bosch.com

Abstract: Reactions such as gestures, facial expressions, and vocalizations are an abundant, naturally occurring channel of information that humans provide during interactions. A robot or other agent could leverage an understanding of such *implicit* human feedback to improve its task performance at no cost to the human. This approach contrasts with common agent teaching methods based on demonstrations, critiques, or other guidance that need to be attentively and intentionally provided. In this paper, we first define the general problem of learning from implicit human feedback and then propose to address this problem through a novel data-driven framework, EMPATHIC. This two-stage method consists of (1) mapping implicit human feedback to relevant task statistics such as reward, optimality, and advantage; and (2) using such a mapping to learn a task. We instantiate the first stage and three second-stage evaluations of the learned mapping. To do so, we collect a dataset of human facial reactions while participants observe an agent execute a sub-optimal policy for a prescribed training task. We train a deep neural network on this data and demonstrate its ability to (1) infer relative reward ranking of events in the training task from prerecorded human facial reactions; (2) improve the policy of an agent in the training task using live human facial reactions; and (3) transfer to a novel domain in which it evaluates robot manipulation trajectories.

Keywords: Interactive Learning, Learning from Human Feedback

1 Introduction

People often react when observing an agent—whether human or artificial—if they are interested in the outcome of the agent's behavior. We have scowled at robot vacuums, raised eyebrows at cruise control, and rebuked automatic doors. Such reactions are often not intended to communicate to the agent and yet nonetheless contain information about the perceived quality of the agent's performance. A robot or other software agent that can sense and correctly interpret these reactions could use the information they contain to improve its learning of the task. Importantly, learning from such *implicit* human feedback does not burden the human, who naturally provides such reactions even when learning does not occur. We view learning from implicit human feedback (LIHF) as complementary to learning from explicit human teaching, which might take the form of demonstrations [1], evaluative feedback [2, 3], or other communicative modalities [4, 5, 6, 7, 8]. Though we expect implicit feedback to typically be less informative in a fixed amount of time than explicit alternatives and perhaps more difficult to interpret correctly, LIHF has the advantage of using already available reactions and therefore induces no additional cost to the user. The goal of this work is to frame the LIHF problem, propose a broad data-driven framework to solve it, and implement and validate an instantiation of this framework using specific modalities of human reactions: facial expressions and head poses (henceforth referred to jointly as **facial reactions**).

[*] Equally contributing authors

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.

Existing computer vision research has shown success in recognizing basic human facial expressions [9, 10, 11]. However, it is not trivial for a learning agent to interpret human expressions. For example, a smile could mean satisfaction, encouragement, amusement, or frustration [12]. Different interpretation of the same facial expression could result in very different learning behaviors. Recent progress in cognitive science also provides a utilitarian view of facial expressions and suggests that they are also used as tools for regulating social interactions and signaling contingent social action; therefore the interpretation of facial expressions may vary from context to context and from person to person [13, 14, 15, 16]. Further, human reactions often have a variable delay after an event or occur in anticipation of an event, posing an additional challenge of interpreting which (series of) action(s) or event(s) the person is reacting to. Additionally, many natural human reactions involve spontaneous micro-expressions consisting of minor facial muscle movements that last for less than 500 milliseconds [17, 18], which can be hard to detect by computer vision systems trained with common datasets with only exaggerated or acted facial expressions [19, 20]. Lastly, human environments often contain more than the agent and its task environment, and therefore inferring what a person is reacting to at any moment adds further difficulty.

We approach LIHF with data-driven modeling that creates a general **reaction mapping** from implicit human feedback to task statistics. The major contributions of this paper are:

1. We motivate and frame the general problem of Learning from Implicit Human Feedback (LIHF), which aims at leveraging under-utilized data modality that already exists in natural human-robot interactions. This problem is different from traditional interactive robot learning settings that put human and robot in explicit pedagogical settings.
2. We propose a general framework to solve this problem, called Evaluative MaPping for Affective Task-learning via Human Implicit Cues (EMPATHIC), which consists of two stages: (1) learning a mapping from implicit human feedback to known task statistics and (2) using such a mapping to learn a task from implicit human feedback.
3. We experimentally validate an instantiation of the EMPATHIC framework, using human facial reactions as the implicit feedback modality, and rewards as target task statistic:
   - We develop an experimental procedure for collecting data of human reactions to an autonomous agent's behavior. The dataset is recorded while *human observers* watch an autonomous agent performing a task that determines their financial payout. We refer to such tasks as the **training tasks**.
   - We analyze the modeling problem through a *human proxy test*: the authors act as proxies for a reaction mapping by watching the reactions of the human observers and then ranking semantically anonymized events by their inferred reward, which we refer to as the *reward-ranking task*. Moderate success at this human proxy test provides confidence that human reactions could inform an understanding of reward values. This activity also provides critical insight regarding which reaction features are helpful for modeling.
   - Our instantiation of EMPATHIC learns a **reaction mapping** from a proximate time window of human reactions to a probability distribution over reward values. The mapping is learned by using a pre-trained model to extract facial reaction features from video data and training a deep neural network via supervision to predict rewards with the extracted features.
   - We compare the performance of the learned reaction mapping and a random baseline on the reward-ranking task. We also show an initial evaluation of learning the training task *online*, in which an agent updates its belief over possible reward functions from live human reactions and improves its policy in real time.
   - We transfer the learned reaction mapping to a **deployment task**, providing a proof-of-concept of the potential for reaction mappings to generalize across tasks. Specifically, the reaction mapping trained with data from the training task is used to evaluate and rank trajectories from a robotic sorting task.

2 Related Work

Our work relates closely to the growing literature of *interactive reinforcement learning (RL)*, or human-centered RL [2, 21, 22, 23, 24, 25, 26, 27, 28, 29] , in which agents learn from interactions with humans in addition to, or instead of, predefined *environmental* rewards. In the EMPATHIC framework, we use the term *implicit* human feedback to refer to any multi-modal evaluative signals humans naturally emit during social interactions, including facial expressions, tone of voice, head gestures, hand gestures, and other body-language and vocalization modalities not aimed at explicit communication. Others' usage of "implicit feedback" has referred to the *implied* feedback when a human refrains from giving explicit feedback [30, 31], to human biomagnetic signals [32], or to facial expressions [33, 34, 35]. This work focuses on predicting task statistics from human facial features and therefore is also related to the broad area of research on *facial expression recognition*.

Interactive Reinforcement Learning Inspired by clicker-training for animals, the TAMER framework proposed by Knox et al. [2, 3] is the first to explicitly model human feedback in the form of button clicks, thus allowing RL agents to learn from human feedback signals without any access to environmental rewards. Veeriah et al. [36] propose learning a value function grounded only in the user's facial expressions and agent actions, using manual negative feedback as supervision. The corresponding RL agent's policy is only a function of the trainer's facial expression and does not reason about the task state. In the preliminary work of Arakawa et al. [34], the authors adopt an existing facial expression classification system to detect human emotions and use a predefined mapping from emotions to TAMER feedback but do not optimize the mapping to be effective for the downstream task. Similarly, recent work of Zadok et al. [37] models the probability of human demonstrators smiling within a task and then biases an RL agent's behavior to increase the predicted probability of human smiling, improving exploration. Li et al. [28] extend TAMER by interpreting the trainer's facial expressions as positive or negative feedback with a deep neural network. Their results suggest it is possible to learn solely from facial expressions of the trainer. Our proposed method differs from prior work through learning a direct mapping from facial reactions to task statistics independent of states or state-actions, which requires no explicit human feedback at either training or testing time. Our system is the first, to the best of our knowledge, attempting to learn from subjects that are not explicitly told to teach or react.

Facial Expression Recognition (FER) The field of facial expression recognition contains a rich body of research from areas of psychology, neuroscience, cognitive science and computer vision. Fasel and Luettin [10] provide an overview of traditional FER systems and Li and Deng [11] detail recent FER systems based on deep neural networks. Our proposed method does not perform FER explicitly but maps extracted facial features to reward values. Our work is closely related to the problem of dynamic FER, where time-series data are used as input for temporal predictions. Modern FER systems often consist of two stages: data pre-processing and predictive modeling with deep networks [11]. Inspired by techniques from the FER literature, our proposed system leverages an existing toolkit [38, 39, 40] to extract facial features that are sufficiently informative for modeling despite our small dataset, and we explicitly model the temporal aspect of the problem by further extracting features in the frequency domain.

3 The LIHF Problem and The EMPATHIC Framework

Markov Decision Processes (MDPs) are often used to model sequential decision making of autonomous agents. An MDP is given by the tuple $\langle S, A, T, R, d_0, \gamma \rangle$, where: S is a set of states; A is a set of actions an agent can take; $T : S \times A \times S \to [0, 1]$ is a probability function describing state transition based on actions; $R : S \times A \times S \to \mathbb{R}$ is a real-valued reward function; d_0 is a starting state distribution and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi : S \times A \to [0, 1]$ maps from any state and action to a probability of choosing that action. The goal of an agent is to find a policy that maximizes the expected return $E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$ where r_t is the reward at time step t.

The problem of **Learning from Implicit Human Feedback** (LIHF) asks how an agent can learn a task with information derived from human reactions to its behavior. LIHF can be described by the tuple $\langle S, A, T, R^{\mathcal{H}}, X^{\mathcal{H}}, \Xi, d_0, \gamma \rangle$. S, A, T, d_0, and γ are defined identically as in MDPs. The agent receives observations from implicit feedback modalities asynchronously with respect to time steps, and each such observation $x \in X^{\mathcal{H}}$ contains implicit feedback from some human $\mathcal{H}$. An observation function Ξ denotes the conditional probability over $X^{\mathcal{H}}$ of observing x, given a trajectory of states and actions and the human's hidden reward function $R^{\mathcal{H}}$. States in LIHF are generally broader than task states, and include all environmental and human factors that influence the conditional probability of observing x. The goal of an agent is to maximize the return under $R^{\mathcal{H}}$. How to ground observations $x \in X^{\mathcal{H}}$ containing implicit human feedback to evaluative task statistics is at the core of solving LIHF.

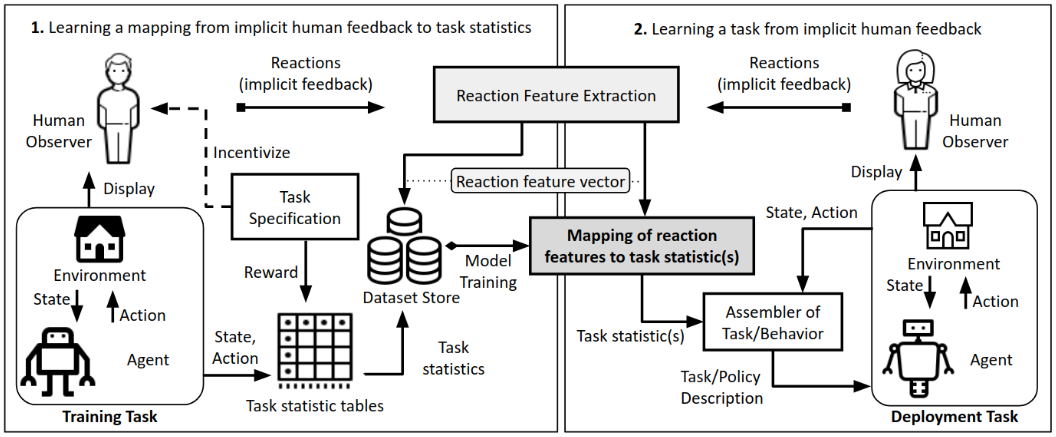


Figure 1: Overview of EMPATHIC

The formulation of LIHF resembles the definition of Partially Observable MDPs, but here the partially observable variable is the human's reward function rather than state. We include a graphical model in Appx.A that describes how LIHF models the data generation process.

