# MARS: Multi-Agent Reinforcement Learning for Spatial-Spectral and Temporal Feature Selection in EEG-based BCI (Supplementary Material)

Dong-Hee Shin, Young-Han Son, Jun-Mo Kim, Hee-Jun Ahn, Jun-Ho Seo, Chang-Hoon Ji, Ji-Wung Han, Byung-Jun Lee, Dong-Ok Won, and Tae-Eui Kam

## I. Interpreting the Output Dimensions of EEGNet

As previously mentioned in the method section, the width and height of the feature map derived from EEGNet represent the temporal and spatial-spectral domains, respectively. To provide a more comprehensive explanation and to illustrate the feature extraction process of EEGNet, we present an overview in Fig. S1. This visualization helps substantiate our statement regarding the interpretation of the feature map dimensions.

Before delving into the explanation of Fig. S1, we first describe the notation used in this context to ensure a clear understanding. Here are the key notations: $f_s$ denotes the sampling frequency; $E$ represents the number of electrodes; $T'$ signifies the number of time points in the input signal; $k$ stands for the length of the kernel; $T$ corresponds to the number of time points in the extracted feature; and $C$ is used to denote the number of channels in the feature map.

In the first convolutional block of EEGNet, temporal kernels are used, which can act as bandpass filters for extracting spectral features from input EEG signals [1], [2]. Specifically, the first convolutional block captures spectral information at 2 Hz and above by setting the length of the kernel as half of the sampling rate. The second block of EEGNet uses depthwise convolution with a size of $(E, 1)$ to learn spatial filters. In detail, the network learns the spatial filter for each temporal kernel by utilizing depthwise convolution, enabling the extraction of frequency-specific spatial information. The last convolutional block uses separable convolution with $C$ kernels that efficiently merges multiple spatial-spectral features generated from the second block.

As a result, the output feature map derived from EEGNet is a combination of spatial-spectral features (along the height, represented by the y-axis), while still preserving temporal information (along the width, represented by the x-axis). Thus, the height of the output feature map can be viewed as the spatial-spectral domain, and the width represents the temporal domain.

## II. Theoretical Justification for Multi-Agent RL

In this section, we present the monotonic policy improvement theorem for cooperative multi-agent systems with a centralized critic, which offers theoretical guarantees for the performance of the technique proposed in this study. These theorems and lemma provide insights into the cooperative learning dynamics within our MARS framework and demonstrate how the centralized critic facilitates monotonic policy improvement in multi-agent settings.

