ANCHORING MOTION, REFINING TRAJECTORIES: GENERALIZABLE CONTINUOUS NEURAL DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Brain activity encodes motor information across multiple temporal scales, enabling continuous neural control beyond discrete action recognition. However, existing methods are typically tailored to specific subjects and tasks, with direct neural-to-kinematic mappings that can entangle subject-specific neural variability with task-dependent motion patterns, limiting generalization. To address these limitations, we introduce AMRT, a shared neural decoding framework for heterogeneous electrode configurations and movement paradigms. Motivated by cortical encoding of movement direction and speed, AMRT adopts a two-stage design that anchors local motion through velocity estimation and refines the resulting trajectories over longer horizons, encouraging reuse of local motion tendencies across subjects and tasks. Across three datasets, AMRT generalizes to unseen trials and subjects and demonstrates transfer potential across neural recordings and movement paradigms, highlighting the potential of shared neural representations and anchor-and-refine decoding for generalizable continuous neural control. Our code is available at `https://anonymous.4open.science/r/AMRT`.

1 INTRODUCTION

Brain-computer interfaces (BCIs) translate neural activity into control commands for external devices, with motor decoding forming the basis of applications (Xu et al., 2021). Electrophysiological motor signals can be recorded from scalp electroencephalography (EEG) or intracranial EEG (iEEG), providing complementary access to movement-related neural activity. Most existing motor-decoding studies focus on **discrete decoding**, classifying imagined or executed movements into predefined action categories (Kobler et al., 2022; Ju et al., 2024; Srinivasan et al., 2025). While these paradigms establish the decodability of motor-related EEG patterns, discrete decisions alone cannot support the moment-to-moment adjustments required for real-world motor control. Continuous decoding instead recovers time-varying motor commands, making it essential for fine-grained control of cursors, robotic effectors, and neuroprostheses (Pandarinath & Bensmaia, 2022).

Korik et al. (2018) showed that EEG contain sufficient information to reconstruct continuous hand trajectories, establishing the feasibility of continuous motor decoding. Building on this foundation, recent advances in deep temporal modeling, including recurrent networks, Transformers, and multi-scale attention mechanisms, have substantially improved continuous kinematic decoding, enabling increasingly accurate reconstruction of hand velocity and trajectories from brain electrical signal (Duan et al., 2025; Giri & Pancholi, 2026; Li et al., 2026b; Zhu et al., 2026). Yet robust continuous decoding across subjects and tasks remains challenging. **(1) Neural distribution shift:** neural signals vary substantially across subjects, sessions, and recording systems, while heterogeneous electrode configurations further hinder representation alignment. **(2) Decoding-target design:** existing methods typically regress position or velocity directly (Kwon et al., 2026; Mc Shane et al., 2023; Zhu et al., 2026), making the decoding target sensitive to differences in movement workspace and task structure. Position preserves global trajectory structure but is closely tied to task-specific workspaces and initial states, whereas velocity describes local movement dynamics but accumulates integration error over time. **(3) Generalization scope:** existing studies largely evaluate within fixed subjects, recording configurations, or datasets, leaving generalization across these settings insufficiently understood. These challenges motivate a unified framework for robust and transferable continuous neural decoding.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

These considerations are consistent with the temporal organization of motor-cortical activity. Motor cortex encodes movement direction and speed (Moran & Schwartz, 1999), making velocity a natural representation of local motion. Meanwhile, cortical population dynamics unfold over longer timescales (Churchland et al., 2010; Kaufman et al., 2014). This motivates separating local motion estimation from longer-horizon trajectory reconstruction rather than learning a single direct neural-to-position mapping. We therefore introduce **AMRT** (**A**nchoring **M**otion, **R**efining **T**rajectories), a two-stage framework that uses velocity as an intermediate motion representation. Stage-0 predicts local motion tendency from neural context and integrates it into a coarse trajectory, providing a motion anchor that captures the local movement direction and approximate progression. Stage-1 then combines the coarse trajectory with temporally extended neural context to predict residual trajectory corrections, explicitly compensating for integration drift and recovering long-horizon trajectory structure. To handle neural heterogeneity across subjects and recording configurations, AMRT further maps different electrode layouts into a shared representation through a channel adapter. Together, channel adaptation and anchor-and-refine decoding address heterogeneity in the neural and motion spaces, respectively. We evaluate AMRT from unseen trials and subjects to cross-dataset transfer across heterogeneous EEG/iEEG recordings. Across three datasets, AMRT generalizes under increasingly challenging shifts, while transfer analyses reveal that reusable information can emerge at different levels of the decoding hierarchy. Our contributions are three-fold:

1. **An anchor-and-refine formulation for generalizable continuous decoding.** We separate local motion tendency estimation from long-horizon trajectory reconstruction, using velocity to anchor moment-to-moment motion dynamics and trajectory-space residual refinement to correct accumulated drift. This decomposition provides a structured intermediate representation for transferring local motion while preserving trajectory-level consistency.
2. **A shared neural decoding framework for heterogeneous domains.** We introduce AMRT, which combines a shared neural encoder with lightweight channel adaptation and a common action-chunk decoder, enabling continuous decoding across subjects and datasets with heterogeneous electrode configurations and movement paradigms.
3. **A hierarchical evaluation of continuous-decoding generalization.** We evaluate generalization from unseen trials and subjects to transfer across datasets with different recording configurations and movement paradigms. To our knowledge, this is the first systematic study of cross-dataset transfer for continuous kinematic decoding across heterogeneous EEG/iEEG recording settings.

2 METHODOLOGY

This section presents AMRT for continuous kinematic decoding from heterogeneous neural recordings. After formulating the problem, we describe heterogeneous neural alignment for shared decoding in Section 2.2, followed by motion anchoring and trajectory refinement in Section 2.3.

2.1 PROBLEM FORMULATION

We consider continuous kinematic decoding from complete movement trials. Let the n-th complete trial of subject s be $\mathcal{D}_{s,n} = \{(\mathbf{X}_t, \mathbf{p}_t, \mathbf{v}_t)\}_{t=1}^{T_{s,n}}$, where $T_{s,n}$ is the number of control steps. At step t, $\mathbf{X}_t \in \mathbb{R}^{C_s \times P}$ is the neural segment aligned with the corresponding control step, where C_s is the subject-specific number of electrodes and P is the number of neural samples per step. The position and velocity are $\mathbf{p}_t, \mathbf{v}_t \in \mathbb{R}^D$, where $D \in \{2, 3\}$ depends on the movement space of the dataset. Velocity labels are derived from the recorded position trajectories during preprocessing. To reduce trial-specific dependence on the initial hand position, we decode trajectories in relative coordinates, using $\mathbf{p}_t^{\mathrm{gt}} = \mathbf{p}_t - \mathbf{p}_1$ as the trajectory-space target.

To provide sufficient temporal context while retaining online compatibility, we perform decoding over consecutive output blocks rather than individual control steps. For a block of length L starting at step a, the model observes an H-step neural context ending at the block endpoint:

$$\mathcal{X}_a^{(H)} = \mathbf{X}_{a+L-H:a+L-1}, \qquad H \geq L, \tag{1}$$

where successive blocks advance by a stride S. Boundary contexts are edge-padded and excluded from the training objective using a validity mask. This formulation is block-causal: no neural activity after step $a + L - 1$ is used, while observations within the current block are jointly available for prediction. The same temporal protocol is used throughout the framework.

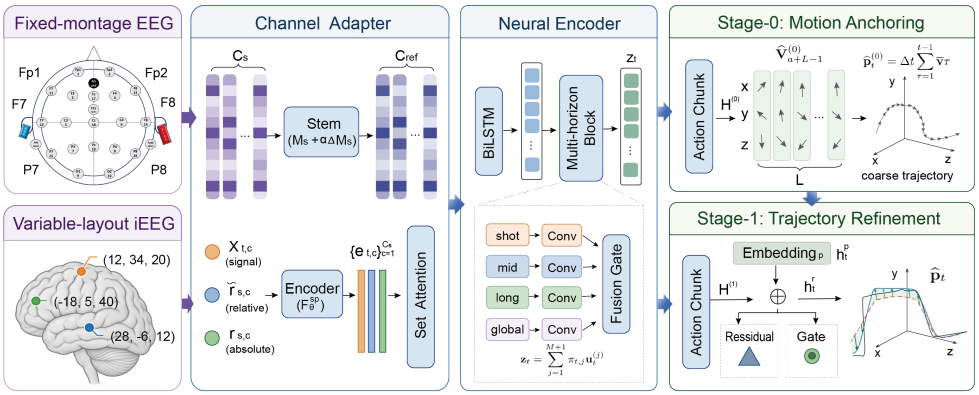

Figure 1: Overview of the Anchoring Motion Refining Trajectory (AMRT) framework.

2.2 Heterogeneous Neural Alignment for Shared Decoding

This subsection introduces a shared decoding architecture for heterogeneous neural recordings. It combines modality-aware channel adaptation, multi-horizon neural encoding, and query-based action-chunk decoding to map EEG/iEEG signals into motion predictions.

