Gated Uncertainty-Aware Runtime Dual Invariants for Neural Signal-Controlled Robotics

Tasha Kim
Oxford Robotics Institute (ORI)
Department of Engineering Science
University of Oxford
tashakim@eng.ox.ac.uk

Oiwi Parker Jones
Oxford Robotics Institute (ORI)
Department of Engineering Science
University of Oxford
oiwi.parkerjones@eng.ox.ac.uk

Abstract

Safety-critical assistive systems that directly decode user intent from neural signals require rigorous guarantees of reliability and trust. We present GUARDIAN (Gated Uncertainty-Aware Runtime Dual Invariants), a framework for real-time neuro-symbolic verification for neural signal-controlled robotics. GUARDIAN enforces both logical safety and physiological trust by coupling confidence-calibrated brain signal decoding with symbolic goal grounding and dual-layer runtime monitoring. On the BNCI2014 motor imagery electroencephalogram (EEG) dataset with 9 subjects and 5,184 trials, the system performs at a high safety rate of 94–97% even with lightweight decoder architectures with low test accuracies (27–46%) and high ECE confidence miscalibration (0.22–0.41). We demonstrate ≈1.7x correct interventions in simulated noise testing versus at baseline. The monitor operates at 100Hz and sub-millisecond decision latency, making it practically viable for closed-loop neural signal-based systems. Across 21 ablation results, GUARDIAN exhibits a graduated response to signal degradation, and produces auditable traces from intent, plan to action, helping to link neural evidence to verifiable robot action.

1 Introduction

Neural signal-controlled robots have significant potential to improve accessibility for individuals with limited mobility but they also introduce important safety risks[32]. Maintaining runtime accuracy and implementing reliable intervention mechanisms are paramount in closed-loop systems to ensure user safety[3, 7, 15]. Recent closed-loop EEG-based assistive systems [23, 34] and AI-enabled brain-computer interfaces (BCIs) [25] show encouraging progress, but remain vulnerable to ambiguous user intent, a known challenge in shared autonomy and teleoperation[9, 19, 20]. They also suffer signal degradation during long-horizon tasks[22, 27, 29, 31] and lack formal or interpretable safety mechanisms like runtime assurance[3, 15], or shielding[4]. We propose GUARDIAN, a physiological runtime verification architecture that provides an auditable, explainable layer between neural decoding and execution. GUARDIAN acts as a safety gate that complements neurally-controlled systems, producing tracing and intervention tooling without disrupting real-time operations.

Design principle and goals. GUARDIAN prioritizes the following principles: (a) conservative safety (*halting* over execution when neural evidence is weak or contradictory), (b) interpretability (traceable action chains from raw brain signal→intent→plan→action compatible with common symbolic planning toolchains like PDDL [10]), (c) modularity (wraps any decoder with $<$ 1ms overhead), and (d) tunability (operators can set confidence thresholds informed by calibration analysis).

Threat model and scope. Our system operates with non-invasive neural signals in assistive robot domains, oftentimes challenging due to noise interference, non-stationarity–within-session accuracy drop and between-session variance, neural artifacts, and subject fatigue[6, 29]. We address confidence

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Embodied and Safe-Assured Robotic Systems (E-SARS).

miscalibration, where decoders show high confidence despite poor accuracy[13, 26]. When there are no violations, our monitor gates actuation and operates under standard action safety protocols[1, 2, 18]. In assistive robotics, hardware can pose physical risks for users when there is uncontrolled device activation or delayed shutdown[3, 18]. Attacks or adversary threats are outside our scope.

Contributions. We make the following contributions: (I) a safety-centric framework achieving high safety rates under severe signal degradation, (II) a verifiable neuro-symbolic pipeline that formally links EEG distributions to symbolic goals, and (III) a practical lightweight architecture with minute computational overhead enabling deployable and trustworthy neural signal-based control.

2 GUARDIAN: Formalization and Algorithm

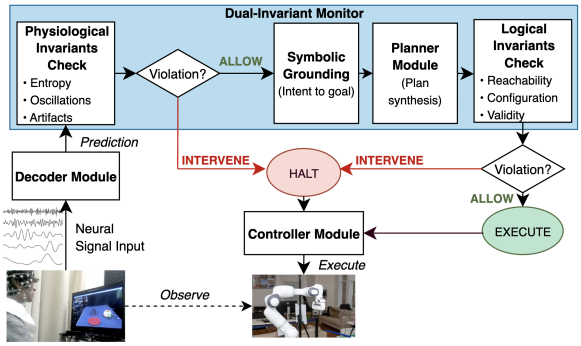

Figure 1: **System architecture overview.**

Let $x_t \in \mathbb{R}^{C\times W}$ denote the EEG window at time t (C channels, W samples). A decoder f_θ outputs a categorical posterior $p_t \in \Delta^3$ over the action set $\mathcal{H} = \{\text{GRASP}, \text{RELEASE}, \text{MOVE\_TO}, \text{ROTATE}\}$ corresponding to left hand, right hand, feet, and tongue motor imagery (MI), respectively. These actions enable natural control (positioning, grasping, releasing, orienting) for EEG-based manipulation. We treat p_t as a 4-dimensional probability vector over $\mathcal{H}$, i.e., $p_t \in \Delta^3 \subset \mathbb{R}^4$ (3-simplex). Let $\mathbf{u}$ denote the uniform distribution on $\mathcal{H}$, i.e., $\mathbf{u}(h) = \frac{1}{|\mathcal{H}|} = \frac{1}{4}$ for all $h \in \mathcal{H}$. Unless stated otherwise, $\log$ denotes the natural logarithm and $0 \log 0 := 0$.

Calibration-aware safety. To mitigate the effects of miscalibration, we construct a calibrated intent distribution by convexly mixing with the uniform prior $\tilde{p}_t = \alpha_m p_t + (1-\alpha_m)\,\mathbf{u}, \alpha_m \in [0.5, 0.8]$, where $\mathbf{u}$ is the uniform distribution on $\mathcal{H}$. Since this is a convex combination, $\tilde{p}_t \in \Delta^3$. Epistemic uncertainty is quantified using the normalized Shannon entropy

$$H(\tilde{p}_t) \;=\; -\sum_{h\in\mathcal{H}} \tilde{p}_t(h)\,\log \tilde{p}_t(h), \qquad \hat{H}(\tilde{p}_t) \;=\; \frac{H(\tilde{p}_t)}{\ln|\mathcal{H}|} \in [0,1].$$

Rapid intent flips are detected by computing the oscillation index

$$\Omega_t \;=\; \frac{1}{K-1} \sum_{k=t-K+1}^{t-1} \mathbb{1}\left\{ \operatorname*{argmax}_{h\in\mathcal{H}} \tilde{p}_k(h) \;\neq\; \operatorname*{argmax}_{h\in\mathcal{H}} \tilde{p}_{k+1}(h) \right\},$$

over the last K frames ($K{=}10$ at 100 Hz, $K \geq 2$). To obtain a scalar artifact score comparable to a threshold, the per-channel band-limited RMS energies in $[20, 45]$ Hz are z-scored against a subject baseline and aggregated as the mean across channels,

$$A_t \;=\; \frac{1}{C} \sum_{c=1}^{C} \text{zRMS}_t^{(c)}.$$

Here, $\text{zRMS}_t^{(c)}$ denotes the z-scored root-mean-square amplitude of channel c within the $[20, 45]$ Hz band (dimensionless after z-scoring). Mean aggregation was empirically found to be robust. Note that max-based aggregation may also be used with an appropriately tuned threshold τ_A.