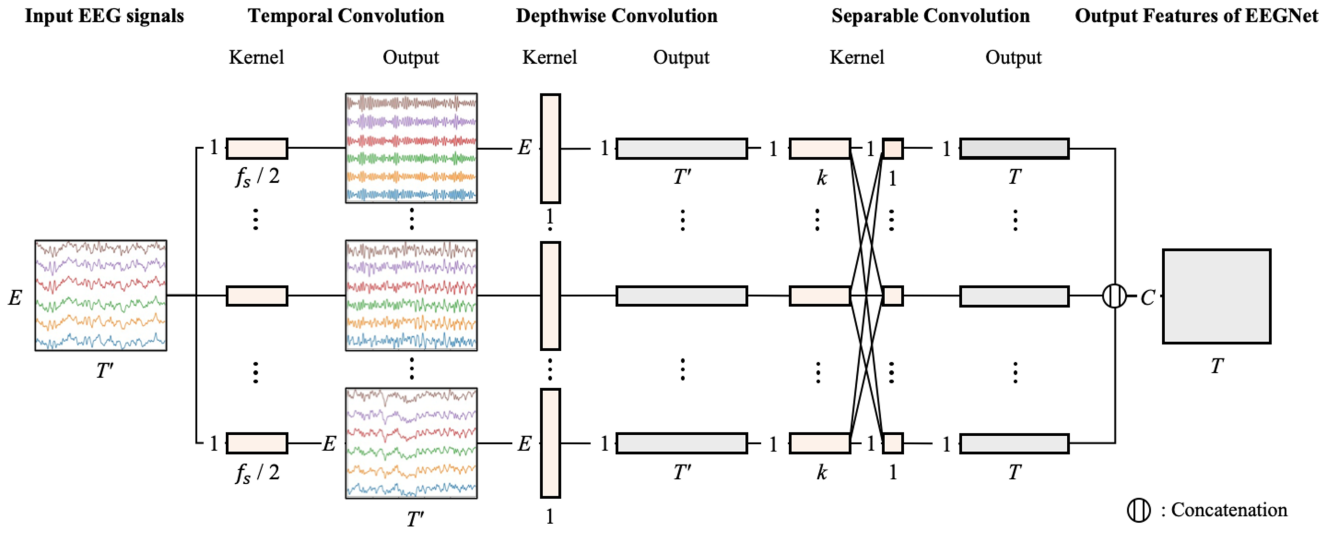


Fig. S1. Overview of the feature extraction process in EEGNet.

Before we delve into the theorem, it's important to mention that we employ the action-value function (Q-function) over the advantage function to simplify our approach in the theorem. Given that the advantage function is derived from the action-value function [3], this choice does not significantly impact the validity of the theorem and allows for a more straightforward representation of our theoretical analysis.

Monotonic policy improvement is a well-established and widely accepted principle in the context of single-agent RL systems [4]. It refers to the notion that as policies are iteratively updated, the performance of agents does not decrease, offering a theoretical guarantee of progress towards better solutions [4]. This policy improvement underpins many policy iteration-based algorithms and is employed in RL theorems to ensure reliable learning. The policy improvement in a single-agent RL can be formally stated as the following lemma.

**Lemma 1** (Policy Improvement in a Single-Agent RL Setting)**.** *Given a single-agent RL environment with state space $\mathcal{S}$, action space $\mathcal{A}$, value function V. If an agent updates its policy $\pi$ to $\pi'$ using the policy improvement operation, then the value function V under the new policy $\pi'$ will not decrease. For all $s \in \mathcal{S}$,*

$$V^{\pi'}(s) = \mathbb{E}_{\pi'}\left[R_t|s_t{=}s\right] \geq \mathbb{E}_{\pi}\left[R_t|s_t{=}s\right] = V^{\pi}\left(s\right), \tag{7}$$

*where $R_t$ represents the reward at time t, and $\mathbb{E}_\pi$ denotes the expectation under policy $\pi$.*

*Proof Sketch:* The proof is based on the principle of optimality in the Bellman equation [5] for $V$, which for a state-action value function $Q$ and policy $\pi$ is given by:

$$V^{\pi}(s) = \sum_{a \in A} \pi(a|s) Q^{\pi}(s,a). \tag{8}$$

Then, let us consider any state $s$, and define $a'$ as the action that maximizes the action-value function for the given state $s$, expressed as $a' = \arg\max_{a \in A} Q^{\pi}(s,a)$. We then have:

$$\begin{aligned} V^{\pi'}(s) &= Q^{\pi'}(s,a') \\ &\geq Q^{\pi}(s,a') \\ &\geq \sum_{a \in A} \pi(a|s) Q^{\pi}(s,a) \\ &= V^{\pi}(s). \end{aligned} \tag{9}$$

This equation indicates that the policy $\pi'$ is derived from the policy improvement operation, which acts to maximize the expected rewards, or equivalently, the value function. Hence, the value $V^{\pi'}(s)$ under the new policy $\pi'$ should be no less than $V^{\pi}(s)$ under the original policy $\pi$.

Note that our MARS framework operates within multi-agent systems, where agents collaborate to achieve common objectives. To formally analyze this multi-agent learning scenario, we now introduce Theorem 1, which extends Lemma 1 from the perspective of a single-agent scenario to a multi-agent setting. This theorem allows us to delve into the dynamic shift that occurs when transitioning from an individual agent to a team of collaborating agents.

**Theorem 1** (Policy Improvement in a Stationary Multi-Agent Setting)**.** *In a stationary multi-agent RL environment with $N$ agents, state space $\mathcal{S}$, action space $\mathcal{A}$, value function V. If each agent $i$, where $1 \leq i \leq N$, updates its policy $\pi_i$ to $\pi_i'$ using the policy improvement operation, while the policies $\pi_{-i}$ of all other agents remain unchanged, then the value function V under the new joint policy $\{\pi_i', \pi_{-i}\}$ does not decrease. For all $s \in \mathcal{S}$,*

$$\begin{aligned} V^{\{\pi_i', \pi_{-i}\}}(s) &= \mathbb{E}_{\{\pi_i', \pi_{-i}\}}\left[R_t|s_t{=}s\right] \\ &\geq \mathbb{E}_{\{\pi_i, \pi_{-i}\}}\left[R_t|s_t{=}s\right] = V^{\{\pi_i, \pi_{-i}\}}(s). \end{aligned} \tag{10}$$

*Proof Sketch:* The value function under the new policy will not decrease if each agent updates its policy based on the policy improvement principle described in Lemma 1. It should be noted that the environment dynamics remain stationary throughout the learning process.

