Intermediate Sensory Feedback Assisted Multi-Step Neural Decoding for Reinforcement Learning Based Brain-Machine Interfaces

Xiang Shen, Xiang Zhang, Yifan Huang, Shuhang Chen, Zhuliang Yu, and Yiwen Wang, *Senior Member, IEEE*

***Abstract*— Reinforcement-learning (RL)-based brain-machine interfaces (BMIs) interpret dynamic neural activity into movement intention without patients' real limb movements, which is promising for clinical applications. A movement task generally requires the subjects to reach the target within one step and rewards the subjects instantaneously. However, a real BMI scenario involves tasks that require multiple steps, during which sensory feedback is provided to indicate the status of the prosthesis, and the reward is only given at the end of the trial. Actually, subjects internally evaluate the sensory feedback to adjust motor activity. Existing RL-BMI tasks have not fully utilized the internal evaluation from the brain upon the sensory feedback to guide the decoder training, and there lacks an effective tool to assign credit for the multi-step decoding task. We propose first to extract intermediate guidance from the medial prefrontal cortex (mPFC) to assist the learning of multi-step decoding in an RL framework. To effectively explore the neural-action mapping in a large state-action space, a temporal difference (TD) method is incorporated into quantized attention-gated kernel reinforcement learning (QAGKRL) to assign the credit over the temporal sequence of movement, but also discriminate spatially in the Reproducing Kernel Hilbert Space (RKHS). We test our approach on the data collected from the primary motor cortex (M1) and the mPFC of rats when they brain control the cursor to reach the target within multiple steps. Compared with the models which only utilize the final reward, the intermediate evaluation interpreted from the mPFC can help improve the prediction accuracy by 10.9% on average across subjects, with faster convergence and more stability. Moreover, our proposed algorithm further increases 18.2% decoding accuracy compared with existing TD-RL methods. The results reveal the possibility of achieving better multi-step decoding performance for more complicated BMI tasks.**

***Index Terms*— Brain-machine interface (BMI), reinforcement learning, medial prefrontal cortex, sensory feedback, multi-step task, temporal difference learning.**

I. Introduction

BRAIN-MACHINE interface (BMI) builds up a communication pathway between cortical areas and external devices [1]. BMI generally collects neural activities from motor-related areas and interprets them into motor intentions using a decoder [2], [3], [4]. The reinforcement learning (RL) method fits the scenario for paralyzed people as it does not need real limb movement to decode. When the trajectory deviates from the target, the subjects must adjust their neural activities to correct the trajectory to approach the target through trial and error [5], [6]. When the subjects accomplish the predefined task, a reward (food or water) will be presented externally at the end of the trial to guide the learning of the task. At the same time, such explicit rewards are utilized for training an RL-based decoder by updating the mapping between neural activities and actions.

Several RL methods have been proposed in the BMI area to learn the state-action mapping with an instantaneous reward [7], [8]. In the classic center-out or reaching experiments, the subjects (rats and non-human primates) were simply required to reach targets within one step once the correct direction was selected, so it was relatively effortless for them to understand the task. Correspondingly, the neural patterns of different directional moving states are distinct for the decoder to separate. And the decoder gets reward information for each time instance and updates the parameters instantaneously. Therefore, it is more efficient for the decoder to establish the mapping between neural activities and actions.

However, tasks are more complicated in real BMI applications as they demand multi-step prosthesis control instead of simple one-step command [9], [10], [11]. For example, the subject needs to avoid the objects in the space or reach the final

Manuscript received 25 January 2022; revised 19 June 2022 and 13 September 2022; accepted 15 September 2022. Date of publication 11 October 2022; date of current version 20 October 2022. The work of Yiwen Wang was supported in part by the Grants from China Brain Project 2021ZD0200403, in part by the National Natural Science Foundation of China under Grant 61836003, in part by the Seed Fund of the Big Data for Bio-Intelligence Laboratory under Grant Z0428, in part by The Hong Kong University of Science and Technology (HKUST)-Guangzhou University Joint Research Collaboration Fund under Grant GZU22EG01, and in part by the Special Research Support from Chao Hoi Shuen Foundation under Grant R9051. *(Corresponding author: Yiwen Wang.)*

Xiang Shen, Xiang Zhang, and Yifan Huang are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong.

Shuhang Chen is with the Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology, Hong Kong.

Zhuliang Yu is with the School of Automation Science and Engineering, South China University of Technology and Pazhou Laboratory, Guangzhou 510641, China.

Yiwen Wang is with the Department of Electronic and Computer Engineering and the Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: eewangyw@ust.hk).

Digital Object Identifier 10.1109/TNSRE.2022.3210700

target within the multiple steps. In this scenario, a reward will be given only by the end of the trial instead of instant reward delivery for each step. The previous one-step RL algorithms have not been applied to tackle the delayed reward efficiently in the multi-step task. Although researchers could calculate the relative position to the target as instantaneous reward information for the decoder training in some direct reaching tasks [9], temporal difference (TD) learning is commonly used as it provides a learning procedure for delayed reward RL problems [12]. It learns from the actual rewards and the predictions of the future rewards that the agent expects to obtain from the next state. The error is backpropagated through time to update the decoder. Tobias *et al.* proposed a novel learning scheme (SAGREL) to update the input-output mapping by a neural network structure with a delayed reward [13]. Here, the error signal (one-step State-Action-Reward-State-Action (Sarsa)-style) only uses information from the previous state with limited temporal history. SAGREL often traps in the local minima due to the nonlinear neural network structure. This algorithm has only been tested on a simple classification while not implemented in the BMI area. SAGREL would face the challenge to find the correct neural-action mapping because the neural inputs are generally noisier, and the state-action space grows larger in a real BMI scenario. Bae *et al*. introduced the kernel temporal difference KTD (λ) error to Q-learning (Q-KTD) for neural decoding in RLBMI to find the mapping between a monkey's neural states and the positions of a computer cursor or a robot arm [14]. The kernel structure in Q-KTD ensures a global optimum within the explored state space. However, such a space is limited due to the adopted the ε-greedy policy for the action selection. In the delayed reward task, the decoder could easily bias the good actions with a low action value in the early stage, and the ε-greedy policy would seldom explore such actions in the later training sessions. The Q-KTD method has been applied in one-step BMI tasks. For the multi-step task, it calculates the extra reward information to assist learning at each time instance, which essentially is the instantaneous reward. Above TD-RL algorithms have not fully utilized the intermediate evaluation from the brain in the multi-step task where the reward is sparse. Additionally, the previous RL decoders are not powerful enough to assign credit over a large state-action space in the multi-step task and thus may fail to explore the optimal mapping between the neural activities and a sequence of actions.

Real-time sensory interaction is vital in BMI scenarios, as it exists all the time to indicate the prosthesis status [15], [16], [17]. Subjects observe the sensory feedback and evaluate it internally to refine the actions. Especially the medial prefrontal cortex (mPFC), including the anterior cingulate cortex (ACC), evaluates the current state based on the sensory feedback, and this internal evaluation will correspondingly guide the subject to obtain future rewards [18], [19], [20], [21]. Note that mPFC has been acknowledged to play a crucial role in decision making, including conflict monitoring [22], error detection [23], executive control [24], reward-guided learning [25], [26], and decision-making about risk and reward [27]. Here, we are interested in investigating if the mPFC responses can indicate the status of the prosthesis before the subject has received the final water reward. In literature, Hajcak et al. performed a gambling task on human subjects to investigate whether the mPFC is related to visual feedback [28]. The visual stimuli are designed to represent the final win or loss. They found that the neural activities of the mPFC responded differently when the visual stimuli were presented to the subject. In other words, the mPFC activities reflect the binary classification of the evaluation on the feedback that leads to bad outcomes versus good outcomes. Warren et al. trained rats to use their nose to poke three ports that were associated with different reward probabilities [29]. The odor cues representing different final rewards would be given before the rats received the external reward. They recorded the local field potential from the rodent ACC in four rats and observed the deflection in the local field potential when the subject encountered no-reward odor feedback. Moreover, Bryden et al. discovered that 38 of 111 recorded ACC neurons in rats significantly increased neural firing by the onset of the odor feedback compared to the baseline with no odor feedback [30]. All these statistical findings verify that the mPFC activities respond differently to the sensory feedback related to future outcomes. However, these neuron patterns of the mPFC response upon the sensory feedback are based on statistical analysis across trial averages and have not been utilized to derive the intermediate evaluation in the single-trial analysis. Not to mention that this information has been utilized in BMI scenarios.