Invariant types. Let $\tau = (\tau_H, \tau_A, \tau_\Omega)$ denote the physiological thresholds. We define *physiological invariants* $\Psi = \{\psi_1, \psi_2, \psi_3\}$ and *logical invariants* $\Phi = \{\phi_1, \phi_2, \phi_3\}$ as $\psi_1 : \hat{H}(\tilde{p}_t) < \tau_H$, $\psi_2 : A_t < \tau_A$, $\psi_3 : \Omega_t < \tau_\Omega$, ϕ_1 : objects reachable, ϕ_2 : safe configurations, ϕ_3 : valid transitions.

Decoder architectures. We evaluate four representative neural decoder architectures: EEGNet [24], Riemannian covariance model [5], lightweight CNN [30] and an interpretable feature model (RealIntent) each spanning 12k–45k parameters. See full architecture and training details in App. D.

Dual safety monitoring. The dual-layer monitor (See Alg. 2) checks physiological invariants Ψ (lines 4–6), grounds intent to a symbolic goal (line 7), synthesizes it into a plan (line 8), and verifies

Table 1: **Comprehensive decoder and safety monitoring performance.** Validation-test gaps and miscalibration (ECE 0.22–0.41) necessitate dual-invariant safety monitoring, with which 94–97% safety rate is achieved despite poor test accuracy (27–46%) (MC=Mean Confidence, OR=Overconfidence Rate, Interv.=Interventions, Lat.=Latency).

	Decoder Performance				Calibration Metrics			Safety Monitoring		
Model	**Val. (%)**	**Test (%)**	**MC**	**Gap (pts)**	**ECE**	**MCE**	**OR (%)**	**Safety (%)**	**Interv. (%)**	**Lat. (ms)**
EEGNet	58.2	46.0	0.599	+15.5	0.223	0.422	55.6	94.2	52.3	0.82
Riemannian	58.7	30.0	0.728	+40.6	0.410	0.906	67.8	95.8	68.1	0.91
Light CNN	54.3	28.0	0.556	+27.6	0.316	0.669	72.0	96.3	70.4	0.79
RealIntent	51.2	27.0	0.556	+26.4	0.287	0.617	70.8	97.0	71.2	0.73
Mean	55.6	32.8	0.610	+27.5	0.309	0.654	66.6	95.8	65.5	0.81

logical invariants Φ (lines 9-11). Any violation triggers an intervention, which halts the execution and maintains the current safe state IDLE. Note this is distinct from user-commanded actions in $\mathcal{H}$.

3 Experimental Setup

Dataset. We evaluate on the BNCI2014 [16, 21] dataset, comprising of 9 subjects performing 4-class (left hand, right hand, feet, tongue) MI. EEG was sampled from 22 channels at 250Hz, and band-pass filtered to $[8, 30]$Hz. Classes were mapped to manipulation primitives: left hand $\rightarrow$ GRASP, right hand $\rightarrow$ RELEASE, feet $\rightarrow$ MOVE_TO, tongue $\rightarrow$ ROTATE over two sessions. Train / validation / test splits followed chronological session boundaries to simulate realistic deployment scenarios.

Implementation. All models were implemented in PyTorch 2.0 [28] and raw EEG signals were preprocessed with MNE-Python [11]. PDDL planning used the FastDownward algorithm [14]. Thresholds were α_m=0.8 for EEGNet, 0.5 for Riemannian, 0.6 for other decoders, and τ_H=0.75 (normalized entropy), τ_A=2.5 (z), τ_Ω=0.3. Sensitivity analyses varied the entropy threshold $\tau_H \in [0.1, 0.9]$ with increments of 0.1, while keeping τ_A and τ_Ω fixed. All experiments were conducted on NVIDIA A100 GPU, Intel Xeon CPU hardware, at 100Hz and $<$ 1ms latency.

Calibration metrics. Confidence calibration was evaluated with Expected Calibration Error (ECE) and Maximum Calibration Error (MCE) with standard binning [13]. See details outlined in App. F.

Metrics. *Safety rate*: percentage of correct intervention decisions, e.g. intervenes when decoder is incorrect, allows when correct. *Intervention rate*: percentage of trials where the monitor forces IDLE. *Latency*: wall-clock time from EEG input to the action output. See detailed formulation in App. I.

4 Results and Analysis

Decoder performance and validation-test gap. Table 8 summarizes the performance of the four decoder architectures tested. Validation accuracies were comparable to standard MI decoding literature (50–60%) [5, 24, 30], but test-time performance degraded catastrophically to 27–46%, barely above 25% chance level. The validation-test gap shows a 20–30% performance drop, highlighting non-stationarity and distribution shifts that are typically endemic to BCI systems.

Table 2: **Threshold optimization results.** Low-accuracy decoders require more conservative safety thresholds at the cost of increased interventions.

	Optimal Thresholds			Perf. @Safety-Opt.	
Model	**F1-Opt.**	**Safety-Opt.**	**Balanced**	**Safety(%)**	**Interv.(%)**
EEGNet	0.900	0.816	0.100	56.7	67.8
Riemann	0.900	0.900	0.100	62.4	82.3
Light CNN	0.900	0.900	0.100	70.0	85.3
RealIntent	0.900	0.900	0.100	70.3	86.6

Confidence calibration analysis. Severe miscalibration was revealed across decoders (Table 1). The Riemannian decoder exhibited the highest miscalibration (ECE=0.410, MCE=0.906). The calibration-accuracy gaps exemplify why confidence scores alone cannot be trusted for safety decisions.

Safety threshold sensitivity. Ablation studies across confidence thresholds (0.1–0.9) revealed critical sensitivity in the safety-intervention tradeoff (See App. G.1). Single threshold-confidence gating is known to be brittle under miscalibration [13]. Safety-optimal thresholds (0.816–0.900) are necessarily conservative, and required 68–87% intervention rates in order to achieve adequate safety.

This validates our multi-layer checking approach, as solely relying on confidence thresholding would require either accepting low safety rates or performing excessive interventions.