2.2.1 Modality-aware channel adapter.

Neural recordings exhibit distinct forms of spatial heterogeneity. EEG typically follows standardized montages with stable channel identities, whereas iEEG uses subject-specific electrode sets without consistent index correspondence across patients. We therefore adopt two complementary channel adapters: an **exact channel adapter** for fixed-montage recordings and a **geometry-aware channel adapter** for variable electrode layouts. In modality-specific experiments, we use the corresponding adapter for EEG and iEEG; for cross-dataset transfer, the geometry-aware adapter provides a shared interface that decouples downstream decoding from dataset-specific electrode indexing.

Exact projection for fixed montages. Standardized EEG montages provide reproducible electrode layouts and stable channel identities, enabling reliable channel correspondence across subjects (Hh, 1958; Tucker, 1993). We anonicalize the labels and construct a deterministic matrix $\mathbf{M}_s \in \mathbb{R}^{C_{\text{ref}} \times C_s}$ for montage, where $\mathcal{C}_s$ is the set of valid channels. Each observed channel is mapped to its matching reference-montage entry, with unmatched entries set to zero. We optionally augment this nominal correspondence with a small learnable residual:

$$\widetilde{\mathbf{X}}_t = (\mathbf{M}_s + \alpha\, \Delta\mathbf{M}_s)\, \mathbf{X}_t, \tag{2}$$

where $\Delta\mathbf{M}_s$ is initialized to zero and α controls its scale. The residual is shared across source subjects to correct errors beyond the fixed canonical mapping. It is frozen for zero-shot transfer and adapted only with labeled target data, enabling lightweight cross-subject adaptation.

Geometry-aware encoding for variable electrode sets. Unlike EEG, iEEG varies across subjects in channel count and cortical coverage (Holdgraf et al., 2019), making channel correspondence unreliable. We therefore treat valid contacts as a coordinate annotated set, where a shared encoder embeds each electrode segment and coordinate encoders incorporate centered local geometry and global implantation location. Let $\mathbf{r}_{s,c} \in \mathbb{R}^3$ denote the coordinate of electrode c for subject s. To obtain a subject-normalized representation of the local electrode layout $\widetilde{\mathbf{r}}_{s,c}$, we center the coordinates and normalize their scale by the RMS spatial radius (Gower, 1975; Qi et al., 2017). Specifically, for electrode coordinate $\mathbf{r}_{s,c}$, we use $\widetilde{\mathbf{r}}_{s,c} = (\mathbf{r}_{s,c} - \boldsymbol{\mu}_s)/(\rho_s + \epsilon)$, where $\boldsymbol{\mu}_s$ and ρ_s denote the mean coordinate and RMS radius of the valid electrodes, respectively. For each electrode, we jointly encode its neural activity, subject-normalized relative geometry, and absolute spatial location:

$$\mathbf{e}_{t,c} = F_\theta^{\text{sp}}\left(\mathbf{X}_{t,c}, \widetilde{\mathbf{r}}_{s,c}, \mathbf{r}_{s,c}\right). \tag{3}$$

The normalized coordinates capture the relative electrode configuration, while the original coordinates retain the absolute recording location removed by canonicalization. These geometry-aware electrode embeddings form an unordered set. We aggregate them using a small set of learned spatial

queries with masked multi-head attention (Lee et al., 2019), followed by a fusion layer to obtain the shared neural representation $\mathbf{h}_t$. Invalid or padded channels are excluded by the attention mask. This set-based aggregation is permutation invariant and naturally accommodates different electrode counts and layouts without assuming cross-subject channel identity.

2.2.2 MULTI-HORIZON NEURAL ENCODER

Movement-related neural activity spans multiple temporal scales. We therefore encode the channel-aligned context using complementary local and global temporal operators. After an initial temporal stem and a bidirectional recurrent layer that captures sustained trends, the sequence is processed by stacked multi-horizon blocks. Given the same input sequence, each block applies J temporal convolution branches with different receptive fields and one global attention branch, producing branch features $\mathbf{u}_t^{(j)} \in \mathbb{R}^d$ at control step t. We adaptively fuse these features with a learned gating network:

$$\boldsymbol{\pi}_t = \mathrm{softmax}\left(\phi_\pi\left([\mathbf{u}_t^{(1)}; \ldots; \mathbf{u}_t^{(J+1)}]\right)\right), \qquad \mathbf{z}_t = \sum_{j=1}^{J+1} \pi_{t,j}\, \mathbf{u}_t^{(j)}, \tag{4}$$

where ϕ_π outputs $J+1$ fusion weights, $\mathbf{u}_t^{(1)}, \ldots, \mathbf{u}_t^{(J)}$ are features from the temporal branches, and $\mathbf{u}_t^{(J+1)}$ is the global-attention feature. Since the fusion weights are conditioned on the current branch responses, the model can emphasize different temporal scales at different control steps. Stacking the fused features over time yields the memory $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_H] \in \mathbb{R}^{H \times d}$, which combines local neural dynamics with longer-range movement context. Multi-scale temporal modeling has proven effective for EEG decoding (Altaheri et al., 2023; Zhu et al., 2026), while convolution and attention provide complementary local and global temporal biases (Song et al., 2022). All operations are restricted to the H-step context defined in Sec. 2.1, preserving the block-causal protocol.

2.2.3 QUERY-BASED ACTION-CHUNK DECODER

Rather than decoding each control step independently, we jointly predict an L-step motion chunk. Following action chunking (Zhao et al., 2023), each relative position is assigned a learned query. The queries interact through self-attention and attend to the encoded neural memory, enabling joint prediction across the chunk. For stage $s \in \{0, 1\}$, we define stage-specific temporal queries $\mathbf{Q}^{(s)} = [\mathbf{q}_1^{(s)}, \ldots, \mathbf{q}_L^{(s)}]$ and compute $\mathbf{H}^{(s)} = \mathrm{ChunkDec}^{(s)}\left(\mathbf{Q}^{(s)}, \mathbf{Z}\right)$, where $\mathbf{H}^{(s)} = [\mathbf{h}_1^{(s)}, \ldots, \mathbf{h}_L^{(s)}]$ is the chunk representation used by the corresponding stage. $\mathrm{ChunkDec}^{(s)}$ uses self-attention over queries and cross-attention to $\mathbf{Z}$. Each query corresponds to a fixed temporal position in the output chunk, so no matching is required (Carion et al., 2020).

2.3 MOTION ANCHORING AND TRAJECTORY REFINEMENT

Given the decoded chunk representation, we reconstruct continuous trajectories in two stages. Stage-0 represents local motion with velocity and integrates it into a coarse trajectory, while Stage-1 refines long-horizon trajectory structure through residual correction.

2.3.1 STAGE-0: MOTION ANCHORING

Stage-0 captures local movement dynamics by predicting an L-step velocity chunk from the decoded representation: $\widehat{\mathbf{V}}^{(0)}_{a:a+L-1} = \phi_0\big(\mathbf{H}^{(0)}\big)$, where ϕ_0 is the velocity head. We refer to $\widehat{\mathbf{V}}^{(0)}$ as **motion anchors**, as they provide locally coherent estimates of movement direction and speed for subsequent trajectory reconstruction. Stage-0 is optimized with a Huber loss (Huber, 1992) over valid velocity targets, excluding padded or invalid elements with the corresponding mask. After training, Stage-0 remains frozen during trajectory refinement.

As movement unfolds, Stage-0 produces overlapping velocity anchors that are integrated into a coarse trajectory. Predictions are first denormalized using training-set statistics, then combined by Hann-weighted overlap-add and integrated, yielding the velocity estimate $\overline{\mathbf{v}}_t$. The coarse trajectory is then obtained by cumulatively integrating these estimates from the beginning of the trial:

$$\widehat{\mathbf{p}}_1^{(0)} = \mathbf{0}, \qquad \widehat{\mathbf{p}}_t^{(0)} = \Delta t \sum_{\tau=1}^{t-1} \overline{\mathbf{v}}\tau, \quad t > 1, \tag{5}$$

where $\Delta t = 1/f_c$ and f_c is the control sampling rate. Thus, $\widehat{\mathbf{p}}_t^{(0)}$ summarizes the motion anchors accumulated up to the current step and provides the trajectory state that Stage-1 subsequently refines.

Errors in the local velocity estimates naturally accumulate through this integration, making long-horizon drift explicit in trajectory space.

