2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Maximizing BCI Human Feedback using Active Learning

Zizhao Wang*, Junyao Shi*, Iretiayo Akinola* and Peter Allen

***Abstract*— Recent advancements in *Learning from Human Feedback* present an effective way to train robot agents via inputs from non-expert humans, without a need for a specially designed reward function. However, this approach needs a human to be present and attentive during robot learning to provide evaluative feedback. In addition, the amount of feedback needed grows with the level of task difficulty and the quality of human feedback might decrease over time because of fatigue. To overcome these limitations and enable learning more robot tasks with higher complexities, there is a need to maximize the quality of expensive feedback received and reduce the amount of human cognitive involvement required. In this work, we present an approach that uses active learning to smartly choose queries for the human supervisor based on the uncertainty of the robot and effectively reduces the amount of feedback needed to learn a given task. We also use a novel multiple buffer system to improve robustness to feedback noise and guard against catastrophic forgetting as the robot learning evolves. This makes it possible to learn tasks with more complexity using lesser amounts of human feedback compared to previous methods. We demonstrate the utility of our proposed method on a robot arm reaching task where the robot learns to reach a location in 3D without colliding with obstacles. Our approach is able to learn this task faster, with less human feedback and cognitive involvement, compared to previous methods that do not use active learning.**

I. Introduction

Learning from human feedback (LfHF) is an effective way to teach robot agents new skills. In this learning paradigm, an artificial agent receives feedback signals from a human expert that is watching the agent learn [1][2][3][4]. However, LfHF requires a human to be present during the learning process to provide the evaluative feedback. Depending on the difficulty of the task, the learning process might require a large amount of feedback to effectively learn the task, which translates to a significant amount of human time.

Human feedback can be collected via a variety of means including mouse clicks [3][5], facial expressions [6], finger pointing [7] and via human physiological signals like brain signals [8][9] among others. Learning from physiological signals is appealing in that the human does not need to perform any extra actions like mouse clicking etc, besides watching and evaluating the robot. On the other hand, learning from measured physiological signals such as brain signals measured via electroencephelography (EEG) presents a unique challenge; the physics of the EEG devices limit their signal-to-noise ratio which consequently results in noisy decoding of the evaluative feedback. More feedback might be needed to overcome increased noise levels in it; this solution is undesirable due to its high time cost.

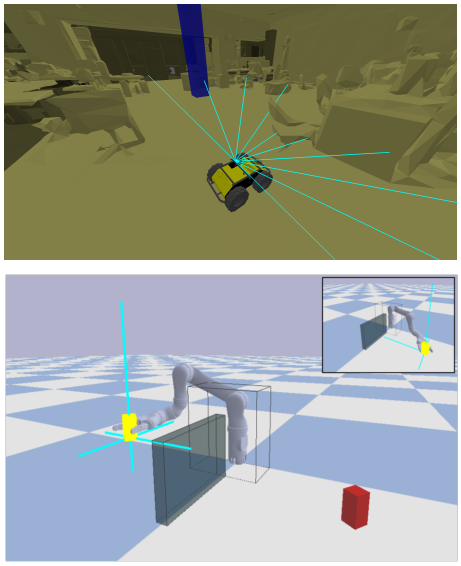

Fig. 1: Task environments. **Top**: For the navigation task, the robot agent learns an obstacle-avoidance policy to reach the goal (blue). The agent senses its environment with laser scans (10 cyan rays). **Bottom**: In the reaching task, the robot arm moves its end-effector to place the yellow block at the goal (red). The challenge is to navigate around the obstacle (slightly transparent black box) to reach the goal. Inset is an example of successful reach. For both tasks, in the sparse reward RL setting, the agent is unable to avoid obstacles and reach the goal. Compared with the navigation task, the large 3D state space in the reaching task would require a large amount of human feedback. Our proposed method reduces feedback required and is able to learn this task using evaluative feedback decoded from human EEG.

In this work, we investigate ways to reduce the amount of human feedback decoded from physiological signals needed to teach robot agents new tasks. This would reduce the amount of human hours required to teach the robot and also enable learning more complex tasks that previously required a large amount of feedback. We propose an active-learning-like approach that selects which queries to ask the human rather than ask the user at every move at every time-step. While a simulated robot can move very fast during the learning process, we optimally choose when to slow the robot down to get human feedback. Otherwise, the robot is able to speedily step through many steps when not asking for feedback. We use a special replay buffer to keep consensus feedback data to enable robustness to noise and guard against catastrophic forgetting.

Experiments using simulated feedback signals from a

*Equal Contribution.