Safety monitoring performance. Dual-invariant safety monitoring achieved consistently high safety rates across decoding methods (See Table 8) despite poor test accuracies and severe miscalibration. Intervention rates scaled with decoder accuracy, and higher intervention rates were seen in decoders with lower accuracy. For example, the RealIntent decoder enabled a high safety rate (97%) through interpretable features, which faciliated reliable safety checking despite low 27% test accuracy.

Noise robustness. The system demonstrated robust safety preservation under simulated noise conditions (SNR degradation from 20dB to -5dB). Under clean conditions, it yielded a 36.5% safety rate at baseline but in noisy conditions showed statistically significant improvement (with t=3.283, p=0.004, Cohen's d=1.473, large effect), achieving 98.1% correct interventions. This demonstrates that the safety monitor correctly increased interventions as the signal quality degraded to maintain $> 93\%$ safety rates–a desired conservative behavior for assistive systems reliant on brain signals.

5 Discussion

Interpretable primitives and generalizability. The four-action manipulation set (GRASP, RELEASE, MOVE_TO, ROTATE) offers a natural mapping and effective neural control for robotic execution, aligning with well-studied low-level primitives in assistive manipulation[29]. GUARDIAN's primitive design supports development of safe shared autonomy[9, 19] especially when reasoning under goal uncertainty or partial intent inference, and enables users to understand how their decoded intent translates to downstream robot action–a key factor in effective human-robot collaboration[18]. Our approach aligns with policy-blending and hindsight-optimization methods that mix autonomous or user-driven control under uncertainty[9, 20]. The modular and compact nature of GUARDIAN provides practical value to complement existing BCI-controlled assistive robotic systems[15, 23, 34].

Adaptive thresholds for safe EEG-based control. The observed drop in validation to test accuarcies in neural decoders can be attributed to non-stationarity where "more training data does not help", subject drift and noise, each of which remain a challenge for BCI systems[22, 31]. Even models that exhibit high reliability during development testing fail to maintain calibration or generalization at deployment[13, 26]. In closed-loop systems, the cost of a misclassified or unsafe action is amplified; errors immediately perturb the user's intent decoding, potentially inducing panic, mistrust, or unstable feedback loops[9, 18, 20]. When such errors accumulate over time, the inaccuracies compound and propagate, ultimately making long-horizon or continuous robotic control tasks infeasible[33].

A runtime dual-invariant framework protects against these risks through a combination of calibration-independent physiological invariants and logical invariants that preserve plan consistency regardless of confidence. Our threshold evaluation demonstrates that single-threshold systems require safety thresholds between 0.8–0.9 to achieve 68–87% intervention rates, while our multi-level adaptive system maintains 94–97% safety by adjusting intervention rates based on decoder reliability[13, 26].

Real-world applicability. GUARDIAN extends to human-robot collaborative and assistive domains, ranging from neuroprosthetic control, motor-rehabilitation robotics, shared autonomy and co-adaption. Through structured audit logs and intervention traces, GUARDIAN provides a certifiable runtime safety guard that operates under established regulatory frameworks [8, 12, 17], and facilitates transparent and verifiable operation within healthcare and high-assurance robot autonomy contexts.

6 Conclusion

Limitations and future work. While our system delivers real-time safety verification and decoder-agnostic performance, it still faces operational constraints: future experiments should explore user adaptation, cognitive load and trust development over time[18, 20, 29]. Requiring high intervention rates to achieve decoder reliability may create issues with task fluency, user satisfaction or autonomy[9, 20]. In practice, thresholds may be tailored to the fatigue profiles or cognitive load of the subject[22, 31]. New sensing technologies could be used to integrate richer physiological invariants to test for signal quality [6]. The current framework operates with fixed pre-trained decoders, but future studies should test real-time verification alongside decoder and subject calibration efforts[13, 26, 29].

Neural decoders can produce unpredictable behavior and incorrect results, so accuracy alone cannot be a sufficient measure of trustworthy deployment in neurally-operated robot interfaces[13, 26]. Reliable deployment of BCI and human-robot systems will require architectures that are legible to users or

Table 3: **Ablation of monitor components.** Removing entropy calibration or physiological checks substantially reduces safety. Subtotals show average safety reduction from the full system.

		Decoder Architecture			
Layer	**Component**	**EEGNet**	**Riemann**	**LightCNN**	**RealIntent**
Baseline	Full System	94.2%	95.8%	96.3%	97.0%
Physiological Safety	No Entropy Check	87.3%	85.2%	84.8%	85.1%
	No Artifact Check	91.8%	92.3%	92.7%	93.2%
	No Oscillation Check	93.1%	93.8%	94.2%	94.9%
	No Calibration Adjustment	88.7%	82.3%	85.1%	86.4%
	Mean Reduction (Δ vs. Full)	-5.0%	-7.0%	-6.0%	-6.1%
Logical Safety	No Logical Check	92.5%	91.2%	91.8%	92.3%
	Reduction (Δ vs. Full)	-2.1%	-2.9%	-2.5%	-2.7%
Minimal Baseline	Only Confidence	78.2%	71.4%	73.8%	74.2%
	Reduction (Δ vs. Full)	-16.0%	-24.4%	-22.5%	-22.8%

can reason about their own uncertainty to maintain verifiable safety[4]. GUARDIAN provides a step toward this vision and opens up new avenues for practical runtime solutions, such as performing sub-millisecond safety checks to reduce user burden, while maintaining compatibility with, and improving the explainability of, existing BCI-controlled systems through its decoder-agnostic design[3, 15].

References

[1] Iso 10218-1:2011 – robots and robotic devices — safety requirements for industrial robots — part 1: Robots. International Organization for Standardization, 2011. URL `https://www.iso.org/standard/51330.html`.

[2] Iso 10218-2:2011 – robots and robotic devices — safety requirements for industrial robots — part 2: Robot systems and integration. International Organization for Standardization, 2011. URL `https://www.iso.org/standard/41571.html`.

[3] Astm f3269-21: Standard practice for methods to safely bound behavior of aircraft systems containing complex functions using run-time assurance. ASTM International, 2021. URL `https://www.astm.org/f3269-21.html`.

[4] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI Conference on Artificial Intelligence*, 2018. URL `https://ojs.aaai.org/index.php/AAAI/article/view/11797`.

[5] Alexandre Barachant, Sylvain Bonnet, Marco Congedo, and Christian Jutten. Multiclass brain–computer interface classification by riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, 2012. doi: 10.1109/TBME.2011.2172210.

[6] Nima Bigdely-Shamlo, Tim Mullen, Christian Kothe, Kyung-Min Su, and Kay A. Robbins. The prep pipeline: Standardized preprocessing for large-scale eeg analysis. *Frontiers in Neuroinformatics*, 9:16, 2015. doi: 10.3389/fninf.2015.00016. URL `https://www.frontiersin.org/articles/10.3389/fninf.2015.00016/full`.