We propose a data-driven solution to the LIHF problem that infers relevant task statistics from human reactions. As Fig. 1 shows, the EMPATHIC framework has two stages: (1) learning a mapping from implicit human feedback to relevant task statistics and (2) using such a mapping to learn a task. In the first stage, human observers are incentivized to want an agent to succeed—to align the person's $R^{\mathcal{H}}$ with a known task reward function R—and they are then recorded while observing the agent. Task statistics are computed from R for every timestep to serve as supervisory labels, which train a mapping from synchronized recordings of the human observers to these statistics. Task state and action are *not* inputs to the reaction mapping, allowing it to be deployed to other tasks. In the second stage, a human observes an agent attempt a task with sparse or no environmental reward, and the human observer's reaction to its behavior is mapped to otherwise unknown task statistics to improve the agent's policy, either directly or through other usage of the task statistics, such as guiding exploration or inferring the reward function $R^{\mathcal{H}}$ that describes the human's utility. In general, any instantiation of EMPATHIC can be achieved through specifying these elements:

- the reaction modality and the target task statistic(s);
- the end-user population(s);
- training task(s) for stage 1 and deployment task(s) for stage 2;
- an incentive structure for stage 1 to align human interests with task performance; and
- policies or RL algorithms to control the observed agent in both stages.

Any specific task or person can optionally be part of both stages. Note that EMPATHIC is defined broadly enough to include instantiations with varying degrees of personalization—from learning a single reaction mapping applicable to all humans to training a person-specific model—and of across-task generalization. We hypothesize that a single reaction mapping will be generally useful but that training to specific users or tasks will yield even more effective mappings. Such personalized training may also guard against negative effects of potential dataset bias from the first stage of EMPATHIC if it is used amongst underserved populations.

This paper presents one instantiation of EMPATHIC, using facial reactions as the modality for implicit human feedback. Sections 4 and 5 provide the instantiation details.

4 Data Collection Domains and Protocol

In this section we describe the experimental domains and data collection process of our instantiation of EMPATHIC.

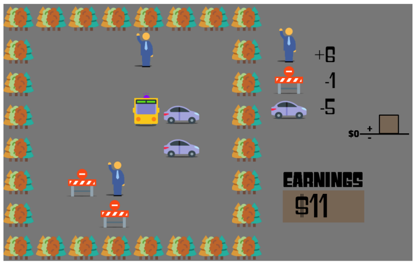


Figure 2: *Robotaxi* environment

Robotaxi We create *Robotaxi* as a simulated domain to collect implicit human feedback data with known task statistics. Fig. 2 shows the visualization viewed by the human observer. An agent (depicted as a yellow bus) acts in a grid-based map. Rewards are connected to objects: $+6$ for picking up a passenger; -1 for crashing into a roadblock; and -5 for crashing into a parked car. Reward is 0 otherwise. An object disappears after the agent moves onto it, and another object of the same type is spawned with a short delay at a random unoccupied location. An episode starts with two objects of each type.

Robotic Sorting Task A robotic manipulation task is a deployment domain for test transfer of the learned reaction mapping across task domains. The physical setup of the task is shown in Fig. 3. The robot's task is to sort the aluminum cans into the recycling bin. Reward is $+2$ upon recycling a can, -1 upon recycling any other object, and 0 at all other times. The episodes are short (< 20 seconds), containing predetermined trajectories with at most a single non-zero reward event. Further details are in Appx.B.

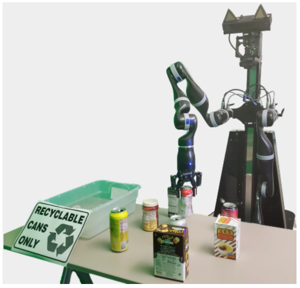


Figure 3: Robotic sorting task

Data Collection We recruited participants to interact with autonomous agents in both tasks. Before human participants observed the agents executing a task, they were informed that their financial compensation for the study would be proportional to the agent's earnings. The payment structure creates a direct mapping between the ground-truth reward label and its financial value to the human subject, intending to align human interests with the task and therefore connecting their reactions to task statistics. To minimize explicit feedback (i.e., intended to influence the agent), participants were told that their "reactions are being recorded for research purposes", and nothing more was said regarding our intended usage of their reactions. This experimental setup contrasts with prior related work [28, 34, 36], in which human participants were explicitly asked to teach with their facial expressions, and aligns with a key motivation for the LIHF problem, which is to leverage data that is already being generated in existing human-agent interactions. 17 human participants observed 3 episodes of *Robotaxi*, and 14 of the participants observed 7 episodes of the robotic task. Experiments occurred in an isolated room and videos were recorded as the human participants watched the agents execute suboptimal behavior trajectories that were predefined. All data collection was conducted after obtaining the participant's consent and the participants were debriefed at the end of their sessions. See Appx.B for further details.

5 Reaction Mapping Design

Human Exploration of the Data To better understand the task of training a reaction mapping, the authors serve as proxies for a mapping. Specifically, we view a semantically anonymized version of each agent trajectory alongside a synchronized recording of the human participant's reactions; after this viewing, we attempt to rank the reward values of the 3 object types. Fig. 4 shows the interface. Each human proxy watched one truncated episode from each of the 17 participants. To measure performance, Kendall's rank correlation coefficient $\tau \in [-1, 1]$ [41] is used to compare a human proxy's inferred ranking with ground truth (a higher τ value indicates a higher correlation between two rankings). Table 1 shows mean τ scores for the human proxy test across 17 participants, with a mean for each author. Wilcoxon signed-rank tests [42] compare each human proxy's 17 τ scores with the expected value $\tau = 0$ for uniformly random reward ranking, and corresponding p-values are also in Table 1. In this test, 5 out of 6 humans outperformed random ranking, and 1 human author did so significantly even after adjusting a $p < 0.05$ threshold for multiple testing to $p < 0.0083$ using a Bonferroni correction [43]. This person's success suggests that the reactions contain sufficient information to rank object rewards, though humans vary in their ability to harness the information. With our experience as proxies for the reaction mapping, we identify 7 common reaction gestures

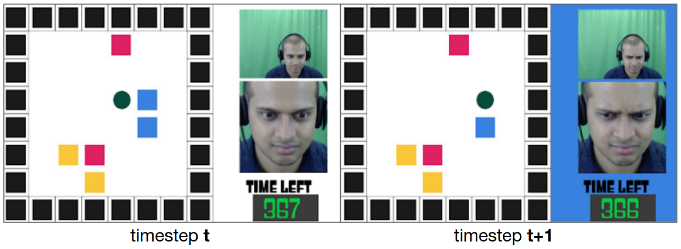


Figure 4: Human proxy's view: semantics are hidden with color masks; the dark green circle is the agent; observer's reaction is displayed; detected face is enlarged; background is colored by last pickup. The left frame is the same game state shown in Fig 2.

	Avg. τ	p-value
Human Proxies	.569	.004
	.216	.185
	.098	.319
	-.176	.179
	.255	.123
	.294	.059
Avg.	.209	.078

Table 1: Human proxy test result: average τ values across participants are displayed; a baseline that randomly picks rankings has an expected τ value of 0.

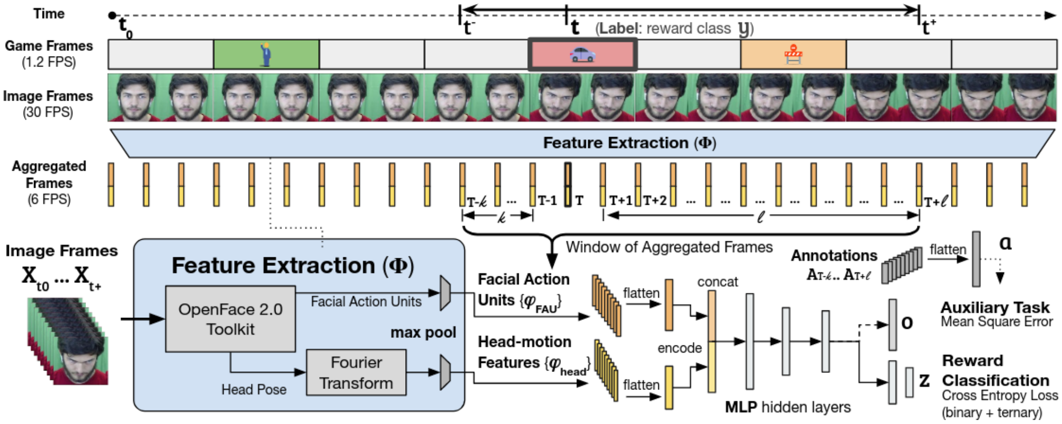


Figure 5: The feature extraction pipeline and architecture of the reaction mapping

that helped us infer reward rankings: smile, pout, eyebrow-raise, eyebrow-frown, (vertical) head nod, head shake, and eye-roll. The collected video data was annotated with frame onsets and offsets of these 7 gestures as well as the general positive, negative, or neutral sentiment of the gesture. The corresponding trajectories were not viewed during annotation. Appx.C contains a detailed analysis of the annotations, which informed our model design.

Reaction Mapping Architecture To demonstrate that the implicit feedback signal in human facial reactions can be computationally modeled, we construct a reaction mapping that takes a temporal series of extracted features as input and outputs a probability distribution over reward classes. We extract facial features from video data with a pre-trained model and train a deep neural network on predicting rewards with the extracted features in a supervised way.[1] The feature extraction pipeline and architecture of the proposed deep network model is shown in Fig. 5. OpenFace 2.0 [38, 39, 40] is used to extract features from raw videos of human reactions. Raw videos consist of 30 image frames per second. For each image frame, OpenFace extracts head pose and activation of facial action units (FAUs). For detecting head nods and shakes, we explicitly model the head-pose changes by keeping a running average of extracted head-pose features and subtract it from each incoming feature vector. Frequencies of changes in head-pose are then computed through fast Fourier transform, and the coefficients of frequencies are used as head-motion features. To allow the series of input features to cover a large enough temporal window of reactions, feature vectors of consecutive image frames are combined through max pooling of each dimension, resulting in temporally aggregated feature vectors of the same size. Refer to Appx.D for full details about feature extraction.