In this paper, we are interested in building an RL framework that utilizes the intermediate reward signal upon the sensory feedback to assist decoders in learning the neural-action mapping in a multi-step task. Specifically, we will utilize mPFC neural activities to generate intermediate guidance upon the sensory feedback. Instead of only using the final reward, we can update the decoder with this extra evaluation information during the trial. To effectively assign credit for the multi-step task, we further propose a decoding algorithm that incorporates the temporal difference method into a quantized attention-gated kernel reinforcement learning algorithm (TD-QAGKRL), which achieves the global minimum of input-output mapping by exploring the space expanded not only spatially over neural states in the Reproducing Kernel Hilbert Space (RKHS) but also over a time sequence of the movement. Our algorithm projects the sequence of neural input data into RKHS and builds a universal approximation between spatial-temporal neural features and action values. The sequence of actions is selected probabilistically using a softmax policy. A new learning rule is developed to assign the intermediate credit over the space of the neural firing pattern in RKHS and the time sequence of the movement that leads to future rewards.

To validate our proposed method on the platform of the BMI, we trained two rats to learn to brain control the cursor to reach the target area within multiple steps. The rats had already mastered the one-lever-press manual control task. In the new task, the rats needed to adjust their neural activities to control the cursor to reach the start area, and an audio tone will be given to indicate the start of the trial. Then it had to control the cursor continuously moving to enter the success area to

receive a water reward at the end of the trial. The intermediate evaluation is extracted from the mPFC of rats upon the sensory feedback. Then, we implement a support vector machine (SVM) combined with a confidence metric to classify the mPFC neural activities when hearing sensory feedback versus receiving no sensory feedback. The intermediate guidance with high confidence is used to train the TD-QAGKRL decoder for the multi-step task. In comparison, we also implement the SAGREL and Q-KTD methods as the decoders. First, to verify the advantage of using intermediate reward extracted from the mPFC activities, we compare our approach using intermediate guidance with the scenario only using reward at the final step of a trial. Moreover, we want to validate the decoding performance of TD-QAGKRL with the SAGREL and Q-KTD when using the same reward information. The evaluation is the correct rate regarding the ground truth for each step.

The rest of the paper is organized as follows. Section II A introduces the experiment design and data collection. Section II B illustrates the online multi-step decoding framework, including the characterization of the mPFC activities as the internal evaluation of the sensory feedback and the structure of our proposed decoder. In Section III, we visualize the neural patterns of the mPFC activities upon the sensory feedback and compare our proposed method with the other existing methods in terms of decoding performance and reconstructed prosthesis trajectory. In the last section, conclusions and discussions are presented.

II. Method

A. Experiment Design and Data Collection

We use two male Sprague Dawley (SD) rats in our experiment. The BMI experimental paradigm was designed and implemented at the Hong Kong University of Science and Technology. All animal handling procedures were approved by the Animal Ethics Committee of the Hong Kong University of Science and Technology, strictly complying with the Guide for Care and Use of Laboratory Animals.

The whole experiment paradigm consists of two stages. In the manual control (MC) stage, the rats were trained to press the lever in the behavioral box using its right limb after hearing a start tone [31]. The rat would be rewarded with a water drop and presented with a success tone when the task was accomplished. When the subjects achieved an average success rate of over 80%, they entered the second stage: brain control (BC) without a lever. In the BC stage, the primary motor cortex (M1) and the mPFC neural activities were fed into an online Kalman filter (KF) decoder to generate the continuous cursor trajectory, as indicated in Fig. 1(a). The input of the Kalman filter is formed with the multiple channel firing rates considering the 400 ms firing history, and the output is the continuous cursor trajectory every 100 ms. Kalman filter parameter is pre-trained with the data collected from the well-trained manual control data with an average decoding accuracy of 0.82 in correlation coefficient (CC). Compared with the control experiments as the baseline, where we feed the shuffled M1 activity into the KF, the decoding performance is much lower (0.089 in CC). In the subplot of the

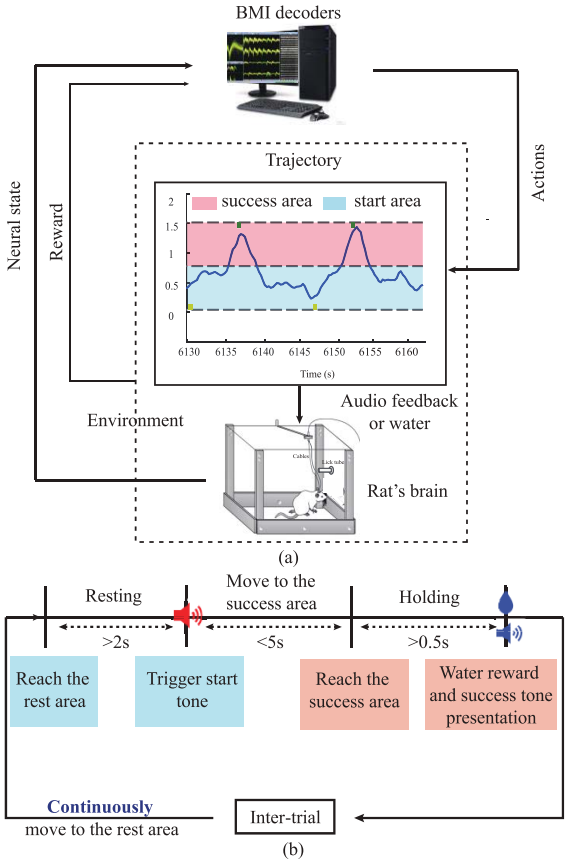

Fig. 1. BMI experiment of a multi-step brain control task. (a) An SD rat adjusts its neural activities to control the decoding trajectory to complete the position reaching task in a behavioral box. (b) Behavioral procedure of the brain control position reaching task across trials.

trajectory (the dashed square), we defined a start (blue area) and success range (red area). In each trial, the subject needed to adjust its neural activities and let the decoded trajectory reach the start range and stay for over 2 s to trigger the 900 ms start tone, shown as a red speaker in Fig. 1(b). And the subject needed to reach and stay within the success range for 500 ms. A water drop would be presented by the end of the trial together with 90 ms success audio tone (blue speaker). The maximum allowed duration of each trial was 5s. After receiving the reward, the subject had to brain control the trajectory to return to the start area to trigger the next start tone. The whole movement trajectory consists of reaching the success range and returning to the rest range. If the subject cannot trigger the start or reach the success area, an autocue will be given after 10 s.

For each rat, two 16-channel microelectrodes were implanted into the M1 and the mPFC area on the left hemisphere, respectively. Neural signals from the two cortical areas were recorded simultaneously by a Plexon (Plexon Inc, Dallas, Texas). The raw signal was sampled at the 40 kHz frequency and was high-passed at 500 Hz with a 4-pole Butterworth filter. Here we used a threshold criterion ($Thr = -3 \sim -5\sigma_0$, where σ_0 is the standard deviation of the histogram of the amplitudes) to detect the spikes. An offline sorter (Plexon Inc, Dallas, Texas) was utilized to sort the single neuron from each channel, and the spike timing information was restored.