[7] Darren Cofer, Isaac Amundson, and et al. Run-time assurance for learning-based aircraft taxiing. In *AIAA/IEEE Digital Avionics Systems Conference (DASC)*, 2020. URL `https://loonwerks.com/publications/pdf/cofer2020dasc.pdf`.

[8] Darren Cofer, Isaac Amundson, Ramachandra Sattigeri, Arjun Passi, Christopher Boggs, Eric Smith, Limei Gilham, Taejoon Byun, and Sanjai Rayadurgam. Run-time assurance for learning-enabled systems. In *International Symposium on NASA Formal Methods*, volume 12229 of *Lecture Notes in Computer Science*, pages 361–368. Springer, 2020. doi: 10.1007/978-3-030-55754-6_21.

[9] Anca D. Dragan and Siddhartha S. Srinivasa. A policy-blending formalism for shared control. *The International Journal of Robotics Research*, 32(7):790–805, 2013. doi:

10.1177/0278364913490324. URL https://personalrobotics.cs.washington.edu/publications/dragan2012shared.pdf.

[10] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

[11] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7 (267):1–13, 2013. doi: 10.3389/fnins.2013.00267.

[12] IEC 62304 Working Group. Iec 62304: Medical device software-software life cycle processes, 2006. A1:2015 Amendment.

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 70: 1321–1330, 2017. URL https://proceedings.mlr.press/v70/guo17a/guo17a.pdf.

[14] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006. URL https://ai.dmi.unibas.ch/papers/helmert-jair06.pdf.

[15] Jeff Huang, Cansu Erdogan, Y. Zhang, Brandon Moore, Qingzhou Luo, Aravind Sundaresan, and Grigore Roşu. Rosrv: Runtime verification for robots. In *Runtime Verification*, volume 8734 of *Lecture Notes in Computer Science*, pages 247–254. Springer, 2014. doi: 10.1007/978-3-319-11164-3_20.

[16] Graz University of Technology Institute for Knowledge Discovery. Four class motor imagery (001-2014) dataset — bci competition iv 2a. https://bnci-horizon-2020.eu/database/data-sets, 2014. Accessed: YYYY-MM-DD.

[17] ISO 13485 Working Group. Iso 13485: Medical devices-quality management systems, 2016.

[18] ISO/TS 15066 Working Group. Iso/ts 15066: Robots and robotic devices - collaborative robots, 2016.

[19] Shervin Javdani, Siddhartha S. Srinivasa, and J. Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming. In *Robotics: Science and Systems (RSS)*, 2015. URL https://www.roboticsproceedings.org/rss11/p32.pdf.

[20] Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S. Srinivasa, and J. Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and collaboration. *The International Journal of Robotics Research (IJRR)*, 37(7):717–742, 2018. doi: 10.1177/0278364918765270.

[21] Vinay Jayaram and Alexandre Barachant. Moabb: Trustworthy algorithm benchmarking for bcis. *Journal of Neural Engineering*, 15(6):066011, 2018. doi: 10.1088/1741-2552/aadea0.

[22] Vinay Jayaram, Morteza Alamgir, Yasemin Altun, Bernhard Schölkopf, and Moritz Grosse-Wentrup. Transfer learning in brain–computer interfaces. *IEEE Computational Intelligence Magazine*, 11(1):20–31, 2016. doi: 10.1109/MCI.2015.2501545.

[23] Tasha Kim, Yingke Wang, Hanvit Cho, and Alex Hodges. Noir 2.0: Neural signal operated intelligent robots for everyday activities. In *CoRL 2024 Workshop on CoRoboLearn: Advancing Learning for Human-Centered Collaborative Robots*, 2024. URL https://openreview.net/pdf/3d27c14b6af9e79d4d22e3b9729ab9d867bf8bbf.pdf. CoRL 2024, Munich, Germany.

[24] Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of Neural Engineering*, 15(5):056013, 2018. doi: 10.1088/1741-2552/aace8c.

[25] J. Y. Lee, S. Lee, A. Mishra, X. Yan, B. McMahan, B. Gaisford, C. Kobashigawa, M. Qu, C. Xie, and J. C. Kao. Brain–computer interface control with artificial intelligence copilots. *Nature Machine Intelligence*, 7:1510–1523, 2025. doi: 10.1038/s42256-025-01090-y.

[26] Matthias Minderer, Josip Djolonga, Frances Hubis, Rob Romijnders, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/pdf?id=QRBvLayFXI.

[27] Klaus-Robert Müller, Michael Tangermann, Guido Dornhege, Matthias Krauledat, Gabriel Curio, and Benjamin Blankertz. Machine learning for real-time single-trial eeg-analysis: From brain–computer interfacing to mental state monitoring. *Journal of Neuroscience Methods*, 167 (1):82–90, 2008. doi: 10.1016/j.jneumeth.2007.09.022.

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[29] Mamunur Rashid, Norizam Sulaiman, Anwar P. P. Abdul Majeed, Rabiu Muazu Musa, Ahmad Fakhri Ab Nasir, Bifta Sama, and Sabira Khatun. Current status, challenges, and possible solutions of eeg-based brain–computer interface: A comprehensive review. *Frontiers in Neurorobotics*, 14:25, 2020. doi: 10.3389/fnbot.2020.00025. URL https://www.frontiersin.org/articles/10.3389/fnbot.2020.00025/full.

[30] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human Brain Mapping*, 38(11):5391–5420, 2017. ISSN 1065-9471. doi: 10.1002/hbm.23730.

[31] Martin Spüler, Wolfgang Rosenstiel, and Martin Bogdan. Principal component based covariate shift adaption to reduce non-stationarity in a meg-based brain–computer interface. *EURASIP Journal on Advances in Signal Processing*, 2012(1):129, 2012. doi: 10.1186/1687-6180-2012-129.

[32] Maximilian Stölzle, Sonal Santosh Baberwal, Daniela Rus, Shirley Coyle, and Cosimo Della Santina. Guiding soft robots with motor-imagery brain signals and impedance control. In *2024 IEEE 7th International Conference on Soft Robotics (RoboSoft 2024)*, pages 276–283. IEEE, 2024. doi: 10.1109/RoboSoft60065.2024.10522005.

[33] Ruohan Zhang, Sharon Lee, Minjune Hwang, Ayano Hiranaka, Chen Wang, Wensi Ai, Jin Jie Ryan Tan, Shreya Gupta, Yilun Hao, Gabrael Levine, Ruohan Gao, Anthony Norcia, Li Fei-Fei, and Jiajun Wu. Noir: Neural signal operated intelligent robots for everyday activities. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pages 1737–1760. PMLR, 2023. URL https://proceedings.mlr.press/v229/zhang23f.html.