Let $\{X_{t_0}, ..., X_t\}$ denote the sequence of raw input image frames from time t_0 to t. Time t_0 is the start of an episode, and t is the time of the last image frame for the T-th aggregated frame being calculated. Aggregated FAU features $\varphi_{FAU} \in \mathbb{R}^m$ and head-motion features $\varphi_{head} \in \mathbb{R}^n$ are extracted by the feature extractor Φ: $(\varphi_{FAU}, \varphi_{head})_T = \Phi(\{X_{t_0}, ..., X_t\})$. A window of consecutive aggregated frame features is used as input for a data sample, which is labeled with the reward category (i.e., -5, -1, or $+6$) received during the time step containing the T-th aggregated frame. The window of aggregated frames begins at the (T-k)-th and ends at the (T+ℓ)-th aggregated frame. Since some reactions happen after an event, future data is needed to make a prediction for the current event; hence the prediction has a fixed time delay defined by the window. FAU features and the head-motion features are encoded separately: the temporal series for each is flattened into a single vector and then encoded with a linear layer. The two encodings are then concatenated into a single vector, which is input to a multilayer perceptron (MLP). We include an auxiliary task of predicting the corresponding annotations $\{A_{(T-k)}, ..., A_{(T+\ell)}\}$ as a single flattened vector $\boldsymbol{a} \in \{0, 1\}^{10(k+\ell+1)}$, in which each binary element of A indicates whether a reaction gesture is occurring. This auxiliary task is intended to speed representation learning and act as a regularizer. Empirically, use of this auxiliary task achieves the best test loss but is unnecessary for better-than-random performance in the reward-ranking task (see Section 6). We also use a binary classification loss that combines

[1] Our proposed approach could instead model other task statistics or be trained end-to-end with a convolutional neural network (removing the feature extraction module). *For this dataset*, however, modeling non-zero reward classes with a pre-trained feature extractor is empirically more effective than either of these strategies. More details can be found in Appx.E.3 and K.

the two negative reward classes as one, which reintroduces the ordinality of the reward classes by additionally penalizing predictions with the wrong sign. Let $g_\theta(\cdot)$ represent the MLP-based network, $\boldsymbol{z} \in \mathbb{R}^c$ be the output (unnormalized log probabilities of the c classes with a corresponding ground-truth one-hot label $\boldsymbol{y} \in \{0, 1\}^c$), and $\boldsymbol{o}$ denote the output of the auxiliary task. $\boldsymbol{y}_{bin}$ is the ground-truth binary class, and $\boldsymbol{z}_{bin}$ denotes the corresponding binary prediction computed from $\boldsymbol{z}$. Therefore, $(\boldsymbol{z}, \boldsymbol{o})_T = g_\theta(\{(\varphi_{FAU}, \varphi_{head})_{T-k}, ..., (\varphi_{FAU}, \varphi_{head})_{T+\ell}\})$.

The loss to be optimized is then expressed as:

$$\mathcal{L}(\theta) = -\boldsymbol{y} \cdot \log(\text{softmax}(\boldsymbol{z})) - \lambda_1 \boldsymbol{y}_{bin} \cdot \log(\text{softmax}(\boldsymbol{z}_{bin})) + \lambda_2 ||\boldsymbol{a} - \boldsymbol{o}||_2$$

The neural network is trained with Adam [44]. We employ random search [45] to find the best set of hyper-parameters to use, including the input's window size (k and ℓ), learning rate, dropout rate, loss coefficients (λ_1 and λ_2), and the depth and widths of the MLP. The set of candidate window sizes for random search was informed by ad hoc analysis of the annotations of high-level human facial reactions (Appx.C). Since our dataset is small, we employ k-fold cross validation for the random search of hyper-parameters after randomly sampling one episode of data from each subject into a *holdout* set for final evaluation. Each set of randomly sampled parameters is evaluated across train-test data folds, and the set with the lowest average test loss is selected. Details of the random search process and an ablation study of the reaction mapping design can be found in Appx.E.

6 Results and Evaluation

To validate that the learned mappings from our instantiation of stage 1 effectively enable task learning in stage 2, we test the following hypotheses (in which we refer to observers from stage 1 who have created data in the training set as "*known subjects*"):

Hypothesis 1 [deployment setting is the same as training setting]. The learned reaction mappings will outperform uniformly random reward ranking, using reaction data from *known subjects* watching the *Robotaxi* task.

Hypothesis 2 [generalizing **H1** to online data from novel subjects]. The learned reaction mappings will improve the online policy of a *Robotaxi* agent via updates to its belief over reward functions, based on *online* data from *novel* human observers;

Hypothesis 3 [generalizing **H1** to a different deployment task]. The learned reaction mappings can be adapted to evaluate robotic-sorting-task trajectories and will outperform uniformly random guessing on return-based rankings of these trajectories, using reaction data from *known subjects*.

Reward-ranking Performance in the Robotaxi Domain The learned reaction mappings are evaluated on the reward-ranking task. Let q be the random variable for reward event and x be the variable for human reactions. Let m be the discrete random variable over possible reward functions, which in the *Robotaxi* task can be considered a reward ranking. The model $g_\theta(\cdot)$ effectively models $\text{P}(q \mid x, m)$, which is the probability of an event given the human's reaction and a fixed reward ranking m. The goal is to find the posterior distribution over m: $\text{P}(m \mid q, x) \propto \text{P}(q \mid x, m)\text{P}(m)$ (see proof in Appx.F). Given a uniform prior over m, we can find $\text{P}(m \mid q, x)$ using prediction of the mapping $g_\theta(\cdot)$. The maximum a posteriori reward ranking is chosen as the learned mapping's single estimation after incorporating mappings from all human reaction data in an episode. To reduce the effect of stochasticity in training neural networks, we train 4 times and report the mean performance. Fig. 6 shows the learned reaction mappings' (per-subject) performance on the episodes in

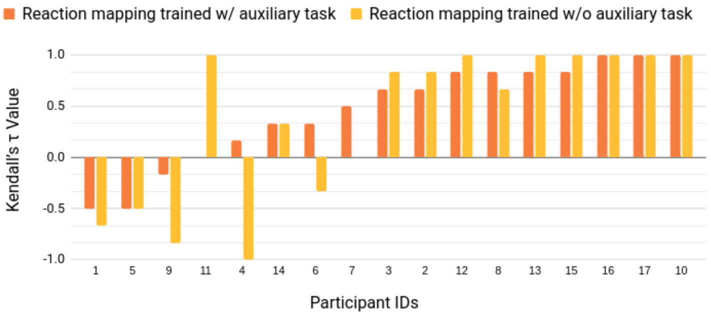


Figure 6: Sorted per-subject Kendall's τ for *Robotaxi* reward-ranking task

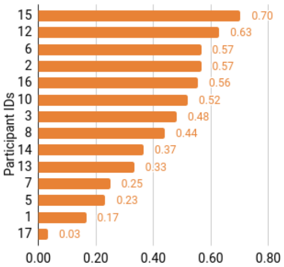


Figure 7: Sorted per-subject Kendall's τ for evaluating robotic sorting trajectories

the *holdout* test. Using Wilcoxon Signed-Rank test, the mappings' performance on the *holdout* set is significantly better than uniformly random guessing ($\tau = 0$), supporting **H1**; $p = 0.0024$ with the annotation-reliant auxiliary task and $p = 0.0207$ without it.

Online Learning in the Robotaxi Domain The learned reaction mapping can interactively improve an agent's policy: the agent updates its belief over all possible reward rankings using human reactions to its recent behavior and then follows a policy that is approximately optimal with respect to the most likely reward function. To test such online policy learning, all data collected in stage 1 trains a single reaction mapping, and this reaction mapping is used for single-episode sessions with human observers, none of whom created data within the stage-1 training set. 9 of the 10 participants' interactions achieved a better return than that of a random policy, and 7 of the 10 participants' interactions ended with the probability of reward mappings that lead to optimal behaviors being the highest, moderately supporting **H2**. Details of this preliminary evaluation can be found in Appx.J.

Trajectory Ranking in Robotic Sorting Domain To generalize the reaction mapping trained in the *Robotaxi* domain to the robotic sorting task, we modify the original loss function by removing the multi-class reward classification loss and interpret the reaction mapping's binary output as a "positivity score" for an aggregated frame. Each human participant observed 7 trajectories (an episode each), chosen from 8 distinct predetermined trajectories. Each trajectory accrues return of $+2$ (recycling a can); -1 (recycling any other object); or 0 (nothing placed in the bin). This return enables ground-truth rankings of trajectories. Because we suspect humans react to higher-level actions in this task—to *pick and place object X* rather than to the joint torques applied at 25 ms time steps—the window size of the *Robotaxi* reaction mapping is too small to contain all relevant facial reactions. To address this apparent temporal incompatibility, we compute a per-trajectory positivity score as the mean of the positivity scores of its aggregate frames. A derivation of this approach is in Appx.L with further details of the trajectory design. Fig. 7 shows Kendall's τ values for per-participant rankings of trajectories. For each trajectory, we compute an overall (cross-subject) positivity score as the mean of the trajectory's per-subject positivity scores. After ranking the 8 trajectories by these scores, Kendall's τ independence test yields $\tau = 0.70$ ($p = 0.034$); this test implicitly compares to uniformly random guessing, since its $\tau = 0$. This result supports **H3**.

7 Discussion and Conclusion

In this paper we introduce the LIHF problem and the EMPATHIC framework for LIHF. We demonstrate that our instantiation interprets human facial reactions in both the training task and the deployment task. We now discuss the limitations of this work and directions for future investigation.

Experimental Design We validate our instantiation of EMPATHIC with a single training task and similar testing tasks. An important future extension is to generalize this method to tasks with varying temporal characteristics and reward structures. In our current setup, agent actions do not have large long-term consequences on the expected return, however changes in human expectations could significantly affect their reactions. One way to incorporate such information into our current modeling approach is to craft corresponding task environments to explore the use of human facial reactions in predicting the long-term returns of agent behaviors.

Human Models Data collected in this work allow us to study reactions of human observers who fix their attention on the agent, whereas in real-world settings human observers are often attending to their own tasks. A natural next step is to extend our experiment setup to a more general scenario, in which we also need to infer the relevance of human reactions to the agent's behavior. Additionally, our instantiation assumes that human reactions were influenced by recent and anticipated agent experience but not by other likely factors, such as changing expectations of agent behavior; explicitly modeling such latent human state may further improve LIHF.

Data Modalities This work maps from facial reactions to discrete rewards. In future work, other forms of human implicit feedback, such as gaze and gestures, could be included to get a more accurate mapping to different task statistics and better performance in a variety of real-world tasks.

The above limitations notwithstanding, this paper takes a significant step towards the goal of enabling an agent to learn a task from implicit human feedback. It does so by successful application of a learned mapping from human facial reactions to reward types for online agent learning and for evaluating trajectories from a different domain.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[2] W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16. ACM, 2009.

[3] W. B. Knox, P. Stone, and C. Breazeal. Training a robot via human feedback: A case study. In *International Conference on Social Robotics*, pages 460–470. Springer, 2013.

[4] S. Chernova and A. L. Thomaz. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121, 2014.

[5] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, 2017.

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

[7] O. Kroemer, S. Niekum, and G. Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *arXiv preprint arXiv:1907.03146*, 2019.

[8] H. Admoni and B. Scassellati. Social eye gaze in human-robot interaction: a review. *Journal of Human-Robot Interaction*, 6(1):25–63, 2017.

[9] P. Ekman. Facial expressions. *Handbook of cognition and emotion*, 16(301):e320, 1999.

[10] B. Fasel and J. Luettin. Automatic facial expression analysis: a survey. *Pattern recognition*, 36(1): 259–275, 2003.

[11] S. Li and W. Deng. Deep facial expression recognition: A survey. *arXiv preprint arXiv:1804.08348*, 2018.

[12] M. E. Hoque, D. J. McDuff, and R. W. Picard. Exploring temporal patterns in classifying frustrated and delighted smiles. *IEEE Transactions on Affective Computing*, 3(3):323–334, 2012.

[13] J. Panksepp and D. Watt. What is basic about basic emotions? lasting lessons from affective neuroscience. *Emotion review*, 3(4):387–396, 2011.

[14] C. Crivelli and A. J. Fridlund. Facial displays are tools for social influence. *Trends in Cognitive Sciences*, 22(5):388–399, 2018.

[15] R. E. Jack and P. G. Schyns. The human face as a dynamic tool for social communication. *Current Biology*, 25(14):R621–R634, 2015.

[16] M. N. Dailey, C. Joyce, M. J. Lyons, M. Kamachi, H. Ishi, J. Gyoba, and G. W. Cottrell. Evidence and a computational explanation of cultural differences in facial expression recognition. *Emotion*, 10(6):874, 2010.

[17] T. Pfister, X. Li, G. Zhao, and M. Pietikäinen. Recognising spontaneous facial micro-expressions. In *2011 international conference on computer vision*, pages 1449–1456. IEEE, 2011.