Department of Computer Science, Columbia University, New York. This work was supported in part by a Google Research grant and National Science Foundation grant IIS-1527747.

noisy oracle show that our algorithm is able to learn a complex reaching task (with a large state and action space) in a feedback-efficient way. A baseline method would require a larger amount of feedback and a prohibitively long amount of time to learn the task from real human feedback. In summary, our method addresses the challenge of robot learning from limited (in quantity and quality) human feedback. Short human attention span and the imperfect signal-to-noise ratio of many brain-computer-interface devices restrict the quantity and quality of the evaluative feedback obtained from human physiological signals (such as brain signals). We propose to alleviate some of the challenges algorithmically. Our main contributions include:

- We introduce an active query approach that models a robot's decision uncertainty with a Dirichlet distribution and queries the human for feedback only when the robot is unsure.
- We present the novel use of a *purified* buffer to enable robustness to highly noisy human feedback.
- We demonstrate the application of our method to a challenging task which has large state and action spaces and is otherwise difficult to learn from small amount of noisy feedback.

II. Related Work

A. Learning from Human Feedback

Significant research effort has been directed toward learning from human feedback (LfHF) and its combination with reinforcement learning (RL) to teach robots different skills [1][2][3][4]. A natural approach is to use feedback as the reward signal to train a reinforcement learning agent, such as TAMER [1][4]. A potential drawback of using feedback purely for RL is that inconsistent feedback can yield sub-optimal performance. This can be addressed by combining human feedback with hand-crafted markov decision process (MDP) reward functions [10][11]. In a two-stage process, the policy first learned from human feedback is then fine-tuned by RL. This assumes that the learned feedback policy from the first stage provides good initialization for RL. Although we also use this two stage approach, our work is orthogonal in that we focus on reducing the amount of feedback needed to have a good LfHF policy to guide RL stage.

Another approach to learning from feedback is to derive a reward function from the feedback [3] rather than using the feedback directly as rewards. The derived reward function is then used for RL. In that work, human feedback is provided in the form of preferences between pairs of trajectory segments indicated by mouse clicks. In contrast, we focus on learning from physiological signals that do not require mouse clicks and obtain human evaluative feedback for each state-action pair chosen by the agent. We use active learning to reduce the amount of feedback needed in this setting.

B. Active Learning

According to a survey[12], active learning is a learning algorithm that can attain superior performance with fewer data by selecting which data to learn from. This approach

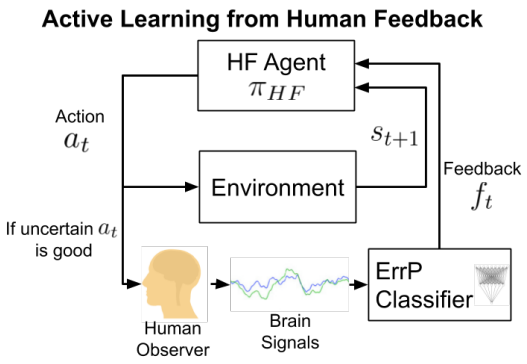

Fig. 2: Active LfHF. Instead of query labels for every action, we only ask for feedback that π_{HF} is unsure to be good. This improves information gain from each query and boosts feedback-efficiency.

selects data points that are optimal with respect to some information-seeking criteria in a way that minimizes the total number of queries required. Several works have used this technique in machine learning settings [13][14][15] and especially for robotics [16][17][18]. We use this key insight of active learning to reduce the amount of required feedback and enable learning hard tasks directly from the brain. Asides from our distinct application, we differ from existing works in the idea of selectively choosing to slow down some robot actions for the user to observe while the robot speeds through other actions during training. In addition, we use a Dirichlet distribution to measure robot uncertainty and actively choose actions with high uncertainties.

C. Brain-Computer Interface (BCI) Robot Learning

Error-related potentials (ErrPs) occur in the human brain when a human observes an error during a task [19] and a few works have explored using this as evaluative feedback to train robot in the context of robot learning. For example, [20] used ErrPs as reward signals in the reinforcement learning setting to teach a robot a reaching task. The task was limited to 2D task by limiting the points to a plane of 5x5 possible points. In contrast, the reaching task in this work is 3D with an obstacle that the robot has to learn to avoid. More recently, in our previous work [9], we developed an algorithm to use evaluative feedback from human brain to enhance learning a navigation task in a sparse reward setting. In that work, we trained a supervised learning model using noisy evaluative feedback from decoded ErrP signals. The obtained supervised policy is suboptimal but provides useful coarse guidance for the subsequent RL learning stage that uses a sparse reward to achieve an optimal policy. Now, we build on our previous work and address the unique issues of learning from brain signals. These include dealing with inherent noisiness of the feedback obtained via BCI devices and reducing the overall amount of feedback needed to learn a task. Our proposed solution enables learning more complex tasks with higher dimensional state and action spaces.

III. Method

There are three stages in our Active Brain-Guided RL algorithm: train an EEG classifier so that we can infer the

human feedback from brain signals; learn a Human Feedback (HF) policy based on the actively queried feedback; improve RL policy learning from sparse rewards by learning to reproduce good decisions generated by the HF policy.

