# Error-Related Potential-Based Shared Autonomy via Deep Recurrent Reinforcement Learning

Xiaofei Wang, Hsiang-Ting Chen, Chin-Teng Lin, *Fellow, IEEE*

***Abstract*—*Objective.*** **Error-related potential (ErrP)-based brain-computer interfaces (BCIs) have received a considerable amount of attention in the human-robot interaction community. In contrast to traditional BCI, which requires continuous and explicit commands from an operator, ErrP-based BCI leverages the ErrP, which is evoked when an operator observes unexpected behaviours from the robot counterpart. This paper proposes a novel shared autonomy model for ErrP-based human-robot interaction. *Approach.* We incorporate ErrP information provided by a BCI as useful observations for an agent and formulate the shared autonomy problem as a partially observable Markov decision process (POMDP). A recurrent neural network-based actor-critic model is used to address the uncertainty in the ErrP signal. We evaluate the proposed framework in a simulated human-in-the-loop robot navigation task with both simulated users and real users. *Main results.* The results show that the proposed ErrP-based shared autonomy model enables an autonomous robot to complete navigation tasks more efficiently. In a simulation with 70% ErrP accuracy, agents completed the task 14.1% faster than in the no ErrP condition, while with real users, agents completed the navigation task 14.9% faster. *Significance.* The evaluation results confirmed that the shared autonomy via deep recurrent reinforcement learning is an effective way to deal with uncertain human feedback in a complex human-robot-interaction task.**

## I. Introduction

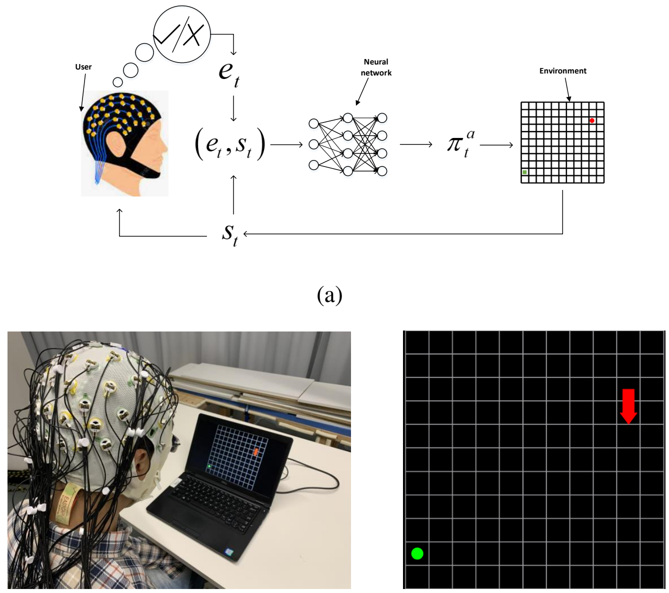
Error-related potential (ErrP)-based brain-computer interfaces (BCIs) have been widely used in human-robot interactions in recent works [1, 2]. The ErrP is an event-related potential (ERP) that are involuntarily evoked when a human perceives unexpected errors in an environment [3, 4]. The ErrP phenomenon was first reported in choice-reaction tasks [5]. After the participant was aware of an erroneous response made by herself, a negative potential approximately 80 ms and a sustained positivity in the time interval between 200 and 500 ms were observed [3, 6]. It was later found that ErrP was also evoked 250 ms after the user observed an unexpected event in the external environment [4]. Due to the nature of ErrP signals, this type of brain activity is particularly useful as supervision or feedback signals during human-robot interactions tasks. ErrP signals can enhance the scalability of a system in cases in which a user can assess a device's actions as correct or incorrect. The agent takes advantage of the implicit brain signals acquired from the human user when determining the appropriate agent action. Thus, the human user does not need to explicitly send action commands, significantly reducing the burden on the human user [7, 8]

The shared autonomy in human-robot interaction leverage the strengths of both human and robots, where robots can no longer act solitarily, but must share part of their autonomy space with human. In most traditional shared control tasks, the user needs to provide explicit input, such as keyboard or mouse commands [9–11], during interactions. BCI systems offer new channels that allow shared autonomy by integrating user intent directly according to the ongoing brain activity, thus eliminating the need to exploit muscular control [12, 13]. The use of shared-autonomy schemes may allow error-related potentials to be used as complementary signals in BCI systems. Due to the natural uniqueness of ErrPs, ErrP-based shared autonomy can leverage the advantages of human-robot collaboration without interrupting the user's main workflow. However, due to the uncertainty of EEG signals, a direct mapping of ErrPs to robot actions is not sufficient for optimal behavior. For example, a misclassification of EEG signal will lead wrong robot action. On the other hand, to train a shared autonomy model via deep neural network need a large data set. But real ErrPs data collections can be very time-consuming [14] and have other drawbacks, such as overfitting if there is not enough data.

In this paper, we propose a shared autonomy framework that incorporates ErrP-based BCI via deep recurrent reinforcement learning. Considering the uncertainty of ErrP, we formulate the shared autonomy as a Partially Observed Markov Decision Process (POMDP). Unlike the Markov Decision Process (MDP), where the agent decides actions based on the direct observation of the full underlying state, POMDP allows the agent to make optimal decisions based on a history of partial observations or uncertain inputs [15, 16]. We consider the uncertainty of the ErrP signal similar to an agent's imperfect sensing of the environment. A BCI module might incorrectly infer the user's intention because of a noisy ErrP signal; similarly, a robot might wrongly identify the direction of an arrow sign due to the noisy image captured from an imperfect camera module. In other words, observations of the actual environmental state could differ and be represented using probabilistic models [17, 18]. Thus in our experiment, instead of real EEG data, we simulate ErrP as a binary input of 0 or 1 and represent its uncertainty as a Bernoulli distribution with a probability $P$ of observing the true state.

Similar with previous works [12, 19–21], an agent accumulatively changes the decision probability over time, in this

This work was supported in part by the Australian Research Council (ARC) under discovery grant DP180100656, DP210101093, and DP220100803. Research was also sponsored in part by the Australia Defence Innovation Hub under Contract No. P18-650825, US Office of Naval Research Global under Cooperative Agreement Number ONRG - NICOP - N62909-19-1-2058, and AFOSR – DST Australian Autonomy Initiative agreement ID10134. We also thank the NSW Defence Innovation Network and NSW State Government of Australia for financial support in part of this research through grant DINPP2019 S1-03/09 and PP21-22.03.02.




(a)

(b) (c)

Figure 1: An overview of our method for ErrP-based real-time shared autonomy and deep reinforcement learning, where the user's ErrP and robot observation of the environment were concatenated as the neural network input(a). We evaluated our method in a navigation task with real human participants (b). The red arrow with an arrow indicates the agent, and the green dot indicates the target (c).

paper, we use recurrent neural networks (RNN) to approach the POMDP. The RNN is an approach that involves stacking the memory history and is robust to partial observations [22]. To solve the neural network training issue with a large data set, we use binary value (0 or 1) to simulate the decoded results of the ErrP classifier, instead using real EEG data to train the model. This simulation enables us to train our model without real users. Our approach builds upon the shared autonomy framework [9] As shown in Figure 1, we apply our method in a navigation task. Our studies with both simulated users and real human participants suggest that ErrP-based shared autonomy can successfully improve task performance.

Our contributions in this work can be summarized as follows:

- A novel ErrP-based reinforcement learning for shared autonomy.
- Demonstration the feasibility of the proposed shared control paradigm with simulated ErrP.
- Evaluation the ErrP-based shared autonomy with real human participants in a navigation task with a pretrained shared autonomy model.

## II. Related works

### A. *ErrP-based BCI for Human-Robot Interaction*

Recently, the ErrP-based BCI has been widely used in Human-Robot Interaction tasks[1, 23, 24]. Salazar et al. [1] proposed a closed-loop system that used the ErrP as an implicit input to guide a robotic arm in a binary bin-sorting task. Kim et al. [2] used the ErrP as an implicit reward of a robot to learn the mapping between human gestures and actions. Stefan K. Ehrlich et al. [23] demonstrated the applicability of ErrPs as human feedback signals for real-time mediating coadaptation in human-robot interactions. Lopes-Dias et al. [24] showed the feasibility of online asynchronous decoding of ErrP signals and used the resulting decoded signals as feedback to guide a robotic arm towards a target after the robot was halted at an unexpected moment. These works show that ErrPs can be used to decode human intention during human-robot interactions.