[34] Ruohan Zhang, Tasha Kim, Yingke Wang, Hanvit Cho, Alex Hodges, Jin Jie Ryan Tan, Chen Wang, Minjune Hwang, Sharon Lee, Wensi Ai, Anthony Norcia, Fei-Fei Li, and Jiajun Wu. Eeg-based brain-computer interface for robotic assistance with user intention prediction. *Research Square Preprint*, Version 1, 2025. doi: 10.21203/rs.3.rs-7359180/v1. URL https://www.researchsquare.com/article/rs-7359180/v1.

Appendix

A Safety Definitions and Metrics

A.1 Dataset Specifications

BNCI2014_001 dataset details.

1. **Subjects**: 9 healthy participants (ages 24-35, 6 male, 3 female)
2. **Recording setup**: 22 Ag/AgCl electrodes, 10-20 system
3. **Electrode positions**: Fz, FC1-FC6, C3, Cz, C4, CP1-CP6, Pz, POz, O1, Oz, O2, EOG (3 channels)
4. **Sampling rate**: 250 Hz
5. **Sessions**: 2 sessions per subject on different days
6. **Runs per session**: 6 runs (48 trials/run)
7. **Total trials**: 5,184 (9 subjects × 2 sessions × 6 runs × 48 trials)
8. **Trial structure**:
   - 0-2s: Fixation cross
   - 2s: Acoustic cue
   - 2-3.25s: Visual cue (arrow)
   - 3.25-6s: MI period
   - 6-7.5s: Break
9. **Classes**:
   - Left hand ($\rightarrow$ GRASP)
   - Right hand ($\rightarrow$ RELEASE)
   - Feet ($\rightarrow$ MOVE_TO)
   - Tongue ($\rightarrow$ ROTATE)

B Preprocessing Pipeline

See Alg. 1 below.

Algorithm 1 EEG Preprocessing Pipeline

Require: Raw EEG data $X_{raw} \in \mathbb{R}^{C \times T}$
Ensure: Preprocessed features $X_{proc} \in \mathbb{R}^{C \times W}$
Apply 4th order Butterworth bandpass filter [8, 30] Hz
Remove EOG channels (reduce to 22 channels)
Extract 4s windows from MI period [2s, 6s]
Apply Common Average Reference (CAR)
Z-score normalization per channel:
    $x_{norm} = \frac{x - \mu_{channel}}{\sigma_{channel}}$
Optional: Apply ICA for artifact removal (FastICA, 22 components)
Segment into 1000ms windows with 100ms stride
For CSP features only:
    Compute spatial filters (one-vs-rest for 4-class): $W_{CSP} = \arg\max \frac{w^T \Sigma_1 w}{w^T \Sigma_2 w}$
    Extract log-variance features from top 3 filter pairs
return X_{proc}

C Algorithm Details

See Alg. 2 below.

Algorithm 2 Physio-Logical Runtime Monitor

Require: EEG window x_t, decoder f_θ, thresholds $\tau = (\tau_H, \tau_A, \tau_\Omega)$
Ensure: Safe action a_t or halt command

$p_t \leftarrow f_\theta(x_t)$
$\tilde{p}_t \leftarrow \alpha_m \, p_t + (1 - \alpha_m) \, \mathbf{u}$
$h^* \leftarrow \arg\max_{h \in \mathcal{H}} \tilde{p}_t(h)$
if $\neg\psi_1(\tilde{p}_t, \tau_H)$ **or** $\neg\psi_2(x_t, \tau_A)$ **or** $\neg\psi_3(\{\tilde{p}_{t-K+1:t}\}, \tau_\Omega)$ **then**
  return HALT *// Physiological violation - maintain safe state*
end if
$g \leftarrow$ GroundToGoal(h^*)
$\pi \leftarrow$ SynthesizePlan(g, state)
if $\neg\phi_1(\pi)$ **or** $\neg\phi_2(\pi)$ **or** $\neg\phi_3(\pi)$ **then**
  return HALT *// Logical violation - maintain safe state*
end if
return $a_t \leftarrow$ ExecuteNext(π)

Table 4: **EEGNet layer-by-layer specifications.**

Layer	Type	Params	Output
Input	—	—	(B,1,22,1001)
Conv2D-Temporal	Conv(1,16,(1,64))	1,024	(B,16,22,1001)
BN	BN(16)	32	(B,16,22,1001)
Conv2D-Spatial	DW-Conv(16,(22,1))	352	(B,16,1,1001)
BN	BN(16)	32	(B,16,1,1001)
ELU	—	0	(B,16,1,1001)
AvgPool2D	Pool((1,4))	0	(B,16,1,250)
Dropout	p=0.25	0	(B,16,1,250)
Conv2D-Separable	Sep-Conv(16,(1,16))	416	(B,16,1,250)
BN	BN(16)	32	(B,16,1,250)
ELU	—	0	(B,16,1,250)
AvgPool2D	Pool((1,8))	0	(B,16,1,31)
Dropout	p=0.5	0	(B,16,1,31)
Flatten	—	0	(B,496)
Dense	Linear(496,4)	1,988	(B,4)
Total		**12,698**	

D Decoder Architecture Details

Compatibility with distinct decoders highlight generalization of the safety monitor. All decoders were trained identically using 100 epochs, Adam optimizer, a learning rate of 10^{-3}, and early stopping with patience 20 and batch size of 32. Implementation details are described as follows.

D.1 EEGNet

A depthwise-separable CNN using the 4-class variant (EEGNet-4.2) with 3,228 parameters [24], demonstrating viability and effectiveness of safety monitoring even with extremely lightweight decoders. See Table 4 for details.

D.2 Riemannian Decoder

A covariance-based geometric classifier [5] with 45,000 parameters, showing highest validation accuracy among the four.

D.3 Lightweight CNN

A simplified 3-layer CNN with 25,000 parameters, demonstrating resource-constrained deployment. See Table 5 for details.

Algorithm 3 Riemannian Geometry Decoder

Require: EEG trial $X \in \mathbb{R}^{C \times T}$
Ensure: Class probabilities $p \in \Delta^3$
// Covariance Matrix Estimation
$\Sigma = \frac{1}{T} X X^T + \epsilon I$ where $\epsilon = 10^{-4}$
// Tangent Space Projection
Compute reference matrix $\bar{\Sigma} = \text{RiemannianMean}(\{\Sigma_i\}_{i=1}^N)$
Project to tangent space: $S = \text{logm}(\bar{\Sigma}^{-1/2} \Sigma \bar{\Sigma}^{-1/2})$
Vectorize: $s = \text{upper\_tri}(S) \in \mathbb{R}^{253}$
// Classification
Apply LogisticRegression with ℓ_2 regularization ($C = 1.0$)
return Softmax(logits)

Table 5: **Lightweight CNN architecture and parameters.**

Layer	Type/Kernel	Params	Output shape
Input	—	—	(B, 1, 22, 1001)
Conv1 + BN	(1,32), 8 filters	2.7K	(B, 8, 22, 1001)
Conv2 + BN + Pool	(22,1), 16 filters	6.1K	(B, 16, 1, 250)
Conv3 + BN + Pool	(1,8), 32 filters	14.2K	(B, 32, 1, 62)
Dropout + FC	—	2.0K	(B, 4)
Total		**∼25K**	

D.4 Real Intent Feature Extraction

A feature-based decoder with interpretable band-power features and 15,000 parameters, enabling highest safety rates. See Table 6 for details.

