ThoughtLink

Closed-Loop Brain-to-Robot Control via
Non-Invasive EEG Decoding and Humanoid Simulation

Technical Report — Kernel & Dimensional VC Track

Joshua (ML Engineer) Mourad (Simulation Engineer) Dimitri (Full-Stack Developer)

February 2026

Abstract

ThoughtLink decodes non-invasive brain signals (EEG) into discrete robot commands and demonstrates closed-loop brain-to-robot control in a MuJoCo humanoid simulation. The system processes 6-channel EEG recordings through a compact convolutional neural network (EEGNet, 12.6K parameters), applies temporal stabilisation for flicker suppression, and dispatches commands to a simulated G1 humanoid at 4 Hz. A web dashboard provides real-time visualisation of the full pipeline. This report details the machine learning implementation, simulation bridge, and frontend/backend architecture.

Contents

1 System Overview

Section to be completed.

2 Dataset

The dataset comprises 900 `.npz` recordings from 6 subjects across 20 sessions. Each file contains a 15-second trial with EEG sampled at 500 Hz across 6 channels (shape 7499×6) and TD-fNIRS moment features (shape $72 \times 40 \times 3 \times 2 \times 3$, unused in this work). Five motor-imagery classes are perfectly balanced at 180 trials each.

Table 1: Class definitions and robot command mapping.

Index	Brain Label	Robot Command	Behaviour
0	Right Fist	`ROTATE_RIGHT`	Turn right
1	Left Fist	`ROTATE_LEFT`	Turn left
2	Both Fists	`STOP` / Gear-dep.	Halt / grab
3	Tongue Tapping	`FORWARD` / `SHIFT_GEAR`	Walk / shift
4	Relax	`IDLE`	Stand still

Trial durations: mean 10.0 ± 0.48 s (range 8.4–11.3 s). Stimulus onset occurs at $t = 3$ s (sample 1500). Zero files were flagged as corrupted or malformed. Subject trial counts ranged from 90 to 180 per subject.

3 Machine Learning Pipeline

3.1 Preprocessing

Bandpass filter. A 4th-order Butterworth bandpass filter (1–40 Hz) is applied per-channel using `scipy.signal.filtfilt` for zero-phase distortion. This passband captures the mu (8–12 Hz), beta (13–30 Hz), and theta (4–8 Hz) rhythms associated with motor imagery while rejecting DC drift below 1 Hz and high-frequency EMG artefacts above 40 Hz.

Normalisation. Per-channel z-score normalisation ($\mu = 0$, $\sigma = 1$) is applied after filtering, with a guard condition ($\sigma < \epsilon \Rightarrow \sigma = 1$) to prevent division by zero on flat channels. A final `nan_to_num` pass removes any residual NaN or Inf values.

Windowing. Fixed 1-second windows (500 samples) are extracted at a 125-sample stride (0.25 s):

- **Active classes** (0–3): Windows from the stimulus region, starting at sample 1500. Trials shorter than 2 s are discarded.
- **Relax** (class 4): Random non-overlapping windows from the full 15 s signal (~20 per trial, seed-controlled for reproducibility).

This yields **13,100 windows** for binary and **28,772 windows** for 5-class classification.

Data splitting. A random stratified 80/10/10 split is used. An initial experiment with subject-based splitting (4 train / 1 val / 1 test out of 6 subjects) revealed severe cross-subject generalisation failure: training accuracy reached ~60% while validation accuracy remained at chance (~50% binary, ~20% 5-class). The random split assumes a calibrated-per-subject deployment model, which is standard practice in BCI systems where per-user calibration data is available.

3.2 Baseline Models

Feature extraction uses Welch's PSD (nperseg=256) decomposed into four frequency bands per channel:

$$\mathbf{f} = \Big[\underbrace{P_\theta, P_\alpha, P_\mu, P_\beta}_{\text{4 bands}} \times \underbrace{6\ \text{ch}}_{=24}\Big] \oplus \Big[\underbrace{\tfrac{P_\mu}{P_\beta}}_{\text{6 ratios}}\Big] \oplus \Big[\underbrace{\text{Var}(x_{\text{ch}})}_{\text{6 variances}}\Big] = 36 \text{ features} \tag{1}$$

Table 2: Baseline classifier accuracy (%).

Model	**Binary**	**5-Class**
Logistic Regression (balanced)	51.1	20.4
SVM-RBF (balanced)	52.6	23.1
Chance level	50.0	20.0

Both baselines barely exceed chance. Hand-crafted PSD features collapse the temporal structure of each 1-second window into summary statistics, discarding the sequential dynamics that distinguish motor imagery classes. This motivates learned convolutional features.