### B. *Shared autonomy using BCI*

Shared control is a widely used technology in human-robot interactions. BCI systems provide new channels that allow shared control by integrating the user intent directly according to the ongoing brain activity, eliminating the need to exploit muscular control [12, 13, 25]. Various methods have been used in BCI-based shared autonomy systems. Previous studies [26–28] have proposed flexible self-paced BCI systems that switch between automatic and subject control methods. While the switch model is efficient, only one control command can be executed at a time. Thus, this kind of method cannot take advantage of both human inputs and robot autonomy. Some research [29, 30] has used shared control in hierarchical systems, with the brain signal providing high-level commands via BCIs as the robot performs low-level tasks, such as grasping, navigation, and manipulation. In [30], steady-state visually evoked potentials (SSVEPs) were used to select a target while a robot arm performed a specific grasping action. However, this shared control method subdivides tasks into separate modules for the human user and the robot. Recently, deep reinforcement learning (RL) frameworks incorporate user inputs and agent observations to achieve shared autonomy [9]. This shared control scheme opens the door to the use of ErrP signals as an alternative or complementary signal in BCI systems.

### C. *ErrP-based Human-in-the-loop reinforcement learning*

ErrP has been widely used in human-in-the-loop RL systems. In these systems, the ErrP signal is used as a positive or negative reward to accelerate the training of autonomous agents [31–33]. In [19], ErrPs was used as negative reinforcers of the actions to infer the optimal control strategies. In [20], ErrP was used to learn the reward function in an inverse reinforcement learning control to the robot to avoid obstacles. In [12], inverse RL based on ErrP signals was used to infer the goal position in a virtual grid. In [31], ErrP-based RL was used to update the reward to determine a policy in a route learning strategy. ErrP has also been used in RL to choose the correct target among several possible targets. In [34], ErrPs served as the reward in a reinforcement learning approach to train an intelligent neuroprosthesis controller. The objective in this work was to improve the control policy. In [2], ErrP was used to train a robot to learn human gestures through a reinforcement learning strategy based on the leap motion and ErrP features. However, when the ErrP signal was used as a reward, while the ErrPs accelerated learning, the signals operated independently of the system during testing [31–33].

Unlike others works where human-in-the-loop reinforcement learning frameworks leverage human feedback to train autonomous agents that operate independently of the user at test time [31–33], In our paper, We combine user input (ErrP) and robot observation as inputs of the deep model for mapping optimal actions. The shared autonomy will always need to leverage user input to accomplish the task both at training and test time.

### D. Formulating Human-Robot Interaction as POMDP

A POMDP can handle sequential decisions with various uncertainties arising from human feedback errors and sensing noise. The POMDP formulates a problem in which the state measurements are partial observations in sequential decisions. Recently, the POMDP has emerged as a popular approach in human-robot collaboration tasks [35–38].

In [35], human-robot collaboration was formulated as a POMDP by characterizing the robot's information and human's intention as the state space. In [36], human-robot collaboration was formulated as a POMDP to learn the human model via Bayesian nonparametric learning to determine the human state. Moreover, in [37], the observation model, dynamic machine model, and human model were combined in one framework and formulated as a POMDP model for the human-in-the-loop system. In [38], human-computer interactions were formulated as a consequence of a POMDP and used to model human perception during interactions. In summary, the POMDP does not assume that the system state is fully observable, and the POMDP's ability to represent uncertainties arising from different sources makes it a suitable model in human-robot collaboration applications. In our paper, ErrP uncertainty is represented by a Bernoulli distribution with a probability $P$ of observing the truth. As a result, our system can be considered a partially observable Markov decision process with uncertain observations. The POMDP allows for optimal decision-making under uncertain input conditions.

## III. Method

### A. Overview

In this section, we first introduce background knowledge on the POMDP. We then introduce the ErrP-based shared framework, neural network architecture and reinforcement learning, task environment, and input feature to the neural network.

### B. POMDP background

A Markov decision process (MDP) assumes that an agent can fully observe an environment. Otherwise, the agent senses the environment with limited or uncertain observations. If the observations are uncertain, the state signal is no longer Markovian, violating a key assumption of most reinforcement learning techniques [39]. A POMDP allows for optimal decision making even when the agent's observation is partially [16]. A partially observable Markov decision process is a tuple $\langle S, A, \Omega, T, O, R\rangle$ in which S is a finite set of states, A is a finite set of actions, $\Omega$ is a finite set of observations, T is a transition function defined as T: $S \times A \times S \rightarrow [0,1]$, O is an observation function defined as O: $S \times A \times \Omega \rightarrow [0,1]$, and R is a reward function defined as R: $S \times A \times S \rightarrow R$.

The discrete set of observations $\Omega = \left\{o^1, \ldots\ldots, o^M\right\}$ represents the agent's observation, which depends on the next state $s^{'}$ and is sometimes conditioned on its action $a$. This set can be determined with the observation function O: $S \times A \times \Omega \rightarrow [0,1]$. The probability of observing o in state $s^{'}$ after an action is $O\left(s^{'}, a, o\right)$. This requires that $O\left(s^{'}, a, o\right) \geq 0$ and $\sum_{o \in \Omega} O(s^{'}, a, o) = 1$. In our paper, the discrete partial observation is $\Omega = \{0, 1\}$, which represents the decoded ErrP result. The probability follows the Bernoulli distribution. If $P = 0.7$, the probability can be modelled as follows: $O\left(s^{'}, a, o^1\right) = 0.7$, $O\left(s^{'}, a, o^2\right) = 0.3$, or $O\left(s^{'}, a, o^2\right) = 0.7$, $O\left(s^{'}, a, o^1\right) = 0.3$. In this case, the agent has a 70% chance to observe the true environment state. Thus, an agent with uncertain ErrP feedback conforms to Partially Observable Markov Decision Processes.

### C. ErrP-based framework

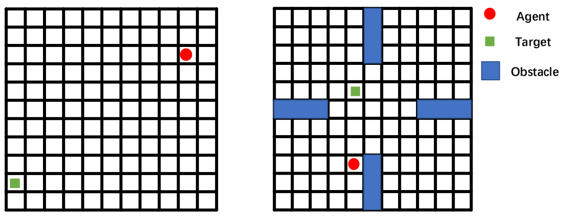
The classification of ErrP signals collected from humans is not perfect due to misclassification. ErrP uncertainty can be regarded as an agent's imperfect sensing of the true state of the environment. We use a deep reinforcement learning agent that maps observations from sensors (including ErrP) to actions. We incorporate the ErrP information provided by a BCI as useful observations for the agent. Our method jointly embeds the ErrP information $e_t$ acquired from the user and the agent's observations of the environment $s_t$ by concatenating the values.

$$\tilde{s}_t = \begin{bmatrix} e_t \\ s_t \end{bmatrix}$$

### D. Network architecture and reinforcement learning

Our network architecture builds on the one proposed by Sutton et al. [40]. The actor consists of 64-bit gated recurrent units (GRUs) that use fully connected layers to process the input and produce the output values of the hidden states, $h_t^a$. The action probabilities are produced by the final layers, $z$, via a bounded softmax distribution: $P(u) = (1-\varepsilon) soft\max{(z)}_u + \varepsilon / |U|$, where $\varepsilon / |U|$ lower-bounds the probability of any given action. We anneal $\varepsilon$ linearly from 0.5 to 0.05 across 5500 training episodes and set it to 0 during the test. The critic is a feedforward network with multiple ReLU layers and fully connected layers.

We choose the widely used advantage actor-critic (A2C) algorithm [41, 42] to stabilize the training by reducing the variance. We train the critic with this policy to estimate the Q value using TD($\lambda$) [41], which is adapted for use in deep neural networks. We train the actor with advantage function $A(\tau^a, u^a) = Q(\tau^a, u^a) - V(\tau^a)$, where $Q(\tau^a, u^a)$ is action value function and $V(\tau^a)$ is value function. The update direction is defined by the gradient $g = E_T\left[\sum_{t=0}^{T-1} \nabla_{\theta\pi} \log \pi\left(u_t \left| s_t\right.\right) G_t\right]$, where




(a) (b)

Figure 2: The environment without obstacles (a) and obstacles (b).

$G_t$ is empirical returns. At each time step, the policy architecture is fed the ErrP, agent's local observation and step number and is tasked with estimating the Q-value function and policy at each point.

### E. Task statement

We test our method in two environments. As shown in Figure 2, the navigation environment is described by a grid map. The first environment is a grid map without obstacles, and the second environment is a grid map that includes several obstacles. The layout of the map and the positions of the obstacles were fixed during training and testing. The second environment simulates a real-world environment where an agent's observation is blocked by obstacles. The use of two environments demonstrates the generalizability of the proposed shared autonomy framework.