2.3.2 STAGE-1: TRAJECTORY RESIDUAL REFINEMENT

Stage-0 reconstructs a coarse trajectory from local motion anchors, but velocity errors accumulate into long-horizon drift. Stage-1 therefore shifts the decoding objective from local motion to trajectory-space correction. Rather than predicting trajectories from scratch, Stage-1 learns how the coarse trajectory should be corrected. A separate refinement decoder combines the frozen neural memory with the current coarse trajectory state. For each control step, the refinement module conditions on both the coarse trajectory state and the corresponding neural representation. Specifically, we first embed the coarse position and fuse it with the refinement feature:

$$\mathbf{h}_t^r = \mathrm{Fuse}\left(\mathbf{h}_t^{(1)}, \mathbf{h}_t^p\right), \qquad \mathbf{h}_t^p = \mathrm{Emb}_p\left(\widehat{\mathbf{p}}_t^{(0)}\right) \tag{6}$$

where $\widehat{\mathbf{p}}_t^{(0)}$ is the coarse position, Emb_p maps the position into the feature space, and $\mathbf{h}_t^{(1)}$ provides the neural evidence used for trajectory refinement. From the fused representation, we predict both a candidate residual correction and a gate that modulates how strongly this correction is applied:

$$\Delta\mathbf{r}_t = f_\Delta\left(\mathbf{h}_t^r\right), \qquad \mathbf{g}_t = \sigma\left(f_g\left(\mathbf{h}_t^r\right)\right). \tag{7}$$

Here, f_Δ predicts the direction and magnitude of the trajectory correction, while f_g determines how strongly the correction is applied at the current control step. We refer to the resulting residual as a **gated correction increment**. Since adjacent decoding blocks overlap in time, predictions associated with the same control step are first aggregated by overlap-add:

$$\overline{\Delta\mathbf{r}}_t = \mathrm{OA}\left(\mathbf{g}_t \odot \Delta\mathbf{r}_t\right), \qquad \mathbf{r}_t = \sum_{\tau=1}^{t-1} \overline{\Delta\mathbf{r}}_\tau, \qquad \widehat{\mathbf{p}}_t = \widehat{\mathbf{p}}_t^{(0)} + \mathbf{r}_t, \tag{8}$$

where $\mathrm{OA}(\cdot)$ denotes overlap-add aggregation across blocks, and $\mathbf{r}_t$ is the accumulated trajectory correction up to control step t. Predicting correction increments rather than absolute positions preserves the coarse motion structure established by Stage-0, while allowing long-horizon neural evidence to progressively compensate for accumulated trajectory errors. The gate allows the model to modulate correction magnitude. Accordingly, Stage-1 is optimized at the trial level with trajectory reconstruction, gap, smoothness, and gating-utility losses. This two-stage design directly realizes our **anchor-and-refine** principle: Stage-0 establishes transferable local motion anchors, while Stage-1 correct accumulated drift and restore trajectory-level consistency.

3 EXPERIMENT

We organize our experiments into a hierarchy of increasingly challenging generalization settings, from unseen trials to unseen subjects and cross-dataset transfer. Section 3.1 describes the evaluation setup. Section 3.2 evaluates within-dataset generalization to unseen trials and subjects, while Section 3.3 extends the evaluation across heterogeneous datasets and recording settings, and movement paradigms. Section 3.4 further analyzes the motion-to-trajectory design.

3.1 EXPERIMENTAL SETUP

Datasets. We evaluate our framework on three public continuous motor decoding datasets spanning different movement paradigms and neural recording modalities. WAY-EEG-GAL (Luciw et al., 2014) contains EEG and 3D kinematics during grasp-and-lift movements under varying object conditions, we use wrist trajectory as the decoding target. U-Limb (Averta et al., 2021) provides synchronized EEG and motion capture during diverse upper-limb movements, from which we decode the hand-dorsum trajectory. Joystick-Track, included in the ECoG library of Miller (2019) and originally introduced by Schalk et al. (2007), contains intracranial recordings during 2D joystick tracking. Together, these datasets span diverse upper-limb movements across EEG and iEEG recordings.

Preprocessing. We convert all datasets into temporally aligned neural–motor sequences. Neural signals are resampled to 500 Hz, notch-filtered at 50 or 60 Hz, common-average referenced, and channel-wise z-score normalized within each chunk. We follow the original trajectory construction and coordinate processing for each dataset. Velocity is causally estimated from position using a third-order trailing Savitzky-Golay derivative (Savitzky & Golay, 1964), while the original trajectory

is retained as the position target. For each dataset and kinematic axis, velocity is standardized using training-set statistics: $\widetilde{\mathbf{v}}_t = (\mathbf{v}_t - \boldsymbol{\mu}_d) \oslash \boldsymbol{\sigma}_d$, where $\boldsymbol{\mu}_d$ and $\boldsymbol{\sigma}_d$ are the per-axis mean and standard deviation computed from valid training labels.

Baselines and implementation. We compare our method with three recent continuous EEG decoding approaches: M3T-Attention (Zhu et al., 2026), Copilot-Assisted (Li et al., 2026b), and the Transformer-Based Regressor (TBR) (Dangi & Meena, 2026). All methods were originally evaluated on WAY-EEG-GAL. We reproduce each baseline following its reported preprocessing pipeline, model architecture, and training configuration as closely as possible, while evaluating all methods using the same data partitions and evaluation metric as our framework. We report trial-level trajectory Pearson correlation coefficient (`traj_PCC`), computed between predicted and ground-truth trajectories over time and averaged across test trials. Unless specified otherwise, results are reported as mean±std across runs. Statistical significance is assessed using a two-sided paired t-test against the best-performing baseline, with *, **, and *** denoting $p < 0.05$, $p < 0.01$, and $p < 0.001$.

3.2 Generalization Across Trials and Subjects

We first assess AMRT at two levels of generalization, from unseen trials of observed subjects to unseen subjects with potentially shifted neural distributions and electrode configurations. We begin with cross-trial decoding, then evaluate cross-subject generalization under leave-one-subject-out (LOSO) setting and further examine the benefit of few-shot calibration.

3.2.1 Cross Trial Decoding

We begin with the first level of our generalization hierarchy, evaluating unseen trials under the `ctri-mix` protocol, which pools trials across subjects to mitigate the instability of within-subject evaluation when per-subject trials are scarce. Complete trials are pooled across subjects and split into training, validation, and test sets at an 8:1:1 ratio, with splitting performed at the trial level to avoid window leakage. Results are aggregated over 10 random seeds. For fixed-channel baselines, we introduce `JT-common`, truncating Joystick-Track recordings to a common channel count across subjects. We additionally report within-subject `ctri` results on WAY-EEG-GAL for comparison with prior work (Jain & Kumar, 2023a; 2025; 2023b; Li et al., 2026b), and on Joystick-Track to evaluate decoding with all available channels retained.

Across the cross-trial settings, AMRT consistently achieves the highest PCC, with its advantage becoming more pronounced as the decoding setting becomes more heterogeneous (Table 1). On WAY-EEG-GAL, where strong baselines already reach high PCC, AMRT provides a modest but consistent improvement under both `ctri-mix` and `ctri`. The gap widens substantially on U-Limb, where diverse upper-limb movements increase behavioral variability, with AMRT improving PCC from 0.314 to 0.530 over the best baseline. This advantage also extends to Joystick-Track, indicating that AMRT remains effective under a distinct iEEG recording setting. We further examine the contribution of individual model components under the `ctri-mix` protocol in Appendix E.3.

3.2.2 Generalization to Unseen Subjects

We then evaluate cross-subject generalization under a subject-disjoint setting, where the test subject is not observed during training and may exhibit substantially different neural signal distributions.

Table 1: Comparison of baselines and AMRT. Results are trajectory PCC. For Joystick-Track under `ctri-mix`, JT-common uses the channel subset shared across subjects, enabling matched comparison with fixed-channel baselines, while Joystick-Track uses all available electrodes.

	`ctri-mix`				`ctri`	
	WAY-EEG-GAL	JT-common	Joystick-Track	U-Limb	WAY-EEG-GAL	Joystick-Track
M3T-Attention	0.813±0.036	0.464±0.097	–	0.256±0.041	0.869±0.058	0.575±0.128
Copilot-Assisted	0.674±0.022	0.378±0.084	–	0.314±0.025	0.785±0.083	0.465±0.303
TBR	0.248±0.053	–	0.012±0.051	0.033±0.028	0.464±0.096	0.037±0.035
AMRT	**0.837±0.045**	**0.707±0.037*****	**0.778±0.076*****	**0.530±0.029*****	**0.910±0.028***	**0.829±0.106****

Table 2: Comparison of baselines, the full AMRT, and the channel-adapter ablation (AMRT w/o CA). Best results are shown in bold. Significance markers are computed against the best baseline.

	WAY-EEG-GAL	U-Limb	Joystick-Track	JT-common
M3T-Attention	0.785±0.128	0.160±0.061	–	0.334±0.110
Copilot-Assisted	0.539±0.159	0.085±0.078	–	0.094±0.079
TBR	0.263±0.100	0.033±0.028	0.013±0.041	–
AMRT w/o CA	0.789±0.103	0.360±0.105	0.680±0.117	–
AMRT	**0.841±0.055**	**0.430±0.073*****	**0.713±0.120*****	**0.687±0.125*****

We first assess AMRT under LOSO evaluation and then examine whether limited target-subject calibration can provide further gains.

Cross-subject Decoding under LOSO. We adopt a leave-one-subject-out protocol (`csub`). For each fold, one subject is held out for testing, a disjoint subject is used for validation and checkpoint selection, and all remaining subjects are used for training. The procedure is repeated until every subject has served as the test subject once. Results are macro-averaged across held-out subjects, with no test-subject data used for training or model selection. As shown in Table 2, AMRT achieves the highest PCC across all three datasets. On WAY-EEG-GAL, strong baselines remain competitive on average but vary considerably across held-out subjects, indicating less consistent transfer. AMRT achieves stronger and more stable performance across all datasets, including Joystick-Track, where subject-dependent electrode configurations prevent fixed-channel baselines from being directly applied. We further ablate channel adapter. AMRT w/o CA still outperforms the baselines, but consistently underperforms the full model in cross-subject decoding. This suggests that the shared decoder captures much of the transferable motion structure, while channel alignment provides a complementary improvement by mitigating subject-specific differences in electrode configurations.