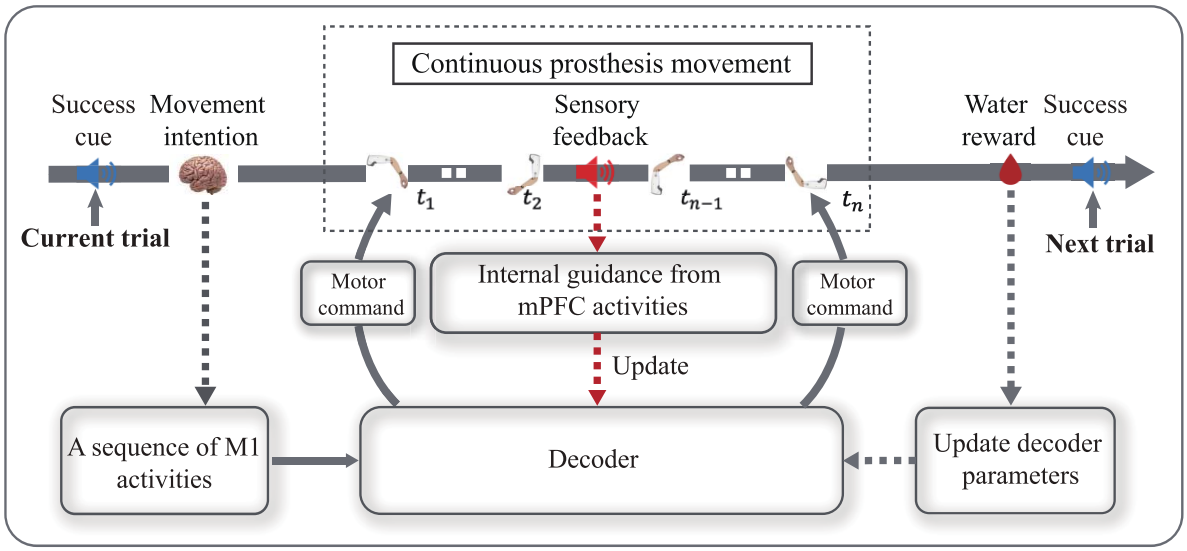

Fig. 2. Diagram of the online multi-step decoding framework. For each trial, the decoder receives neural activities from M1 to generate a sequence of actions, and the neural prosthesis will correspondingly change its status. Meanwhile, the sensory feedback will be given to indicate the prosthesis status. Upon hearing the sensory feedback, the mPFC activities are interpreted as the intermediate guidance to update the decoder before the final reward is given. Then the prosthesis continues to move until reaching the target, and the final reward will also be utilized to update the parameter of the decoder.

Spike firing rates were counted with a nonoverlapping 100 ms time window. Meanwhile, the corresponding behavior events were acquired by a behavior recording system (Lafayette Instrument, USA) and synchronized through a Plexon digital input. A total of 196 and 186 trials for two rats, respectively, on the well-trained days were recorded for analysis.

B. Intermediate Guidance Assisted Multi-Step Decoding Framework

We propose an RL framework that employs the intermediate guidance information from mPFC activities upon the sensory feedback to assist the subjects in accomplishing a multi-step task in BMI scenarios. Moreover, we embed a TD method into QAGKRL as a decoder to effectively reach the optimization over the spatial-temporal space for the multi-step task. This framework is designed in an online manner, as shown in Fig. 2. The decoder receives a sequence of neural activities from M1 for each coming trial and generates actions that continuously move the prosthesis. External sensory feedback is presented in the middle of the trial process to indicate the prosthesis status. The mPFC activities post the sensory feedback are put into the classification model to generate the intermediate guidance, which updates the decoder parameter upon the sensory feedback before the subject gets the final external reward. This classifier will be pre-trained with the data collected from the previous day or using the first several trials on the same day. The decoder continues to generate actions for the subsequent trial. When the subject adapts to the prosthesis control using the neural activities, the decoder in parallel learns to interpret the neural activities to output the actions. In our work, the decoder takes in the sorted data each time and simulates the online scenario to generate the output accordingly. The details are explained in the following sub-sections.

1) Internal Evaluation of the Sensory Feedback From the mPFC: The sensory feedback generally exists during the interaction between subjects and the neuro-prosthesis. It is not directly associated with the final external water reward but indicates the status of the prosthesis [28], [29], [30]. Therefore, the multi-step decoding online scenario mimics the closed-loop BMI setting in our experiment. To extract the intermediate guidance from the mPFC activities upon the sensory feedback, we label mPFC activities 500 ms before the triggered sensory feedback (start tone), which represent the state of trying but have not succeeded, as 0; and label the mPFC activities 500 ms post the sensory feedback, which represent the successful trigger of the trial start, as 1. To discriminate the two cases, we take in every 100 ms neural activity with a history of 300 ms within the duration as SVM input. The output is the action labels.

The kernel SVM is implemented with LIBSVM [32], [33]. Here we denote $\hat{r}$ as

$$\hat{r} = h\left(\omega^T \phi(x_t^L) + b\right), \tag{1}$$

where h is a threshold function. x_t^L is the neural input formed with a size of 4^*N by 1 (N is the number of channels; 4^*N includes the 300 ms history and current firing of N channels; t represents the time instance taken from the duration of 500 ms before and after the feedback in the trial L). ω denotes weight vector sizing of 4^*1, which is obtained using cross-validation. b is a constant as a bias in the hyperplane. $\phi(\cdot)$ is a radial basis function, which converts the original data into the RKHS. The value of $\omega^T \phi\left(x_t^L\right) + b$, derived from the SVM classification result, measures the distance from the data samples to the classification hyperplane. This distance distribution of training samples is used to set the threshold *thr,* out of which 90% of SVM classification results exist [34]. In the testing phase, we put the mPFC neural activities every 100 ms after the sensory feedback into the pre-trained SVM

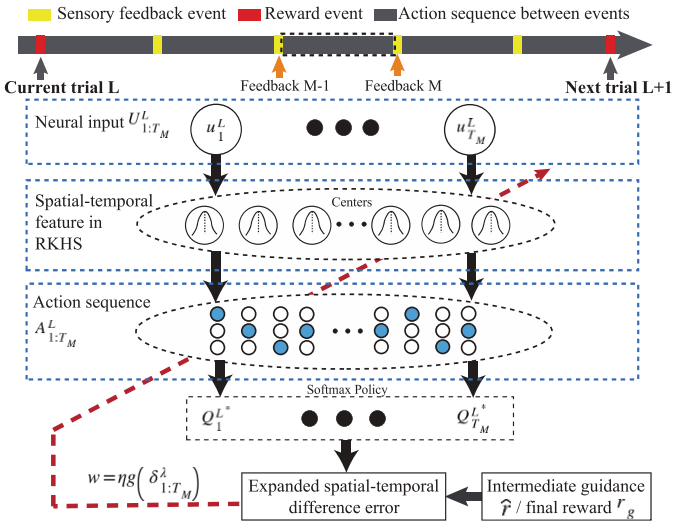

Fig. 3. The network structure of TD-QAGKRL. For each trial, the decoder projects the sequence of neural input data into RKHS feature space, builds a nonlinear mapping between spatial-temporal neural features and action values. The action sequence (blue dots) is selected probabilistically using a softmax policy. An expanded spatial-temporal error is utilized to update the decoder parameter.

model to get the intermediate guidance to update the RL decoder. The dominating classification results within 500 ms will be the final intermediate guidance for this trial. If the SVM values lie within the range between *-thr* and *thr*, we take the classified results as low confidence. If low confidence SVM results dominate the trial, we will not update the decoder parameters when the subjects observe the sensory feedback in the multi-step task.

2) TD-QAGKRL Decoding and Updating: Here we propose a new TD-RL method to learn the neural-action mapping for the multi-step task. The structure is shown in Fig. 3. The decoder is fed with a sequence of neural activities and generates sequential actions to reach the target within a trial. Our new learning method addresses credit assignment over space and time. Here, space and time represents credit assignment spanned in the RKHS and time sequence of the motion in one trial, respectively. Using the intermediate guidance extracted from the mPFC, we decompose the whole trial into several segments separated by sensory feedback. We assume that exploring the optimal mapping in each segment contributes to the efficient convergence to the global spatial-temporal optimization of the neural-state mapping for the whole trial.

We use $u_t^L \in R^{1\times N+1}$ to represent the neural input from the M1 and the mPFC considering a background firing as bias at each time instant within trial L. N is the channel number. T is the time length of trial L.$U_{1:T}^L \in R^{T\times N+1}$ represents a sequence of the neural input transformed to RKHS to form the spatial-temporal feature by a kernel method, $\kappa\left(u_t^L, u_j^L\right) = < \phi(u_t^L), \phi\left(u_j^L\right) >$, which is commonly used as a Gaussian kernel:

$$\kappa\left(u_t^L, u_j^L\right) = \exp\left(-\frac{\| u_t^L - u_j^L \|^2}{2\sigma^2}\right), \tag{2}$$

where $u_t^L, u_j^L \in U_{1:T}^L$, and σ decides the flatness of the Gaussian kernel.

