Draft Report: Integrating Brain-Computer Interfaces with Adaptive AI

Student ID: 11314389

December 8, 2025

1 Introduction

Research in motor imagery Brain–Computer Interfaces (BCIs) has delivered impressive offline classification accuracies, yet two structural barriers continue to block real-world deployment. First, electroencephalogram (EEG) signals are notoriously *non-stationary.* Any static decoder—whether handcrafted CSP+LDA or deep convolutional networks—tends to overfit the calibration session; as soon as the neural state drifts, accuracy collapses. Second, even when classification works, the control pipeline remains *open-loop*: the predicted label is mapped directly to a rigid command, making it difficult to generate smooth, corrective actions for robotic end-effectors.

Classical pipelines rely on OVR-CSP to maximise class variance ratios, followed by linear classifiers that assume a stationary feature manifold. Deep neural nets improve representation learning but still operate as one-shot recognisers, lacking the feedback needed to adapt to drift or to arbitrate between task goals and noisy sensory evidence. As a result, conventional decoders struggle to produce continuous, stable trajectories suitable for robot control.

This project explores a new hypothesis: by reframing BCI decoding as a *sequential decision-making* problem, we can leverage reinforcement learning (RL) to close the loop. Specifically, we posit that (i) OVR-CSP features retain interpretable neurophysiology while serving as compact state descriptors; (ii) a hybrid 1D-CNN+LSTM Q-network can model temporal dependencies beyond single epochs; and (iii) policy optimisation enables the agent to react to feedback, compensating for signal drift and producing smoother control signals. Figure 1 sketches the resulting pipeline. The remainder of this report details the methodology, implementation status, and evaluation plan that operationalise this hypothesis.

2 Methodology

This chapter documents the end-to-end system devised to translate raw EEG into robotic control commands. The current implementation contains three fully operational phases—data processing, feature engineering, and reinforcement-learning training—plus a planned execution phase for simulation and robot deployment.

2.1 Phase 1: Data Acquisition & Preprocessing

2.1.1 Dataset and Neurophysics

We adopt the BCI Competition IV-2a dataset [5]. Motor imagery (MI) elicits **Event-Related Desynchronization/Synchronization (ERD/ERS)** in the **Mu (8–13 Hz)** and **Beta (13–30 Hz)** bands over the contralateral sensorimotor cortex [1]. These modulations provide the electrophysiological signatures exploited downstream.

2.1.2 Spectral Filtering Strategy

MNE-Python implements a zero-phase FIR bandpass:

$$x_{\text{band}}(t) = \mathcal{F}^{-1}\{H(\omega)\mathcal{F}[x(t)]\}, \quad H(\omega) = \begin{cases} 1, & \omega \in [8, 30] \text{ Hz} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The passband suppresses EOG-dominated low frequencies and EMG-rich high frequencies while retaining ERD/ERS dynamics.

2.1.3 Artifact Removal via ICA

Independent Component Analysis isolates ocular artefacts:

$$X = AS, \qquad \hat{S} = WX. \tag{2}$$

Components strongly correlated with dedicated EOG channels form the rejection set $\mathcal{I}_{\text{bad}}$, yielding cleaned signals

$$\tilde{X} = A\tilde{S}, \quad \tilde{S}_i = 0 \text{ if } i \in \mathcal{I}_{\text{bad}}. \tag{3}$$

2.1.4 Epoching and Output Artifacts

Cue-locked windowing produces tensors $X \in \mathbb{R}^{N \times C \times T}$. Each subject's cleaned raw, ICA report, and epoch data are persisted in `.fif` format. Amplitudes remain on the physical scale; optional normalisation is reserved for future experiments.

[Insert System Overview Diagram Here]
EEG Raw → Preprocessing → CSP/LDA → DQN Agent → Control.

Figure 1: Overview of the proposed BCI control framework.

2.2 Phase 2: Feature Engineering & Baseline

2.2.1 One-vs-Rest CSP

For each class k, Common Spatial Patterns solve

$$\Sigma_k w = \lambda(\Sigma_k + \Sigma_{\neg k})w, \qquad J(w) = \frac{w^\top \Sigma_k w}{w^\top \Sigma_{\neg k} w}, \tag{4}$$

producing filters W and log-variance features f_i. The inverse matrix $P = W^{-1}$ supports topographic verification.

2.2.2 Baseline Hypothesis with LDA

CSP features feed an LDA classifier evaluated via stratified K-fold cross-validation. The resulting metrics, filters, and transformed features are exported (`csp_features.npz`, `csp_filters.npy`, `csp_patterns.npy`, `csp_metrics.json`), ensuring the RL module consumes identical representations.