Few-shot calibration. We next investigate whether limited target-subject supervision can further improve the LOSO model. For each held-out subject, we use $K \in \{1, 3, 5, 10\}$ labeled trials for calibration, where one shot corresponds to one complete trial. Calibration sets are nested across shot counts, and all models are evaluated on the same blind test set. We compare full fine-tuning **Full-FT**, which updates Stage-0 and subsequently Stage-1, with **Stage-1-only adaptation**, which freezes the channel adapter and Stage-0 and updates only the trajectory-refinement module. Both strategies use a fixed adaptation schedule without target-subject checkpoint selection. Their comparison indicates whether limited calibration mainly benefits motion estimation or trajectory refinement.

As shown in Figure 2(a) and Table 7, the effect depends on both the zero-shot baseline and calibration size. On WAY-EEG-GAL, where LOSO performance is already strong, calibration yields only marginal gains. On U-Limb, performance improves progressively with more target trials, with Full-FT becoming increasingly beneficial as supervision grows. On Joystick-Track, Stage-1-only adaptation consistently improves over the LOSO model, whereas Full-FT is less stable in the low-shot regime. Overall, Stage-1-only adaptation is more reliable when calibration data are scarce, suggesting that limited supervision is often better used for trajectory refinement than for relearning the neural-to-motion mapping. As more target data become available, updating Stage-0 becomes increasingly beneficial in settings with larger subject-specific shifts.

3.3 Transfer Across Heterogeneous Datasets

We next investigate whether AMRT can reuse learned representations across datasets with different recording modalities, electrode configurations, and movement paradigms. We focus primarily on single-source transfer to expose source-target compatibility, while dual-source transfer and heterogeneous-data scaling analyses are reported in Appendix E.2.

For each target dataset, we construct repeated subject-disjoint training/validation/test splits with an approximate 8:1:1 ratio, adjusted to the available cohort size, and repeat the procedure over six random seeds. Within each split, we subsample $f \in \{0.10, 0.25, 0.50, 1.00\}$ of the target training trials while keeping the validation and test sets fixed. Velocity denormalization uses control statistics from the full target training split in all settings. At each fraction, we compare training from scratch

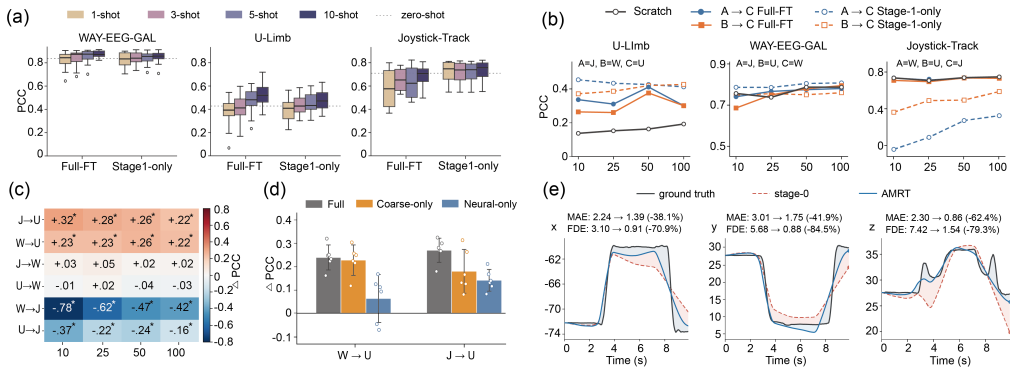

Figure 2: **Summary of the main experimental results.** **(a)** Few-shot adaptation across calibration sizes. **(b)** Cross-dataset transfer across target-data fractions. **(c)** Stage-1-only gains over matched scratch training. **(d)** Stage-1 input ablation with Full, Coarse-only, and Neural-only variants. **(e)** Representative trajectory reconstructions from Vel-Int and AMRT.

with transfer. The **scratch** model is trained only on the selected target subset, whereas the **transfer** model is first pretrained on the source dataset and then adapted to the same target subset. We use both Full-FT and Stage-1-only adaptation, following the cross-subject setting. U-Limb, WAY-EEG-GAL, and Joystick-Track are abbreviated as U, W, and J.

Cross-dataset pretraining selectively improves target-data efficiency. The strongest transfer gains are observed on U-Limb. With Stage-1-only adaptation, both J→U and W→U outperform scratch training at every target-data fraction and across all split-fraction comparisons. With only 10% of the U-Limb training data, they achieve PCC of 0.454 and 0.371, exceeding 0.192 obtained by scratch training with the full target data. This suggests that cross-dataset pretraining captures reusable structure that is difficult to learn from limited target data. However, these gains are not uniform across source-target pairs, motivating a closer examination of what determines transferability.

AMRT transfers both motion tendencies and reusable neural representations. Successful transfer does not necessarily require the source-domain motion prediction to remain directly valid. For W→U, the frozen Stage-0 decoder already achieves a PCC of 0.310 without target-domain parameter updates, indicating that the learned motion tendency remains useful in the target domain. In contrast, J→U has near-zero calibrated zero-update Stage-0 PCC, yet reaches an average PCC of 0.432 by adapting only Stage-1. Thus, even when the source motion output is poorly calibrated to the target, the frozen source stack can retain information that remains reusable after refinement.

To identify which information Stage-1 reuses during transfer, we ablate its two refinement inputs in Eq. 6: the neural feature $\mathbf{h}_t^{(1)}$, derived from the frozen neural representation, and the coarse-trajectory embedding $\mathbf{h}_t^p$. **Coarse-only** removes $\mathbf{h}_t^{(1)}$, whereas **Neural-only** removes $\mathbf{h}_t^p$. As shown in Figure 2(d), W→U is largely coarse-dominant: coarse-only nearly matches the full model ($+0.23$ vs. $+0.24$), whereas neural-only yields only a small and less consistent gain ($+0.06$). In contrast, both inputs contribute to J→U, with coarse-only and neural-only each improving over scratch ($+0.18$ and $+0.14$) and the full model achieving the largest gain ($+0.27$). These results suggest that W→U mainly reuses source-derived motion information, while J→U benefits from complementary coarse motion and target-readable neural representations. Together, these results show that AMRT transfers both motion information and reusable neural representations across datasets.

Transfer is strongly directional. In our experiments, Joystick-Track provides the strongest initialization for U-Limb despite differences in recording modality and control space, showing that shared modality is not necessary for effective transfer. In contrast, neither EEG source transfers effectively to Joystick-Track even with Stage-1-only adaptation, suggesting that EEG-to-iEEG transfer is particularly challenging in our current setting. The channel adapter enables a common decoder to operate across heterogeneous electrode configurations by removing the requirement for one-to-one channel correspondence. In particular, the successful J→U transfer indicates that motion-related

Table 3: Trajectory PCC under different motion-to-trajectory reconstruction strategies. The best results are highlighted in bold.

Setting	Dataset	DIRECT-POS	POS-VEL	VEL-INT	AMRT
ctri	W	0.934±0.022	**0.935±0.024**	0.886±0.040	0.910±0.028
	J	0.849±0.065	**0.860±0.050**	0.823±0.114	0.829±0.106
ctri-mix	W	0.831±0.020	0.831±0.020	0.785±0.038	**0.837±0.045**
	U	0.259±0.053	0.187±0.061	0.471±0.020	**0.530±0.029*****
	J	0.702±0.018	0.694±0.020	0.770±0.076	**0.778±0.076**
csub	W	0.825±0.066	0.836±0.062	0.800±0.076	**0.841±0.055**
	U	0.268±0.120	0.282±0.123	0.369±0.107	**0.430±0.073***
	J	0.361±0.060	0.357±0.052	0.689±0.132	**0.713±0.120**

representations learned from iEEG can remain useful after alignment to EEG, while the adapter primarily provides the structural alignment needed to expose this transferable information.

3.4 MOTION-TO-TRAJECTORY DECODING

Having evaluated AMRT under increasingly challenging generalization settings, from unseen trials to unseen subjects and datasets, we next examine which reconstruction formulation gives rise to these gains. Under the same model architecture, training configuration, and data splits, we compare DIRECT-POS, which directly regresses trajectories; VEL-INT, which predicts velocity and reconstructs trajectories by temporal integration; POS-VEL, which jointly predicts position and velocity while using the predicted position for reconstruction; and our full formulation. The three variants share the same Stage-0 architecture and differ only in their prediction objective, while the comparison between VEL-INT and our full model isolates the contribution of trajectory refinement. We evaluate all variants under the same `ctri-mix`, `ctri`, and `csub` protocols used above.