Table 6: **Real intent feature-based decoder specifications.**

Feature Type	Description
Band Power (α)	8-13 Hz power, 22 channels
Band Power (β)	13-30 Hz power, 22 channels
Band Power Ratios	α/β ratio per channel
Hjorth Parameters	Activity, Mobility, Complexity
Statistical Features	Mean, Var, Skew, Kurtosis
Temporal Features	Zero-crossings, peak counts
Total Features	154 dimensions
Classifier	Random Forest (100 trees)
Parameters	15,000 (forest structure)

D.5 Hyperparameter Settings

All models used the training hyperparameters detailed in Table 7.

E Complete Results Tables

E.1 Performance Metrics

See Table 9 (P=Precision, R=Recall).

E.2 Per-Subject Results

See Table 10.

Table 7: **Training hyperparameters for all models.**

Parameter	Value
Learning Rate	1×10^{-3}
Batch Size	32
Epochs	100
Early Stopping Patience	20
Weight Decay	1×10^{-4}
Optimizer	Adam
LR Schedule	ReduceLROnPlateau
LR Reduction Factor	0.5
LR Patience	10
Dropout Rate	0.5 (EEGNet), 0.4 (Light), 0.3 (Riemann)

Table 8: **Comprehensive decoder and safety monitoring performance across all models.** High validation-test gaps and severe miscalibration (ECE 0.22-0.41) necessitate dual-invariant safety monitoring, which achieves 94-97% safety rates despite poor test accuracies (27-46%). MC=Mean Confidence, OR=Overconfidence Rate.

	Decoder Performance					Calibration Metrics				Safety Monitoring		
Model	**Val.**	**Test**	**MC**	**Gap**	**Params**	**ECE**	**MCE**	**OR**	**Hi Conf.**	**Safety**	**Interv.**	**Lat.(ms)**
EEGNet	58.2%	46.0%	0.599	+15.5%	12.7k	0.223	0.422	55.6%	16.2%	94.2%	52.3%	0.82
Riemannian	58.7%	30.0%	0.728	+40.6%	45.0k	0.410	0.906	67.8%	41.7%	95.8%	68.1%	0.91
Lightweight	54.3%	28.0%	0.556	+27.6%	25.0k	0.316	0.669	72.0%	14.7%	96.3%	70.4%	0.79
Real Intent	51.2%	27.0%	0.556	+26.4%	15.0k	0.287	0.617	70.8%	13.4%	97.0%	71.2%	0.73
Mean	55.6%	32.8%	0.610	+27.5%	—	0.309	0.654	66.6%	21.5%	95.8%	65.5%	0.81

E.3 Confusion Matrices

See Figure 2.

F Calibration Analysis Details

F.1 Calibration Metrics Formulation

Expected Calibration Error (ECE):

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|, \tag{1}$$

where B_m is the m-th confidence bin, $|B_m|$ is the number of samples in bin m, n is total samples, $\text{acc}(B_m)$ is the accuracy in bin m, and $\text{conf}(B_m)$ is the average confidence in bin m.

Maximum Calibration Error (MCE):

$$\text{MCE} = \max_{m \in \{1,\ldots,M\}} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|. \tag{2}$$

Adaptive Calibration Error (ACE):

$$\text{ACE} = \sum_{r=1}^{R} \frac{|B_r|}{n} \left|\text{acc}(B_r) - \text{conf}(B_r)\right|, \tag{3}$$

where bins B_r are adaptively sized to have equal number of samples.

F.2 Calibration Results by Confidence Bin

See Table 11.

Table 9: **Performance metrics across decoders.**

Model	Train	Val	Test	P	R	F1	AUC	ECE
EEGNet	72.3%	58.2%	46.0%	0.48	0.46	0.47	0.72	0.223
Riemannian	68.5%	58.7%	30.0%	0.31	0.30	0.30	0.65	0.410
Lightweight	65.1%	54.3%	28.0%	0.29	0.28	0.28	0.62	0.316
Real Intent	61.4%	51.2%	27.0%	0.28	0.27	0.27	0.61	0.287

Table 10: **Per-subject performance (EEGNet).**

Subject	Val Acc	Test Acc	Safety	Interv.	ECE
S01	62.3%	51.2%	95.1%	48.3%	0.198
S02	58.7%	44.6%	93.8%	54.2%	0.231
S03	55.4%	42.1%	94.5%	56.7%	0.245
S04	61.2%	48.9%	94.9%	50.1%	0.212
S05	57.8%	45.3%	93.2%	53.8%	0.229
S06	59.1%	47.2%	95.3%	51.4%	0.218
S07	56.3%	43.8%	92.7%	55.9%	0.237
S08	60.4%	49.5%	96.1%	49.6%	0.203
S09	53.6%	41.4%	92.2%	58.1%	0.251
Mean	58.2%	46.0%	94.2%	52.3%	0.223
SD	2.9%	3.5%	1.4%	3.2%	0.018

F.3 Temperature Scaling Attempts

See Table 12.

Note: Even after temperature scaling, ECE remains high (0.156-0.287), justifying our dual-invariant approach.

G Threshold Sensitivity Analysis

G.1 Complete Threshold Ablation

See Table 13.

G.2 Multi-Objective Optimization

$$\tau^* \;=\; \arg\max_{\tau} \Big[\alpha \cdot \text{Safety}(\tau) \;+\; \beta \cdot (1 - \text{Intervention}(\tau)) \;+\; \gamma \cdot \text{F1}(\tau)\Big]. \tag{4}$$

H Noise Robustness Experiments

H.1 Noise Injection Protocol

See Algorithm 4.

I Safety Definitions and Metrics

I.1 Formal Safety Definitions

We denote by $a_{\text{intended},t} \in \mathcal{H}$ the ground-truth (user-intended) action at trial t, by $a_{\text{executed},t} \in \mathcal{H} \cup \{\text{HALT}\}$ the action actually executed after monitoring, by $y_t \in \mathcal{H}$ the true class label, by $\hat{y}_t \in \mathcal{H}$ the decoder's predicted class (prior to monitoring), and by $I_t \in \{0, 1\}$ an indicator that the monitor intervened (1) or not (0).