3.3 EEGNet Architecture

The model adapts the EEGNet architecture [1]. Input shape: $(B, 1, 500, 6)$.

Table 3: EEGNet layer structure. B = batch size.

#	**Operation**	**Output**	**Notes**
1	Conv2d$(1, 32, (64, 1))$ pad$(32, 0)$	$(B, 32, 500, 1)$	Temporal, kernel = 128 ms
2	DepthwiseConv2d$(32, 64, (1, 6))$ groups= 32	$(B, 64, 500, 1)$	Spatial filter
3	BN $\to$ ELU $\to$ AvgPool$(4, 1)$ $\to$ Drop(0.3)	$(B, 64, 125, 1)$	Regularisation
4	SepConv2d$(64, 64, (16, 1))$	$(B, 64, 125, 1)$	Depthwise + pointwise
5	BN $\to$ ELU $\to$ AvgPool$(8, 1)$ $\to$ Drop(0.3)	$(B, 64, 15, 1)$	Second regularisation
6	Flatten $\to$ Linear$(960, C)$	(B, C)	$C \in \{2, 5\}$

Total trainable parameters: **9,730** (binary) / **12,613** (5-class). The flat dimension (960) is computed dynamically via a dummy forward pass through the feature extractor.

3.4 Training Protocol

Table 4: Training hyperparameters.

Parameter	Value
Optimiser	AdamW
Learning rate	1×10^{-3}
Weight decay	1×10^{-4}
Scheduler	CosineAnnealingLR ($T_{\max} = 100$)
Batch size	64
Max epochs	100
Early stopping	Patience = 25 epochs
Loss	CrossEntropyLoss (class-weighted, label smoothing = 0.1)
Gradient clipping	max_norm = 1.0
Augmentation	Gaussian noise ($\sigma = 0.05$) + amplitude scaling (0.9–1.1×)
Hardware	NVIDIA RTX 3070 Ti (8 GB)

Class weights are computed via inverse frequency to compensate for the Relax class having fewer extracted windows. Label smoothing at 0.1 prevents softmax saturation, which benefits the downstream confidence-gating mechanism by preserving calibrated uncertainty.

3.5 Results

3.5.1 Binary Classification (Right Fist vs Left Fist)

Table 5: Binary EEGNet results (9,730 parameters, 100 epochs).

Class	Precision	Recall	F1	Support
Right Fist	0.621	0.688	0.653	654
Left Fist	0.651	0.581	0.614	656
Overall accuracy		**63.4%**		
Best validation accuracy		63.5% (epoch 96)		

Confidence analysis: correct predictions averaged 0.662 confidence vs 0.600 for incorrect (gap = 0.062). The narrow gap is a known property of label smoothing.

3.5.2 Five-Class Classification

Table 6: Five-class EEGNet results (12,613 parameters, 93 epochs, early stopped).

Class	Precision	Recall	F1	Support
Right Fist	0.339	0.279	0.306	680
Left Fist	0.283	0.300	0.291	616
Both Fists	0.319	0.305	0.312	652
Tongue Tapping	**0.509**	**0.564**	**0.536**	668
Relax	0.207	0.237	0.221	262
Overall accuracy		**35.2%**		
Best validation accuracy		36.8% (epoch 68)		

Tongue Tapping is the most separable class (F1 = 0.536, recall = 56.4%), likely because tongue movement produces distinct mu-desynchronisation patterns in central electrodes. Left/Right Fist form the most confused pair, consistent with overlapping contralateral motor cortex activations on a 6-channel montage. Relax is weakest (F1 = 0.221): the absence of motor imagery has no distinctive spectral signature at this channel resolution.

Confusion matrix analysis reveals the largest off-diagonal mass between Right Fist, Left Fist, and Both Fists (the three motor imagery classes sharing overlapping cortical representations), while Tongue Tapping draws relatively few false positives.

3.6 Binary vs Multi-Class Decoding

Binary decoding (63.4%) substantially outperforms the corresponding 2-class subset from the 5-class model. The binary model concentrates its 9.7K parameter budget on a single decision boundary, while the 5-class model distributes 12.6K parameters across 10 pairwise boundaries.

For robot control, a **hierarchical approach** is suggested: first classify Tongue Tapping vs motor imagery vs Relax (3-class, leveraging Tongue Tapping's separability), then refine motor imagery into Left/Right/Both. Per-class F1 scores suggest this would improve overall accuracy by 5–10%. This was not implemented due to hackathon time constraints.