2.3 Phase 3: RL Framework Definition

We reformulate control as an MDP [3]; Table 1 summarises the specification.

Table 1: **Specification of the Markov Decision Process (MDP).**

Component	Notation	Definition / Description
State Space	s_t	Stack of the latest L CSP feature vect ($s_t \in \mathbb{R}^{L \times N_{\mathrm{CSP}}}$), aligned with the repl buffer tensors.
Action Space	a_t	Discrete end-effector comman {Left, Right, Up, Down}.
Reward	r_t	Label-aligned signal $r_t \in \{+1, -\lambda\}$ us for current offline training; environme derived rewards will replace this duri closed-loop deployment.
Transition	$\mathcal{P}$	Offline updates sample (s_{t+1}) fro stored transitions; simulators/hardwa will provide $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$.
Discount	γ	Default $\gamma = 0.99$ balances short- a long-term objectives.

2.3.1 Function Approximation

The Q-function $Q(s, a; \theta)$ is implemented as shown in Fig. 2. Layer specifications correspond to `dqn_model.py` (Table 2).

[Insert 1D-CNN + LSTM Architecture Diagram Here]
Q-network with convolutional front-end and recurrent head.

Figure 2: Hybrid Q-network combining 1D-CNN and LSTM.

Table 2: **Architecture of the 1D-CNN+LSTM Q-Network.**

Layer	Type / Params	Activation	Output Shape
Input	CSP feature stack	–	(B, L, N_{CSP})
1	Conv1D($N_{\text{CSP}} \to 64$, $k = 3$, $p = 1$)	ReLU	$(B, L, 64)$
2	Conv1D($64 \to 64$, $k = 3$, $p = 1$)	ReLU	$(B, L, 64)$
3	LSTM(hidden= 128, layers= 1)	tanh /σ	$(B, L, 128)$
4	Dense($128 \to 128$)	ReLU	$(B, 128)$
Output	Dense($128 \to \lvert\mathcal{A}\rvert$)	Linear	$(B, 4)$

Note: B denotes batch size, L sequence length.

2.3.2 Training Mechanism

Experience Replay, a soft-updated target network, and Huber loss

$$L_\delta(y, Q) = \begin{cases} \frac{1}{2}(y - Q)^2, & |y - Q| \le \delta \\ \delta(|y - Q| - \frac{1}{2}\delta), & \text{otherwise} \end{cases} \quad (5)$$

stabilise training, with $y = r_t + \gamma \max_{a'} Q_{\text{target}}(s_{t+1}, a'; \theta^-)$.

2.4 Phase 4: Execution & Evaluation (Planned)

The repository currently outputs metrics (accuracy, confusion matrix, performance gap vs. CTNet baseline). Deployment to PyBullet/Gym and hardware is planned; Table 3 summarises the intended robustness tiers.

Table 3: **Robustness Evaluation Scenarios and Success Criteria (planned).**

Tier	Condition	Evaluation Goal
Tier 1: Baseline	Clean IV-2a data (current)	Establish performance ceiling and confirm policy convergence.
Tier 2: Noisy	Additive white Gaussian noise, SNR $\in [5, 15]$ dB (planned)	Stress-test robustness when channel quality degrades.
Tier 3: Artifact	Simulated eye-blink artefacts superimposed on epochs (planned)	Validate ICA preprocessing and learned policy against non-stationary artefacts.

2.4.1 Engineering Metrics & Iterative Optimisation

Control smoothness

$$S = \frac{1}{T - 1} \sum_{t=2}^{T} \|a_t - a_{t-1}\|_2,$$

task success rates, and end-effector error will guide future reward shaping. Should S remain high, an additional penalty $-\beta\|a_t - a_{t-1}\|_2$ will be integrated in the RL objective.

References

[1] I. Hameed *et al.*, "Enhancing motor imagery EEG signal decoding through machine learning," *Computers in Biology and Medicine*, 2025.

[2] W. Zhao *et al.*, "A Brain-Computer Interface Control System Design Based on Deep Learning," University of Manchester, 2024.

[3] S. Nallani and G. Ramachandran, "RLEEGNet: Integrating Brain-Computer Interfaces with Adaptive AI," *arXiv:2402.09465*, 2024.

[4] D.-H. Shin *et al.*, "MARS: Multiagent reinforcement learning for spatial-spectral and temporal feature selection," *IEEE T-SMC*, 2024.

[5] J.-H. Jeong *et al.*, "Multimodal signal dataset for 11 intuitive movement tasks," *GigaScience*, 2020.