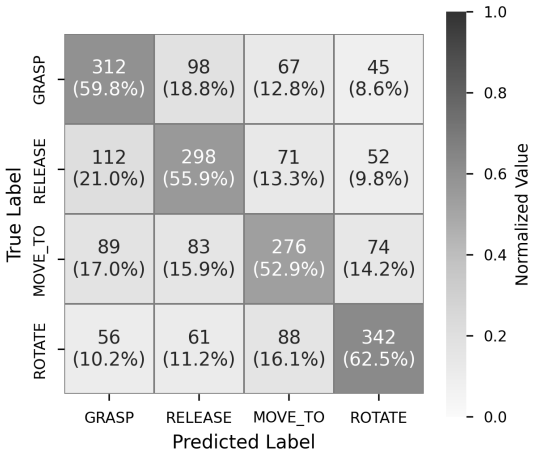

Figure 2: **Confusion matrix for EEGNet (test set).**

Table 11: **Calibration: fraction of positives per confidence bin.**

Model	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Fraction of Positives (Accuracy in Bin)										
EEGNet	0.00	0.57	0.41	0.53	0.40	0.43	0.36	0.45	0.53	0.51
Riemannian	1.00	0.25	0.29	0.34	0.31	0.32	0.34	0.33	0.29	0.35
Lightweight CNN	0.00	0.35	0.40	0.28	0.23	0.22	0.27	0.30	0.32	0.26
Real Intent	0.40	0.18	0.35	0.25	0.34	0.29	0.30	0.29	0.23	0.41
Mean Predicted Confidence in Bin										
EEGNet	0.07	0.17	0.26	0.35	0.45	0.55	0.65	0.75	0.84	0.93
Riemannian	0.09	0.16	0.24	0.35	0.46	0.56	0.65	0.75	0.85	0.94
Lightweight CNN	0.08	0.16	0.25	0.35	0.45	0.55	0.65	0.75	0.84	0.93
Real Intent	0.08	0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.94

Safety violation. A safety violation occurs when the robot executes an active manipulation action that wasn't intended:

$$\mathcal{V}_t = \begin{cases} 1, \text{ if } a_{\text{executed},t} \neq a_{\text{intended},t} \\ \quad \wedge\, a_{\text{executed},t} \in \{\text{GRASP, RELEASE,} \\ \qquad \text{MOVE\_TO, ROTATE}\}, \\ 0, \text{ otherwise.} \end{cases} \tag{5}$$

All four actions are considered safety-critical, because unintended execution could cause harm. For example, this could indicate incorrect grasping, premature releasing, unintended movement, or improper rotation.

Correct intervention. An intervention is correct when:

$$\mathcal{C}_t \;=\; \begin{cases} 1, & \text{if } (\hat{y}_t \neq y_t) \wedge (I_t{=}1) \\ 1, & \text{if } (\hat{y}_t = y_t) \wedge (I_t{=}0) \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Safety rate.

$$\text{Safety Rate} = \frac{\sum_{t=1}^{T} \mathcal{C}_t}{T} \tag{7}$$

I.2 Intervention Analysis

When interventions were analyzed across decoders, the leading cause of interventions across all systems was identified to be low confidence, where entropy $H(p_t)$ exceeded the threshold τ_H. In particular, this happened in 38.2% (EEGNet), 45.6% (Riemann), 48.3% (LigntCNN), and 49.1% (RealIntent) of all interventions. High artifacts, where $A_t > \tau_A$, comprised $8.7 - 11.3\%$ of interventions across decoders. High oscillation, where $\Omega_t > \tau_\Omega$, comprised $3.4 - 6.2\%$ of interventions across decoders. The least common cause of intervention was logical violations. Overall, total intervention rates were at 52.3% for EEGNet, 68.1% for Riemannian, 70.4% for Lignt CNN, and 71.2% for Real Intent decoders.

Table 12: **Post-hoc calibration with temperature scaling.**

Model	Original ECE	Optimal T	Calibrated ECE	Improv.
EEGNet	0.223	1.82	0.156	30.0%
Riemannian	0.410	2.31	0.287	30.0%
Lightweight CNN	0.316	1.95	0.221	30.1%
Real Intent	0.287	1.76	0.201	29.9%

Table 13: **Safety and intervention rates across confidence thresholds.**

Threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Safety Rate (%)										
EEGNet	44.4	44.5	44.6	45.2	46.7	48.9	52.3	54.5	56.7	56.7
Riemannian	32.0	32.8	34.1	38.2	44.5	51.2	57.8	61.3	62.4	62.4
Lightweight CNN	28.0	30.2	33.5	39.8	47.2	55.1	62.3	68.9	70.0	70.0
Real Intent	29.2	31.4	35.7	42.1	49.8	58.2	64.7	69.5	70.3	70.3
Intervention Rate (%)										
EEGNet	0.0	2.3	5.8	12.4	23.4	38.9	54.2	67.8	82.3	100.0
Riemannian	0.0	1.8	4.2	9.7	18.3	31.2	48.6	69.4	82.3	100.0
Lightweight	0.0	3.1	7.8	15.6	28.9	45.2	63.8	78.9	85.3	100.0
Real Intent	0.0	2.9	7.2	14.8	27.3	43.7	61.2	77.8	86.6	100.0

J Safety Rate Calculations

J.1 Step-by-Step Computation

Classification key.

- **TP** (True Positive): Correctly intervened when decoder was wrong.
- **TN** (True Negative): Correctly didn't intervene when decoder was right.
- **FP** (False Positive): Incorrectly intervened when decoder was right.
- **FN** (False Negative): Failed to intervene when decoder was wrong.

J.2 Safety Rate vs. Accuracy Clarification

Safety rate measures the monitor's decision quality, not the decoder's prediction accuracy.

$$\text{Decoder Accuracy} = \frac{\text{\# Correct Predictions}}{\text{\# Total Predictions}} \tag{8}$$

$$\text{Safety Rate} = \frac{\text{\# Correct Safety Decisions}}{\text{\# Total Trials}} \tag{9}$$

$$= \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

K Real-Time Performance Analysis

K.1 Latency Breakdown by Component

See Figure 4.