A. EEG Classifier Training

Similar to [9], to evaluate human feedback from EEG signals, we collect EEG data and corresponding ErrP labels in an offline session. Then with collected data, we train a classifier, denoted as $g(\cdot;\theta_{EEG})$, which enables us to detect ErrPs when the human observes the robot choose wrong actions. In detail, first, the human subject gets familiar with the paradigm in a short training. Then the subject will watch the robot conducting a random policy while EEG signals and the judgments of actions (good/bad) are simultaneously recorded. In our reaching task paradigm, we obtain ground truth judgments as labels with the Dijkstra algorithm. For tasks where ground truth cannot be easily scripted, a human expert or the subject can provide the labels. In our experiments, the EEG signals are recorded at 2048 Hz using 64 channels of the BioSemi EEG Headset and around 600 data points are collected.

After data collection, we preprocess EEG data and train the classifier. We filter EEG data to 1-30 Hz, apply baseline correction with $[-0.2, 0]$s before stimuli as the baseline interval, and extract EEG trails at $[0, 0.8]$s after the robot takes actions. Then we train the classifier modified from EEGNet [21] using the cross-entropy loss $\mathcal{L}_{EEG}$, where detailed architecture modifications are described in [9]. During experiments, the testing accuracies among different subjects range between $0.55 - 0.73$.

B. Human Feedback Policy

With the trained EEG classifier, when subjects observe the robot's action a_t at state s_t, we can determine the corresponding feedback f_t from their brain signals. Note that in our work f_t can be quite noisy as the EEG classifier has low accuracy. To learn a human feedback policy, we assume the subject first forms a policy she/he thinks is optimal and then judges if a_t is good by checking if a_t is consistent with her/his optimal policy at s_t. Formally, we denote this judgement process as a function $F : S \times A \to [0, 1]$, where $0/1$ denotes a_t is bad/good respectively. During an online session, we collect feedback data (s_t, a_t, f_t) in a replay buffer, and simultaneously we learn an approximator of F which is denoted as $\hat{F}$ and derive the HF policy as:

$$\pi_{HF}(s) = \arg\max_a \hat{F}(s, a) \tag{1}$$

In our work, we use a fully connected network as the approximator and $\hat{F}$ is continually trained with the feedback pairs (s_t, a_t, f_t) generated by π_{HF}. Despite the straightforward setup, there are two challenges to learn a good HF policy: (1) the limited amount of human feedback (≤ 1000 labels) to learn a long-horizon task (2) the highly inconsistent feedback due to the low accuracy $(0.55 - 0.73)$ of the EEG classifier. To overcome these challenges, beyond having a light network (one hidden layer of 32 units) to reduce parameters to learn, we adopt the following two strategies:

1) Active Feedback Query: In our previous work [9], the agent asks for feedback on every action it takes, but this wastes feedback, especially when $\hat{F}$ is well trained for the given (s_t, a_t). Instead, as shown in Fig 2, the agent measures its confidence that a_t is good and then only query feedback if it is not confident enough (less than the threshold ϵ_{AQ}). Inspired by active learning, this strategy enables the agent to learn most from each query and improves feedback efficiency. However, we need to notice learning the HF policy is still different from active learning scenarios since it is solving a sequential decision making problem rather than classifying data points. In detail, we adopt Evidential Deep Learning (EDL) [22] for $\hat{F}$. Rather than output a deterministic probability for a_t to be good, $\hat{F}$ generates a Dirichlet distribution over all such probabilities from which we can measure the prediction confidence. Given an input s_t, $\hat{F}$ outputs the evidences that each possible action a_i is either good (denoted as $e_{ig} \geq 0$) or bad ($e_{ib} \geq 0$). Then the Dirichlet distribution D_i over all possible classification probability, $\boldsymbol{p_i} = [p_{ig}, p_{ib}]$, is formed with the parameter $\boldsymbol{\alpha_i} = [\alpha_{ig}, \alpha_{ib}] = [e_{ig} + 1, e_{ib} + 1]$ as

$$D_i(\boldsymbol{p_i}|\boldsymbol{\alpha_i}) = \frac{1}{B(\boldsymbol{\alpha_i})} p_{ig}^{\alpha_{ig}-1} p_{ib}^{\alpha_{ib}-1}, \tag{2}$$

where $B(\boldsymbol{\alpha_i})$ is two-dimensional beta function and p_{ig}, p_{ib} are the probabilities that action a_i is good or bad respectively.