[18] W.-J. Yan, Q. Wu, J. Liang, Y.-H. Chen, and X. Fu. How fast are the leaked facial expressions: The duration of micro-expressions. *Journal of Nonverbal Behavior*, 37(4):217–230, 2013.

[19] X. Li, T. Pfister, X. Huang, G. Zhao, and M. Pietikäinen. A spontaneous micro-expression database: Inducement, collection and baseline. In *2013 10th IEEE International Conference and Workshops on Automatic face and gesture recognition (fg)*, pages 1–6. IEEE, 2013.

[20] A. K. Davison, W. Merghani, and M. H. Yap. Objective classes for micro-facial expression recognition. *Journal of Imaging*, 4(10):119, 2018.

[21] C. Isbell, C. R. Shelton, M. Kearns, S. Singh, and P. Stone. A social reinforcement learning agent. In *Proceedings of the fifth international conference on Autonomous agents*, pages 377–384. ACM, 2001.

[22] P. M. Pilarski, M. R. Dawson, T. Degris, F. Fahimi, J. P. Carey, and R. S. Sutton. Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7. IEEE, 2011.

[23] H. B. Suay and S. Chernova. Effect of human guidance and state space size on interactive reinforcement learning. In *2011 Ro-Man*, pages 1–6. IEEE, 2011.

[24] G. Warnell, N. Waytowich, V. Lawhern, and P. Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[25] G. Li, R. Gomez, K. Nakamura, and B. He. Human-centered reinforcement learning: A survey. *IEEE Transactions on Human-Machine Systems*, 2019.

[26] R. Zhang, F. Torabi, L. Guan, D. H. Ballard, and P. Stone. Leveraging human guidance for deep reinforcement learning tasks. *arXiv preprint arXiv:1909.09906*, 2019.

[27] J. Lin, Z. Ma, R. Gomez, K. Nakamura, B. He, and G. Li. A review on interactive reinforcement learning from human social feedback. *IEEE Access*, 8:120757–120765, 2020.

[28] G. Li, H. Dibeklioğlu, S. Whiteson, and H. Hung. Facial feedback for reinforcement learning: a case study and offline analysis using the tamer framework. *Autonomous Agents and Multi-Agent Systems*, 34 (1):1–29, 2020.

[29] J. MacGlashan, M. K. Ho, R. Loftin, B. Peng, D. Roberts, M. E. Taylor, and M. L. Littman. Interactive learning from policy-dependent human feedback. *arXiv preprint arXiv:1701.06049*, 2017.

[30] R. Loftin, B. Peng, J. MacGlashan, M. L. Littman, M. E. Taylor, J. Huang, and D. L. Roberts. Learning something from nothing: Leveraging implicit human feedback strategies. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 607–612. IEEE, 2014.

[31] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *ACM SIGIR Forum*, volume 51, pages 4–11. Acm New York, NY, USA, 2017.

[32] D. Xu, M. Agarwal, F. Fekri, and R. Sivakumar. Playing games with implicit human feedback.

[33] N. Jaques, J. McCleary, J. Engel, D. Ha, F. Bertsch, R. Picard, and D. Eck. Learning via social awareness: Improving a deep generative sketching model with facial feedback. *arXiv preprint arXiv:1802.04877*, 2018.

[34] R. Arakawa, S. Kobayashi, Y. Unno, Y. Tsuboi, and S.-i. Maeda. Dqn-tamer: Human-in-the-loop reinforcement learning with intractable feedback. *arXiv preprint arXiv:1810.11748*, 2018.

[35] V. Veeriah. Beyond clever hans: Learning from people without their really trying. 2018.

[36] V. Veeriah, P. M. Pilarski, and R. S. Sutton. Face valuing: Training user interfaces with facial expressions and reinforcement learning. *arXiv preprint arXiv:1606.02807*, 2016.

[37] D. Zadok, D. McDuff, and A. Kapoor. Affect-based intrinsic rewards for learning general representations. *arXiv preprint arXiv:1912.00403*, 2019.

[38] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 59–66. IEEE, 2018.

[39] T. Baltrušaitis, M. Mahmoud, and P. Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 6, pages 1–6. IEEE, 2015.

[40] A. Zadeh, Y. Chong Lim, T. Baltrusaitis, and L.-P. Morency. Convolutional experts constrained local model for 3d facial landmark detection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2519–2528, 2017.

[41] H. Abdi. The kendall rank correlation coefficient. *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, pages 508–510, 2007.

[42] R. Woolson. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3, 2007.

[43] E. W. Weisstein. Bonferroni correction. *https://mathworld. wolfram. com/*, 2004.

[44] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[45] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305, 2012.

[46] P. M. Niedenthal, M. Mermillod, M. Maringer, and U. Hess. The simulation of smiles (sims) model: Embodied simulation and the meaning of facial expression. *Behavioral and brain sciences*, 33(6):417, 2010.

A Problem Formulation

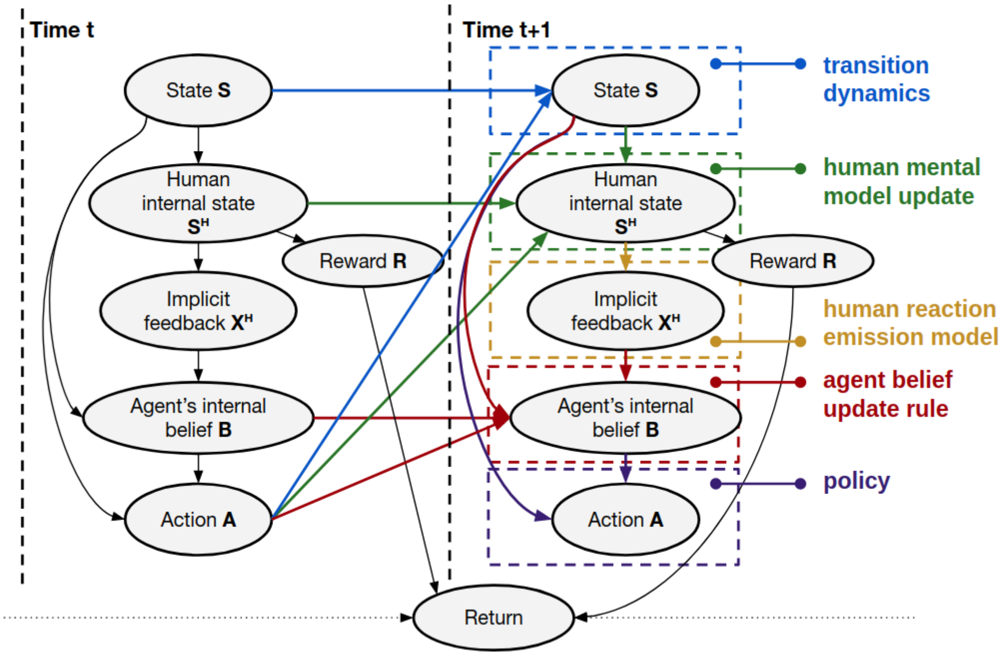

Time t
Time t+1
State **S**
Human internal state **S^H**
Reward **R**
Implicit feedback **X^H**
Agent's internal belief **B**
Action **A**
Return
transition dynamics
human mental model update
human reaction emission model
agent belief update rule
policy

Figure 8: Graphical model for LIHF (colored boxes and their identically colored labels represent conditional probability tables)

The graphical model for LIHF is shown in Figure 8. We assume the human $\mathcal{H}$'s reward function $R^{\mathcal{H}}$ is a temporally invariant element of the human's internal state $S^{\mathcal{H}}$. However, the observation $X^{\mathcal{H}}$ containing implicit human feedback to an action or a trajectory can change over time since it is influenced by the human's mental model of the task and the agent's policy at a particular time step. Given an observation $x \in X^{\mathcal{H}}$, the current state $s \in S$, and the previous action $a \in A$, the agent constructs its belief $b \in B$ as a probabilistic memory of arbitrary form and scope over the task domain. A belief could include, for example, the probability distribution over possible reward functions, which the agent could use to generate a policy (and therefore an action given the current state) that maximizes expected return (aggregated single-step rewards $r \in R$) under the unobserved human reward function $R^{\mathcal{H}}$. Note that reward is not directly dependent on state and action—it is determined by the human entirely (who can, for generality, internally maintain a history of states and actions, and therefore can give non-Markovian rewards).

B Experimental Domains and Data Collection Details

B.1 Robotaxi

Agent Transition Dynamics In the 8×8 grid-based map, the agent has three actions available at each timestep: maintain direction, turn left, or turn right. When the agent runs into the boundary of the map, it is forced to turn left or right, in the direction of the farther boundary.

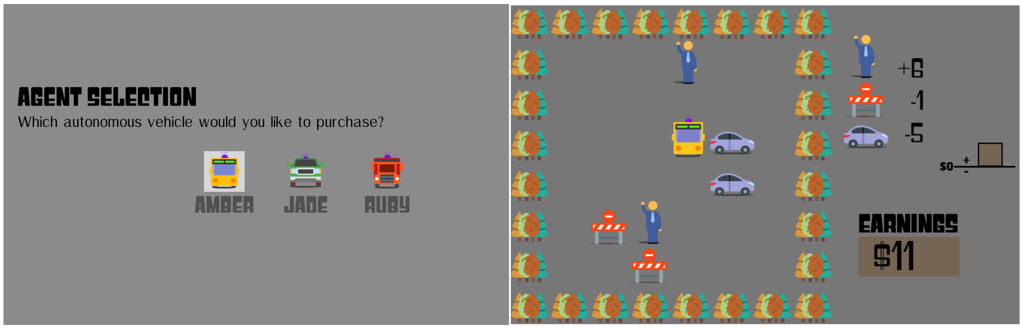
Rewards There are three different types of objects associated with non-zero rewards when encountered in the *Robotaxi* environment: if the agent picks up a passenger, it gains a large reward of $+6$; if it runs into a roadblock, it receives a small penalizing reward of -1; if the agent crashes into a parked car, it receives a large penalizing reward of -5. All other actions result in 0 reward.

Object Regeneration At most 2 instances of the same object type are present in the environment at any given time. An object disappears after the agent moves onto it (a "pickup"), and another object of the same type is spawned at a random unoccupied location 2 time steps after the corresponding pickup.

Agent Policy The agent executes a stochastic policy by choosing from a set of 3 pseudo-optimal policies under 3 different reward mappings from objects to the 3 reward values:

- Go for passenger: $\{$passenger: $+6$, road-block: -1, parked-car: $-5\}$
- Go for road-block: $\{$passenger: -1, road-block: $+6$, parked-car: $-5\}$
- Go for parked-car: $\{$passenger: -1, road-block: -5, parked-car: $+6\}$

The pseudo-optimal policies are computed in real time via value iteration (discount factor $\gamma = 0.95$) on a *static* version of the current map, meaning that objects neither disappear nor respawn when the agent moves onto them. We simplify the state space in this manner because the true state space is too large to evaluate and would create too large of a Q function to store, yet this simplification finds an near-optimal policy that almost always takes the shortest path to an object of the type that its corresponding reward function considers to be of highest reward. At the start of an episode and after each pickup, the agent selects 1 of these 3 policies. The agent follows the selected policy until the next pickup, except that there is a 0.1 probability at each time step that the agent will reselect from the 3 policies. This 0.1 probability of the agent changing its plans, in a rough sense, was included because we speculated that it would help increase human reactions by making the agent typically exhibit plan-based behavior but sometimes change course, violating human expectations. All selections among the 3 policies are done uniformly randomly.


AGENT SELECTION

Which autonomous vehicle would you like to purchase?