Table 3 reveals a clear interaction between reconstruction strategy and generalization regime. Under subject-specific `ctri`, direct position regression remains competitive, showing that absolute trajectories can be learned effectively when the subject-specific movement space is fixed. AMRT is not designed to dominate this setting; its advantage emerges as the decoding distribution shifts across subjects and recording conditions. Under `ctri-mix` and particularly `csub`, AMRT consistently achieves stronger trajectory reconstruction, with clearer gains on U-Limb and Joystick-Track. The weaker performance of POS-VEL further shows that auxiliary velocity supervision alone does not explain these gains. Using velocity as an intermediate representation emphasizes local motion trends that are less tied to workspace, providing a more transferable basis for trajectory reconstruction.

Comparing VEL-INT with AMRT isolates the effect of trajectory refinement. While velocity captures local motion well, direct integration accumulates errors over time. Stage-1 mitigates this drift by refining the trajectory over longer horizons. As illustrated in Figure 2(e), VEL-INT follows the overall movement but gradually deviates from the ground truth, whereas AMRT corrects this drift while preserving the motion pattern. Together with Table 3, these results support the two-stage design: velocity anchors local motion, while trajectory refinement improves long-horizon consistency.

4 CONCLUSION

We introduce AMRT, a channel-adaptive framework for generalizable continuous neural decoding. It combines a neuroscience-inspired two-stage formulation for motion anchoring and trajectory refinement with a shared decoding framework for heterogeneous neural recordings. Across a hierarchical evaluation from unseen trials and subjects to cross-dataset transfer, AMRT learns reusable neural representations and maintains strong decoding performance under increasingly challenging domain shifts. These results highlight the potential of combining transferable local motion with long-horizon trajectory modeling for generalizable continuous neural control. Future work can extend this framework to motor behaviors and larger-scale heterogeneous neural data.

5 AI USE STATEMENT

Generative AI tools were used to assist with writing, literature discovery, and research implementation. For writing, AI was used to revise and polish English expressions, improve clarity and conciseness, and refine the presentation of sections, captions, and experimental descriptions. For retrieval and discovery, AI was used to help identify potentially relevant prior work and organize related literature, with references subsequently checked by the authors. For research execution, AI was primarily used as a coding assistant to implement and debug experimental scripts and improve data processing and visualization code. All methodological decisions, experimental design, result interpretation, and final manuscript content were determined and verified by the authors.

6 ETHICS STATEMENT

We use only publicly available, de-identified neural datasets and conduct no new human-subject data collection. All recordings were collected under the ethics and consent procedures of the original studies. We follow the corresponding data-use requirements and report no additional identifiable participant information.

7 REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed dataset descriptions, preprocessing procedures, and evaluation protocols in Appendix C, together with the model architecture, optimization settings, and training configurations in Appendix D. The source code and experiment configurations are publicly available at `https://anonymous.4open.science/r/AMRT`.

REFERENCES

Hamdi Altaheri, Ghulam Muhammad, and Mansour Alsulaiman. Dynamic convolution with multi-level attention for eeg-based motor imagery decoding. *IEEE Internet of Things Journal*, 10(21): 18579–18588, 2023.

Giuseppe Averta, Federica Barontini, Vincenzo Catrambone, Sami Haddadin, Giacomo Handjaras, Jeremia PO Held, Tingli Hu, Eike Jakubowitz, Christoph M Kanzler, Johannes Kühn, et al. U-limb: A multi-modal, multi-center database on arm motion control in healthy and post-stroke conditions. *GigaScience*, 10(6):giab043, 2021.

Davide Borra, Valeria Mondini, Elisa Magosso, and Gernot R Müller-Putz. Decoding movement kinematics from eeg using an interpretable convolutional neural network. *Computers in Biology and Medicine*, 165:107323, 2023.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Mark M Churchland, John P Cunningham, Matthew T Kaufman, Stephen I Ryu, and Krishna V Shenoy. Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron*, 68(3):387–400, 2010.

Pietro Cimarosto, Kyriaki Kostoglou, Luca Tonin, and Gernot Müller-Putz. Decoding eeg signals during the observation of robotic arm movements. *IEEE Access*, 12:195731–195744, 2024.

Parth G Dangi and Yogesh Kumar Meena. Towards simultaneous decoding of kinetic and kinematic movement parameters during grasp and lift task by noninvasive brain imaging. *arXiv preprint arXiv:2607.24081*, 2026.

Shengcai Duan, Le Wu, Aiping Liu, Ruobing Qian, and Xun Chen. An eeg-semg asynchronous time-frequency progressive fusion model for hand trajectory estimation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2025.

Rongrong Fu, Yuan Fang, Fangyao Xu, Changchun Hua, and Chengcheng Hua. Sand: Spectral-attention neural decoding of hand kinematics from low-frequency eeg dynamics. *IEEE Transactions on Biomedical Engineering*, 2026a.

Xi Fu, Weibang Jiang, Rui Liu, Gernot R Müller-Putz, and Cuntai Guan. Eeg-to-gait decoding via phase-aware representation learning. *Neural Networks*, pp. 108608, 2026b.

Amita Giri and Sidharth Pancholi. Neurokinect: A deep learning framework for hand kinematics reconstruction from eeg signals. *IEEE Transactions on Medical Robotics and Bionics*, 2026.

John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

Jasper Hh. The ten-twenty electrode system of the international federation. *Electroencephologr Clin Neurophysiol*, 10:371–375, 1958.

Christopher Holdgraf, Stefan Appelhoff, Stephan Bickel, Kristofer Bouchard, Sasha D'ambrosio, Olivier David, Orrin Devinsky, Benjamin Dichter, Adeen Flinker, Brett L Foster, et al. ieeg-bids, extending the brain imaging data structure specification to human intracranial electrophysiology. *Scientific data*, 6(1):102, 2019.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Anant Jain and Lalan Kumar. Eeg cortical source feature based hand kinematics decoding using residual cnn-lstm neural network. In *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 1–4. IEEE, 2023a.

Anant Jain and Lalan Kumar. Subject-independent trajectory prediction using pre-movement eeg during grasp and lift task. *Biomedical Signal Processing and Control*, 86:105160, 2023b.

Anant Jain and Lalan Kumar. Esi-gal: Eeg source imaging-based trajectory estimation for grasp and lift task. *Computers in Biology and Medicine*, 186:109608, 2025.

Ce Ju, Reinmar Kobler, Liyao Tang, Cuntai Guan, and Motoaki Kawanabe. Deep geodesic canonical correlation analysis for covariance-based neuroimaging data. In *International Conference on Learning Representations*, volume 2024, pp. 36994–37029, 2024.

Matthew T Kaufman, Mark M Churchland, Stephen I Ryu, and Krishna V Shenoy. Cortical activity in the null space: permitting preparation without movement. *Nature neuroscience*, 17(3):440–448, 2014.

Radin Khayyam and Mohammad B Shamsollahi. Comparison of methods for hand trajectory estimation in 3d space using eeg signals. In *2024 31st National and 9th International Iranian Conference on Biomedical Engineering (ICBME)*, pp. 261–268. IEEE, 2024.

Pegah Khoshkavandi, Mohammad B Shamsollahi, and Ali Motie Nasrabadi. Decoding movement kinematics from eeg using cnn based features. In *2025 11th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pp. 134–138. IEEE, 2025a.

Pegah Khoshkavandi, Mohammad B Shamsollahi, and Ali Motie Nasrabadi. Hand movment decoding from eeg signals using kalman filter with parameters estimated via neural networks and least squares method. In *2025 33rd International Conference on Electrical Engineering (ICEE)*, pp. 620–624. IEEE, 2025b.

Reinmar Kobler, Jun-ichiro Hirayama, Qibin Zhao, and Motoaki Kawanabe. Spd domain-specific batch normalization to crack interpretable unsupervised domain adaptation in eeg. *Advances in Neural Information Processing Systems*, 35:6219–6235, 2022.

Attila Korik, Ronen Sosnik, Nazmul Siddique, and Damien Coyle. Decoding imagined 3d hand movement trajectories from eeg: evidence to support the use of mu, beta, and low gamma oscillations. *Frontiers in neuroscience*, 12:130, 2018.

Attila Korik, Naomi Du Bois, Jose Sanchez Bornot, Niall McShane, Christoph Guger, Alessandra Del Felice, Olive Lennon, and Damien Coyle. Decoding the variable velocity of lower-limb stepping movements from eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2025.

Byoung-Hee Kwon, Ji-Hoon Jeong, and Seong-Whan Lee. E2t: Eeg-to-trajectory transformer for motor imagery-based fully-dof motion prediction. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2026.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Jiamian Li, Niall McShane, Attila Korik, Naomi du Bois, Karl McCreadie, Leen Jabban, Benjamin Metcalfe, Özgür Şimşek, and Damien Coyle. Learning residual kinematic corrections for continuous neural decoding via reinforcement learning. *arXiv preprint arXiv:2607.11530*, 2026a.

Yizhe Li, Shixiao Wang, and Jian K Liu. Copilot-assisted second-thought framework for brain-to-robot hand motion decoding. In *2026 Joint International Conference on Automation-Intelligence-Safety (ICAIS) & International Symposium on Autonomous Systems (ISAS)*, pp. 1–6. IEEE, 2026b.

Matthew D Luciw, Ewa Jarocka, and Benoni B Edin. Multi-channel eeg recordings during 3,936 grasp and lift trials with varying weight and friction. *Scientific data*, 1(1):140047, 2014.