Note the dependence of the generated distribution D_i on the state since its parameter $\boldsymbol{\alpha_i}$ is a function of the evidences which depend on the state via $\hat{F}$. For a given state, the probability that each action a_i is good is evaluated as the mean of the Dirichlet distribution: $\hat{p}_{ig} = \frac{\alpha_{ig}}{\alpha_{ig}+\alpha_{ib}}$. The confidence of each prediction is measured as the difference between the generated D_i and the Dirichlet distribution having uniform probability density, i.e. the one with parameters $\boldsymbol{\alpha} = [1, 1]$. The intuition is that the uniform Dirichlet distribution represents high uncertainty, and a more confident Dirichlet distribution D_i would be further from uniform in the parameter space. This distance in the parameter space (confidence) is measured and normalized to $[0, 1)$ as:

$$c_i = \frac{\alpha_{ig} + \alpha_{ib} - 1 - 1}{\alpha_{ig} + \alpha_{ib}} = \frac{e_{ig} + e_{ib}}{e_{ig} + e_{ib} + 2}. \tag{3}$$

Finally, the parameters are learned by minimizing the expectation of cross-entropy loss, with KL divergence from the uniform Dirichlet distribution as the regularization

$$\begin{aligned}\mathcal{L} = \int -(\mathbb{1}_{f_t=0} \log(p_{ib}) + \mathbb{1}_{f_t=1} \log(p_{ig})) D_i(p_i|\boldsymbol{\alpha}_i) d\boldsymbol{p}_i \\ + \lambda KL\left[D_i(\cdot|\boldsymbol{\alpha}_i) \parallel D(\cdot|[1, 1])\right],\end{aligned} \tag{4}$$

where λ is the regularizing coefficient increases from 0 to 1.

We also tried Gaussian Process as the predictor which naturally provides both prediction and confidence measures. However, it easily overfits the inconsistent feedback and is much more computationally expensive than neural networks.

2) Purified Buffer (PB): Beyond improving the feedback efficiency, the confidence measure from EDL also provides a tool to mitigate the significant inconsistency in the feedback. The key observation is simple: after training, $\hat{F}$ should fit consistent feedback data better than inconsistent

ones, as consistent ones still take up the majority despite low feedback accuracy. In other words, for feedback pairs (s_t, a_t, f_t) that are consistent, $\hat{F}(\cdot|s_t, a_t)$ should have the same prediction as the label f_t, as well as good confidence measures $c_t \geq \epsilon_{PB}$, where ϵ_{PB} is the confidence threshold. Hence, apart from the main replay buffer collecting all (s_t, a_t, f_t) pairs, we hold a second "purified" buffer to store the feedback pairs satisfying this condition. In this way, it will contain feedback of much higher accuracy and can stabilize the π_{HF} learning against feedback noise. As a result, every time we train $\hat{F}$, we will sample batches from both normal replay buffer and the purified buffer. Besides, the purified buffer also helps the agent review important past feedback and reduces the chance of catastrophic forgetting.

Algorithm 1: Active Brain-Guided RL

Data: offline EEG signals $x_{1:M}$ and labels $f_{1:M}$
Train the EEG classifier.

$$\theta^*_{EEG} = \arg\min_{\theta_{EEG}} \frac{1}{M} \sum_{i=1}^{M} [\mathcal{L}_{EEG}(g(x_i; \theta_{EEG}), f_i)].$$

Train the HF policy.
initialize the feedback replay buffer $B_{HF} = \emptyset$.
while *not received K feedback labels* **do**
  execute an action $s_t, a_t, c_t \sim \pi_{HF}(a_t|s_t)$.
  if $c_t \geq \epsilon_{AQ}$ **then**
    do not query the feedback.
    continue to execute the next action;
  query feedback by classifying EEG signal $f_t = g(x_t; \theta^*_{EEG})$.
  construct the purified buffer $B_{PB} = \{(s_t, a_t, f_t) \in B_{HF} | \hat{F}(s_t, a_t) = f_t, c_t > \epsilon_{PB}\}$.
  add (s_t, a_t, f_t) to B_{HF}.
  update $\hat{F}$ with batches from B_{HF} and B_{PB}.
Train the RL policy.
replay and episode buffer $B_{RL} = \emptyset, B_{EP} = \emptyset$.
for $t = 1, \ldots, t_{RL}$ **do**
  if $t \leq 0.2 \cdot t_{RL}$ **then**
    execute an action $s_t, a_t, r_t \sim \pi_{HF}(a_t|s_t)$
  else
    execute an action $s_t, a_t, r_t \sim \pi_{RL}(a_t|s_t)$
  add (s_t, a_t, r_t) to B_{EP}.
  if s_t *is the end of the episode* **then**
    add $(s_t, a_t, R_t = \sum_{k=t}^{\infty} \gamma^{k-t} r_k)$ to B_{RL} for t in B_{EP}.
    clear the episode buffer $B_{EP} = \emptyset$.
  optimize $\mathcal{L}^{PPO}$ with on-policy samples.
  optimize $\mathcal{L}^{imit}$ with batches from B_{RL}.