The size of the grid was $11 \times 11$. The locations of the robot and the target were simplified as grid coordinates. The horizons of the robot were limited to the four corners of its neighbourhood. The robot can move north, south, west, or east during each time step. The robot cannot move towards the barriers or out of the grid. The robot is surrounded by a $1 \times 1$ horizon in which it can detect the target. The agent's task is to identify the goal location within the map. The agent has a limited sensing range that is assumed to be substantially smaller than the size of the maze. The target will be detected when the target is in the sensing range. The goal location and agent start position are randomized (spawned) in a constant static map in each episode during training and testing. After the goal is achieved, a new episode begins. To encourage short trajectories, each time step has a step cost (penalty) of 0.01. A typical sparse terminal reward (20) and the step cost are provided to encourage the agent to reach the target position in the minimal number of steps.

### F. Input features of the neural network

The input features include ErrP feedback from the human user and the agent's observations of the environment. The agent's observations of whether the target is in its current position and four adjacent positions. The step number and most recent agent action are also included as features. All features are normalized by their maximum values. Information about the target position was not included in the input.

*1) Last action:* The coupled action is a useful input feature because the ErrP signal is cued by the agent's last action.

*2) Mark of visited grads:* For a stationary target, an optimal search strategy is trivially represented by a path that attempts to cover the entire environment without revisiting any location. The marker was often used as a reward in learning a policy to encourage the agent to explore unvisited locations [43, 44]. However, the uncertainty of ErrP feedbacks could cause the agent to make incorrect decision. In such case, revisiting an explored location might allow a correction. Indeed, we found that the use of the visited marker as a reward limited the optimal policy and thus yielded slightly suboptimal policies. We found that using visited marker as an input allows the model to learn an optimal strategy.

*3) ErrP information:* To eliminate the gap between the simulated EEG data and the real EEG data collected from a human user, we simplified the EEG data as a binary variable, which corresponds to the decoding output of the ErrP classifier. During model training, we use the binary values 0 and 1 to simulate the output of the ErrP binary classifier. To generate the ErrP values, we calculated the shortest path towards the target position at each step. The shortest path [45] was computed according to the full map environment. This approach follows a environment in which the human user has a global view of the environment. If the current shortest path is larger than the previous shortest path, we considered the current step to be bad action and assigned an ErrP label of 1; otherwise, we assigned an ErrP label of 0.

## IV. Experiment 1: Simulated users

We begin our experiments with simulated users. Then, we evaluate the shared autonomy with real human participants. We use a binary value (0 or 1) to simulate the decoded results of the ErrP classifier.

### A. Experiment Design

We first consider the ErrP as a full observation with 100% accuracy and then consider ErrP as a partial observation with different accuracy levels. We use an autonomous agent without ErrP feedback as our baseline. Our central hypothesis is that our method can improve the agent's performance despite the partial ErrP observations. We use simulated pilots, which enables us to more thoroughly consider different aspects of our method (such as the effects of the ErrP accuracy level on training an effective shared control model and gradient analyses with different ErrP accuracies at various positions). Moreover, we use a simulated ErrP to train the shared control model that is used to test with real human users.

*1) Partially observable ErrP and without ErrP:* We first trained an autonomous agent without ErrP feedback as the baseline. We then trained six agents receiving ErrP feedback with different levels of accuracy ranging from 65% to 100%. We evaluated the agents in 20000 episodes with random starting and target positions. Figure 3a shows the training curve of the agents and Figure 3b shows the average number of steps used by each agents to reach the target.

*2) Trained with full observation and evaluated with partial observation:* To test the robustness of the POMDP model to uncertainty, we compared two model: one was trained with partial observations (75% ErrP accuracy) and another one was trained with full observations (100% accuracy). We evaluated the models with incrementally more complete observations (ranging from 70% to 100% accuracy).

*3) Gradient analysis on ErrP with different observation levels:* The gradient computes the derivatives of the outputs of a model with respect to the input variables and identifies which input variables are important for predicting the outputs. The gradient-based method is a natural and popular attribution method [46] for explaining deep neural network decisions. This method uses the learned model to determine how important the input dimension is for the output. To better understand the mechanisms that allow the POMDP model to perform well in uncertain environments, we analysed the performance and gradients of POMDP models with different accuracies. More specifically, we compared the gradient at 70%, 75%, and 80% accuracies. We found that when the ErrP accuracy is greater than 80%, the learned policy is the same as that learned when the accuracy is 100%. This result indicates that the ErrP gradients are the same when the accuracy is greater than 80%. Therefore, we compared the gradients of models trained with accuracies less than 80%.

*4) ErrP gradient analysis at different positions:* During the test, we found that ErrP has a greater effect on the outputs in the central area than on those in the edge area. We visualized the ErrP gradient map of the model in the two environments to assess whether the ErrP has different effects on the outputs at various positions. The computation of the gradient map is extremely quick since it requires only one backpropagation pass. The gradient map encodes the effect of the ErrP signal on the agent's action at different locations. The colours represent different gradient values.

*5) Agent performance analysis:* The performance was operationalized according to the step number and the success rate of the agent in reaching the target position from the start position. We compared the performance of agents with and without ErrP feedback. Even without human assistance, the agent would eventually reach the target. To evaluate the search ability of different distance ranges, we varied the initial distance between 2 and 20. Each distance was evaluated over 10000 runs. We compared the agent performance in the two environments with no ErrP, 70% ErrP accuracy, and 80% ErrP accuracy.

## B. Result

*1) Partial observation ErrP and without ErrP:* Figure 3 shows that the agent with 100% accurate ErrP feedback performs better than the baseline agent without ErrP feedback. The result also suggested that higher ErrP accuracy corresponds to fewer steps required to reach the target position.

*2) Trained with full observations and evaluated with partial observations:* Figure 4 shows the average number of steps used during the test with models trained with full observations (100% ErrP accuracy) and partial observations (75% ErrP accuracy). The average number of steps decreased as the correct probability increased for both conditions. However, when the accuracy was less than 80%, the model trained with partial observations used fewer steps than the model trained with full observations. In contrast, when the accuracy was greater than 80%, the model trained with partial observations used more steps than the model trained with full observations. The POMDP model allows the performance to scale linearly as a function of the observation quality. Note that when the accuracy was 70%, while both models exhibited a reduced performance, the MDP model decreased to approximately 40 steps, while the POMDP model decreased to approximately 27 steps. The performance of the model trained with full observations declined considerably when presented with incomplete observations. When the accuracy was 100%, the POMDP model used approximately 12 steps, reaching near-perfect levels (approximately ten steps).

*3) Gradient analysis of ErrP with different observation level:* As shown in Figure 5, the ErrP gradient increases as the ErrP accuracy increases. This result indicates that more accurate ErrPs have more important effects on the outputs than ErrPs with low accuracy. In contrast, when the ErrP has a larger effect on the output, the effect of other input variables on the output should be decreased. In other words, the gradients of the agent's observations, such as the position variables, decrease. As shown in Figure 5, the position gradient decreased as the ErrP accuracy increased. These results demonstrate that human feedback gradually induces more effects, while agent observations have fewer effects, as the ErrP accuracy increases during training.

*4) ErrP gradient analysis at different positions:* Figure 6 shows model gradient maps of the two maze environments. In general, the ErrP gradient is large in the central area and small in the edge areas, which indicates that the ErrP has a substantial effect on the central position. In other words, the agent rely more on human feedback in central area than in edge area. In future research, more advanced interpretation methods, such as integrated gradients [47] and SmoothGrad [48], could be used for further analysis.

*5) Agent performance analysis:* The average number of steps were 51.2, 34.8, and 24.0 for the no ErrP condition, 70% accurate ErrP condition and 80% accurate ErrP condition in environment 1 and 50.7, 40.8, and 25.7 in environment 2, respectively. The average number of steps gradually increased as the initial distance increased for both the ErrP conditions and the no ErrP condition in environments 1 and 2.