Niall Mc Shane, Karl McCreadie, Darryl Charles, Attila Korik, and Damien Coyle. Decoding motion trajectories in an upper limb bci: Linear regression vs deep learning. In *2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)*, pp. 1039–1044. IEEE, 2023.

Kai J Miller. A library of human electrocorticographic data and analyses. *Nature human behaviour*, 3(11):1225–1235, 2019.

Daniel W Moran and Andrew B Schwartz. Motor cortical representation of speed and direction during reaching. *Journal of neurophysiology*, 82(5):2676–2692, 1999.

Miho Ohsaki, Minoru Inoue, and Kimiaki Shirahama. Adaptation and evaluation of eeg classifiers for regression-based hand trajectory decoding. In *2026 IEEE Conference on Artificial Intelligence (CAI)*, pp. 1991–1996. IEEE, 2026.

Chethan Pandarinath and Sliman J Bensmaia. The science and engineering behind sensitized brain-controlled bionic hands. *Physiological reviews*, 102(2):551–604, 2022.

Yulong Peng, Yijian Pan, Yuqi Yang, Nenggan Zheng, Weidong Chen, Xiaoling Hu, and Shaomin Zhang. Eeg2motion: Towards open-vocabulary human motion synthesis from non-invasive brain signals. *arXiv preprint arXiv:2608.14754*, 2026.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

Gerwin Schalk, Jan Kubanek, Kai J Miller, NR Anderson, Eric C Leuthardt, Jeffrey G Ojemann, Dave Limbrick, Daniel Moran, Lester A Gerhardt, and Jonathan R Wolpaw. Decoding two-dimensional movement trajectories using electrocorticographic signals in humans. *Journal of neural engineering*, 4(3):264–275, 2007.

Yonghao Song, Qingqing Zheng, Bingchuan Liu, and Xiaorong Gao. Eeg conformer: Convolutional transformer for eeg decoding and visualization. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:710–719, 2022.

Kaviyan Jayalaksshme Srinivasan, Panneerselvam Periasamy, and Sasikala Gunasekaran. Motor imagery and motor execution: a narrative review of electroencephalographic (eeg) signatures, methodological consistency, and translational applications. *Cureus*, 17(9):e93011, 2025.

Yuting Tang, Neethu Robinson, Xi Fu, Kavitha P Thomas, Aung Aung Phyo Wai, and Cuntai Guan. Reconstruction of continuous hand grasp movement from eeg using deep learning. In *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1–4. IEEE, 2024.

Miao Tian, Shurui Li, Ren Xu, Andrzej Cichocki, and Jing Jin. An interpretable regression method for upper limb motion trajectories detection with eeg signals. *IEEE Transactions on Biomedical Engineering*, 72(10):2961–2971, 2025.

Don M Tucker. Spatial sampling of head electrical fields: the geodesic sensor net. *Electroencephalography and clinical neurophysiology*, 87(3):154–163, 1993.

Pengpai Wang, Zhongnian Li, Peiliang Gong, Yueying Zhou, Fang Chen, and Daoqiang Zhang. Mtrt: Motion trajectory reconstruction transformer for eeg-based bci decoding. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:2349–2358, 2023.

Lichao Xu, Minpeng Xu, Tzyy-Ping Jung, and Dong Ming. Review of brain encoding and decoding mechanisms for eeg-based brain–computer interface. *Cognitive neurodynamics*, 15(4):569–584, 2021.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Lei Zhu, Peng Jiang, Aiai Huang, Jianhai Zhang, and Peng Yuan. M3t-attention: a multi-level multi-scale temporal attention transformer for eeg hand movement trajectory decoding. *Cognitive Neurodynamics*, 20(1):33, 2026.

A RELATED WORK

A.1 CONTINUOUS NEURAL DECODING OF MOVEMENT KINEMATICS

Early continuous movement decoders relied on hand-crafted neural features and classical regressors. Linear regression and partial least squares mapped spectral or slow-cortical-potential features to limb kinematics (Khayyam & Shamsollahi, 2024), often combined with Kalman filtering for temporal smoothing (Cimarosto et al., 2024; Khoshkavandi et al., 2025b), while Gaussian processes and gradient-boosted trees (Tian et al., 2025) provided nonlinear alternatives. Hybrid approaches further replaced parts of state-space models with neural networks, such as learned transition dynamics (Khoshkavandi et al., 2025b) or CNN-based EEG denoising (Khoshkavandi et al., 2025a). More recent methods learn spatial-temporal representations directly from neural signals. CNN-GRU and CNN-LSTM architectures have improved hand, wrist, and lower-limb decoding over classical or purely convolutional models (Khayyam & Shamsollahi, 2024; Jain & Kumar, 2023a; Korik et al., 2025), with later extensions incorporating interpretable spectral-spatial weighting (Borra et al., 2023), lightweight preprocessing and correlation-aware objectives (Giri & Pancholi, 2026), cortical-source features (Jain & Kumar, 2023a), and attention for fine-grained finger-joint reconstruction (Tang et al., 2024). Despite these advances, most methods still rely on fixed convolutional receptive fields or sequential recurrent dynamics for temporal modeling.

A.2 ATTENTION-BASED TEMPORAL MODELING

Continuous movement depends on neural dynamics across multiple temporal scales. Recurrent models capture temporal context but are sequential and difficult to parallelize (Wang et al., 2023), while temporal convolutions are limited by fixed receptive fields (Kwon et al., 2026). Transformer-based methods address these limitations through self-attention, enabling sequence-level trajectory reconstruction (Wang et al., 2023), frame-wise velocity decoding (Kwon et al., 2026), multi-scale temporal attention (Zhu et al., 2026), and spectral-domain attention (Fu et al., 2026a). Related work has also extended Transformers to jointly decode heterogeneous kinetic and kinematic variables (Dangi & Meena, 2026) and to synthesize whole-body motion from EEG (Peng et al., 2026). However, attention alone is not always superior: under controlled comparisons, models retaining temporal convolutions can outperform attention-only variants (Ohsaki et al., 2026). This suggests that local neural dynamics and long-range context are complementary, although most existing methods still combine them within a single pathway for direct position, velocity, or trajectory regression.

A.3 GENERALIZATION ACROSS SUBJECTS

Generalization remains challenging because neural recordings vary across subjects, electrode layouts, and acquisition settings. Li et al. (2026a) first regresses kinematic trajectories from EEG and then applies RL-based residual correction, but the correction relies only on first-stage predictions without revisiting neural representations, leaving cross-subject and cross-dataset generalization largely unaddressed. Fu et al. (2026b) improves unseen-subject gait decoding through phase-aware EEG-motion alignment and domain-aware decoding, while SAND enhances cross-subject robustness by combining low-frequency spectral representations with temporal attention (Fu et al., 2026a). However, these methods largely operate with fixed channel organizations and task-specific kinematic spaces, without explicitly addressing heterogeneous electrode layouts or transferable motion representations across recording modalities and movement paradigms. AMRT addresses this gap through a channel-adaptive neural interface and a velocity-anchored, neural-conditioned trajectory refinement framework for generalization across subjects and datasets.

B BASELINE

We compare AMRT against three recent EEG-based continuous movement decoding methods: COPILOT, TBR, and M3T-ATTENTION. We select these baselines based on three criteria: they support continuous kinematic or trajectory regression, are evaluated on publicly available datasets rather than exclusively on proprietary recordings, and provide open-source implementations that allow faithful reproduction and adaptation to our evaluation protocol. Together, they represent recent

Table 4: Summary of the datasets used in our experiments.

Dataset	Modality	Task	Subjects	Channels	Trial	Mean duration
WAY-EEG-GAL	EEG	6 conditions grasp-and-lift	12	32	3,528	9.2 s
U-Limb	EEG	30 movement tasks	20	128	764	12.6 s
Joystick-Track	ECoG	Circular joystick tracking	4	48-64	150	6.3s

approaches based on multi-scale convolutional modeling, Transformer-based temporal regression, and multi-scale temporal attention.

Copilot (Li et al., 2026b) employs a hybrid convolution-attention architecture for EEG-based hand-motion decoding. Its EEG decoder extracts temporal patterns at multiple receptive-field scales, applies squeeze-and-excitation modules to adaptively reweight channel information, and uses multi-head self-attention to capture long-range temporal dependencies. The resulting representation is directly mapped to continuous hand kinematics.

TBR (Dangi & Meena, 2026) formulates movement decoding as Transformer-based temporal regression over EEG-derived event-related desynchronization/synchronization (ERDS) features. A Transformer encoder models temporal dependencies within the feature sequence and jointly predicts multiple kinetic and kinematic movement variables. We use its regression architecture as a representative feature-based Transformer baseline for continuous decoding.

M3T-Attention (Zhu et al., 2026) is a multi-level, multi-scale temporal attention model designed for EEG-based hand-trajectory decoding. It extracts temporal representations at multiple scales and integrates them through cross-scale attention, allowing the model to capture movement-related neural dynamics over different temporal ranges before predicting continuous hand kinematics.

C Dataset Details

This appendix provides additional details on the three datasets summarized in Section 3.1, including their recording setups, movement paradigms, data organization, and preprocessing procedures. A summary of their key characteristics is provided in Table 4.