After learning with human feedback, the policy π_{HF} has a rough knowledge about which actions are good/bad. In spite of the low success rate to finish the task, this HF policy still provides better exploration than random, maximum entropy exploration which most RL algorithms use.

C. Sparse-Reward RL with Guided Exploration and Imitation Learning

The final stage is to learn an RL policy π_{RL} efficiently in an environment with sparse rewards. Typical exploration strategies struggle to stumble on positive rewards that provide learning signals. To address this, we use π_{HF} as the initial behavior policy during RL learning. Even though π_{HF} may be far from perfect, it guides the exploration towards the goal and increases the chances of getting positive rewards. In addition to learning from rewards, we let our RL agent reproduce previous good decisions (by π_{HF} or π_{RL}), using imitation learning [23]. To do this, we store past episodes and their cumulative rewards $(s_t, a_t, R_t = \sum_{k=t}^{\infty} \gamma^{k-t} r_k)$ in a replay buffer. We use a filtered imitation learning loss that selectively learns from good experiences only. In the actor-critic framework, the loss has two components :

$$\mathcal{L}^{imit} = \mathcal{L}^{imit}_{actor} + \mathcal{L}^{imit}_{critic}$$

$$\mathcal{L}^{imit}_{actor} = [-\log \pi_{RL}(a_t|s_t)(R_t - V(s_t))] \cdot \mathbb{1}_{R_t > V(s_t)} \quad (5)$$

$$\mathcal{L}^{imit}_{critic} = \frac{1}{2} \|R_t - V(s_t)\|^2 \cdot \mathbb{1}_{R_t > V(s_t)}. \quad (6)$$

With the filter $\mathbb{1}_{R_t > V(s_t)}$, the RL agents will only learn to choose a_t chosen in the past at s_t if return R_t is greater than current value estimate $(R_t > V(s_t))$. This enables the RL agent to focus on successful episodes from π_{HF} in the initial training stage. Moreover, when π_{RL} is close to convergence, the filter guarantees that π_{RL} will not be constrained to the suboptimal performance of π_{HF} and can outperform it.

Implementation-wise, we can choose any off-policy actor-critic RL algorithm. Our method even can be applied to on-policy Deep RL algorithms like PPO [24] which we adopt as π_{RL}. The policy and value networks have the same architecture as π_{HF}, except for the output layers as they have different output dimensions. For the first 20% of training, we choose π_{HF} as the behavior policy to generate good episodes for imitation learning. Then we switch to the π_{RL}. Our full algorthim for Active Brain-Guided RL is given in Alg 1.

IV. Experiments

To test our method, we implemented navigation and reaching tasks in Gibson [25] and pybullet environment respectively. For the navigation task, the goal is to drive the robot (shown in Fig 1, top) to the fixed goal represented by the blue pillar. The environment is a $11 \times 12m^2$ area with multiple obstacles. The state space is chosen as $s_t = (l_t, d_t, \phi_t) \in \mathbb{R}^{13}$ where $l_t \in \mathbb{R}^{10}$ is laser range observations evenly spaced from 90° left to 90° right of the robot, $d_t \in \mathbb{R}^2$ is the distance and relative angle from goal, and ϕ_t is the robot yaw angle. The action space A is discretized to help human subject identify actions and determine their optimality. Three actions: moving $0.3m$ forward, turning 30° left or right.

For the reaching task, the goal is to control the 6-DOF Mico Robotic arm (shown in Fig 1, bottom) to bring the yellow object to the fixed goal location shown as the red block. To this end, the arm needs to get around the grey obstacle while avoiding self-collision (indicated by the transparent box). The arm moves in a $0.5 \times 1.0 \times 0.675m^3$ space, which covers most reachable space for the arm. The state space is chosen as $s_t = (p_t, l_t) \in \mathbb{R}^9$ where $p_t \in \mathbb{R}^3$ is the Cartersian coordinate of the end effector and $l_t \in \mathbb{R}^6$ is laser range observations along three coordinate axes. The

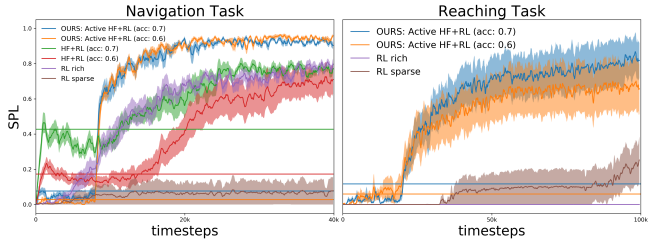

Fig. 3: Simulated Feedback Results. Using the SPL metric, we compare the performance of our method (Active HF+RL) at two feedback accuracies (Blue: 0.7, Orange: 0.6) with RL sparse (Purple), RL rich (Brown). The plot shows the mean and 1/3 of the standard deviation over 10 different runs. We also show the average performance of π_{HF} as blue and orange horizontal lines. Despite its suboptimal performances, π_{HF} accelerates reinforcement learning in sparse reward settings. Meanwhile, RL with sparse/rich rewards struggles or even fails to learn the tasks. Notice that RL rich has SPL = 0 all the time for the 3D reaching task which has a high dimensional state space. For the navigation task, we also show the result of a baseline method [9] (Green: 70%, Red: 60%) that does not use active learning and purified buffer. Active HF+RL outperforms this baseline with better RL policy after convergence.