(a) Car choice screen

(b) *Robotaxi* game view

Figure 9: *Robotaxi* environment

B.2 Robotic Sorting Task

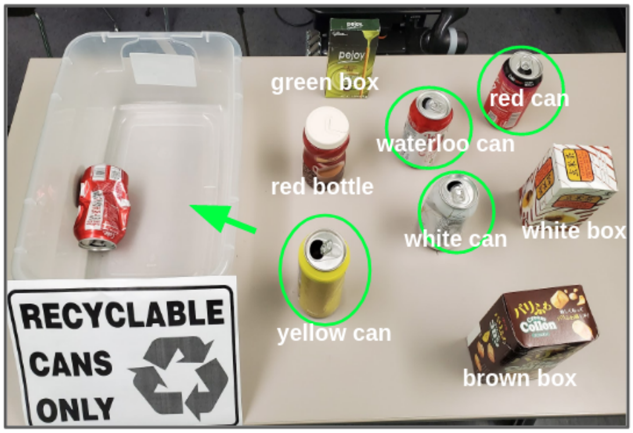


Figure 10: Robotic task table layout with object labels, from the perspective of the human observer

In the robotic sorting task, the robot executes trajectories programmed through key-frame based kinesthetic teaching. The 7-DOF robotic arm is controlled at 40Hz with torque commands. Fig. 10 shows the table layout at the beginning of the task: the robot's task is to sort the aluminum cans into the recycling bin; objects in the green circles give $+2$ rewards when moved into the bin and others

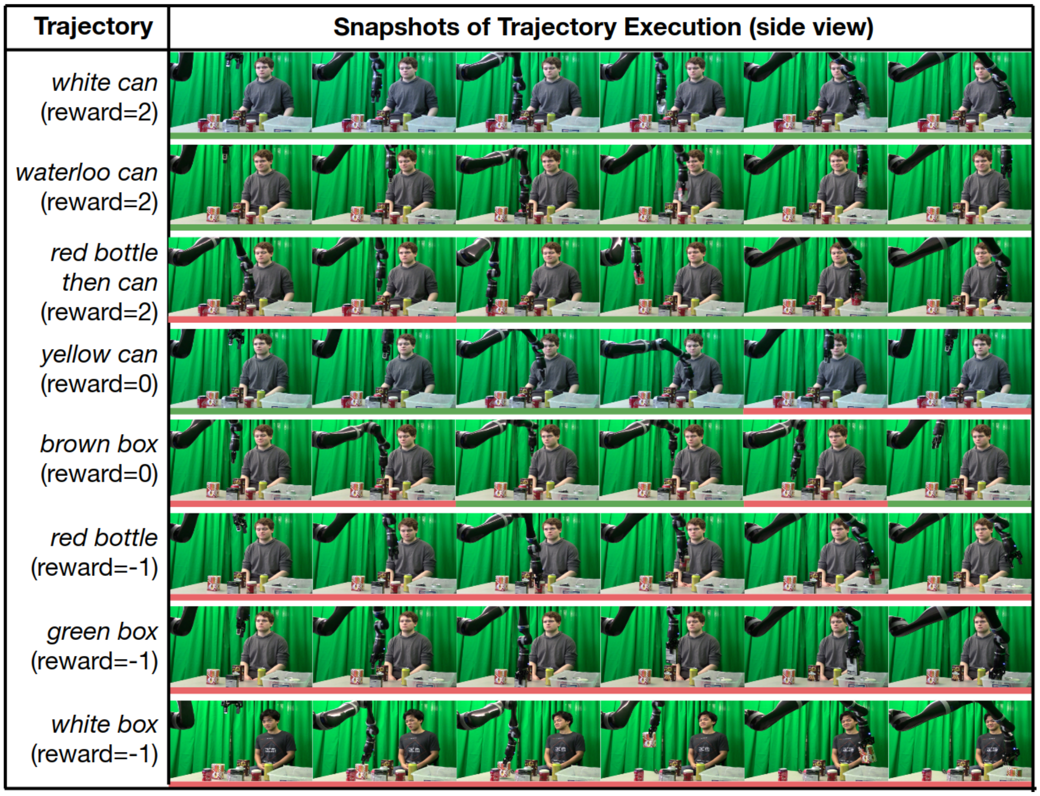


Figure 11: Robotic sorting task trajectories with optimality segmentation

give -1 rewards. Fig. 11 shows snapshots of the set of 8 arm trajectories we used in the robotic sorting task; each arises from a fixed sequence of torque commands. These fixed torque command sequences produce small variations in the actual trajectory, and any qualitative departure—like an object not being grasped successfully—results in that trajectory being removed from our dataset. The 8 episodes each involves 1 or 2 target objects, and ends with a reward of -1, 0 or $+2$. Each trajectory of an episode can be further segmented into reaching, grasping, transferring, placing and retracting sub-trajectories. The relative optimality of these sub-trajectories can be determined by whether the projected outcome is desired. For example reaching for a correct object and retracting from picking up a wrong object are both considered optimal while reaching for and transferring a wrong object are both sub-optimal. The optimality of sub-trajectories is also annotated in Fig. 11 under each trajectory. Note that our algorithm does not use any such segmentations, which are only for illustration.

B.3 Experimental Design

The instructions we give the participants in *Robotaxi* are as follows:

- Hello human! Welcome to Robo Valley, an experimental city where humans don't work but make money through hiring robots!
- You'll start with \$12 for hiring *Robotaxi*, and after each session you will be paid or fined according to the performance of the autonomous vehicle or robot.
- Your initial \$12 will be given to you in poker chips. After each session, we will add or take away your chips based on your vehicle's score. At the end, you can exchange your final count of poker chips for an Amazon gift card of the same dollar value.
- For the 3 sessions with a *Robotaxi*, you begin by choosing an autonomous vehicle to lease.
- The cost to lease one of these vehicles will be \$4 each session.

- The vehicle earns \$6 for every passenger it picks up, but it will be fined \$1 each time it hits a roadblock and fined \$5 each time it hits a parked car.
- You will watch the *Robotaxi* earn money for you, and your reactions to its performance will be recorded for research purposes.
- You will have a chance to practice driving in this world, but the amount earned during the test session won't count towards your final payout.

The instructions we give the participants in robotic sorting task are as follows:

- For the robotic task, the robot is trying to sort recyclable cans out of a set of objects.
- You will earn \$2 for each correct item it sorts and get penalized for \$1 for each wrong item it puts in the trash bin.
- You will watch the robot earn money for you, and your reactions to its performance will be recorded for research purposes.

The participants first control the agent themselves for a test session to make themselves familiar with the *Robotaxi* task, removing a source of human reactions changing in ways we cannot easily model. For the agent-controlled sessions, the participants select an agent at the beginning of each episode of *Robotaxi*. Fig. 9a shows the view of this agent selection. Unbeknownst to the subject, their selection of a vehicle only affects its appearance, not its policy. This vehicle choice was included in the experimental design as a speculatively justified tactic to increase the subject's emotional investment in the agent's success, thereby better aligning R and $R^{\mathcal{H}}$ as well as increasing their reactions. At the start of the session, participants are given \$12, which they must soon spend to purchase a *Robotaxi* agent before it begins its task. To make their earnings and losses more tangible (and therefore, we speculate, elicit greater reactions), participants are given poker chips equal to their current total earnings. After each session they are paid or fined according to the performance of the agent: their chips are increased or decreased based on the score of *Robotaxi*. At the end of the entire experimental session, participants exchange their final count of poker chips for an Amazon gift card of the same dollar value.

B.4 Participant Recruitment

The participants we recruited are mostly college students in the computer science department. Each participant filled out an exit survey of their backgrounds after completing all episodes of observing an agent. The statistics of these 17 subjects are given below:

- Gender: 10 participants are male and 7 are female.
- Age: The participants' average age is 20. Ages range from 18 to 28 (inclusive).
- Robotics/AI background: 1 participant is not familiar with AI/robotics technologies at all. 2 have neither worked in AI nor studied it technically, but are familiar with AI and robotics. 13 have not worked in AI but have taken classes related to AI or otherwise studied it technically. Only 1 has worked or done major research in robotics and/or AI.
- Ownership of robotics/AI-related products: 7 participants own robotics or AI-related products, while 10 do not. The products include Google Home, Roomba, and Amazon Echo.

C Annotations of Human Reactions

C.1 The Annotation Interface

To gain a better understanding of the dataset and the LIHF problem as a whole, two of the authors annotated the collected dataset. These annotations are not intended to serve as ground truth and are only used as labels for an auxiliary task in our training of reaction mappings. Therefore, training/calibration of annotators, evaluation of annotators via inter-rator reliability scores, etc. are not important. The interface for annotating the data is displayed in Fig. 12. A human annotator marks whether facial gestures and head gestures are occurring in each frame, effectively marking the onset and offset of such gestures. Annotation is performed without any visibility of the corresponding

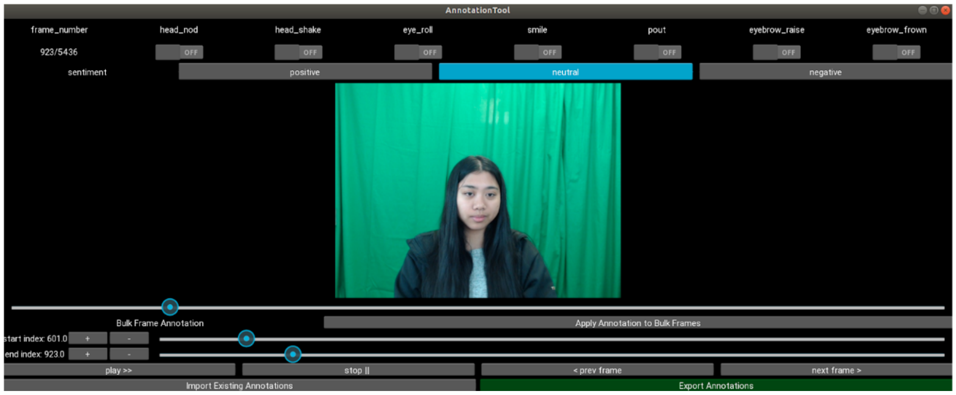


Figure 12: View of the annotation interface. The corresponding trajectory of *Robotaxi* is not displayed.