C.1 WAY-EEG-GAL

WAY-EEG-GAL (Luciw et al., 2014) contains synchronized 32-channel scalp EEG and kinematic recordings acquired during grasp-and-lift movements. Each trial involves reaching toward an object, grasping and lifting it, and returning the hand to the resting position, under varying object weights and surface conditions. We decode the three-dimensional wrist trajectory, corresponding to the fourth position marker in the provided kinematic stream. For preprocessing, each `WS_P*_S*.mat` file is unpacked into individual trials, with EEG and kinematic signals aligned at 500 Hz. The prediction target is represented as $[\dot{x}, \dot{y}, \dot{z}, x, y, z]$, where (x, y, z) denotes the wrist position and velocity is computed causally using a third-order Savitzky-Golay derivative with a 31-sample trialing window. Following recent continuous trajectory decoding studies (Giri & Pancholi, 2026; Zhu et al., 2026), we additionally apply a 0.5-12 Hz band-pass filter to the EEG signals.

C.2 U-Limb

U-Limb (Averta et al., 2021) provides synchronized high-density EEG and motion-capture recordings during a diverse set of upper-limb movements. The protocol comprises 30 movement tasks, each performed repeatedly. We use all 128 EEG channels and decode the three-dimensional trajectory of the hand dorsum. Following the dataset-provided marker definitions and rigid-pose procedure, we recover the hand reference frame and define the hand-dorsum position as the centroid of the four associated markers. We first apply basic quality control to retain valid trials. Trials are required to contain all four hand-dorsum markers, and those exhibiting implausible hand speeds above 3 m/s, typically caused by marker-tracking discontinuities, are removed. We further retain only participants with at least 20 valid trials to ensure sufficient data for training and evaluation. Kinematics are sampled at 100 Hz and temporally aligned with the 500-Hz EEG recordings. We

retain the hand-dorsum position (x, y, z) and compute velocity using a causal third-order Savitzky–Golay derivative with a seven-sample trialing window at 100 Hz. The resulting prediction target is represented as $[\dot{x}, \dot{y}, \dot{z}, x, y, z]$.

C.3 JOYSTICK-TRACK

Joystick-Track (Miller, 2019; Schalk et al., 2007) is an intracranial ECoG dataset, comprising recordings from four participants performing continuous two-dimensional cursor tracking along a circular target trajectory. The number of available ECoG channels varies across participants. The original task consists of repeated circular-tracking epochs, whereas the released data are stored as continuous recordings without explicit trial annotations. We downsample the ECoG signals from 1 kHz to 500 Hz and the cursor coordinates are resampled to 100 Hz. The decoding target is represented as $[\dot{x}, \dot{y}, x, y]$, where (x, y) denotes the observed cursor position and velocity is computed using a causal third-order Savitzky-Golay derivative with a seven-sample trialing window at 100 Hz. For trial-wise evaluation, we recover complete circular-tracking trials by unwrapping the target phase and discard incomplete boundary segments and duration outliers.

D IMPLEMENTATION DETAILS

D.1 MODEL CONFIGURATION

Unless otherwise specified, we use the same model configuration across datasets, with only modality-dependent input handling and output dimensionality adapted to each setting. The encoder token rate, input context length, prediction horizon, and stride are configurable temporal hyperparameters. In our default setting, recordings are represented at 100 Hz, with a 5-s context ($H = 500$), a 1-s prediction horizon ($L = 100$), and a stride of 0.5s tokens ($S = 50$).

Encoder. The channel adapter uses a hidden dimension of 128, 4 learned spatial queries, and 8 attention heads. With each electrode token contains $\lfloor \frac{H}{L} \rfloor$ (set to 5 by default) raw samples. The waveform encoder consists of LayerNorm, a $5 \to 128$ linear projection, GELU, dropout, and a $128 \to 128$ linear layer. Local and global electrode coordinates are encoded by separate two-layer MLPs with dimensions $3 \to 64 \to 128$. The four spatial-query outputs are concatenated and projected to 128 dimensions. The temporal encoder uses one bidirectional LSTM layer with hidden size 64 per direction, followed by four multi-horizon temporal blocks. Each block operates at hidden dimension 128 and contains depthwise temporal convolutions with receptive fields of 5, 25, 51, and 101 tokens. The global branch uses one learned query with 8-head attention. The feed-forward expansion factor is 4, and the maximum supported context length is 4096 tokens.

Decoder. The Stage-0 decoder uses L learned output queries, hidden dimension 128, 8-head query self-attention, and 8-head cross-attention to the encoder memory. The output head is LayerNorm–Linear–GELU–Dropout–Linear. The decoder predicts three velocity dimensions (v_x, v_y, v_z); for two-dimensional targets, the third dimension is omitted. The Stage-1 refinement decoder uses the same hidden dimension (128), number of queries (L), and number of attention heads (8) as Stage-

Table 5: Architecture hyperparameters of AMRT.

Component	Configuration
Hidden dimension	128
Dropout	0.15
Spatial queries	4
Attention heads	8
BiLSTM hidden size	64 per direction
Temporal blocks	4
Temporal receptive fields	5, 25, 51, 101
Stage-0 output queries	L
Stage-1 output queries	L
Max context length	4096

0. The coarse trajectory is projected from D dimensions to 128 dimensions, concatenated with the decoder state, and fused through a $256 \rightarrow 128$ projection. Separate linear heads predict the residual increment and sigmoid gate. Overlapping predictions are merged using weighted overlap-add. Unless otherwise specified, Stage-1 receives both encoder memory and the coarse trajectory.

D.2 Training Configuration

Stage-0 is trained with AdamW using a learning rate of 2×10^{-4} and weight decay 10^{-4}. Stage-1 freezes Stage-0 and is optimized with AdamW using a learning rate of 10^{-3}, weight decay 10^{-4}. One complete trial is used per optimization step, while Stage-0 windows are materialized in mini-batches of 8 for memory efficiency.

Stage-1 objective combines the trajectory reconstruction loss, gap-increment supervision loss, temporal smoothness penalty, and gate-utility loss. Their weights are 1, 0.5, 0.02, and 0.2, respectively; the gate-utility margin is set to 0.01. Let $\mathbf{b}_t$ denote the coarse Stage-0 trajectory and $\mathbf{y}_t$ the ground-truth trajectory. Stage-1 predicts a correction $\mathbf{c}_t$, yielding $\hat{\mathbf{y}}_t = \mathbf{b}_t + \mathbf{c}_t$. We optimize the refined trajectory with a SmoothL1 reconstruction loss, where D indexes the kinematic control dimension:

$$\mathcal{L}_{\text{traj}} = \frac{1}{|\mathcal{V}|} \sum_{(t,D)\in\mathcal{V}} \text{SmoothL1}(\hat{y}_{t,D}, y_{t,D}), \tag{9}$$

and directly supervise the accumulated raw correction $\bar{\mathbf{c}}_t$ toward the residual gap $\mathbf{y}_t - \mathbf{b}_t$:

$$\mathcal{L}_{\text{gap}} = \frac{1}{|\mathcal{V}|} \sum_{(t,D)\in\mathcal{V}} \text{SmoothL1}(\bar{c}_{t,D}, y_{t,D} - b_{t,D}). \tag{10}$$

To encourage temporally coherent corrections, we penalize consecutive changes,

$$\mathcal{L}_{\text{smooth}} = \frac{1}{|\mathcal{V}_\Delta|} \sum_{(t,D)\in\mathcal{V}_\Delta} |c_{t,D} - c_{t-1,D}|. \tag{11}$$

We additionally supervise the gate $g_{t,D}$ to activate only when the predicted correction improves the coarse trajectory. Specifically,

$$u_{t,D} = \mathbb{I}[|b_{t,D} + \bar{c}_{t,D} - y_{t,D}| + \delta < |b_{t,D} - y_{t,D}|], \qquad \delta = 0.01, \tag{12}$$

and define

$$\mathcal{L}_{\text{utility}} = \frac{1}{|\mathcal{V}|} \sum_{(t,D)\in\mathcal{V}} \text{BCE}(g_{t,D}, u_{t,D}). \tag{13}$$

The complete Stage-1 objective is

$$\mathcal{L}_{\text{Stage-1}} = \mathcal{L}_{\text{traj}} + 0.5\,\mathcal{L}_{\text{gap}} + 0.02\,\mathcal{L}_{\text{smooth}} + 0.2\,\mathcal{L}_{\text{utility}}. \tag{14}$$

Within-dataset and cross-subject training. For within-dataset experiments, both stages are trained on the corresponding training partition, with Stage-0 initialized from scratch and Stage-1 trained from the selected Stage-0 checkpoint. For subject-disjoint evaluation, Stage-0 is trained on source subjects and evaluated directly on the held-out target subject. In cross-subject experiments, for each held-out subject, few-shot adaptation draws $K \in 1, 3, 5, 10$ complete trials from a pool of 10 target-subject training trials under three different selection seeds, and results are averaged. Test trials are never used for adaptation, normalization, or checkpoint selection. In the full-adaptation protocol, Stage-0 is first fine-tuned for 15 epochs with a learning rate of 5×10^{-5}, followed by 15 epochs of Stage-1 refinement-head training with a learning rate of 10^{-5}. In the Stage-1-only protocol, the transferred Stage-0 model remains frozen and only Stage-1 is fine-tuned for 15 epochs at 10^{-5}.