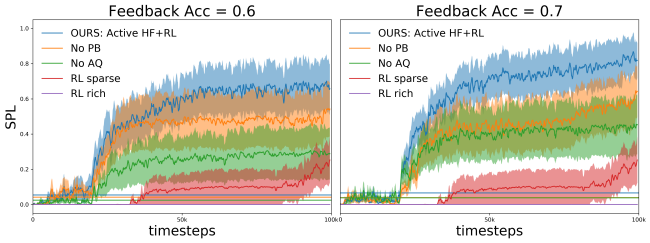

Fig. 4: Ablation Results. **Left**: $\mathcal{C} = 0.6$, **Right**: $\mathcal{C} = 0.7$. We compare how removing Active Query (NO AQ) or Purified Buffer (No PB) affects the learning of π_{HF} and π_{RL}. Again, the plot shows the mean and 1/3 of the standard deviation over 10 different runs, and the horizontal lines represent the average performance of π_{HF} for each method. Without Active Query, we can see a $55\%/43\%$ drop in the π_{HF} for $\mathcal{C} = 0.6/0.7$ respectively. This confirms our algorithm achieves same level of learning performance with less amount of the human feedback. Meanwhile, the π_{HF} degraded by $23\%/40\%$ for "No PB".

action space A contains six actions: moving along both directions of the x/y/z-axis with constant step lengths of $0.025/0.05/0.0675m$ respectively. In this case, the state space is split to a $21 \times 21 \times 11$ grid.

For both tasks, the sparse reward **RL sparse** is -0.05 for every step, except for a +10 bonus when reaching the goal or -5 penalty for collision. Instead, a reward function providing richer learning signals, **RL rich**, can be:

$$r_{rich}(s_t) = c_d \cdot d_t + c_\theta \cdot \theta_t \qquad \text{if no collision occurs}$$

where d_t is the distance from the goal, θ_t is the difference between the current orientation and the orientation to the goal (only for the navigation task), and $c_d = -0.01$ (and $c_\theta = -0.003$) are the empirically selected hyperparameters. This rich reward motivates the robot to get close to (and head towards) the goal, which leads to a more efficient exploration.

For the navigation task, the robot's initial position is uniformly sampled in a $0.2 \times 0.2m^2$ area. For the reaching task, the beginning location of the end-effector is sampled in a $0.2 \times 0.1 \times 0.125m^3$ space, which is a $9 \times 3 \times 3$ grid. For both navigation and reaching tasks, depending on the initial position, the optimal path consists of $17 - 19$ and $33 - 41$ steps respectively. The episode ends if any of the following happens: the goal is reached, a collision occurs with surrounding obstacles or the maximum episode length of 120/80 steps is reached for the navigation/reaching tasks.

V. Results

With two challenging obstacle-avoidance tasks, we evaluate our proposed method. To ensure repeatability, we first use simulated feedback from a scripted oracle. Then, we assess the performance of our system with real human feedback from 6 subjects. For both experiments with simulated and real feedback, we compare our method (Active HF+RL) with RL algorithms learning from sparse/rich rewards (RL sparse/RL rich). We also compare with a baseline method [9] in the navigation task with simulated feedback. We select the Success weighted by (normalized inverse) Path Length (SPL) [26] as the metric, which considers both success rate and path optimality. For a fair comparison, we use the same architecture and hyperparameters for the RL part across all methods. Moreover, RL sparse and RL rich also keep a buffer of their good experiences during training and use self-imitation learning component presented in our method.

A. Learning from Simulated Feedback

For simulated feedback, we test two accuracies: $\mathcal{C} = 0.6/0.7$. It evaluates how well the HF policy assists the RL learning with noisy feedback. More consistent feedback requires fewer labels to learn a good π_{HF}, hence we query $1000/450$ feedback labels for $\mathcal{C} = 0.6/0.7$ respectively in the navigation tasks and $1000/300$ feedback labels in the reaching task. Meanwhile, we select the value of $\epsilon_{AQ} = 0.4/0.5, \epsilon_{PB} = 0.5/0.6$ using grid search.