However, in real-world multi-agent RL systems, the environment dynamics are inherently non-stationary. This means that the conditions or state of the system can change over time as each agent performs its own actions and influence the policies of other agents. Consequently, the policy improvement principle mentioned in Theorem 1 is not always guaranteed in such non-stationary environments. To address this issue and promote effective cooperation between agents, we introduce a centralized critic. The centralized critic ensures the consistency of Q-value evaluation and facilitates joint policy improvement, which is further elaborated in Lemma 2 and Theorem 2.

**Lemma 2** (Role of Centralized Critic in Cooperative Multi--Agent Settings)**.** *Given a cooperative multi-agent RL environment involving $N$ agents and the individual agent policies $\pi_i(\theta_i)$ parameterized by $\theta_i$, and the joint policy $\pi_{jt}$. Let $Q^{\mathcal{C}}$ denote the centralized critic, which takes as input the joint action and joint state, and outputs an evaluation of the quality of this joint action in this joint state. In a non-stationary environment, if agents follow policies that maximize the evaluation output by $Q^{\mathcal{C}}$, then their joint policy will converge towards an optimal policy that maximizes the expected joint reward.*

*Proof Sketch:* The proof relies on the policy gradient theorem under the cooperative multi-agent setting [6]. If agents update their policies by following the policy gradient ascent in the direction that maximizes the evaluation output by $Q^{\mathcal{C}}$:

$$\begin{aligned} \theta_{i,t+1} &= \theta_{i,t} + \alpha \nabla_{\theta_i} J(\theta_{i,t}) \\ &= \theta_{i,t} + \alpha \mathbb{E}_{\tau \sim \pi_{\theta_i}}\left[\nabla_{\theta_i} \log \pi_{\theta_i}(a_{i,t}|s_{i,t}) Q^{\mathcal{C}}(s_{jt}, a_{jt})\right], \end{aligned} \tag{11}$$

where $J(\theta_i) = \mathbb{E}_{\tau \sim \pi_{\theta_i}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$ represents the expected return under the policy $\pi_{\theta_i}$, $\alpha$ denotes the learning rate, and the expectation is taken over trajectories $\tau$ sampled from policy $\pi_{\theta_i}$. Assuming sufficient exploration, ensured by having actions selected according to:

$$a_t = \begin{cases} \text{random action} & \text{with probability } \epsilon, \\ \arg\max_a \ Q^{\mathcal{C}}(s_t, a) & \text{with probability } 1-\epsilon, \end{cases} \tag{12}$$

and assuming ergodicity of the environment (the possibility to reach any state from any other state), and that the learning rate $\alpha$ meets the Robbins-Monro conditions [7]:

$$\sum_t \alpha_t = \infty, \quad \sum_t \alpha_t^2 < \infty. \tag{13}$$

In a non-stationary environment, their joint policy may continue to be updated to maximize the expected joint reward.

**Theorem 2** (Cooperative Multi-Agent Monotonic Policy Improvement Theorem with Centralized Critic)**.** *In a cooperative multi-agent RL environment with a centralized critic $Q^{\mathcal{C}}$, if each agent $i$, where $1 \leq i \leq N$, independently improves its policy $\pi_i(\theta_i)$ using the policy improvement operation based on $Q^{\mathcal{C}}$, while the policies of all other agents remain unchanged, then the joint policy under the new policies does not decrease in terms of expected joint reward.*