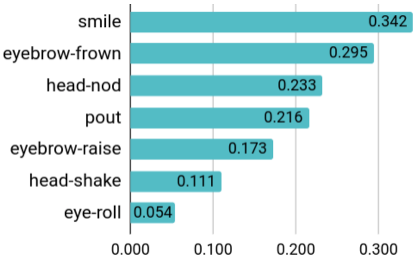


Figure 13: Proportion of annotated gestures

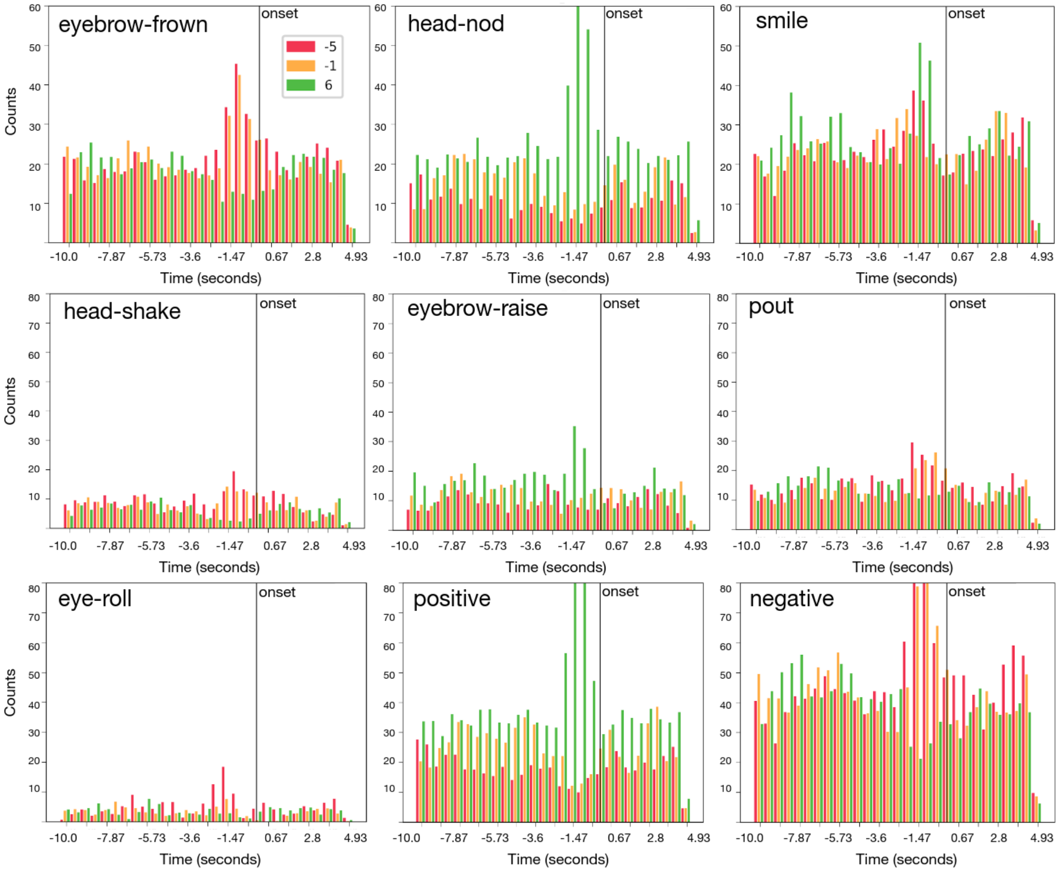


Figure 14: Histograms of non-zero reward events around feature onset

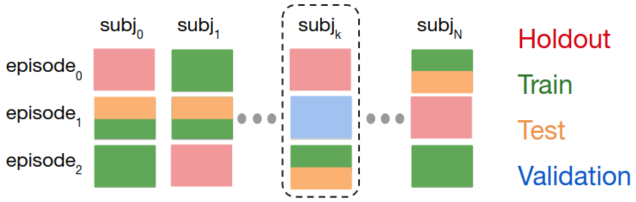


Figure 15: Diagram of data split for subject k

game trajectory. The proportion of 7 reaction gestures in the annotation is shown in Fig. 13. Annotations provide several benefits in this study: in our search for a modeling strategy, we found our first successful reaction mapping while using annotations directly as the only supervisory labels; annotations provide labels for an auxiliary task to regularize training of and speed representation learning by (both important for a small dataset) the reaction mapping from the features extracted automatically via OpenFace [38, 39, 40]; and an annotation-based analysis of our data helped us find a temporal window of reaction data around an event that is effective for inferring the reward types for that event.

C.2 Visualizations of Annotated Data

The annotations can be used to visualize the temporal relationship between reaction onsets and events (rewards). Fig. 14 shows example histograms of reward events binned into time windows around feature onsets. As demonstrated by Fig. 14, the onsets of certain gestures such as eyebrow-frown and head-nod correlate with negative or positive events respectively (peaking around 1.47s before the onset). While smile accounts for a large portion of overall gestures (Fig. 13), it does not correlate strongly with either positive or negative events, contradicting the assumption made by several prior studies that smile should always be treated as positive feedback [28, 34, 37, 36]. While this observation could be specific to our experimental setting or domain, it agrees with established research on the emotional meanings of smiles as shown in the work of Hoque et al. [12] and Niedenthal et al. [46].

In these histograms, the contours of red and yellow bars are strikingly similar in most subplots of Fig. 14, which suggests that although an individual may react differently to the events that provide -1 and -5 reward, it may be hard to distinguish between them through single gestures. We also find that reactions (across all gestures) are likely to occur between 2.8s before and 3.6s after an event (shown as a peak in the histograms), which we use as a prior for designing the set of candidate time windows that random hyperparameter search draws from (see Appx.E.2).

D Feature Extraction

The specific output data we use from OpenFace [38, 39, 40] are: [success, AU01_c , AU02_c , AU04_c , AU05_c , AU06_c , AU07_c , AU09_c , AU10_c , AU12_c , AU14_c , AU15_c , AU17_c , AU20_c , AU23_c , AU25_c , AU26_c , AU28_c , AU45_c , AU01_r , AU02_r , AU04_r , AU05_r , AU06_r , AU07_r , AU09_r , AU10_r , AU12_r , AU14_r , AU15_r , AU17_r , AU20_r , AU23_r , AU25_r , AU26_r , AU45_r , pose_Tx , pose_Ty , pose_Tz , pose_Rx , pose_Ry , pose_Rz].

The AUx_c signals are outputs from classifiers of activation of facial action units (FAU) and AUx_r are from regression model that are designed to capture the intensity of the activation of facial action units. The pose_T and pose_R signals are detected head translation and rotation with respect to the camera pose. Since the camera pose and relative position of a person with the camera varies from training time to application time, we explicitly model the change in the detected person's head pose by maintaining a running average and subtract the average from all incoming pose features. We then use a time window of the past 50 feature frames and compute the Fourier transform coefficients as the head-motion features we feed into the neural network.

E Model Design

E.1 Data Split of k-fold Cross Validation for Random Search

During our search for hyperparameters that learn an effective reaction mapping from data gathered in *Robotaxi*, we used a data-splitting method designed to avoid overfitting and to have relatively large training sets despite our small dataset size. Recall that each participant observe and react to 3 episodes. Of these 3 episodes, 1 is randomly chosen as a holdout episode and is not used for training or testing except for final evaluation. With the remaining 2 episodes per subject, we split data such that different train-test-validation sets are created for each subject, as shown in Fig. 15. Specifically, we construct a train-test-validation set for each subject by assigning one episode of the target subject as the validation set, randomly sampling half (either the first half or the second) of an unused episode from each subject into the test set, and using the remaining data in the training set. For a target subject, a model is trained on the subject's corresponding training set and tested after each epoch on the test set. The epoch with the best cross-validation loss is chosen as the early stopping point, and the model trained at this epoch is then evaluated on the validation set. The performance of a hyperparameter set is defined as the mean of the cross entropy losses across each subject's validation set. The hyperparameter set with the lowest such mean cross entropy loss is selected for evaluation on the holdout set. The data split for evaluation on the holdout set is similar but simpler. From the 2 episodes per subject that are not in the holdout set, half an episode is randomly sampled into the test set and the rest are in the training set. A single model is trained (stopping with the lowest cross-entropy loss on test set) and then evaluated on the holdout set.

E.2 Hyperparameters

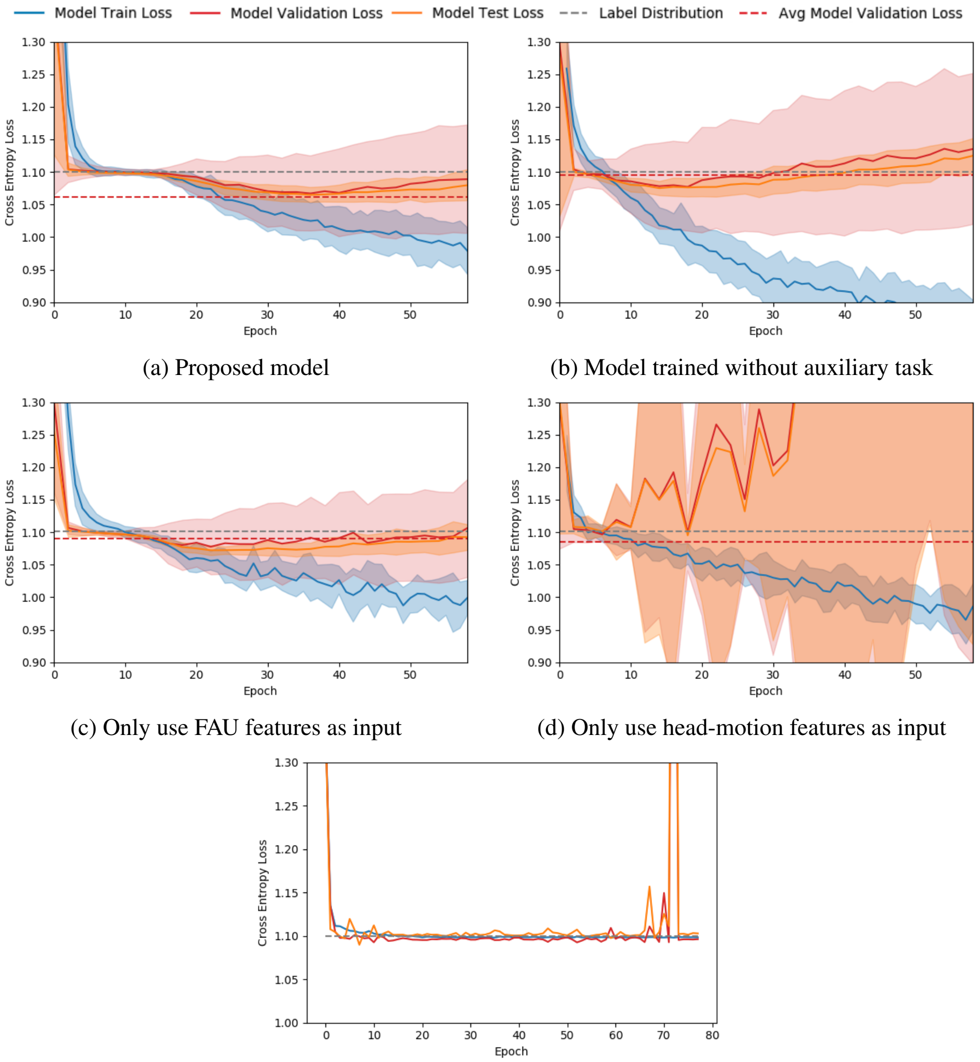
Random search is used to find the best set of hyper-parameters, including input window size (k and l), learning rate, dropout rate, loss coefficients (λ_1 and λ_2), depth and widths of the MLP hidden layers. Fig. 14 indicates that reactions are likely to onset between 2.8s before and 3.6s after an event. Therefore, we convert the corresponding range of temporal window into the number of aggregated frames before and after a particular prediction point (aggregated frame) and use that as the range to sample the input window. Each set of randomly sampled parameters is evaluated on all 17 train-test folds and the set with the lowest average test loss is selected. For each model, the weights with the lowest test loss are saved and evaluated on the validation set.