Then, the action value is computed by linearly combining spatial-temporal features with the weights as follows:

$$\begin{aligned} Q_k\left(u_t^L\right) &= \sum_{j=1}^{t-1} w_{k,j}\left\langle \phi\left(u_t^L\right), \phi\left(u_j^L\right)\right\rangle \\ &= \sum_{j=1}^{t-1} w_{k,j}\kappa\left(u_t^L, u_j^L\right), \end{aligned} \tag{3}$$

where $w_{k,j}$ is the coefficient between the j^{th} Gaussian kernel center in the RKHS and the k^{th} action. t represents all the preceding time instances across all previous L-1 trials.

This network has an inherently growing structure as it allocates a new center for each coming data sample in each trial, which causes a linearly growing computational complexity. Here we adopt a quantization approach to decrease kernel centers [35]. We explore an optimal quantization threshold ξ_U according to the distribution of Euclidean distances between the pairs of input vectors. The value of action k can be calculated with quantized centers as in Eq. 4. $Q_k(U_{1:T}^L)$ is formed with a sequence of action values in trial L.

$$Q_k(U_{1:T}^L) = \sum_{j=1}^{|C_{L-1}|} w_{k,j}\kappa(U_{1:T}^L, u_j^q) \tag{4}$$

$$d_{min}^C = \min_{1\le p\le |C_{L-1}|} \| u_j^L - C_p^L \|, \tag{5}$$

where $|C_{L-1}|$ is the size of the centers in RKHS, including the input sequences of the preceding samples across all the previous L-1 trials. u_j^q is the j^{th} center after quantization, whose minimal distance d_{min}^C in Eq.5 to all the previous centers C_p^L, which is updated over trials, is larger than ξ_U. If the distance of u_t to all the previous centers is larger than ξ_U, we assign a new kernel center to this input u_t. Otherwise, the centers remain unchanged.

After the action values are calculated for each time instance within trial L, we will probabilistically choose the action sequence $A_{1:T}^L$ based on the softmax policy. $A_{1:T}^L$ is formed with a sequence of chosen action k^* (blue dots in Fig. 3), defined as $P\left(Z_{k^*} = 1\right) = \frac{exp(Q_{k^*}(u_t^L)/\tau)}{\sum_{k'=1}^{K} exp(Q_{k'}(u_t^L)/\tau)}$, where τ is the temperature parameter and K is the action set.

In the multi-step task, the explicit reward is only given at the end of the trial. However, during the interaction with the prosthesis, the subjects receive sensory feedback before they reach the final target, which indicates the decoder has stayed within the start area long enough to trigger an audio tone, which indicates the successful start of the trial. Like the existing multi- step decoders, if the decoded action sequence reaches the final target successfully, our decoder will get a ground truth reward $r_g = 1$. Otherwise, it receives no reward $r_g = 0$. The difference is that we utilize the extra guidance information when receiving the sensory feedback to update the decoder. If the subjects accomplish a part of the task, the decoder will receive an additional intermediate guidance $\hat{r}$ derived from the classification on mPFC activity as described

in Eq.1. Here, we formulate the reward function as:

$$r = \begin{cases} r_g, & \textit{Task ends} \\ \hat{r}, & \textit{Sensory feedback is given} \\ 0, & \textit{Otherwise} \end{cases} \tag{6}$$

Based on the reward function, a multi-step task can be decomposed by the sensory feedback (yellow bars) into smaller segments, as indicated in Fig. 3. Instead of all the data within a whole trial, we analyze a sequence of neural data within the interval between the adjacent $(M-1)^{th}$ and M^{th} feedback. To assign the credit over space and time, we propose a TD error via backpropagation for the QAGKRL as in Eq. 7.

$$\delta_{t_M} = r^L_{t_M+1} + \gamma\, Q(u^L_{t_M+1}, a^L_{t_M+1}) - Q(u^L_{t_M}, a^L_{t_M}) \tag{7}$$

$$\delta^\lambda_{t_M} = \delta^L_{t_M} + \sum\nolimits_{n=1}^{T_M-1} (\gamma\,\lambda)^n\, \delta^L_{t_M+n}, \tag{8}$$

where t_M represents the time index in segment M. T_M represents the length of segment M. $r^L_{t_M+1}$ is the reward received at time t_M+1, which is obtained from Eq.6. $Q(u^L_{t_M+1}, a^L_{t_M+1})$ and $Q(u^L_{t_M}, a^L_{t_M})$ are the future rewards expected to be obtained at time t_M+1 and the current time t_M, respectively. λ is the eligibility trace-decay parameter, and γ is the discount factor.

Different from the previous TD methods, Q values are calculated from current input and the quantized centers in RHKS, which assigns the credit assignment over space. And we also consider the temporal difference as in Eq.7. In this way, we assign the credit over both space and time. Note that this error is accumulated only within the current segment of the trial to update the decoder, as shown in Eq. 8. In addition, a global error-based expansive function is defined to enhance the learning when an unexpected reward comes, shown as:

$$g\left(\delta^\lambda_{t_M}\right) = \begin{cases} \dfrac{\delta^\lambda_{t_M}}{1-\delta^\lambda_{t_M}+\epsilon}, & 0 \le \delta^\lambda_{t_M} \le 1 \\ \delta^\lambda_{t_M}, & \textit{otherwise} \end{cases} \tag{9}$$

where ε = 1e-4 here, which is a small constant to eliminate the singularity when $\delta^\lambda_{t_M} = 1$.

This expanded error will be used to efficiently update the weights of TD-QAGKRL for every time instance within the trial when sensory feedback is presented. For a new center in RKHS that cannot be quantized, we assign a new kernel center to this input u^L_t with the weight $\omega_{k,l}$. If the input is quantized to the closest center p, we locally update the center's coefficient accordingly, as in Eq.10. This spatial-temporal error helps to optimize the neural-state mapping for the multi-step task.

$$\begin{cases} \omega_{k,l} = \eta g\left(\delta^\lambda_t\right), & l : \textit{new center index} \\ \omega_{k,p} = \omega_{k,p} + \eta g\left(\delta^\lambda_t\right), & p : \textit{closest center index} \end{cases} \tag{10}$$

III. Result

In this section, we will first visualize the firing patterns of M1 neural activities during the position-reaching task and mPFC neuron patterns before and after the onset of the triggered sensory feedback. We will also demonstrate the classification results using an SVM across multiple segments from two subjects. Then, we embed this intermediate guidance from the mPFC activity into an RL framework for the multi-step task. We compare the decoding performance and reconstructed trajectory of the proposed method using the intermediate reward with the case only using the final reward. To further validate the decoding ability of the proposed method, we compare it with the other TD-RL methods given the same reward information.

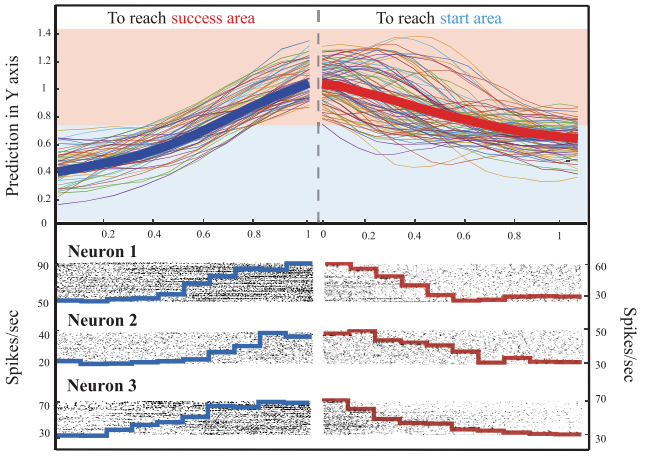

Fig. 4. Average trajectory (top) and raster plots (bottom) of three typical M1 neurons. The average firing rate across trials is shown in solid line.