Sixty steps was taken as the maximum number of steps; that is, if the agent successfully reaches the target position within 60 steps, it is considered a success. Otherwise, the agent has failed. Figure 7 shows the success rate to reach the target position within 60 steps for each initial distance. The success rate gradually decreased as the initial distance increased. The average success rates were 79.74%, 83.19% and 95.86% for the no ErrP, 70% accurate ErrP and 80% accurate ErrP conditions for environment 1 (Figure 7a) and 69.43%, 76.70% and 94.21% for the no ErrP, 70% accurate ErrP and 80% accurate ErrP conditions for environment 2 (Figure 7b). The success rate gradually decreased as the initial distance increased for both the ErrP conditions and the no ErrP condition in environments 1 and 2, except when the initial distance was the maximum value for the no ErrP condition in environment 2. The start and target positions were limited to the four corner positions, which allow the initial distance to be maximum value. We found that when the agent start position was (0, 0), the agent moved in the direction of the opposite corner. If the agent start position and target position were (0, 0) and (10, 10), 20 steps were used to reach the target position, which is the optimal number of steps. These results indicate that even when the ErrP signal is 70% accurate, the success rate is higher than the success rate in the no ErrP condition.

### C. Discussion

*1) Performance of the shared control model:* The simulation experiment indicates that the shared control model can greatly improved the task efficiency compared with autonomous agent. As shown in Figure 7, even when the human feedback was partially inaccurate, the success rate of 70% and 80% of ErrP accuracy on the success rate to reach the target position is larger than in the case of no ErrP. The integration of the agent observations and human perception help the agent gains more information about the environment than an autonomous agent. Besides, the higher success rate of 80% ErrP accuracy compared with 7% one demonstrated that the agent can make better decisions with more accurate observation of the environment.

In the shared control policy with 100% accuracy, we found that if the ErrP signal is provided, the agent changes its search direction to the left in an anticlockwise search approach, as shown in Figure 8a. The agent changes its trajectory in real-time to adapt to the human feedback as previous study [49]. The same performance was observed when the ErrP accuracy was greater than 80%. However, when the ErrP accuracy was less than 80%, the agent learned a different accuracy. In this case, the agent did not change its search direction immediately after an ErrP signal was provided. Instead, the agent changed its direction when it was more confident. Thus, we hypothesize that the confidence level is related to the ErrPs of the previous steps and the current position. For the no ErrP condition, the agent's trajectory followed an anticlockwise search. The trajectory was fixed and depended only on the agent's starting point, as shown in Figure 8b.

*2) ErrPs with different accuracies:* We investigated the performance of agents trained with various noise levels during training. We provide the input accuracy during training. Figure 3a shows that the agent learned different policies during training with different ErrP accuracy levels, demonstrating that the ErrP accuracy could be learned by the model during training. Figure 3b shows that the less uncertain the human feedback, the better decision the agent can make. In addition, we investigated the threshold of the ErrP accuracy that is sufficient for training an efficient shared control model. We found that if the ErrP accuracy is greater than 70%, the model trained with this ErrP accuracy performs better than an autonomous agent. However, if the ErrP accuracy is less than 70%, the shared control performance was not considerably different from that of a sole autonomous agent, as the sole agent could learn a search policy without human feedback. Thus, we take 70% as the threshold for training an effective model. This result provided a new perspective on human feedback accuracy in shared control critic models. Therefore, we selected participants with offline accuracies greater than 70% for the online test.

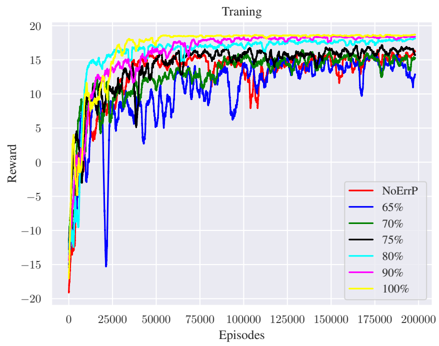
*3) Model robustness:* As shown in Figure 4, the model trained with high ErrP accuracy was more sensitive to ErrP input than the other models. The agent is more likely to change its search direction when the human user provides negative feedback. In other words, human users have a more significant effect on the agent's action in a more accurate ErrP model than in a less accurate ErrP model. The performance declines dramatically when using the model trained with full observations and tested with partial observations. However, the model trained with partial observations is more capable of handling partial observability when the observation quality changes during the evaluation. The results are consistent with the results of [22]: the model trained with partial observations is robust towards missing game screens and remains scalable, improving the performance as more data become available. Furthermore, the model trained with partial observations was more robust to uncertainty during evaluation, despite the fact that the two learned models used the same neural network architecture. In addition, the model is scalable enough to improve performance as the observation accuracy increases. Therefore, during the test with real human participants, we chose the shared control model trained with ErrP accuracy, which is similar to real EEG classification accuracy with cross-validation.

## V. Experiment 2: Real-World User Study

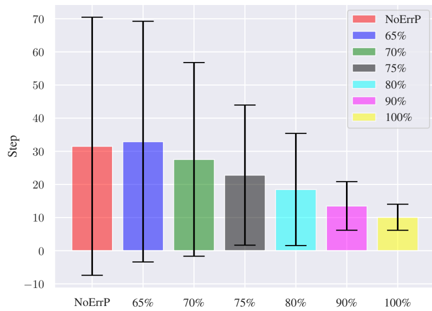
In this section, we evaluate our method during the test phase with real human participants. Our model was pretrained with simulated EEG data. We want to validate the feasibility of using the model trained on simulated EEG data with real human participants in the same task environment. We validate the feasibility of the learned model in two environments: a environment without obstacles and a environment with obstacles.

### A. Experiment Design

*1) Interaction environment design:* To evoke ErrP signals, the interaction environment, especially the stimulus, needs to be carefully considered [50]. The environment design was based on the design presented in [21], which includes a grey grid with a red agent and a green target on a black background. The agent's start and target positions were generated under the condition that their distance be larger than one grid (the agent's observation ability). At each step, the agent moved from its current position to one of the four adjacent positions. A 1 s animation within the agent served as a countdown to draw the participants' attention. The agent then jumped instantaneously to the next position, with an arrow directed towards the position. This arrow remained visible for 1 s.




(a)




(b)

Figure 3: Training curve with different ErrP accuracies conditions as well as no ErrP condition (a). The average number of steps used to reach the target position (b).

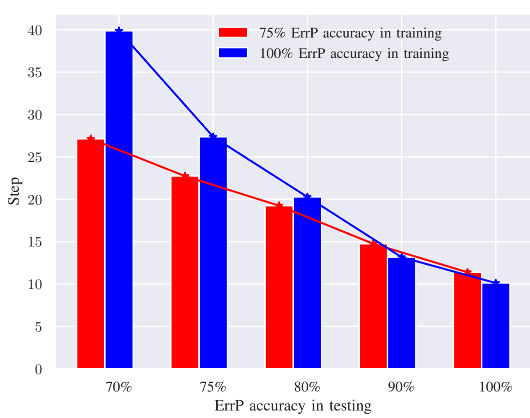



Figure 4: The average number of steps used to reach the target position for different accuracy levels with models trained on partial and full observations.

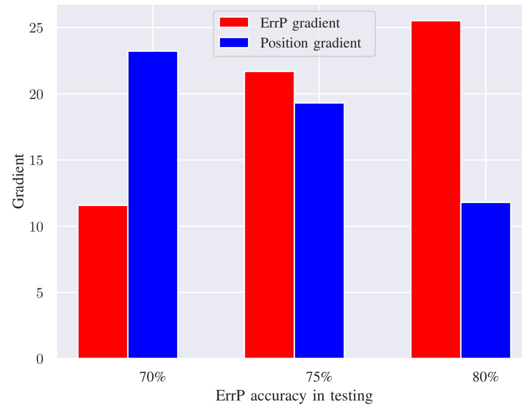



Figure 5: ErrP and position gradients with different ErrP accuracies.

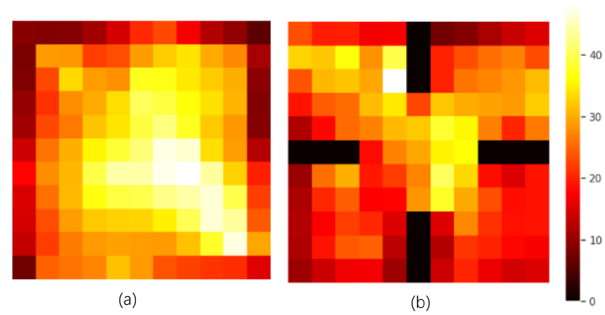



Figure 6: ErrP gradient distribution at different positions in the maps of the two environments. The color indicates the gradient value at different positions

Then, the highlights disappeared, and the agent remained at its new position for 1 s before its next step.