The best hyper-parameters found through random search are: $\{$learning_rate = 0.001, batch_size = 8, k = 0, l = 12, dropout_rate = 0.6314, λ_1 = 2, λ_2 = $1\}$. Below is the best model architecture found through random search:

```
(facial_action_unit_input): Linear(in_features=455, out_features=64, bias=True)
(head_pose_input): Linear(in_features=702, out_features=32, bias=True)
(hidden): ModuleList(
    (0): Linear(in_features=96, out_features=128, bias=True)
    (1): BatchNorm1d(128, eps=1e-05, momentum=0.1)
    (2): LeakyReLU(negative_slope=0.01)
    (3): Dropout(p=0.63, inplace=False)
    (4): Linear(in_features=128, out_features=128, bias=True)
    (5): BatchNorm1d(128, eps=1e-05, momentum=0.1)
    (6): LeakyReLU(negative_slope=0.01)
    (7): Dropout(p=0.63, inplace=False)
    (8): Linear(in_features=128, out_features=64, bias=True)
    (9): BatchNorm1d(64, eps=1e-05, momentum=0.1)
    (10): LeakyReLU(negative_slope=0.01)
    (11): Dropout(p=0.63, inplace=False)
    (12): Linear(in_features=64, out_features=8, bias=True)
    (13): BatchNorm1d(8, eps=1e-05, momentum=0.1)
    (14): LeakyReLU(negative_slope=0.01)
    (15): Dropout(p=0.63, inplace=False))
(out): Linear(in_features=8, out_features=3, bias=True)
(auxiliary_task): Linear(in_features=128, out_features=130, bias=True)
```

E.3 Ablation Study for Predictive Model Design

To validate the effectiveness of our model design, we conduct ablation study on the use of auxiliary task and input features. Fig. 16 shows the loss profiles during training across 17 subject train-test-validation sets for the proposed model, the model without auxiliary loss, the model using only FAU features, and the model using only head-motion features respectively. Each of them uses its own set



(a) Proposed model

(b) Model trained without auxiliary task

(c) Only use FAU features as input

(d) Only use head-motion features as input

(e) End-to-end model (Resnet+LSTM architecture)

Figure 16: Loss profiles for training different models (each model has its own set of best hyper parameters found through random search except the end-to-end model)

of hyperparameters found through random parameter search. All models are evaluated using 17-fold cross validation based on each subject, and the set with the lowest average test loss is selected. As shown in Fig. 16, our best full model has the best average loss on the test set, and also has the lowest mean and variance of validation loss compared with the other three models. We also tested training an end-to-end model with a Resnet-18 CNN as feature extractor and an LSTM model for processing features within a window. The CNN-LSTM model's training loss did not decrease to be lower than that obtained by outputting the label distribution. Given the size of this end-to-end model, we could not efficiently conduct an extensive hyperparameter search and have to rely on manual tuning. We speculate that as a main factor of failure. Meanwhile, we may not have enough data to effectively train a CNN-based feature extractor. Leveraging existing models such as OpenFace [38, 39, 40] for extracting features alleviates our modeling burden with limited amount of data.

F Computing Reward Ranking with Learned Reaction Mapping

q is the random variable for reward event and x is the variable representing human reactions. m is the discrete random variable over possible reward functions (i.e. reward rankings). The learned mapping effectively models $\mathrm{P}(q \mid x, m)$, which is the probability of an event given the human's reaction and a fixed reward ranking m. The goal is to find the posterior distribution over m: $\mathrm{P}(m \mid q, x)$.

Below is the proof for $\mathrm{P}(m \mid q, x) \propto \mathrm{P}(q \mid x, m)\mathrm{P}(m)$:

$$\begin{aligned} \mathrm{P}(q, x, m) &= \mathrm{P}(q \mid x, m)\mathrm{P}(x \mid m)\mathrm{P}(m) && (1) \\ &= \mathrm{P}(m \mid q, x)\mathrm{P}(q, x) && (2) \end{aligned}$$

$$\mathrm{P}(m \mid q, x)\mathrm{P}(q \mid x)\mathrm{P}(x) = \mathrm{P}(q \mid x, m)\mathrm{P}(x \mid m)\mathrm{P}(m) \quad (3)$$

x and m are conditionally independent since the human observes the reward, therefore:

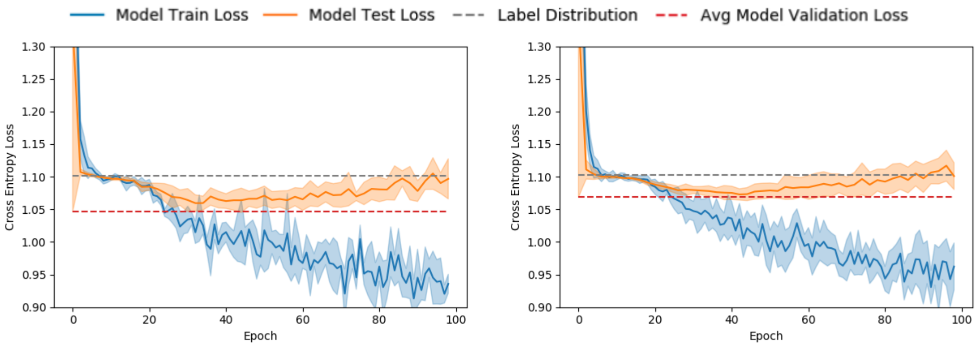
$$\mathrm{P}(x \mid m) = \mathrm{P}(x) \quad (4)$$

Hence,

$$\mathrm{P}(m \mid q, x)\mathrm{P}(q \mid x) = \mathrm{P}(q \mid x, m)\mathrm{P}(m) \quad (5)$$

$\mathrm{P}(q \mid x)$ is constant across all values of m, therefore: $\mathrm{P}(m \mid q, x) \propto \mathrm{P}(q \mid x, m)\mathrm{P}(m)$.

G Reaction Mapping Training Profile



(a) Holdout data as evaluation set

(b) Human proxy episodes as evaluation set

Figure 17: Loss profiles for training final models for reward ranking evaluation.

Fig. 17 shows the loss profiles (across 4 repetitions) for training final (single-model) mappings to be evaluated on our stage-2 instantiations. The red dotted line shows the average validation loss across 4 repetitions using the model selected with the lowest testing loss.

H Reward Ranking Performance on Human Proxy Test Episodes

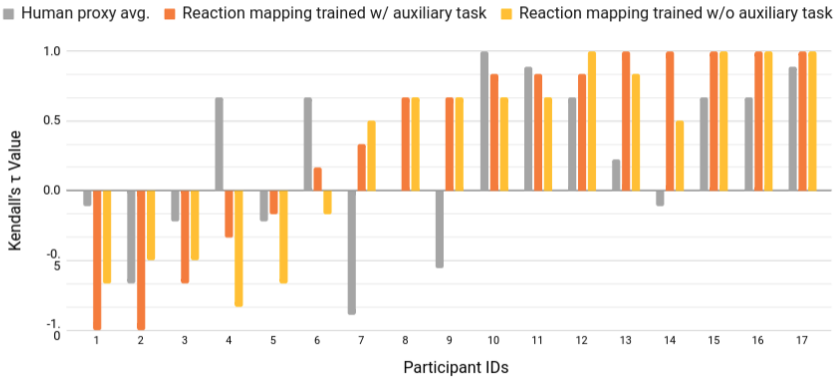


Figure 18: Sorted per-subject Kendall's τ for *Robotaxi* reward-ranking task on the human proxy test episodes

The result of performing the same reward ranking task on the human proxy test episodes is shown in Fig. 18. In this setting, the episodes on which the mappings are evaluated are the same as the human proxies viewed, and the rest of the episodes are used as training data. In general, our model's performance is bad on participants that the human proxies also find difficult and good on participants the human proxies performed well on, with a few exceptions.

I Effects of Different Belief Priors

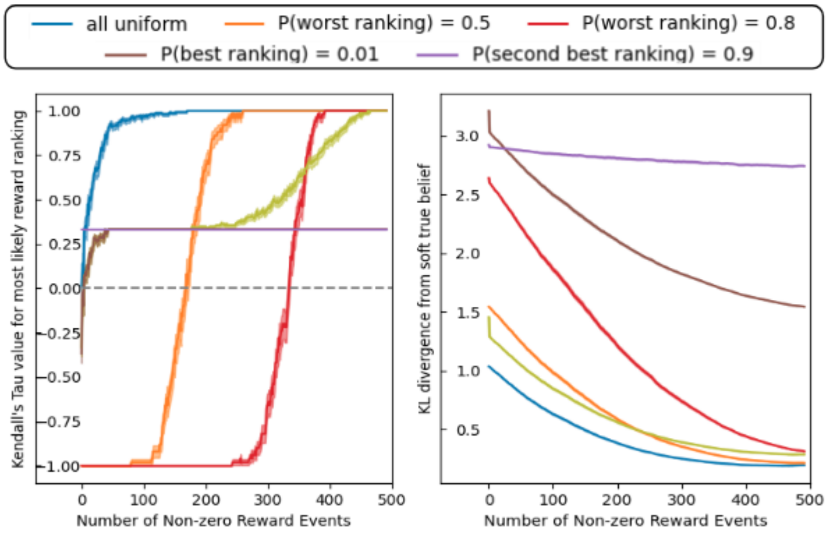


Figure 19: Performance of reward inference starting from different priors over reward rankings.

To test whether our reward learning method can recover from prior beliefs over reward rankings that give low probability to the correct reward ranking, we perform inference over all possible reward rankings starting from different priors. We pool predictions of the learned *Robotaxi* reaction mapping on the holdout dataset. From this pool of likelihoods, we randomly sample without replacement to update the reward belief. Different classes of reward ranking priors are tested:

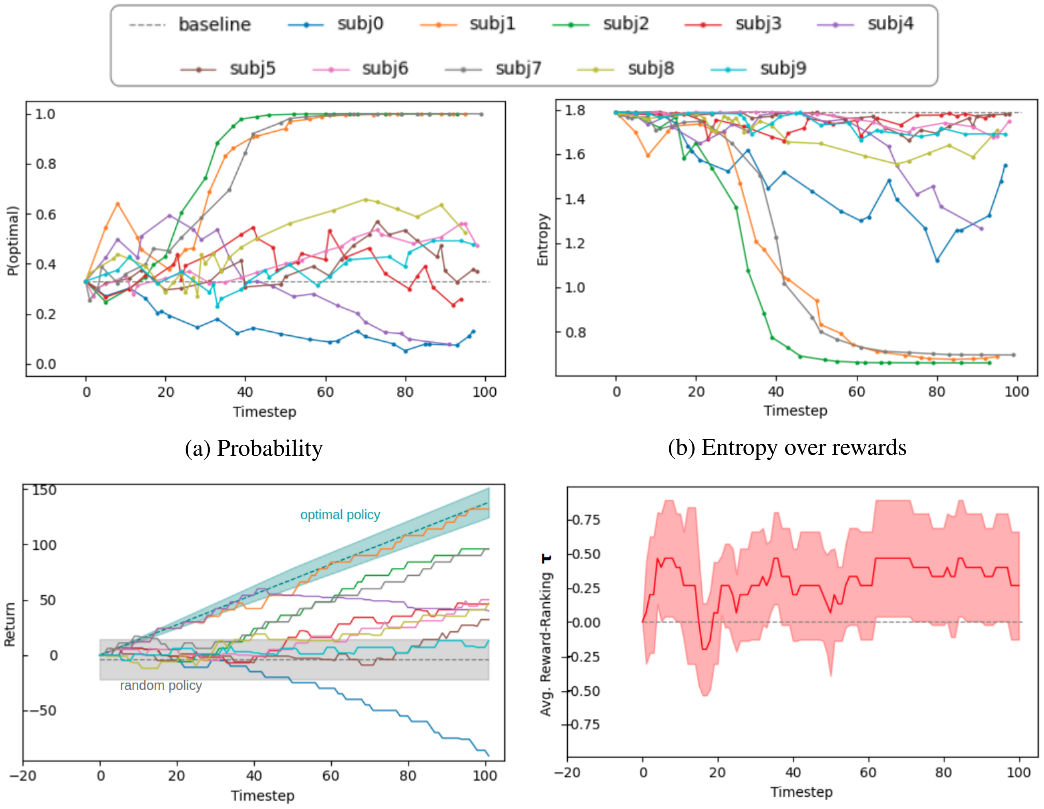
- **all uniform**: uniform prior over all possible reward rankings (used in all other experiments);
- **P(worst ranking) =** p: the reward mapping that ranks events in the *reverse* of the correct ranking has prior probability mass p, and the rest of the reward rankings uniformly share $1-p$ probability mass;
- **P(best ranking) =** p: the correct reward ranking has prior probability mass p, and the rest of the reward rankings uniformly share $1-p$ probability mass; and
- **P(second best ranking) =** p: the reward ranking that correctly ranks the positive-reward event first but incorrectly rank the two negative-reward events has prior probability mass p, and the rest of the reward rankings uniformly share $1-p$ probability mass.