*Proof Sketch:* Based on Lemma 1 and 2, each agent $i$ updates its policy $\pi_i(\theta_i)$ to a new policy $\pi_i'(\theta_{i,t+1})$ using the policy improvement operation based on the centralized critic. The direction of the policy update ensures that the expected return under the new policy, $J(\theta_{i,t+1})$, is at least as much as the expected return under the old policy, $J(\theta_{i,t})$. Hence, we can write:

$$J(\theta_{i,t+1}) \geq J(\theta_{i,t}), \tag{14}$$

which indicates the expected return does not decrease following the policy update. Subsequently, given the mild conditions under which we operate, where the $\arg\max$ operation on the joint policy is feasible, as indicated by equation (12), the joint policy $\pi_{jt} = \{\pi_1(\theta_{1,t}), \pi_2(\theta_{2,t}), ..., \pi_N(\theta_{N,t})\}$ under the new policies of all agents will also not decrease in terms of expected joint reward:

$$Q^{\mathcal{C}}(s_{jt}, a_{jt}|\pi_{jt}) \geq Q^{\mathcal{C}}(s_{jt}, a_{jt}|\pi_{jt-1}), \tag{15}$$

where $\pi_{jt-1}$ denotes the joint policy prior to the update. Again, this proof holds under the assumption that the $\arg\max$ operation on the joint policy is feasible, ensuring monotonic policy improvement in a cooperative multi-agent RL environment with a centralized critic.

## III. Pseudo-code of MARS framework

We provide the pseudo-code for the training process of our MARS framework as shown in Algorithm 1. We want to remark that there are three main components in the MARS framework: the actor network for the spatial-spectral agent $\theta^{\text{SS}}$, another actor network for the temporal agent $\theta^{\text{TP}}$, and the centralized critic network $\psi$. Note that these components are represented by neural network models. The parameters of the centralized critic and actor networks are updated through episodes from the replay buffer based on equation (4) and equation (6) in our main manuscript, respectively.

**Algorithm 1** Pseudo-code for the MARS framework

**Input**: Initial feature map $F_0$
**Input**: Spatial-spectral agent $\theta^{\text{SS}}$, Temporal agent $\theta^{\text{TP}}$, Critic $\psi$
**Output**: Selected feature map $F_T$
Initialize Buffer $B$
Initialize $s_0 \leftarrow F_0$
**for** *time step t=0 to T-1* **do**
  $o_t^{\text{TP}}$, $o_t^{\text{SS}} \leftarrow$ Eq. (1)
  $a_t^{\text{TP}} \leftarrow \theta^{\text{TP}}(o_t^{\text{TP}})$
  $a_t^{\text{SS}} \leftarrow \theta^{\text{SS}}(o_t^{\text{SS}})$
  $\boldsymbol{a_t} \leftarrow a_t^{\text{TP}} \cdot a_t^{\text{SS}}$
  $r_t \leftarrow$ Eq. (2)
  Add episode $(s_t, \boldsymbol{a_t}, r_t)$ to $B$
**end**
**for** *transition* $(s_t, \boldsymbol{a_t}, r_t, s_{t+1}, \boldsymbol{a_{t+1}})$ *in* $B$ **do**
  $\mathcal{L}_t(\psi) \leftarrow$ Eq. (4)
  $\mathcal{L}_t(\theta^{\text{TP}}) \leftarrow$ Eq. (6)
  $\mathcal{L}_t(\theta^{\text{SS}}) \leftarrow$ Eq. (6)
**end**
Update $\psi \leftarrow \psi - \alpha \sum_t \mathcal{L}_t(\psi)$
Update $\theta^{\text{TP}} \leftarrow \theta^{\text{TP}} - \alpha \sum_t \mathcal{L}_t(\theta^{\text{TP}})$
Update $\theta^{\text{SS}} \leftarrow \theta^{\text{SS}} - \alpha \sum_t \mathcal{L}_t(\theta^{\text{SS}})$
Obtain $F_T \leftarrow s_T$

## References


[1] W. Ko, E. Jeon, S. Jeong, and H.-I. Suk, "Multi-scale neural network for EEG representation learning in BCI," *IEEE Comput. Intell. Mag.*, vol. 16, no. 2, pp. 31–45, 2021.

[2] Z. Jia, Y. Lin, J. Wang, K. Yang, T. Liu, and X. Zhang, "MMCNN: A multi-branch multi-scale convolutional neural network for motor imagery classification," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2021, pp. 736–751.

[3] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[4] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun, "Reinforcement learning: Theory and algorithms," *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, vol. 32, 2019.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[6] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[7] E. Moulines and F. Bach, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," *Adv. Neural Inf. Process. Syst.*, vol. 24, 2011.