Before the real-time control experiment, participants were first asked to perform five blocks of 120 trials in environment 1, which contained no obstacles. The agent's initial and target positions were randomly generated. If the agent did not reach the target position after 60 trials, a new run was started. The EEG data collected during these five blocks were used to calibrate the classifier. If the agent's action decreased the distance to the target position, the action was labelled "correct"; if the agent's action increased the distance to the target position, the action was labelled "error". During the experiment, the participants were asked to mentally judge whether the agent's action was correct or an error.

*2) Participants:* Sixteen participants (average age $28.57 \pm 3.11$ years old, two females) participated in the experiment. Seven participants participated in both the offline-BCI and online-BCI experiments. Seven participants participated in only the offline experiment, as their ErrP BCI performance were below the 70% threshold. As described in Section IV.C, the shared autonomy model performs better only when the ErrP classification accuracy is greater than 70%. The remaining two participants were excluded from further analysis, as the participants could not complete the online experiment due to battery power issues. All participants provided informed consent for the study, which was approved by the University of Technology Sydney (UTS) Human Research Ethics Committee (ETH19-3830). All participants had normal vision and did not report any known neurological or psychiatric diseases.

*3) EEG recording and pre-processing:* EEG signals were recorded from 64 locations according to the extended 10/20 system using a LiveAmp wireless EEG system from Brain Vision [51] with a sampling rate of 500 Hz. The reference channel was placed at the FCz channel position, and the ground channel was placed at the forehead position [51]. The signal was resampled to 256 Hz and filtered using a finite impulse response (FIR) bandpass filter with cut-off frequencies 1-50 Hz.. Then, the common average reference was used to reduce signal contamination. Both offline training and online testing used a same EEG signal pre-processing pipeline

*4) ErrP Feature extraction:* Temporal features extracted from time-series data have been used in many ErrP activity studies [1, 50, 52, 53]. It has been reported that the classification results of temporal features are better than those of spectral features for decoding ErrP signals [54]. Thus, temporal features were used for classification in this study. Similar with studies [21, 24], the averaged signal amplitude within a 30-ms-long window between 150 ms and 600 ms at each trial and channel was extracted. Thus, during the time window from 150 ms to 600 ms, there will be $15 = ((600 - 150)/30)$ samples for one channel. The classification between correct and error feedback was performed from all 64 EEG electrodes [21]. Thus, the feature vector length is 64*15=960.

*5) ErrP classifier training:* To enable real-time detection of neural activity during each trial, the classifier must be calibrated to classify the EEG waveform as ErrP or non-ErrP. This ErrP classification is a binary classification task that indicates the agent's action as correct or incorrect.

To minimize overfitting effects, we used tenfold cross-validation to train the classifier with 90% of the data, and the remaining 10% of the data were used for testing. The extracted features include redundant features, and traditional linear discriminant analysis (LDA) has limited flexibility for complex features. Thus, it is necessary to search for a subset of the available features that can improve the classification performance. Shrinkage and selection methods are commonly used feature selection methods. We use shrinkage LDA [55] as the classifier in our paper, which is widely used for decoding ErrP signals [50].

A binary linear classifier can be characterized by a projection vector $\mathbf{w}$ and a bias term $b$ referring to the separating hyperplane $\mathbf{wx} + b = 0$ . The projection vector of LDA is defined as:

$$w = \boldsymbol{S}_w^{-1}(\boldsymbol{u}_a - \boldsymbol{u}_b) \tag{1}$$

Where $\boldsymbol{S}_w^{-1}$ is the covariance or within class variance, $\boldsymbol{u}_a$ and $\boldsymbol{u}_b$ is the mean value of class A and class B.

The empirical covariance of the above is unbiased and has good properties when the number of observations is greater than the dimensionality of variables. However, for high-dimensional data with only few data trials, the estimation covariance may become imprecise because the covariance of matrix estimate is singular and the inverted matrix in imprecise. This phenomenon leads to a systematic error: large eigenvalues of the original covariance matrix are estimated too large, and small eigenvalues are estimated too small [56]. This estimation error makes the performance of LDA in high-dimensional situations far from optimal. Shrinkage is a common method that compensates the systematic bias in the estimated covariance matrix by a regularized covariance matrix $\boldsymbol{S}_b$ :

$$\boldsymbol{S}_b = (1 - \lambda)\boldsymbol{S}_b + \lambda \boldsymbol{D} \tag{2}$$

Where $D$ is a diagonal matrix taking the diagonal elements of $\boldsymbol{S}_b$. Thus, the parameter $\lambda$ forces the extreme eigenvalues towards average [56].

*6) Online test with real participants :* The participants who achieved ErrP accuracy threshold of 70% further the online test. The shared control model that best matched the participant's offline accuracy was used in the online test. For example, if the participant's offline ErrP classification accuracy is 78%, we chose the shared control model that pretrained with 80% ErrP accuracy. The computational cost for the training model is about 40 hours and 30 munites, which is running on a workstation with two Intel Xeon 6132 CPUs and NVIDIA RTX 6000 GPUs, as well as 96GB of RAM.

## B. Results

*1) Electrophysiology analysis:* Figure 9 shows the correct, error, and difference grand average potentials (error minus correct averages) in the Fz channel averaged across all subjects in the online sessions for both environments. The difference grand average was characterized by three components: a negative deflection at approximately 200 ms, a positive deflection at approximately 300 ms, and another negative component at approximately 400 ms.

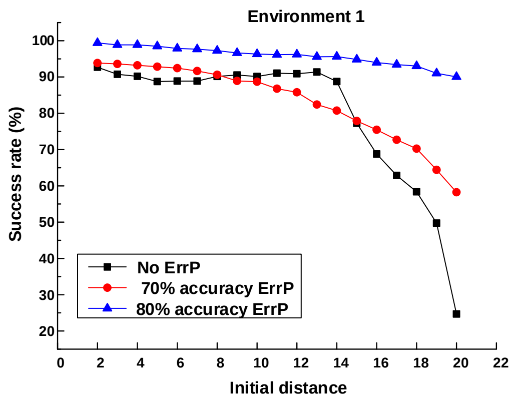
*2) Classification analysis of ErrP:* In this section, we analyse the real-time classification accuracy of the ErrP signals with the classification model calibrated with offline data for the two environments. As mentioned in the simulation section, if the ErrP accuracy is less than 70%, the low ErrP classification accuracy and no ErrP models perform similarly. Table I shows the offline training accuracy using 10-fold cross-validation and the online test accuracy for the two environments. The overall offline training accuracy was 76.65%. The overall online test accuracy were 73.22%, 69.20% for environment 1 and environment 2, respectively.

*3) Agent performance analysis:* In this section, we analysed the success rate and number of steps for real human users to evaluate the feasibility of the shared control model with real human participants. To test the model's target search ability for different initial distances, we chose episodes with initial distances between 2 and 20 (maximum), resulting in a total of 19 episodes with random sequences for each environment. The episodes were pregenerated for all the participants.

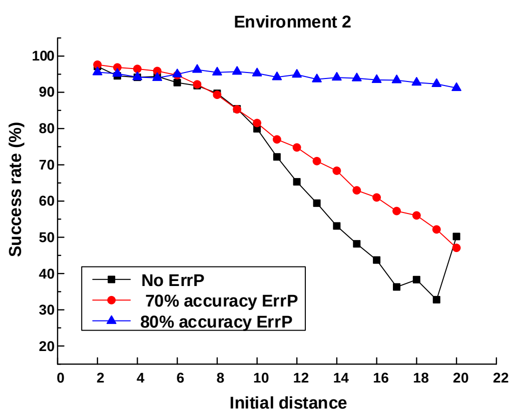
During the online test, the brain signal's classification of the agent's last action as either correct or an error was fed

| Participant | Offline training (%) | Online test of environment 1 (%) | Online test of environment 2 (%) |
|---|---|---|---|
| S2 | 73.33 | 77.68 | 77.73 |
| S4 | 72.83 | 60.88 | 58.02 |
| S5 | 85.17 | 94.75 | 65.02 |
| S6 | 74.67 | 69.01 | 66.54 |
| S8 | 76.17 | 71.91 | 66.11 |
| S10 | 80.17 | 71.81 | 70.25 |
| S15 | 75.33 | 64.28 | 70.11 |
| Average | 76.65 | 73.22 | 69.20 |

Table I: ErrP training accuracy with 10-fold cross-validation and test accuracy for the two environments.