As a function of the number of non-zero reward observations used for inference, we record the following performance metrics: the average Kendall's τ score of the most likely reward ranking, and the KL divergence between the current belief and a soft true belief distribution, defined such that the prior probability of a particular reward ranking is $\exp(\lambda\tau)/Z$, where τ is the Kendall's τ value for that reward ranking. This experiment is repeated 100 times and the mean performance over the number of non-zero reward events is shown in Fig. 19.

Out of the six different priors we tested, the hardest belief prior to recover from is **P(second best ranking) = 0.9**, where the two negative rewards are swapped compared to the correct ranking. In four out of the six cases, the reward inference process is able to recover the true reward ranking after incorporating enough data points. If higher weight is put on the predicted likelihood, the inference will converge faster to the correct reward ranking. The reward inference process in general is sensitive to the prior used. Therefore, when lacking justification for a biased prior, we recommend starting with a uniform prior.

Note that this experiment is conducted offline with data from human observers watching an agent with a fixed policy. We expect the learning dynamics to be different when the agent is updating its behavior policy online according to its existing belief, using live data from the human observer.

J Online Learning Results



(a) Probability

(b) Entropy over rewards

(c) Return

(d) Kendall's τ values

Figure 20: Trials of informal online learning sessions in *Robotaxi*

This online learning evaluation is conducted in informal settings. Due to the practice of social distancing during the world-wide spread of COVID-19, the authors recruited their friends as test subjects and conducted the evaluation in their own homes. Because of this informality, two aspects of the experimental design were not followed. First, participants were unpaid, removing our main mechanism for aligning $R^{\mathcal{H}}$ and R. (Note: R never produces reward observed by the agent but is instead used only for evaluating agent performance.) Second, prior to human subjects acting as observers, they did not control the agent for an episode, an experimental step that had been intended to provide the subjects an understanding of the task. We suspect that the lack of each of these aspects worsens our results; nonetheless, these results are fairly positive.

As shown in Fig. 20, in 9 out of 10 trials the final return is positive (8 out of 10 significantly higher than random policy returns) with $p = 0.0134$ by binomial test, and in 3 out of 10 trials the belief converged to the optimal and the second optimal reward rankings (both ranking passenger pick-up highest) with low entropy (when the probability mass evenly splits between the two rewards, the entropy is $-\ln(0.5) * 0.5 * 2 = 0.69$). Further, the average Kendall's τ value of reward ranking is higher than the random guessing baseline, after a small number of initial timesteps.

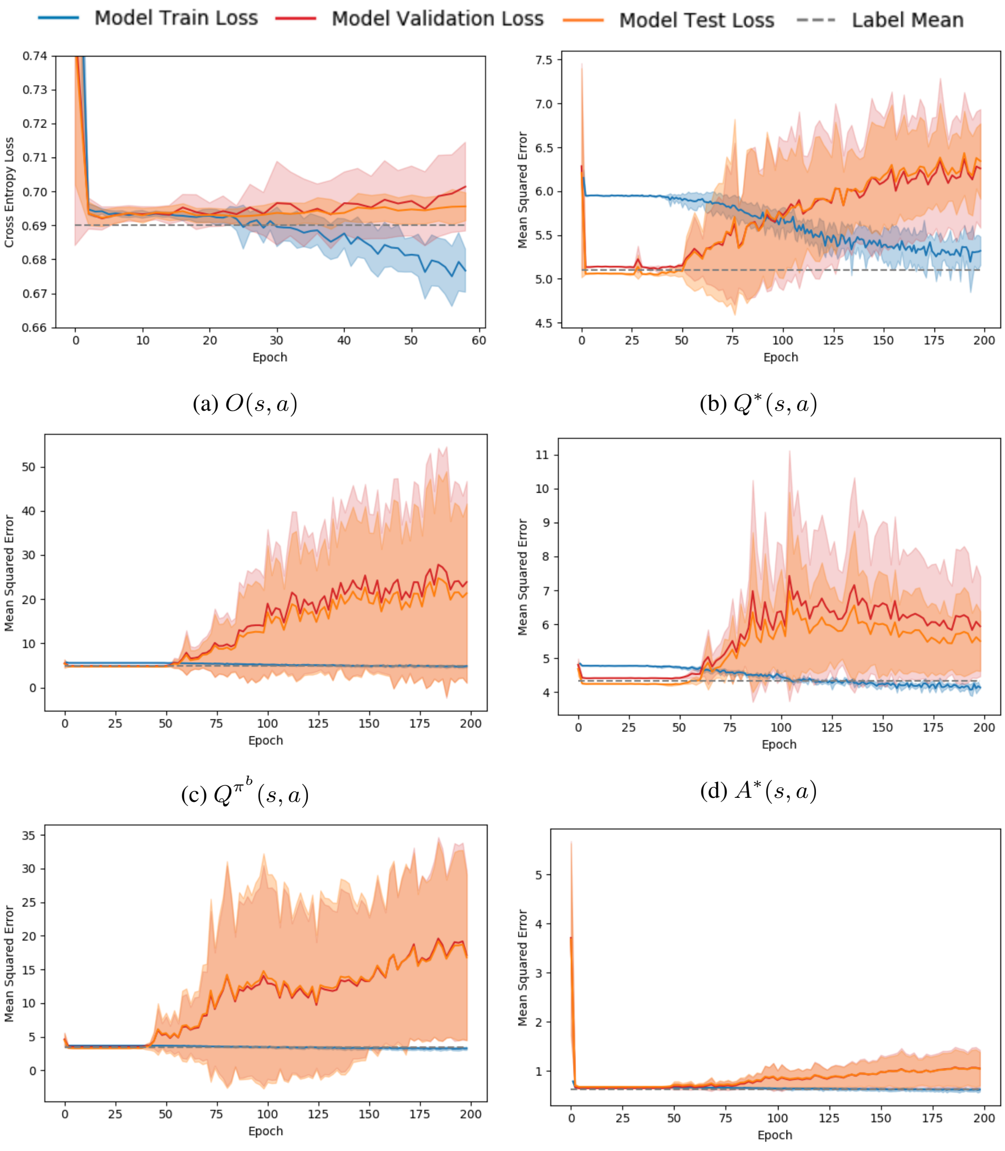
K Preliminary Modeling of Other Task Statistics

We also attempted to model other task statistics including the following ones computed with the agent's behavior policy π^b and the optimal policy π^* under the ground-truth reward function:

- Q-value of an action under optimal policy: $Q^*(s,a) = R(s,a) + V^*(s')$
- Optimality (0/1) of an action ($\mathbb{1}$ is the indicator function): $O(s,a) = \mathbb{1}_{[Q^*(s,a)]}(Q(s,a))$

- Q-value of an action under the behavior policy: $Q^{\pi^b}(s,a) = R(s,a) + V^*(s')$
- Advantage of an action under optimal policy: $A^*(s,a) = Q^*(s,a) - V^*(s)$
- Advantage of an action under behavior policy: $A^{\pi^b}(s,a) = Q^{\pi^b}(s,a) - V^{\pi^b}(s)$
- Surprise modeled as the difference in Q: $S(s,a) = Q^{\pi^b}(s,a) - Q^*(s,a)$

As mentioned previously, for computing the agent's policy in *Robotaxi*, we use an approximate optimal policy by assuming the grid map is static and run value iteration on the gridworld map (we repeat value iteration computation whenever the map changes, i.e. some object was picked up). We believe this policy is optimal as long as there are no more than 2 objects of the same type in the map. We then use Monte Carlo rollouts to estimate the value and Q-value along each trajectory.



(a) $O(s,a)$

(b) $Q^*(s,a)$

(c) $Q^{\pi^b}(s,a)$

(d) $A^*(s,a)$

(e) $A^{\pi^b}(s,a)$

(f) $S(s,a)$

Figure 21: Loss profiles for training with other task statistics

For training on these task statistics, cross entropy loss is used for optimality classification, mean square error is used for all other task statistics, and loss of the auxiliary task is additionally used for all task statistics for consistency. As shown in Fig. 21, the models trained on these task statistics all tend to overfit: as soon as the training loss starts to decrease, the test and validation loss start to increase, both never decreasing below the baseline performance of predicting the label's mean.

We speculate that modeling these task statistics is difficult due to *time-aliasing* in the training data, in which two adjacent training inputs in two consecutive timesteps are very similar but have different labels determined by the timestep's task statistics. Such time-aliasing is less of a problem when modeling only non-zero reward categories since non-zero reward events are often separated by zero-reward steps. An important direction for future research is to find a mechanism to directly address the time-aliasing in data labels.

We've also used a discount factor in computing the task statistics, treating *Robotaxi* as an infinite-horizon MDP while the actual episodes have a finite trajectory length of 200 time steps. This could be another factor that influences our modeling of these task statistics.

L Evaluating Robotic Sorting Task

To evaluate the robotic sorting trajectories. We consider each trajectory as an extended action and assume facial reactions along the trajectory are generated by a single latent state that represents the human's internal model of the robot's action. Therefore, the mean of the positivity score along each trajectory is used as the overall scoring of the trajectory. Fig. 22 shows the positivity score along all 7 trajectories (episodes) of the robotics task from a participant. For each trajectory, the mean positivity score across all participants is then computed. Fig. 23 shows the trajectory ranking related to all sorted items with their corresponding mean positivity score.

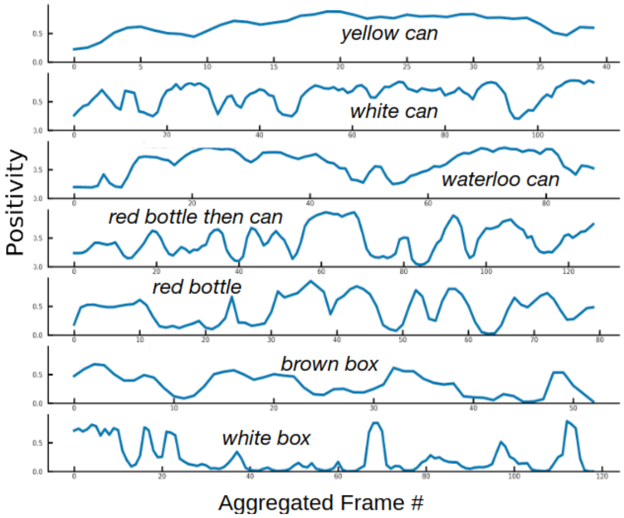
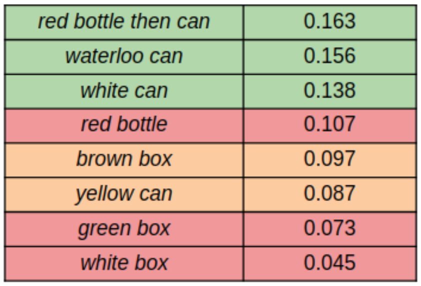


Figure 22: Sample plot of trajectory positivity score over aggregated frames



red bottle then can	0.163
waterloo can	0.156
white can	0.138
red bottle	0.107
brown box	0.097
yellow can	0.087
green box	0.073
white box	0.045

Figure 23: Overall trajectory ranking by mean positivity score across subjects (each entry is colored by the trajectory's final return: green for $+2$, yellow for 0, and red for -1)