First, we visualize the neural modulation of M1 activity when the subject completes the position-reaching task. The top part shows the trajectory of each trial and averaged trajectory (thick blue and red curves) in Fig. 4. The x-axis is the time, and the y-axis is the position. As the length of each trial is different, we pick the same length of data (1 second) from the event onset (start and success). The bottom part shows the raster plot and corresponding histogram (blue and red solid curves) of three typical M1 neurons, respectively. We can see the neural patterns are changing distinctly from the event onset (start and success), which reveals that animals modulate their M1 neural activities when they try to complete the task. In our experiment, the RL decoder fed with only M1 information can already achieve a decoding accuracy of over 90.3%, and adding the mPFC activity can improve by 1.6% [29]. Thus, the brain control trajectory is mainly completed using M1 instead of mPFC neural activities. Then we also validate whether mPFC neurons respond to the sensory feedback. We plot the raster and histogram of two typical mPFC neurons under two circumstances: when trying vs. after successfully trigger the start tone, as shown in Fig. 5. The trying stage includes the periods that the subjects have not obtained the reward (e.g., the audio cursor did not move towards the desired trajectory). Scales on the x-axis represent the time, and units on the y-axis indicate the behavioral trials and occurrence of spikes per second for raster and histogram plots, respectively. The zero point represents the onset of the triggered start tone. Note that no water reward is given after the presence of the audio tone. This setting is different from the cases in [31] and [36], where the water reward is given right after the audio tone that indicates success. It is clearly seen that the neural patterns of mPFC activities change significantly after the start tone is presented to the subject.

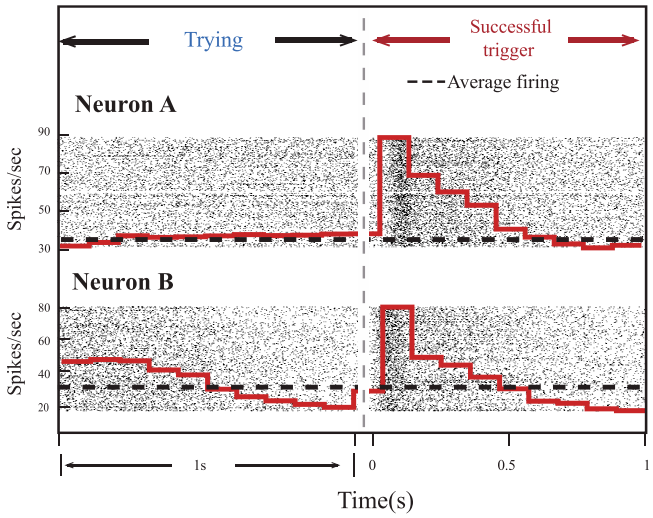

Fig. 5. The raster plots of two typical mPFC neurons in the duration of trying and successfully trigger the start tone. The average firing rate across trials is shown in red.

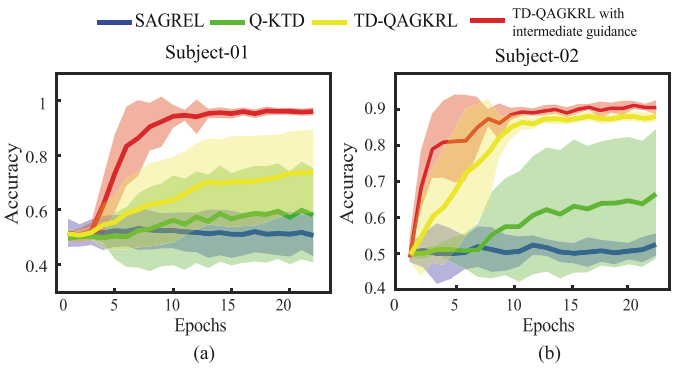

Fig. 6. Learning curves of two subjects using five TD-RL models. The solid line represents the mean value across 10 initializations and shaded areas show the standard deviation. The dashed black represents the Q(λ)-learning, which is commonly used in the one-step BMI decoding task.

Given such observations, we build a classification model using mPFC data from two periods (trying vs. successfully trigger the start tone) during training. For testing, neural activities after the triggered audio tones are put into this model to indicate the intermediate evaluation at each time instance.

Here, we implement a kernel SVM to obtain the single-trial evaluation information via four-fold cross-validation. Specifically, the mPFC neural activities (500 ms in the trying stage vs. successful trigger stage) are put into SVM to distinguish whether the mPFC has interpreted that a part of the task has been accomplished, i.e., the successful trigger of start in our experiment. The average classification accuracy by the SVM reaches 85.5% across subjects. The high classification accuracy ensures that mPFC activities can be an intermediate evaluation of the external sensory feedback.

After obtaining the intermediate evaluation information from the mPFC activities, we train the RL decoder with the real data collected from the brain control position reaching task. The online recording involves the neural data when subjects are not fully engaged in the task, which leads to a long response time. Thus, we cluster the trajectory and segment the neural activity corresponding to the large velocity, which characterizes the duration that the subjects are engaged. In the task, we observe that the subjects need at least 0.7 seconds to complete one subtask (i.e., going up to the success area). In this case, considering a 300 ms history, we segment 400 ms M1 neural activities from going up and down trajectory clusters, respectively. The segmented M1 neural data are reconnected as a trial with a shorter response time for TD-RL training. The output of the decoder is the action sequence. Moreover, we adopt a confidence metric to generate intermediate guidance with high confidence to update decoder parameters [34]. The threshold of confidence metric is set by choosing 90% of the total SVM results far from the hyperplane. Such an intermediate guidance is treated with high confidence and will be used to update the decoder parameters when the subject receives sensory feedback in the multi-step task.

To validate whether incorporating the extra intermediate guidance from the mPFC activity into the decoder improves decoding performance, we compare our method (TD-QAGKRL with intermediate guidance) with the model using the final reward only (TD-QAGKRL). We explore the optimal parameters by picking the values that lead to the highest performance in validation data. In each initialization, 60% of the data is selected for decoder training, 20% for validation, and the rest for testing. We shuffle the data 10 times for each method. Note that the test data has never used in the training phase for each shuffle. With optimal parameters (e.g., we select kernel width $h = 1.6$, quantization threshold $\xi_u = 0.6$, learning rate $\eta = 0.3$, discount factor $\gamma = 0.9$, and eligibility trace rate $\lambda = 0.99$ for TD-QAGKRL), the learning curves of the four models are shown in Fig. 6 for the two subjects, respectively. The x-axis represents the training epochs (each contains 20 trials), and the y-axis is the correct rate. The solid curve shows the mean performance value across 20 data shuffles. The shadow represents the standard deviation of the success rate. We can see that the TD-QAGRKL (yellow curve) has a slower convergence speed (15 epochs late on Subject-01 and 5 epochs late on Subject-02) and lower convergence rate compared with using the extra intermediate guidance (red curve). This result reveals that using the intermediate evaluation as the additional reward information for the decoder can significantly improve the performance.

Moreover, we want to verify that the proposed TD-QAGKRL provides a better decoding ability than existing algorithms given the same reward information. In Fig. 6, Q(λ)-learning [9] is selected as the benchmark of RL decoder and is represented by the black dashed curve. The performance of Q(λ)-learning is poor due to the network structure, which is prone to trap in the local minima. This result is expected and concurred with the results shown in [35], which demonstrated that Q(λ)-learning failed in a multi-step obstacle avoidance task. One extension of Q(λ)-learning is SAGREL, which shares the same structure. Therefore, we use SAGRL as the baseline in the following comparisons. We can see that SAGREL (blue curve) has a large variance around the chance level, which means that it

