ThoughtLink: From Brain to Robot (Tech Report, 1 page)

VC Track supported by Kernel & Dimensional

Problem
Robots can perceive and plan, but when ambiguity occurs humans must intervene.
Today intervention is low-bandwidth (keyboard/dashboard). This does not scale.

ThoughtLink decodes non-invasive brain signals into discrete, high-level robot instructions (not joint control), enabling fast one-to-many supervision.

Data & Command Space
Dataset: ‘robot_control_data/data/*.npz‘ (15s chunks; cue starts at 3s).
Signals: 6-channel EEG at 500 Hz (time-series).
Canonical actions: ‘STOP‘, ‘LEFT‘, ‘RIGHT‘, ‘FORWARD‘, ‘BACKWARD‘.

Approach (Strictly Aligned to Requirement_doc.md)
1) Intent decoder (fast baseline first)
- Windowing: 0.5s windows, 0.1s hop (real-time compatible).
- Features (per channel):
  log-variance + log bandpower (theta 4-8, alpha 8-13, beta 13-30) +
  relative bandpower (4-30). For 6 channels => 42-D features.
- Model (hierarchical):
  Stage1: Binary Logistic Regression (REST vs MOVE).
  Stage2: Softmax Regression (LEFT/RIGHT/FORWARD/BACKWARD on MOVE windows).
- Optional baseline correction: ‘pre_cue‘ delta features to reduce drift.

2) Temporal stability & false-trigger suppression (core hackathon focus)
- EWMA smoothing + hysteresis thresholds + debouncing over time.
- Parameters: ‘p_move_on/off‘, ‘move_on/off_k‘, ‘p_dir‘, ‘dir_k‘, ‘dir_off_k‘, ‘dir_margin‘, ‘stop_on_dir_uncertain‘.
- Auto-tuning (‘examples/tune_stability.py‘) searches parameters under explicit constraints to avoid degenerate "all STOP" solutions.

3) Multi-subject generalization (ELLA-style shared basis)
- Learn a shared linear basis from non-target subjects; fit target with small calibration sessions to adapt quickly.
- Implemented in ‘examples/train_ella_intent.py‘ (basis task = session/subject).

Evaluation (Real-Time & Closed-Loop)
We evaluate in closed-loop replay (50 Hz) and in MuJoCo sim:
- ‘false_rate_global‘: fraction of REST time where predicted action is non-STOP.
- ‘onset_latency‘: time from cue start (3s) to first non-STOP during MOVE.
- ‘switches_per_min‘: temporal stability / flicker proxy.
- ‘inference_ms‘: per-tick inference latency (must support real time).

Internal KPI targets (for demo + scalability reasoning):
- Demo pass: ‘false_rate_global <= 0.05‘.
- 100-robot scale: ‘false_rate_global <= 0.01‘ (otherwise errors accumulate).

Example Result (Illustrative)
Subject ‘37dfbd76‘, session ‘5c71e7df‘, update_hz=50 (ELLA + stabilizer):
- ‘false_rate_global=0.043‘ (meets demo pass), ‘onset_latency_mean=2.50s‘, ‘inference_ms_mean~0.37ms‘ (CPU, numpy).
Tradeoff observed: stricter false-trigger settings reduce ‘move_coverage‘.

Demo Commands (3-5 minutes)
1) Oracle control (proves sim pipeline): ‘examples/intent_policy.py --mode oracle‘
2) Model control (decoder + stabilizer): ‘examples/intent_policy.py --mode model‘
3) Batch report (not cherry-picking): ‘examples/eval_closed_loop.py --top-k 5‘

Key deliverables:
- Design doc: ‘THOUGHTLINK_DESIGN.md‘
- Closed-loop scripts: ‘examples/intent_policy.py‘, ‘examples/eval_closed_loop.py‘