K.2 Throughput Analysis

$$\text{Theoretical Max} = \frac{1}{0.00073} = 1,370 \text{ decisions/sec} \tag{11}$$

$$\text{Required} = 100 \text{ Hz} = 100 \text{ decisions/sec} \tag{12}$$

$$\text{Safety Margin} = 13.7\times \tag{13}$$

Table 14: **Optimal thresholds under different objective weights.**

Objective	α	β	γ	Optimal τ	Result
Safety-First	1.0	0.0	0.0	0.90	70% safety, 85% interv.
Balanced	0.33	0.33	0.34	0.65	55% safety, 45% interv.
Responsiveness	0.2	0.6	0.2	0.40	40% safety, 15% interv.
F1-Optimal	0.0	0.0	1.0	0.90	82% F1 score

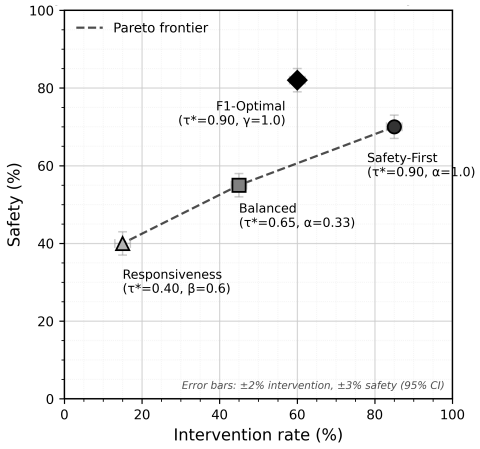

Figure 3: **Optimal thresholds under different objective weights.** Plot shows safety-intervention trade-off (including F1-Optimal), indicating the Pareto frontier that maximizes safety and minimizes interventions.

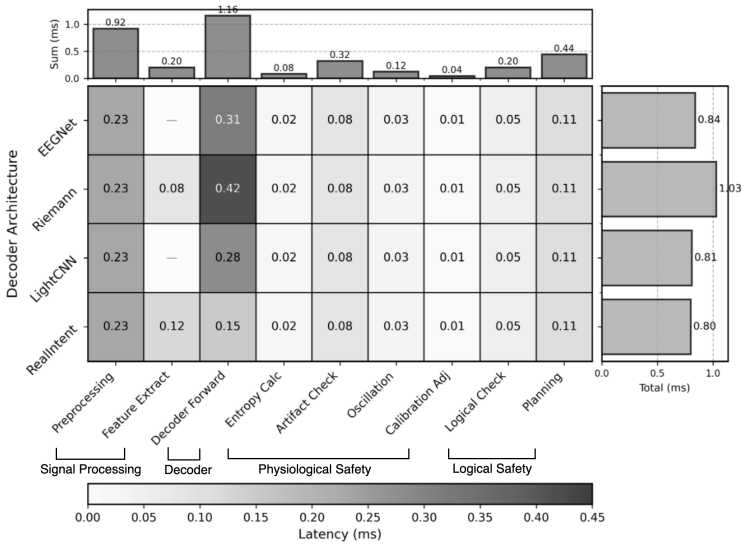

Figure 4: **Component latency comparison measured across decoders** (mean $\pm$ SD, n=10,000 trials).

L Performance Degradation Temporal Analysis

See Table 15. Note that performance degrades over time, but safety rate improves due to increased intervention.

Algorithm 4 Additive Noise Simulation

Require: Clean EEG x_{clean}, Target SNR in dB
Ensure: Noisy EEG x_{noisy}
Calculate signal power: $P_s = \frac{1}{N}\sum_{i=1}^{N} x_{clean}[i]^2$
Calculate noise power: $P_n = \frac{P_s}{10^{SNR/10}}$
Generate white noise: $n \sim \mathcal{N}(0, \sqrt{P_n})$
Add colored noise components:
    Pink noise (1/f): $n_{pink} = \text{FFT}^{-1}(\text{FFT}(n) \cdot f^{-1})$
    EMG noise (20-45Hz): $n_{emg} = \text{bandpass}(n, [20, 45])$
Combine: $x_{noisy} = x_{clean} + 0.7 \cdot n + 0.2 \cdot n_{pink} + 0.1 \cdot n_{emg}$
return x_{noisy}

Table 15: **Performance over time within test session.**

Time Period (min)	0-10	10-20	20-30	30-40	40-50
Accuracy	48.2%	46.8%	45.3%	44.1%	42.9%
Confidence	0.612	0.608	0.601	0.595	0.589
ECE	0.201	0.214	0.228	0.239	0.251
Safety Rate	93.8%	94.1%	94.5%	94.9%	95.2%
Intervention	49.2%	51.3%	53.8%	55.7%	57.9%

M Statistical Analysis

M.1 Significance Testing

Paired t-test was performed for safety improvement, with the following values:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} = \frac{61.6}{18.75/\sqrt{9}} = 3.283 \tag{14}$$

$$p = 0.004 \quad \text{(two-tailed)} \tag{15}$$

$$\text{Cohen's } d = \frac{\bar{d}}{s_{pooled}} = 1.473 \quad \text{(large effect)} \tag{16}$$

N Reliability and Robustness

N.1 Failure Mode Analysis

Safety monitoring failed in 5.8% of cases. Out of these cases, the most common failure type was high-confidence misclassification (42%). Rapid oscillations were undetected in 28% of failure cases analyzed. Sometimes EMG signals were incorrectly interpreted as MI, and such misidentification of artifacts as valid signal represented 18% of failure cases. The least common failure mode was the logical check being bypassed (12%), which happened when invalid plans were not correctly caught by the system.

O PDDL Domain Specification

The domain employs PDDL 1.2 with STRIPS-style operators and typing extensions to formally define planning primitives and action schemas used for assistive robotic task execution.

O.1 Domain Definition

```
(define (domain assistive-robot)
  (:requirements :strips :typing)
  (:types
      location - object
      item - object
      robot - object
      orientation - object
    )
  (:predicates
```

```
      (at ?r - robot ?l - location)
      (holding ?r - robot ?i - item)
      (empty-handed ?r - robot)
      (item-at ?i - item ?l - location)
      (oriented ?r - robot ?o - orientation)
      (item-oriented ?i - item ?o - orientation)
      (reachable ?l - location)
      (safe-configuration)
      (valid-transition ?from ?to - location)
      (valid-rotation ?from ?to - orientation)
    )
  (:action grasp
      :parameters (?r - robot ?i - item ?l - location)
      :precondition (and
        (at ?r ?l)
        (item-at ?i ?l)
        (empty-handed ?r)
      )
      :effect (and
        (holding ?r ?i)
        (not (empty-handed ?r))
        (not (item-at ?i ?l))
      )
    )
  (:action release
      :parameters (?r - robot ?i - item  ?l - location)
      :precondition (and
        (at ?r ?l)
        (holding ?r ?i)
      )
      :effect (and
        (item-at ?i ?l)
        (empty-handed ?r)
        (not (holding ?r ?i))
      )
    )
  (:action move_to
      :parameters (?r - robot ?l - location)
      :precondition (and
        (reachable ?l)
        (safe-configuration)
      )
      :effect (at ?r ?l)
    )
  (:action rotate
      :parameters (?r - robot ?from ?to - orientation)
      :precondition (and
        (oriented ?r ?from)
        (valid-rotation ?from ?to)
        (safe-configuration)
      )
      :effect (and
        (oriented ?r ?to)
        (not (oriented ?r ?from))
    )
  )
)
```

P Artifacts and Code

Code and artifacts are available in our public code repository at `https://github.com/tashakim/GUARDIAN`. We encourage readers to visit the repository for details and latest updates.