Shown in Fig 3, most RL-sparse trials struggle to learn the task as the chance to reach the goal with random actions is very small. For RL-rich, even though it works well in the navigation task, in the harder reaching task, it is even worse than RL sparse as all trials fail as the method greedily maximize the rewards and cannot get over the obstacle. This suggests designing a successful dense reward is a nontrivial task and requires lots of trial and error. Instead, our method (Active HF+RL) solves the reaching task well for most trials. It confirms π_{HF} still provides enough exploration to the goal and good experiences for π_{RL} to learn from, even with feedback with significant noise. Compared with a baseline method [9], the imitation learning buffer enables the π_{RL} to constantly review good exploration experiences from π_{HF}, and thus the RL agent can receive enough learning signals despite the poor performance of π_{HF}.

In Fig 4, in the reaching task, we perform a series of ablation experiments to measure the importance of active feedback query and purified buffer when learning π_{HF}, as well as their effects on the training of π_{RL}. Without the

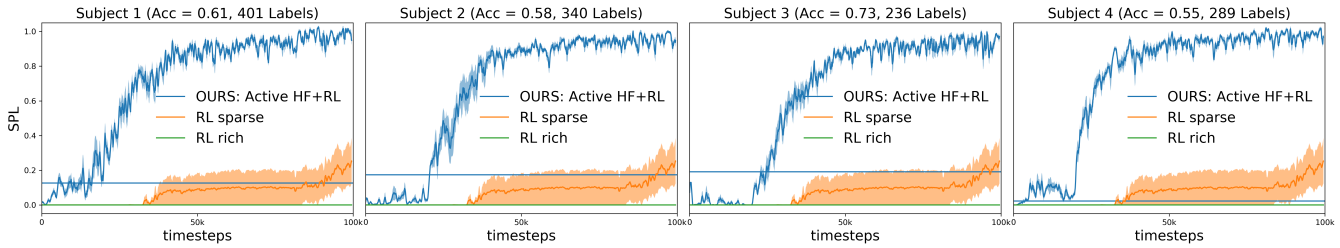

Fig. 5: Real Feedback Results for 4 Successful Subjects on a reaching task. Our method (Blue) uses feedback from human brain signals and accelerates RL learning. Subject 4 has low π_{HF} performance due to high feedback noise; our method still achieves optimal performance.

active query (**No AQ**), we can see a significant drop in the performance of π_{HF} and π_{RL}. Especially, for $\mathcal{C} = 0.6$, the final performance of π_{RL} is close to RL-sparse. To see why, consider the case where the π_{HF} agent already learned to move towards the goal but didn't know how to go over the obstacle yet. When collisions happen and the agent is reset to initial positions, the agent first needs to get close to the obstacle again. However, the "No AQ" will query feedback all the way even in well-learned states, which is a waste of the feedback. Besides, there can be wrong feedback due to noise, and it may harm or even destroy the learned policy. Instead, the "AQ" agent knows what it already knows and won't query feedback until it is near the obstacle. In this way, the queried feedback is more informative and the agent can learn to get over the obstacle more efficiently. For agents without the purified buffer (**No PB**), there is also a drop in performance, especially for $\mathcal{C} = 0.7$. Meanwhile, we find feedback in the purified buffer has a much higher accuracy than the simulated accuracy, with 80% compared with 60%, or 88% compared with 70%. This affirms the purification buffer can assist π_{HF} learning with more accurate labels.

B. Learning from Real Human Feedback

We tested our Active HF+RL method on the reaching task with 6 human subjects providing feedback in the form of EEG signals while watching the simulated robot learn. First, the subject is trained for 5 minutes to get familiar with the paradigm and understand how the arm should move to the goal. Then, the subject has 20 1-min offline sessions to collect data for training the EEG classifier and finally provides feedback during 30 1-min online sessions to train the π_{HF} policy. This π_{HF} is subsequently used to guide the RL similar to the simulation experiments.

Shown in Fig 5, the low performing π_{HF} policies from 4 subjects successfully guide the RL learning. For two other subjects, the EEG classifier accuracies are low ($\sim 51\%$), and thus their feedback is too noisy to train a useful π_{HF} to guide RL. A limitation of our approach is that we depend on the ability to detect error signals via EEG. Since detecting ErrP can vary across different individuals, our system does not work for individuals whose signals cannot be accurately classified. Nevertheless, our extensive simulation experiments show that the proposed method handles significant feedback noise levels and can also be applied to other input modalities that have less variability across subjects.