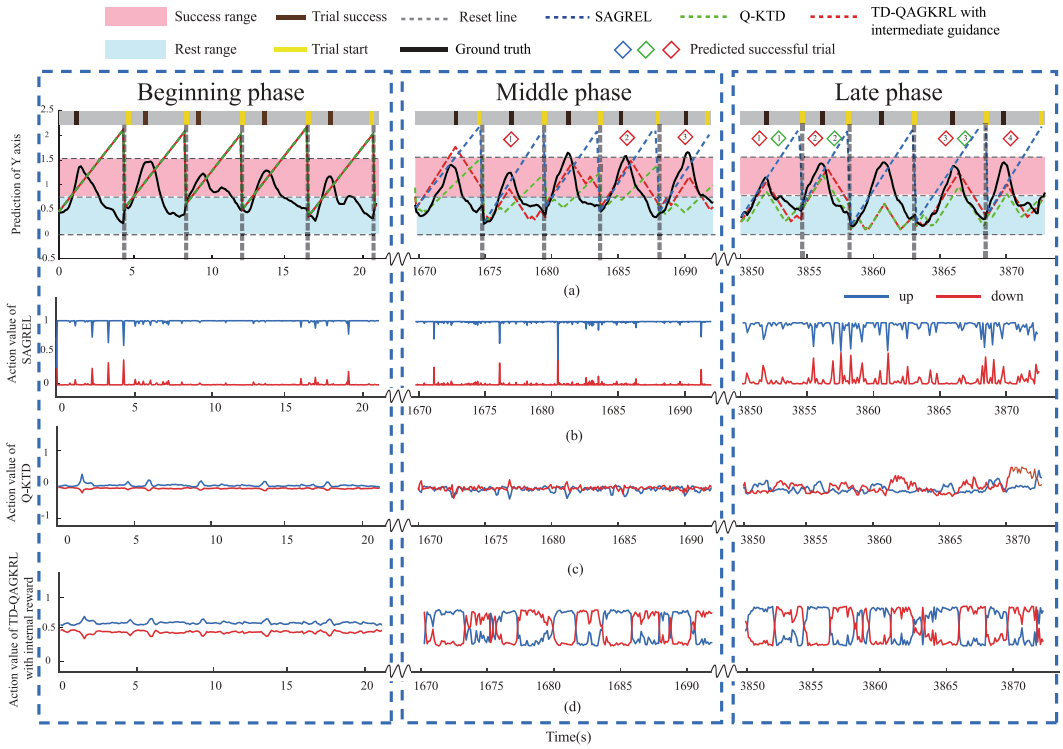

Fig. 7. The learning process of the trajectory reconstruction and the action values of the three decoders. The three columns represent the beginning, middle, and end phases of the decoder training. (a) Trajectory reconstructions by SAGREL (blue dashed line), Q-KTD (green dashed line), and TD-QAGKRL with intermediate guidance (red dashed line) in three stages. The solid black line is the ground truth brain state. The grey dashed line means the manual reset of the position. The red area indicates the success range, the blue area indicates the rest range, the start events are labelled with the yellow bars, and the success events are labelled with the brown bars. (b), (c) and (d) Corresponding output values of two actions (up and down) of the three decoders during learning. The red curve represents the down action, and the blue curve represents the up action.

can hardly find the correct mapping in the multi-step task as it utilizes limited temporal information. TD-QAGKRL (yellow curve) converges faster than the other two methods and obtains a higher convergence rate. Note that Q-KTD (green curve) can achieve similar performance to TD-QAGKRL with good initiations, but it sometimes traps in the local minima due to the time dynamics. Note that the simulation may include more complex movement sequences to characterize the performance across decoding methods. We simulate a four-target center-out task, where the subjects need to take several steps to reach the target as in [37]. The intermediate reward is generated probabilistically based on the real mPFC activity classification results. We find that the proposed algorithms with intermediate guidance perform better than the existing methods. Due to the page limit, here we show the results on the real data only.

We further examine the reconstructed trajectory decoded from the neural states using three models (Q-KTD, SAGREL, and TD-QAGKRL with intermediate guidance). The trajectory can be grouped into two subsets: going up to reach the success area to get the reward and down to the rest area to trigger the start, as indicated in Fig. 1(a). The step size of going up (l_u) and down (l_d) is obtained based on training trials. l_u, l_d are calculated by the Euclidean distance between the start and end position, divided by the average number of steps within the two points, respectively. We present the adaptation process to illustrate how the three decoding models learn the brain control task over time, shown as the beginning, middle and late phases in three columns in Fig. 7, respectively. The trajectory input is the neural activity collected from M1 and the mPFC in the brain control task, and the output is the action sequence (up or down). The cursor position can be calculated by the current action and step size (l_u or l_d). As shown in Fig. 7(a), the grey line on the top covers the events that occurred throughout the trials, including trial start (yellow bar) and trial success (brown bar).

Successful trials are labelled with colored diamonds by three decoders, respectively. At the beginning of training (first column of Fig. 7(a)), the reconstructed trajectory using three decoders cannot follow the ground truth and keep going up. We can see that action values of up action are consistently higher than the down action of all three decoders (first column of Fig. 7(b), (c), and (d)) because of the random initializations. Therefore, we manually reset the position of each trial, as indicated by the dashed grey line. In the middle stage (second column of Fig. 7(a)), our proposed method begins to track the trajectory while the other two still cannot follow. The transitions among the output action values over time are observed after the beginning phase, indicating that the weight of decoders evolves in learning the state-action mapping to take the correct action to complete the task. Specifically, action values using our method begin to show the difference in the correct sequence (second column in Fig. 7(d)), which means it can distinguish between the two actions. However, SAGREL cannot differentiate movements as the "up" action values are still consistently higher than the down action (second column

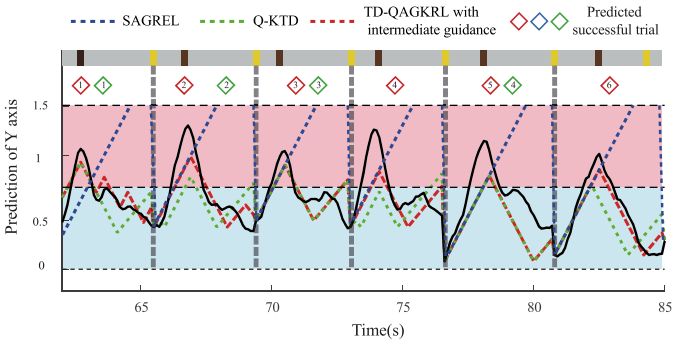

Fig. 8. The trajectory reconstruction on the testing data of three TD-RL decoders.

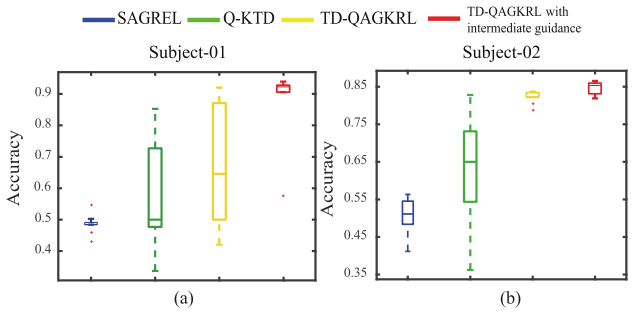

Fig. 9. Box plot of the statistical testing result of four methods for two subjects.

of Fig. 7(b)), and the Q-KTD method starts to learn action policies (second column of Fig. 7(c)) as the action values begin to fluctuate in the trial, but Q-KTD cannot classify actions well. In the late stage (third column of Fig. 7(a)), our method and Q-KTD track the ground truth while SAGREL still fails and keeps going up. The transitions between action values using our method and Q-KTD is almost the same as the ground truth (third column in Fig. 7 (c) and (d)), while SAGREL keeps the same with the previous phases (third column in Fig. 7 (b)) since it is trapped in the local minima. Compared with Q-KTD, our method has a faster convergence speed (learn the mapping in the middle phase) and better reconstructs the trajectory with more correct trials (4 red diamonds).