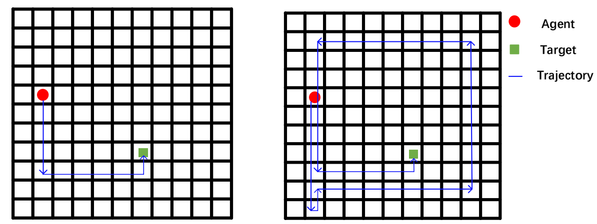



(a)




(b)

Figure 7: Success rate of reaching the target position within 60 steps for each initial distance with the no ErrP, 70% accurate ErrP, and 80% accurate ErrP conditions for environments 1 and 2.




(a) (b)

Figure 8: The agent search policy with 100% accurate ErrP (a) and no ErrP (b).

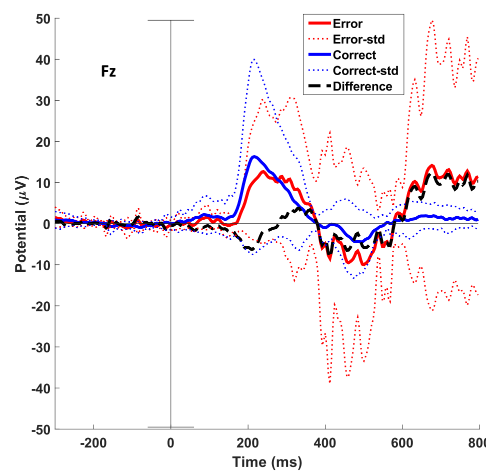



Figure 9: ERP analysis for the correct and error conditions, averaged over the online trial sessions at Fz channel by removing baseline [-300 0]ms. The black line is the difference between correct and error condition. The red and blue dotted lines are the standard deviation for the error and correct conditions respectively.

into the model in real-time to generate the next action. If the agent did not reach the target position after 60 steps, the run was ended, and a new episode was started. The maximum number of steps was set to ensure that participants were not discouraged by long runs. In the experiment with real human participants, the maximum number of steps was set to 60 for each episode. Therefore, episodes with more than 60 steps were not included when calculating the average number of steps.

As shown in Table II, the success rate to reach the target position within 60 steps was 81.20%, and this value was averaged over all participants. The average number of steps was 24.87, which was averaged over all participants by removing failed episodes. The success rate was approximately the same as the success rate of the 70% accurate ErrP condition (83.19%) and was larger than the success rate of the no ErrP condition (79.74%) in the simulations. The average number of steps was almost the same as the number of steps in the 70% accurate

| | Success rate (%) | | Mean and standard of number of steps | |
|---|---|---|---|---|
| | Env 1 | Env 2 | Env 1 | Env 2 |
| S2 | 94.74 | 89.47 | 23.11±13.2 | 20.06±14.38 |
| S4 | 78.95 | 84.21 | 26.07±16.56 | 26.88±13.97 |
| S5 | 94.74 | 84.21 | 21.06±11.73 | 22.69±12.93 |
| S6 | 63.13 | 63.16 | 29±18.69 | 28.42±15.78 |
| S8 | 78.95 | 89.47 | 24.8±14.57 | 17.53±9.91 |
| S10 | 84.21 | 94.74 | 24.94±15.74 | 23.61±12.81 |
| S15 | 73.68 | 89.47 | 28.86±17.59 | 21±13.35 |
| Ave | 81.20 | 84.96 | 24.59±14.66 | 21.92±12.97 |

Table II: Success rate and average number of steps with real human participants.

ErrP condition (24.4) and less than the average number of steps in the no ErrP condition (28.4). With real human participants, the number of steps was 12.43% less than the number of steps in the no ErrP condition.

As shown in Table II, the success rate to reach the target position within 60 steps was 84.96%, and this value was averaged over all participants. The average number of steps was 23.28, which was averaged over all participants by removing failed episodes (failure rate=1-success rate). The success rate was better than that of the 70% accurate ErrP condition (76.70%) and the no ErrP condition (69.43%) in the simulations. The average number of steps was smaller than that in the 70% accurate ErrP condition (25.6) and no ErrP condition (25.4). With real human participants, the number of steps was 8.35% less than the number of steps in the no ErrP condition.

### C. Discussion

*1) Feasibility of simulated ErrP for training:* We demonstrated the feasibility of our method, which involves training with simulated data and testing with real EEG data, with human participants in real time. The key idea is that the simulated data were binary values (0 or 1) based on the ErrP classifier, which has a binary output (0 or 1). The simulated pilot enables us to train the model without real users. Training an RL model requires a vast amount of data, which rendered the capturing of the EEG from real users infeasible. Thus, we use binary values instead of a linear scale between 0 and 1 to increase the similarity between the simulation data and the classification results of real EEG data. The simulated ErrP data can also be scaled linearly between 0 and 1 to train the model. In this case, the classifier's output should scale linearly with the real ErrP data, which is related to the goal congruency, as discussed in [21].

*2) Consider the learning as a POMDP with noisy ErrPs:* The policy learned with clean observations (100% accurate ErrP) is not robust and vulnerable when the environment is inherently noisy during the test. The discrepancy between the clean simulated ErrP data and the real human EEG data contributes to this "reality gap". The real human ErrP feedback cannot match the simulated feedback with 100% certainty. Thus, the shared policy may fail with real human participants because the ErrP signal cannot be decoded with 100% accuracy. We formulate the learning as a POMDP and train the model with simulated noise observations. We find that the model trained with partial observations is more robust to noise during the test than the model trained with full observations.

*3) Area analysis:* As shown in Figure 10, the environment was divided into two areas: the central area and the edge area. Figure 11 shows the ErrPs of online sessions in the central and edge areas. The positive and negative peaks of the ErrP in the central area were larger than those in the edge area. We hypothesize that the participant was more involved in the experiment when the agent was in the central area than when the agent was in the edge area. We also analysed the accuracy in the central and edge areas during environment 2. As shown in Figure 12, the online test accuracy was higher in the central area than in the edge area, except for participant S10, where the accuracy in the edge area was slightly higher than that in the central area, and participant S15, where the accuracy was the same in both the central and edge areas. Both the larger ERP peak amplitude and higher ErrP accuracy in central area demonstrated that the participants give more correct feedback in the central area than in the edge area. These findings indicate that human participants with better performance should be assigned more authority in the critical central area than in the edge area. The simulation result of the gradient map shown in Figure 6 demonstrates that the ErrP acquired from the simulated users has a greater effect in the central area than in the edge area.

*4) ErrP peak analysis:* As shown in Figure 9, the amplitude of negative peak of error condition is bigger than correct condition. However, the correct condition has positive peak amplitude that error condition. Similar result was found in [57], where P300 amplitude following error feedback was not larger than those following correct feedback . We also analyse the difference between the positive peak and negative peak while an agent continuously performed the wrong action. Figure 13 shows that the peak decreased in the first four sequences. We hypothesize that the participant has less expectations of the agent behaviour, as ErrP signals are evoked by unexpected errors. Similar finding was also reported in [58], where the $1^{st}$ and $2^{nd}$ feedback ErrP responses exhibited slight differences in terms of latency and amplitude. However, we could not explain the increase after the fifth sequence. This increase may be related to the participant's emotional state. Future research should attempt to determine how continuous errors affect the ErrP peak. Note that the agent action was determined by the control model and the agent observations, including the environment information and the ErrP signal. Even if the ErrP classification is correct, the agent can still perform incorrect actions, especially in the edge areas, as the ErrP has a smaller effect on the agent actions in this region based on the gradient analysis.

*5) Potential and Limitation of the shared control model:* In addition to testing the shared control model in a simulation environment, one potential breakthrough of this research was to test and demonstrate the shared autonomy in real environment. The ErrP classification accuracy would be a key part of feasibility of the shared control used in real environment. Unlike in previous study of ErrP-based shared control[12, 19, 21], we demonstrated that the model worked successfully when the ErrP classification reached higher accuracy. One

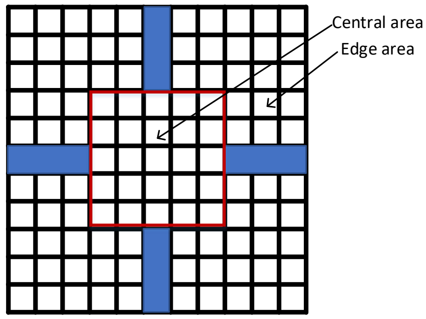



Figure 10: The environment was divided into central areas and edge areas.