VI. Discussion

The experiments on navigation and reaching tasks with either simulated or real feedback show that our Active HF+RL can learn from feedback efficiently and accelerate RL in complex sparse reward environments. In contrast, RL learning from sparse or even rich rewards fails to finish the task. In the ablation studies, with the same amount of feedback, LfHF with Active Query learns much better human feedback policy and RL policy than the one without AQ. This confirms Active Query improves information gain from each query and significantly reduces the amount of expensive feedback needed. Meanwhile, when dealing with feedback inconsistency as large as 40%, the purified buffer can filter out noise and contain feedback of a much higher quality (80% accuracy). This guarantees our method is robust to low ErrP classification accuracy. Finally, even though π_{HF} has suboptimal performance (e.g. subject 4), the imitation learning with Q-value filtering ensures RL will only learn from good guided explorations of π_{HF} and repeatedly reviews them, ensuring RL can achieve optimal performance.

VII. Conclusion

This paper introduces Active Brain-Guided RL, a method to use human feedback evaluated from noisy and expensive EEG signals to bootstrap RL learning in sparse reward settings. We first train a HF policy with Active Query and Purified Buffer, and then the HF policy generates good experiences for RL to learn from. This method demonstrates robustness in three important ways: (1) When using Active Query to smartly select queries, it greatly improves feedback-efficiency and is robust to the limited amount of expensive human feedback. (2) With the Purfied Buffer, it filters out noise and learns from feedback of higher accuracy, making it robust to feedback inconsistency due to noisy EEG signals and poor classification accuracy. (3) It is also robust to the low success rate of the human feedback policy, since the human feedback policy still provides coarse guidance to the goal. Besides, the imitation learning with Q-value filter ensures the RL agent will constantly learn from good experiences of π_{HF} and can outperform it when close to convergence. Different experiments using simulated or real feedback and corresponding ablation studies confirm our method can learn long-horizon tasks from sparse rewards with less amount of feedback than previous methods.

REFERENCES

[1] 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*. ACM, 2009, pp. 9–16.

[2] S. Griffith, K. Subramanian, J. Scholz, C. Isbell, and A. L. Thomaz, "Policy shaping: Integrating human feedback with reinforcement learning," in *Advances in Neural Information Processing Systems*, 2013, pp. 2625–2633.

[3] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, 2017, pp. 4299–4307.

[4] 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.

[5] K. Jagodnik, P. Thomas, A. van den Bogert, M. Branicky, and R. Kirsch, "Training an actor-critic reinforcement learning controller for arm movement using human-generated rewards," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2017.

[6] 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.

[7] F. Cruz, G. I. Parisi, J. Twiefel, and S. Wermter, "Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 759–766.

[8] I. Iturrate, R. Chavarriaga, L. Montesano, J. Minguez, and J. d. R. Millán, "Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control," *Scientific reports*, vol. 5, p. 13893, 2015.

[9] I. Akinola, Z. Wang, J. Shi, X. He, P. Lapborisuth, J. Xu, D. Watkins-Valls, P. Sajda, and P. Allen, "Accelerated robot learning via human brain signals," *arXiv preprint arXiv:1910.00682*, 2019.

[10] W. B. Knox and P. Stone, "Reinforcement learning from simultaneous human and mdp reward," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 475–482.

[11] ——, "Combining manual feedback with subsequent mdp reward signals for reinforcement learning," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. Citeseer, 2010, pp. 5–12.

[12] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[13] A. Agarwal, "Selective sampling algorithms for cost-sensitive multiclass prediction," in *International Conference on Machine Learning*, 2013, pp. 1220–1228.

[14] F. Orabona and N. Cesa-Bianchi, "Better algorithms for selective sampling," in *International conference on machine learning*. Omnipress, 2011, pp. 433–440.

[15] O. Dekel, P. M. Long, and Y. Singer, "Online multitask learning," in *International Conference on Computational Learning Theory*. Springer, 2006, pp. 453–467.

[16] M. Cakmak and A. L. Thomaz, "Designing robot learners that ask good questions," in *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2012, pp. 17–24.

[17] S. Hangl, V. Dunjko, H. J. Briegel, and J. Piater, "Skill learning by autonomous robotic playing using active learning and creativity," *arXiv preprint arXiv:1706.08560*, 2017.

[18] J. Kulick, M. Toussaint, T. Lang, and M. Lopes, "Active learning for teaching a robot grounded relational symbols," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[19] M. Spüler and C. Niethammer, "Error-related potentials during continuous feedback: using eeg to detect errors of different type and severity," *Frontiers in human neuroscience*, vol. 9, p. 155, 2015.

[20] L. Schiatti, J. Tessadori, N. Deshpande, G. Barresi, L. C. King, and L. S. Mattos, "Human in the loop of robot learning: Eeg-based reward signal for target identification and reaching task," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4473–4480.

[21] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 15, no. 5, p. 056013, 2018.

[22] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Advances in Neural Information Processing Systems*, 2018, pp. 3179–3189.

[23] J. Oh, Y. Guo, S. Singh, and H. Lee, "Self-imitation learning," *arXiv preprint arXiv:1806.05635*, 2018.

[24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[25] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese, "Gibson Env: real-world perception for embodied agents," in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[26] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.