We observe the same phenomenon in the testing segments. Here we show the reconstructed trajectory in Fig. 8. The legends remain the same as in Fig. 7. For SAGREL, the trajectory (blue color) fails to reach the target and keeps going up in all trials. Even though Q-KTD (green color) can also partially follow the ground truth trajectory (black color), our method (red color) accomplishes more successful trials (6 diamonds) than Q-KTD. When the decoding performance reaches convergence in the training session, the models are tested on the data that never appears in the respective training. The trained parameters are fixed for the testing to assess how the network retains the information it has learned in the past. To understand how much the proposed method contributes to the decoding other than the intermediate guidance, we also add the performance of TD-QAGKRL with final reward for comparison. Fig. 9 shows the box-plot distributions of four methods' total success rates across all the initializations using the testing data from two subjects. The testing results are shown in Table I. We can find that our proposed method

TABLE I
STATISTICAL TESTING PERFORMANCE OF FOUR DECODING METHODS ACROSS SEGMENTS

	SAGREL	Q-KTD	TD-QAGKRL	TD-QAGKRL with intermediate guidance
Subject-01	51.2±2.7%	57.9±12.3%	65.3±16.3%	84.6%±6.7%
Subject-02	49.4±5.3%	63.7±15.2%	82.2±1.6%	84.7±1.7%

with intermediate guidance (red) ranges much narrower with shorter whiskers, while the Q-KTD ranges (green) have a larger distribution, and SAGREL (blue) remains at the chance level. The results indicate that the decoding performance of Q-KTD is sensitive with the initialization and may trap in the local minima during the search, whereas TD-QAGKRL (yellow) greatly improves performance by 7.4% and 8.5% for two subjects, respectively. With the intermediate guidance, our approach further improves the performance by 19.3% and 2.5% for two subjects, respectively. For each subject, we also perform the right tail paired-sample t-test on 10 test segments. All the tests are performed using Bonferroni correction at an $\alpha = 0.025$ significance level. Under the null hypothesis, the probability of observing an equal or higher value in the test statistics is indicated by the p-value (SAGREL against our method: $p =$ 1.35e-11, and our method against Q-KTD: $p = 0.0034$ for Subject-01; SAGREL against our method: $p = 0.0038$, and our method against Q-KTD: $p = 0.012$ for Subject-02).

In all, our proposed method outperforms the other two algorithms in terms of speed and accuracy and maintains stable performance when reconstructing the trajectory of prosthesis control.

IV. CONCLUSION AND DISCUSSION

Reinforcement learning-based brain-machine interfaces assist paralyzed people in controlling external devices without real limb movement. In real BMI scenarios, the subjects need to control the neural prosthesis with the brain neural activities to accomplish a task. When the trajectory deviates from the target, the subjects need to adjust their neural activities to reach the target. Usually, the task consists of multiple steps but only gives the reward at the final step. Current RL decoder learns the state-action mapping by incorporating the temporal difference method, but is not effective when the task takes too many steps, and the reward is too sparse. In BMI tasks, sensory feedback (visual, audio, etc.) is often given to indicate the status of the prosthesis. Even though this feedback can be observed by the subjects and revealed in the brain cortical areas, this interpretation from the brain has not been investigated and utilized in the RL decoders.

In this study, we propose an RL framework that effectively learns a multi-step decoding task with the assistance of the intermediate evaluation extracted from the mPFC upon the sensory feedback. A new TD-QAGKRL decoder has been proposed to speed up the learning of the neural-state mapping by assigning credit both spatially over neural states spanned in RKHS and over a temporal sequence of movements. We test our framework on the neural data collected when the rats were performing a brain control position reaching task. Sensory

feedback is presented to the rats when they successfully reach the start zone in the middle of the trial. We first model the mPFC neural activities as an internal representation of the sensory feedback by an SVM. The average classification accuracy of distinguishing mPFC neural activities (trying vs. successfully trigger the tone) was over 85% across multiple segments from two subjects. This high classification result indicates that the mPFC activities can be leveraged as the intermediate guidance for the RL decoders in the multi-step task. One thing we need to point out is that, in this work, the time duration of the mPFC activity is different from the work that used the internal neural presentation of the reward (NAcc activity [34], [38], M1 activity [39]). We utilize the mPFC neural response upon the sensory feedback, which is not directly associated with the final reward, but as the intermediate guidance to assist the training of the decoder. In this way, we embed this intermediate guidance into the TD-QAGKRL to split the long multi-step task into smaller task segments upon sensory feedback to effectively learn the state-action mapping. Compared with TD-QAGKRL only using the final reward, leveraging extra intermediate guidance outperforms it in accuracy, convergence speed, and stability. This validates that using intermediate guidance evaluated from the mPFC activities can significantly improve the decoding performance.

Moreover, TD-QAGKRL is more advantageous than the other existing TD-RL methods, given the same reward information for delayed and instantaneous reward cases. First, the sensory feedback or the external reward is provided at certain points (e.g., finish the sub-task or reach the final target). All the models share the same neural inputs and action ensembles. SAGREL inherits a nonlinear neural network and incorporates a Sarsa-style learning signal to update the network, which makes SAGREL prone to trap in the local minima and gain less from the sparse reward information since it only considers information from the previous trial. Q-KTD utilizes an ε-greedy policy to select actions based on the current action values. This policy exploits current knowledge with a high probability and seldom explores the other actions with lower values. It would easily bias the state action policy and limit exploration in the state space. Although it shares the same structure as our method, the input-output mapping can only achieve the optimal within the limited space that has been explored. In comparison, our method explores spatial-temporal optimization and adopts a softmax policy to select the actions according to the probability distribution of all action sequences. Even if the optimal action is not selected, the suboptimal action could be chosen with a higher possibility than the others, which possibly helps prevent the performance from experiencing an abrupt change. Second, when the instantaneous reward is available at each time instance, our method is equivalent to QAGREL [35]. Our method projects input neural data into the RHKS feature space and reaches the global optimum. Prins et al. used a fully connected neural network structure as the actor to select actions [34]. This actor shares the similarity with AGREL, which is also prone to trap in the local minimum. Mahmoudi et al. used a time-delayed neural network with a gamma memory structure to decode actions [38]. This gamma memory structure is utilized to project the input data into the hidden layer. However, a gamma filter is an IIR filter with a restricted or adjustable memory depth [40]. In [41] and [42], it has been pointed out that ensuring stability during IIR adaptation is complex, and the error surface is non-convex. When the gamma filter is extended to the RKHS, it can be generalized to other filters (AR, MR, ARMA) with better performance. Our method (QAGKRL) is optimized in the RHKS feature space. Theoretically, our approach has a better computational capacity than the decoder with a gamma memory structure.

Overall, our proposed method embeds additional intermediate guidance from the mPFC activity upon the sensory feedback to ensure better performance for the multi-step task. This framework is designed in an online manner and can be applied for closed-loop interaction when sensory feedback is available, which has a great potential to improve the performance of clinical BMI applications.

REFERENCES

[1] M. A. Lebedev and M. A. L. Nicolelis, "Brain–machine interfaces: Past, present and future," *Trends Neurosci.*, vol. 29, no. 9, pp. 536–546, 2006.

[2] J. Wessberg *et al.*, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, no. 6810, pp. 361–365, Nov. 2000.

[3] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. L. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nature Neurosci.*, vol. 2, pp. 664–670, Jul. 1999.

[4] C. E. Vargas-Irwin, G. Shakhnarovich, P. Yadollahpour, J. M. K. Mislow, M. J. Black, and J. P. Donoghue, "Decoding complete reach and grasp actions from local primary motor cortex populations," *J. Neurosci.*, vol. 30, no. 29, pp. 9659–9669, Jul. 2010.

[5] A. L. Orsborn, H. G. Moorman, S. A. Overduin, M. M. Shanechi, D. F. Dimitrov, and J. M. Carmena, "Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control," *Neuron*, vol. 82, no. 6, pp. 1380–1393, 2014.

[6] W. Truccolo, G. M. Friehs, J. P. Donoghue, and L. R. Hochberg, "Primary motor cortex tuning to intended movement kinematics in humans with tetraplegia," *J. Neurosci.*, vol. 28, no. 5, pp. 1163–1178, Jan. 2008.

[7] Y. Wang, F. Wang, K. Xu, Q. Zhang, S. Zhang, and X. Zheng, "Neural control of a tracking task via attention-gated reinforcement learning for brain–machine interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 3, pp. 458–467, May 2015.

[8] X. Zhang, C. Libedinsky, R. So, J. C. Principe, and Y. Wang, "Clustering neural patterns in kernel reinforcement learning assists fast brain control in brain–machine interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 9, pp. 1684–1694, Sep. 2019.