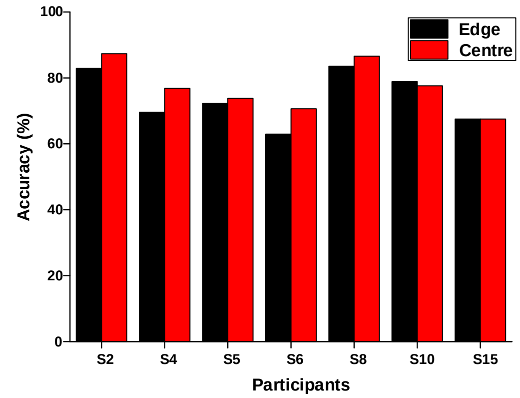



Figure 12: ErrP accuracy of online test for central and edge areas.

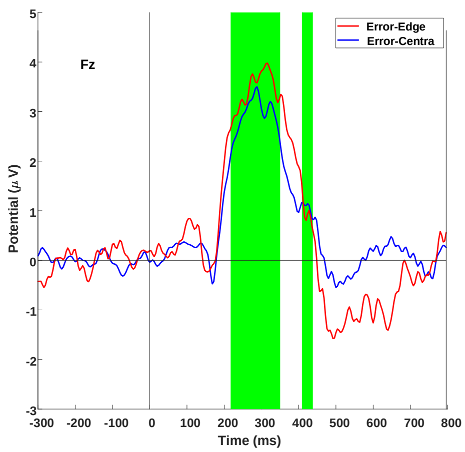



Figure 11: ERP of the error condition in the central and edge areas in Fc channel. Statistically significant difference ($p < 0.05$) was found at green area between error and correct conditions

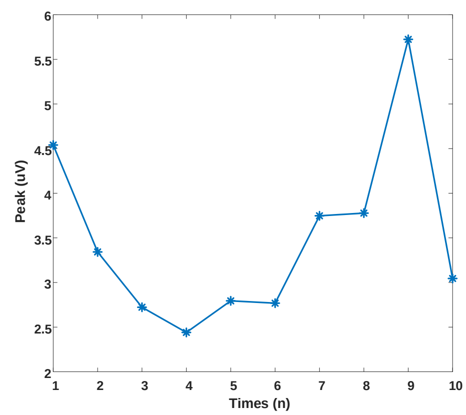



Figure 13: The average difference between the positive and negative peaks versus the error level.

major limitation of the proposed shared control model is that the shared-controlled agent would perform better than an autonomous one only if the ErrP classification accuracy is above 70%. Achieving such classification accuracy in a real-world environment where the user is subject to distractions and noises could be challenging. Further advancements in EEG hardware is required before the proposed framework can be adapted outside a lab environment.The preprocessing and classification could also be optimized to improve the ErrP classification accuracy. The preprocessing used in this paper are not considering possible confounds due to eye movements, which is critical for error-related signals analysis. Moreover, the feature selection and classification algorithm could be optimized to improve classification accuracy.

## VI. Conclusion

We proposed an ErrP-based shared control paradigm with deep recurrent reinforcement learning. To address the low decodability of the ErrP signals, we formulated the learning as a POMDP and used an RNN to solve the POMDP. The shared control model was trained with a simulated ErrP with a Bernoulli distribution with probability $P$ of observing the truth. We validated the proposed model with real-time EEG data obtained from human participants during a navigation task. The agent can adaptively change its search direction based on human feedback. The good performance of the model in simulations and experiments with real human participants suggests that our method is effective in human-robot shared autonomy environments with uncertain noise input, such as neural activities.

## References

[1] Andres F. Salazar-Gomez et al. "Correcting robot mistakes in real time using EEG signals". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 6570–6577.

[2] Su Kyoung Kim et al. "Intrinsic interactive reinforcement learning–Using error-related potentials for real world human-robot interaction". In: *Scientific reports* 7.1 (2017), pp. 1–16.

[3] William J. Gehring et al. "A neural system for error detection and compensation". In: *Psychological science* 4.6 (1993), pp. 385–390.

[4] Hein T. van Schie et al. "Modulation of activity in medial frontal and motor cortices during error observation". In: *Nature neuroscience* 7.5 (2004), pp. 549–554.

[5] M. Falkenstein. "Effects of errors in choice reaction tasks on the ERP under focused and divided attention". In: *Psychophysiological brain research* (1990).

[6] Michael Falkenstein et al. "ERP components on reaction errors and their functional significance: a tutorial". In: *Biological psychology* 51.2-3 (2000), pp. 87–107.

[7] Terence W. Picton et al. "Guidelines for using human event-related potentials to study cognition: recording standards and publication criteria". In: *Psychophysiology* 37.2 (2000). Publisher: Cambridge University Press, pp. 127–152.

[8] Duo Xu et al. "Accelerating reinforcement learning agent with eeg-based implicit human feedback". In: *arXiv preprint arXiv:2006.16498* (2020).

[9] Siddharth Reddy, Anca D. Dragan, and Sergey Levine. "Shared autonomy via deep reinforcement learning". In: *arXiv preprint arXiv:1802.01744* (2018).

[10] Luke Burks et al. "Collaborative human-autonomy semantic sensing through structured POMDP planning". In: *Robotics and Autonomous Systems* 140 (2021), p. 103753.

[11] Jonas Tjomsland, Ali Shafti, and A. Aldo Faisal. "Human-robot collaboration via deep reinforcement learning of real-world interactions". In: *arXiv preprint arXiv:1912.01715* (2019).

[12] Inaki Iturrate, Luis Montesano, and Javier Minguez. "Shared-control brain-computer interface for a two dimensional reaching task using EEG error-related potentials". In: *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2013, pp. 5258–5262.

[13] Katharina Muelling et al. "Autonomy infused teleoperation with application to brain computer interface controlled manipulation". In: *Autonomous Robots* 41.6 (2017), pp. 1401–1422.

[14] Aniana Cruz, Gabriel Pires, and Urbano J. Nunes. "Generalization of ErrP-calibration for different errorrates in P300-based BCIs". In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018, pp. 644–649.

[15] Karl Johan Åström. "Optimal control of Markov processes with incomplete state information". In: *Journal of mathematical analysis and applications* 10.1 (1965). Publisher: Academic Press, pp. 174–205.

[16] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. "Planning and acting in partially observable stochastic domains". In: *Artificial intelligence* 101.1-2 (1998), pp. 99–134.

[17] George E. Monahan. "State of the art—a survey of partially observable Markov decision processes: theory, models, and algorithms". In: *Management science* 28.1 (1982), pp. 1–16.

[18] Matthijs TJ Spaan. "Partially observable Markov decision processes". In: *Reinforcement Learning*. Springer, 2012, pp. 387–414.

[19] Ricardo Chavarriaga and José del R Millán. "Learning from EEG error-related potentials in noninvasive brain-computer interfaces". In: *IEEE transactions on neural systems and rehabilitation engineering* 18.4 (2010), pp. 381–388.

[20] Iason Batzianoulis et al. "Customizing skills for assistive robotic manipulators, an inverse reinforcement learning approach with error-related potentials". In: *Communications biology* 4.1 (2021). Publisher: Nature Publishing Group, pp. 1–14.

[21] Thorsten O. Zander et al. "Neuroadaptive technology enables implicit cursor control based on medial prefrontal cortex activity". In: *Proceedings of the National Academy of Sciences* 113.52 (2016), pp. 14898–14903.

[22] Matthew Hausknecht and Peter Stone. "Deep recurrent q-learning for partially observable mdps". In: *2015 aaai fall symposium series*. 2015.

[23] Stefan K. Ehrlich and Gordon Cheng. "Human-agent co-adaptation using error-related potentials". In: *Journal of neural engineering* 15.6 (2018), p. 066014.

[24] Catarina Lopes-Dias, Andreea I. Sburlea, and Gernot R. Müller-Putz. "Online asynchronous decoding of error-related potentials during the continuous control of a robot". In: *Scientific reports* 9.1 (2019), pp. 1–9.

[25] Yang Xu et al. "Shared control of a robotic arm using non-invasive brain–computer interface and computer vision guidance". In: *Robotics and Autonomous Systems* 115 (2019), pp. 121–129.

[26] Tao Geng, John Q. Gan, and Huosheng Hu. "A self-paced online BCI for mobile robot control". In: *International Journal of Advanced Mechatronic Systems* 2.1-2 (2010), pp. 28–35.

[27] Tao Geng and John Q. Gan. "Motor prediction in brain-computer interfaces for controlling mobile robots". In: *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2008, pp. 634–637.

[28] Abdul R. Satti, Damien Coyle, and Girijesh Prasad. "Self-paced brain-controlled wheelchair methodology with shared and automated assistive control". In: *2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*. IEEE, 2011, pp. 1–8.