3.7 Temporal Context vs Instantaneous Prediction

Each prediction consumes a 1-second window (500 samples). The EEGNet temporal convolution kernel spans 128 ms (64 samples), providing access to local temporal dynamics within each window but no cross-window context. At the 0.25 s stride, the system produces 4 predictions per second.

The **TemporalStabiliser** bridges the gap between noisy per-window predictions and stable robot commands via three cascaded mechanisms:

1. **Confidence gating** (threshold = 0.7): Predictions below 70% softmax confidence are suppressed to `IDLE`. In the 5-class model, mean correct confidence is 0.396—below the gate—so only the model's highest-conviction predictions pass through. This is intentional: missed commands are preferable to false triggers in robot control.
2. **Majority voting** (buffer = 5): A sliding deque of the last 5 high-confidence predictions. A class must win $\geq 3/5$ votes to be emitted, filtering transient spikes.
3. **Hysteresis** (count = 3): The voted class must appear 3 consecutive times before the system switches commands, adding $\sim$0.75 s latency to transitions but eliminating inter-class oscillation.

The combined effect: total latency from neural event to command switch is $\sim$1–2 s. In closed-loop robot control, a 2-second jitter-free command is more operationally useful than sub-second noisy commands.

3.8 Confidence Thresholds and False-Trigger Suppression

Table 7: Confidence distribution statistics.

Metric	Binary	5-Class
Mean confidence (correct)	0.662	0.396
Mean confidence (incorrect)	0.600	0.317
Confidence gap	0.062	0.079

The 0.7 threshold aggressively gates the 5-class model: the majority of even correct predictions fall below it. In practice, only Tongue Tapping (with a long confidence tail reaching > 0.7)

and occasional high-confidence motor imagery predictions pass. The robot therefore defaults to `IDLE/STOP` unless the brain signal is unambiguous.

The 5-class confidence gap (0.079) is larger in relative terms than binary (0.062), indicating the model has calibrated uncertainty—it is measurably less confident when wrong. Label smoothing contributes to this by preventing softmax saturation, which is beneficial for the gating mechanism.

3.9 Model Complexity vs Inference Speed

Table 8: ONNX inference latency (ms), 1000 runs post-warmup.

Model	Params	CPU Mean	CPU P95	CUDA Mean	CUDA P95
Binary	9,730	0.43	0.51	0.90	1.74
5-Class	12,613	0.46	0.59	0.92	1.15

Key findings:

- **CPU outperforms CUDA** at this model scale. CPU–GPU memory transfer overhead dominates when the computation itself takes < 0.5 ms. CUDA only benefits batch inference or larger architectures.
- Adding 30% more parameters (9.7K $\rightarrow$ 12.6K) incurs only **7% additional latency** on CPU. The model is memory-bandwidth-bound, not compute-bound; doubling parameters would be nearly "free" in latency terms.
- Both models are **well within the 10 ms real-time budget**. The full BrainDecoder pipeline (preprocessing + ONNX inference + stabiliser) totals $\sim$3 ms, leaving ample headroom within the 250 ms window stride.
- ONNX verification confirmed PyTorch-to-ONNX fidelity: maximum absolute output difference $< 1.2 \times 10^{-6}$.

3.10 Limitations and Future Work

Channel count. Six EEG channels provide limited spatial resolution. Left/right motor imagery discrimination relies on contralateral differences over C3/C4; a denser montage or source localisation would likely improve the Left vs Right Fist boundary.

Subject generalisation. Subject-based splitting yielded chance-level performance ($\sim$50% binary). The random split assumes per-subject calibration, acceptable for demonstration but not for zero-calibration deployment. Transfer learning and domain adaptation remain open research problems in BCI.

Relax detection. The weakest class (F1 = 0.221). "Absence of motor imagery" lacks a distinctive neural signature. A dedicated rest detector (e.g., alpha power threshold) may outperform the learned classifier for this class.

Hierarchical classification. Per-class results strongly suggest a two-stage approach: detect Tongue Tapping first (highest separability), then subclassify motor imagery. Not implemented due to time constraints, but estimated to yield 5–10% improvement.

4 Simulation Bridge

Section to be completed.

5 Backend Architecture

Section to be completed.

6 Frontend Dashboard

Section to be completed.

7 System Integration

Section to be completed.

8 Conclusion

Section to be completed.

References

[1] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, “EEGNet: A compact convolutional neural network for EEG-based brain–computer interfaces,” *Journal of Neural Engineering*, vol. 15, no. 5, p. 056013, 2018.