[9] J. DiGiovanna, B. Mahmoudi, J. Fortes, J. C. Principe, and J. C. Sanchez, "Coadaptive brain–machine interface via reinforcement learning," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 1, pp. 54–64, Jan. 2009.

[10] B. Wodlinger, J. E. Downey, E. C. Tyler-Kabara, A. B. Schwartz, M. L. Boninger, and J. L. Collinger, "Ten-dimensional anthropomorphic arm control in a human brain–machine interface: Difficulties, solutions, and limitations," *J. Neural Eng.*, vol. 12, no. 1, Dec. 2015, Art. no. 016011.

[11] T. Milekovic *et al.*, "An online brain–machine interface using decoding of movement direction from the human electrocorticogram," *J. Neural Eng.*, vol. 9, no. 4, Jun. 2012, Art. no. 046003.

[12] J. P. O'Doherty, P. Dayan, K. Friston, H. Critchley, and R. J. Dolan, "Temporal difference models and reward-related learning in the human brain," *Neuron*, vol. 38, no. 2, pp. 329–337, Apr. 2003.

[13] T. Brosch, F. Schwenker, and H. Neumann, "Attention-gated reinforcement learning in neural networks - A unified view," in *Proc. Int. Conf. Artif. Neural Netw.*, in Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 8131. Springer, 2013, pp. 272–279.

[14] J. Bae, L. G. Sanchez Giraldo, E. A. Pohlmeyer, J. T. Francis, J. C. Sanchez, and J. C. Príncipe, "Kernel temporal differences for neural decoding," *Comput. Intell. Neurosci.*, vol. 2015, pp. 1–17, Mar. 2015.

[15] M. M. Shanechi, Z. M. Williams, G. W. Wornell, R. C. Hu, M. Powers, and E. N. Brown, "A real-time brain–machine interface combining motor target and trajectory intent using an optimal feedback control design," *PLoS ONE*, vol. 8, no. 4, Apr. 2013, Art. no. e59049.

[16] M. M. Shanechi, A. L. Orsborn, H. G. Moorman, S. Gowda, S. Dangi, and J. M. Carmena, "Rapid control and feedback rates enhance neuroprosthetic control," *Nature Commun.*, vol. 8, no. 1, pp. 1–10, Jan. 2017.

[17] A. J. Suminski, D. C. Tkach, A. H. Fagg, and N. G. Hatsopoulos, "Incorporating feedback from multiple sensory modalities enhances brain–machine interface control," *J. Neurosci.*, vol. 30, no. 50, pp. 16777–16787, Dec. 2010.

[18] T. Spellman, M. Svei, J. Kaminsky, G. Manzano-Nieves, and C. Liston, "Prefrontal deep projection neurons enable cognitive flexibility via persistent feedback monitoring," *Cell*, vol. 184, no. 10, pp. 2750–2766, May 2021.

[19] D. Wu, H. Deng, X. Xiao, Y. Zuo, J. Sun, and Z. Wang, "Persistent neuronal activity in anterior cingulate cortex correlates with sustained attention in rats regardless of sensory modality," *Sci. Rep.*, vol. 7, p. 43101, Feb. 2017.

[20] M. Nakajima, L. I. Schmitt, and M. M. Halassa, "Prefrontal cortex regulates sensory filtering through a basal ganglia-to-thalamus pathway," *Neuron*, vol. 103, no. 3, pp. 445–458, 2019.

[21] J. H. Lui *et al.*, "Differential encoding in prefrontal cortex projection neuron classes across cognitive tasks," *Cell*, vol. 184, no. 2, pp. 489–506, Jan. 2021.

[22] M. M. Botvinick, J. D. Cohen, and C. S. Carter, "Conflict monitoring and anterior cingulate cortex: An update," *Trends Cognit. Sci.*, vol. 8, no. 12, pp. 539–546, 2004.

[23] C. B. Holroyd, M. G. H. Coles, S. Nieuwenhuis, W. J. Gehring, and A. R. Willoughby, "Medial prefrontal cortex and error potentials," *Science*, vol. 296, no. 5573, pp. 1610–1611, May 2002.

[24] M. I. Posner, M. K. Rothbart, B. E. Sheese, and Y. Tang, "The anterior cingulate gyrus and the mechanism of self-regulation," *Cognit., Affect., Behav. Neurosci.*, vol. 7, no. 4, pp. 391–395, Dec. 2007.

[25] K. R. Ridderinkhof, M. Ullsperger, E. A. Crone, and S. Nieuwenhuis, "The role of the medial frontal cortex in cognitive control," *Science*, vol. 306, no. 5695, pp. 443–447, 2004.

[26] M. F. S. Rushworth, M. P. Noonan, E. D. Boorman, M. E. Walton, and T. E. Behrens, "Frontal cortex and reward-guided learning and decision-making," *Neuron*, vol. 70, no. 6, pp. 1054–1069, Jun. 2011.

[27] A. Bechara and A. R. Damasio, "The somatic marker hypothesis: A neural theory of economic decision," *Games Econ. Behav.*, vol. 52, no. 2, pp. 336–372, 2005.

[28] G. Hajcak, J. S. Moser, C. B. Holroyd, and R. F. Simons, "The feedback-related negativity reflects the binary evaluation of good versus bad outcomes," *Biol. Psychol.*, vol. 71, no. 2, pp. 148–154, Feb. 2006.

[29] C. M. Warren, J. M. Hyman, J. K. Seamans, and C. B. Holroyd, "Feedback-related negativity observed in rodent anterior cingulate cortex," *J. Physiol.-Paris*, vol. 109, nos. 1–3, pp. 87–94, Feb. 2015.

[30] D. W. Bryden, E. E. Johnson, S. C. Tobia, V. Kashtelyan, and M. R. Roesch, "Attention for learning signals in anterior cingulate cortex," *J. Neurosci.*, vol. 31, no. 50, pp. 18266–18274, 2011.

[31] X. Shen, X. Zhang, Y. Huang, S. Chen, and Y. Wang, "Task learning over multi-day recording via internally rewarded reinforcement learning based brain machine interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 12, pp. 3089–3099, Dec. 2020.

[32] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *Proc. Neural Netw. Signal Process. VII. IEEE Signal Process. Soc. Workshop*, Sep. 1997, pp. 276–285.

[33] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[34] N. W. Prins, J. C. Sanchez, and A. Prasad, "Feedback for reinforcement learning based brain–machine interfaces using confidence metrics," *J. Neural Eng.*, vol. 14, no. 3, 2017, Art. no. 036016.

[35] F. Wang *et al.*, "Quantized attention-gated kernel reinforcement learning for brain–machine interface decoding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 873–886, Apr. 2017.

[36] X. Shen, X. Zhang, Y. Huang, S. Chen, and Y. Wang, "Modelling mPFC activities in reinforcement learning framework for brain–machine interfaces," in *Proc. 9th Int. IEEE/EMBS Conf. Neural Eng. (NER)*, Mar. 2019, pp. 243–246.

[37] X. Shen, X. Zhang, and Y. Wang, "Kernel temporal difference based reinforcement learning for brain machine interfaces," in *Proc. 43rd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Nov. 2021, pp. 6721–6724.

[38] B. Mahmoudi and J. C. Sanchez, "A symbiotic brain–machine interface through value-based decision making," *PLoS ONE*, vol. 6, no. 3, Mar. 2011, Art. no. e14760.

[39] B. T. Marsh, V. S. A. Tarigoppula, C. Chen, and J. T. Francis, "Toward an autonomous brain machine interface: Integrating sensorimotor reward modulation and reinforcement learning," *J. Neurosci.*, vol. 35, no. 19, pp. 7374–7387, May 2015.

[40] J. C. Principe, B. de Vries, and P. G. de Oliveira, "The gamma-filter—A new class of adaptive IIR filters with restricted feedback," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 649–656, Feb. 1993.

[41] K. Li and J. C. Principe, "The kernel adaptive autoregressive-moving-average algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 334–346, Feb. 2016.

[42] D. Tuia, J. Munoz-Mari, J. L. Rojo-Alvarez, M. Martinez-Ramon, and G. Camps-Valls, "Explicit recursive and adaptive filtering in reproducing kernel Hilbert spaces," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1413–1419, Jul. 2014.