[29] Inaki Iturrate et al. "A noninvasive brain-actuated wheelchair based on a P300 neurophysiological protocol and automated navigation". In: *IEEE transactions on robotics* 25.3 (2009), pp. 614–627.

[30] Iretiayo Akinola et al. "Task level hierarchical system for BCI-enabled shared autonomy". In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 219–225.

[31] Lucia Schiatti et al. "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.

[32] Iretiayo Akinola et al. "Accelerated robot learning via human brain signals". In: *2020 IEEE international*

*conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 3799–3805.

[33] Duo Xu et al. "Accelerating Reinforcement Learning using EEG-based implicit human feedback". In: *Neurocomputing* 460 (2021), pp. 139–153.

[34] Iñaki Iturrate et al. "Teaching brain-machine interfaces as an alternative paradigm to neuroprosthetics control". In: *Scientific reports* 5.1 (2015), pp. 1–10.

[35] Abir-Beatrice Karami, Laurent Jeanpierre, and Abdel-Illah Mouaddib. "Partially observable markov decision process for managing robot collaboration with human". In: *2009 21st IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 2009, pp. 518–521.

[36] Wei Zheng, Bo Wu, and Hai Lin. "Pomdp model learning for human robot collaboration". In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1156–1161.

[37] Chi-Pang Lam and S. Shankar Sastry. "A POMDP framework for human-in-the-loop system". In: *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6031–6036.

[38] Andrew Howes et al. "Interaction as an emergent property of a Partially Observable Markov Decision Process". In: *Computational interaction* (2018), pp. 287–310.

[39] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[40] Jakob Foerster et al. "Counterfactual multi-agent policy gradients". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 2018.

[41] Richard S. Sutton et al. "Policy gradient methods for reinforcement learning with function approximation". In: *Advances in neural information processing systems*. 2000, pp. 1057–1063.

[42] Thomas Degris, Patrick M. Pilarski, and Richard S. Sutton. "Model-Free reinforcement learning with continuous action in practice". In: *2012 American Control Conference (ACC)*. June 2012, pp. 2177–2182. DOI: 10.1109/ACC.2012.6315022.

[43] Piotr Mirowski et al. "Learning to Navigate in Complex Environments". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=SJMGPrcle.

[44] Maximilian Hensel. "Exploration Methods in Sparse Reward Environments". In: *Reinforcement Learning Algorithms: Analysis and Applications*. Springer, 2021, pp. 35–45.

[45] Edsger W. Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pp. 269–271.

[46] Yu Zhang et al. "A survey on neural network interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* (2021).

[47] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks". In: *International Conference on Machine Learning*. PMLR, 2017, pp. 3319–3328.

[48] Daniel Smilkov et al. "SmoothGrad: removing noise by adding noise". In: *CoRR* abs/1706.03825 (2017). arXiv: 1706.03825. URL: http://arxiv.org/abs/1706.03825.

[49] Timothy Zeyl. "Adaptive brain-computer interfacing through error-related potential detection". PhD Thesis. University of Toronto (Canada), 2016.

[50] Xiaofei Wang et al. "Implicit Robot Control using Error-related Potential-based Brain-Computer Interface". In: *IEEE Transactions on Cognitive and Developmental Systems* (2022). Publisher: IEEE.

[51] *LiveAmp 64 ≫ Brain Vision*. en-US. https://brainvision.com/products/liveamp-64/. URL: https://brainvision.com/products/liveamp-64/ (visited on 05/20/2021).

[52] Pierre W Ferrez and Jose del R Millan. "Error-related EEG potentials generated during simulated brain–computer interaction". In: *IEEE transactions on biomedical engineering* 55.3 (2008), pp. 923–929.

[53] Inaki Iturrate, Luis Montesano, and Javier Minguez. "Single trial recognition of error-related potentials during observation of robot operation". In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE, 2010, pp. 4181–4184.

[54] Stefan K Ehrlich and Gordon Cheng. "A feasibility study for validating robot actions using eeg-based error-related potentials". In: *International Journal of Social Robotics* 11.2 (2019), pp. 271–283.

[55] Olivier Ledoit and Michael Wolf. "Honey, I shrunk the sample covariance matrix". In: *The Journal of Portfolio Management* 30.4 (2004), pp. 110–119.

[56] Stefan Haufe et al. "On the interpretation of weight vectors of linear models in multivariate neuroimaging". In: *Neuroimage* 87 (2014), pp. 96–110.

[57] Asako Yasuda et al. "Error-related negativity reflects detection of negative reward prediction error". In: *Neuroreport* 15.16 (2004). Publisher: LWW, pp. 2561–2565.

[58] Aniana Cruz, Gabriel Pires, and Urbano J. Nunes. "Double ErrP detection for automatic error correction in an ERP-based BCI speller". In: *IEEE transactions on neural systems and rehabilitation engineering* 26.1 (2017). Publisher: IEEE, pp. 26–36.


## APPENDIX

| | episode | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | Failed Number |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | distance | 13 | 12 | 6 | 19 | 14 | 9 | 3 | 18 | 17 | 5 | 16 | 4 | 7 | 10 | 15 | 20 | 8 | 2 | 11 | |
| S2 | step | 17 | 14 | 42 | 60 | 32 | 13 | 45 | 20 | 45 | 11 | 44 | 6 | 19 | 12 | 19 | 26 | 22 | 6 | 23 | 1 |
| S4 | step | 60 | 32 | 54 | 31 | 60 | 21 | 3 | 30 | 60 | 17 | 49 | 20 | 17 | 10 | 60 | 42 | 10 | 6 | 49 | 4 |
| S5 | step | 17 | 22 | 8 | 45 | 42 | 23 | 17 | 18 | 21 | 7 | 17 | 4 | 9 | 36 | 19 | 20 | 18 | 36 | 60 | 1 |
| S6 | step | 21 | 24 | 16 | 35 | 56 | 60 | 3 | 60 | 60 | 7 | 56 | 16 | 23 | 60 | 60 | 60 | 60 | 36 | 55 | 7 |
| S8 | step | 23 | 20 | 26 | 57 | 32 | 21 | 5 | 20 | 60 | 5 | 43 | 6 | 41 | 18 | 60 | 24 | 31 | 60 | 60 | 4 |
| S10 | step | 53 | 50 | 60 | 60 | 22 | 28 | 3 | 60 | 25 | 7 | 16 | 4 | 13 | 34 | 31 | 42 | 40 | 14 | 17 | 3 |
| S15 | step | 17 | 60 | 60 | 48 | 38 | 60 | 7 | 23 | 45 | 7 | 55 | 6 | 15 | 42 | 17 | 60 | 50 | 34 | 60 | 5 |

Table A1: Number of steps used in environment 1 with real human participants

| | episode | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | Failed Number |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Distance | 3 | 13 | 14 | 7 | 15 | 4 | 11 | 8 | 9 | 19 | 17 | 6 | 12 | 2 | 18 | 10 | 20 | 5 | 16 | |
| S2 | Steps | 3 | 23 | 48 | 7 | 60 | 12 | 19 | 10 | 12 | 53 | 29 | 6 | 24 | 4 | 28 | 30 | 60 | 17 | 16 | 2 |
| S4 | Steps | 43 | 21 | 30 | 7 | 21 | 4 | 19 | 10 | 29 | 31 | 60 | 38 | 60 | 28 | 26 | 58 | 24 | 41 | 60 | 3 |
| S5 | Steps | 3 | 17 | 28 | 7 | 21 | 12 | 60 | 12 | 40 | 37 | 60 | 8 | 16 | 16 | 38 | 38 | 36 | 60 | 34 | 3 |
| S6 | Steps | 3 | 60 | 58 | 7 | 41 | 60 | 60 | 60 | 30 | 60 | 31 | 36 | 26 | 11 | 38 | 22 | 60 | 60 | 38 | 7 |
| S8 | Steps | 11 | 19 | 22 | 17 | 25 | 6 | 17 | 10 | 60 | 24 | 24 | 8 | 16 | 4 | 60 | 14 | 40 | 7 | 34 | 2 |
| S10 | Steps | 13 | 27 | 60 | 9 | 31 | 6 | 17 | 12 | 37 | 29 | 33 | 10 | 20 | 46 | 44 | 34 | 30 | 5 | 22 | 1 |
| S15 | Steps | 3 | 27 | 18 | 7 | 45 | 60 | 60 | 10 | 12 | 19 | 29 | 6 | 38 | 4 | 24 | 40 | 37 | 18 | 20 | 2 |

Table A2: Number of steps used in environment 2