Cross-dataset transfer. Source models are trained on the full source training split. Target adaptation uses $f \in \{0, 0.10, 0.25, 0.50, 1.00\}$ of the target training trials. The $f = 0$ setting denotes calibrated zero-update transfer: no model parameters are updated on the target domain, while velocity denormalization uses control statistics computed from the full target training split. Stage-0 uses a batch size of 16 and AdamW with learning rate 2×10^{-4}. Two-source pretraining runs for $6{,}000$ optimization steps, while target-only training and transferred Stage-0 fine-tuning use $1{,}500$ steps. Stage-1 is trained for 20 epochs using AdamW with learning rate 10^{-3}, weight decay 10^{-4}, and gradient clipping at 1.0. Stage-0 windows are materialized in batches of 8, while each refinement update uses one complete trial.

Table 6: Default training hyperparameters for cross-trial and cross subject.

Hyperparameters	**Settings**
Seed	42
Stage-0 optimizer	AdamW
Stage-0 learning rate	2×10^{-4}
Stage-0 weight decay	10^{-4}
Stage-0 batch size	32(J), 128(others)
Stage-0 epochs	100
Stage-1 optimizer	AdamW
Stage-1 learning rate	10^{-3}
Stage-1 weight decay	10^{-4}
Stage-1 epochs	150(J), 50(others)
Materialized Windows batch size	8
Increment loss weight	0.5
Smoothness loss weight	0.02

E SUPPLEMENTARY EXPERIMENTS

E.1 ADDITIONAL GENERALIZATION RESULTS

Table 7: Few-shot calibration on unseen subjects. Results are trajectory PCC (mean±std) across held-out subjects. The zero-shot result denotes performance without target-subject calibration. Bold indicates the better adaptation strategy at each shot count.

Dataset	Adaptation	zero-shot	1-shot	3-shot	5-shot	10-shot
WAY-EEG-GAL	Full-FT	0.837±0.056	0.819±0.078	**0.838±0.073**	**0.853±0.061**	**0.873±0.030**
	Stage-1-only		**0.822±0.073**	0.838±0.060	0.841±0.058	0.849±0.057
U-Limb	Full-FT	0.427±0.073	0.381±0.117	0.413±0.113	**0.476±0.105**	**0.525±0.100**
	Stage-1-only		**0.400±0.090**	**0.440±0.084**	0.457±0.086	0.476±0.085
Joystick-Track	Full-FT	0.710±0.117	0.579±0.208	0.653±0.121	0.633±0.172	0.679±0.132
	Stage-1-only		**0.714±0.123**	**0.710±0.122**	**0.710±0.119**	**0.722±0.120**

Table 7 reports the complete few-shot calibration results corresponding to the analysis in the main text. For each held-out subject, we start from the model trained under the subject-disjoint LOSO protocol and reserve 10 target-subject trials as a calibration pool, with all remaining trials used as a fixed test set. Zero-shot uses no calibration trials, while K-shot adaptation uses $K \in 1, 3, 5, 10$ trials from this pool before evaluation on the same test set. We compare Full-FT with Stage-1-only adaptation. The full results further show that the preferred adaptation strategy changes with calibration size. Stage-1-only performs better in the lowest-shot settings, particularly on U-Limb and Joystick-Track, whereas full fine-tuning becomes more effective with additional supervision on WAY-EEG-GAL and U-Limb. On Joystick-Track, full fine-tuning remains below the zero-shot baseline across all shot counts, while Stage-1-only preserves or slightly improves performance.

E.2 CROSS-DATASET TRANSFER

E.2.1 SINGLE-SOURCE TRANSFER

Table 8 reports the complete single-source transfer results across all source-target pairs and target-data fractions, complementing the summary results in Section 3.3. For each target fraction, we compare target-only training from scratch with Full-FT and Stage-1-only transfer from each source dataset. Table 9 provides the complete Stage-1 input ablation for the two successful transfer directions to U-Limb. Stage-1 receives two complementary inputs in the full model: (1) the neural memory produced by the frozen neural pathway, and (2) an embedding of the coarse trajectory reconstructed by Stage-0. Accordingly, **Coarse-only** removes the neural-memory input and refines the trajectory using only the coarse-trajectory embedding, whereas **Neural-only** removes the coarse-

trajectory embedding and relies only on the neural memory. **Full** retains both inputs. These results provide the complete numerical counterpart to Figure 2(d).

Table 8: Complete single-source cross-dataset transfer results. Results are trajectory PCC. Bold indicates the best result for each transfer direction and target-data fraction. J, U, and W denote Joystick-Track, U-Limb, and WAY-EEG-GAL.

Transfer	Adaptation	10%	25%	50%	100%
$J \to U$	Scratch	0.137±0.049	0.152±0.080	0.163±0.120	0.192±0.146
	Full-FT	0.337±0.135	0.311±0.139	0.411±0.132	0.301±0.051
	Stage-1-only	**0.454±0.056**	**0.434±0.048**	**0.426±0.077**	**0.413±0.063**
$W \to U$	Scratch	0.137±0.049	0.152±0.080	0.163±0.120	0.192±0.146
	Full-FT	0.264±0.136	0.261±0.183	0.376±0.037	0.301±0.148
	Stage-1-only	**0.371±0.072**	**0.387±0.067**	**0.419±0.076**	**0.427±0.043**
$J \to W$	Scratch	0.757±0.094	0.738±0.107	0.787±0.083	0.787±0.112
	Full-FT	0.741±0.082	0.767±0.095	0.775±0.105	0.781±0.099
	Stage-1-only	**0.787±0.089**	**0.786±0.099**	**0.805±0.105**	**0.808±0.106**
$U \to W$	Scratch	**0.757±0.094**	0.738±0.107	**0.787±0.083**	0.787±0.112
	Full-FT	0.686±0.084	0.749±0.109	0.780±0.086	**0.795±0.077**
	Stage-1-only	0.747±0.084	**0.756±0.066**	0.750±0.085	0.760±0.087
$U \to J$	Scratch	**0.730±0.087**	**0.702±0.104**	**0.733±0.096**	**0.742±0.109**
	Full-FT	0.703±0.116	0.690±0.097	0.730±0.093	0.727±0.131
	Stage-1-only	0.357±0.226	0.484±0.121	0.489±0.104	0.582±0.137
$W \to J$	Scratch	**0.730±0.087**	0.702±0.104	**0.733±0.096**	**0.742±0.109**
	Full-FT	0.725±0.080	**0.702±0.119**	0.724±0.105	0.729±0.107
	Stage-1-only	-0.047±0.224	0.084±0.190	0.267±0.117	0.320±0.110

E.2.2 Dual-Source Transfer

Table 10 and Figure 3(a) report dual-source transfer, where each target is pretrained jointly on the other two datasets. The benefit remains strongly target-dependent. For U-Limb, both adaptation strategies consistently outperform scratch: Stage-1-only performs best with 10-50% target data, while Full-FT becomes more effective at 100%. For WAY-EEG-GAL, Stage-1-only is the most reliable strategy and achieves the best performance at all target-data fractions, reaching 0.831 at 100%. In contrast, neither adaptation strategy improves Joystick-Track over scratch. Full-FT remains slightly below scratch across fractions, while Stage-1-only degrades substantially, consistent with the weak transfer to Joystick-Track observed in the single-source experiments.

Comparing dual-source with single-source transfer further shows that adding a second source is not uniformly beneficial. For U-Limb, the best single-source models remain stronger than dual-source pretraining at every target-data fraction; in particular, J→U with Stage-1-only achieves 0.454 at 10% target data, compared with 0.347 for dual-source transfer. For WAY-EEG-GAL, dual-source pretraining provides a modest additional benefit over the best single-source transfer. Compared with J→W, combining U and J yields similar performance at 10% target data, but consistently higher PCC from 25% onward, reaching 0.831 versus 0.808 at 100%. For Joystick-Track, dual-source pretraining does not recover the degradation already observed with either EEG source. These results reinforce that transfer depends on source-target compatibility rather than source quantity alone: adding heterogeneous source data can improve transfer, but does not guarantee gains when the transferred representations are poorly matched to the target.

E.2.3 Scaling with Heterogeneous Neural Data

We further study the scaling behavior of joint continuous decoding as the amount of balanced heterogeneous neural data increases. For each dataset, complete trials are first split into training, validation, and test partitions at an $8{:}1{:}1$ ratio. The training and validation partitions are pooled across datasets, whereas each test partition is retained for dataset-specific evaluation. Across scaling settings, we vary only the composition and size of the pooled training set while keeping the validation and test

Table 9: Stage-1 input ablation for single-source transfer to U-Limb under Stage-1-only adaptation. Results are trajectory PCC (mean±std); bold indicates the best result at each target-data fraction.

Transfer	Stage-1 input	10%	25%	50%	100%
$U \to U$	Scratch	0.137±0.049	0.152±0.080	0.163±0.120	0.192±0.146
$J \to U$	Full	**0.454±0.056**	**0.434±0.048**	**0.426±0.077**	**0.413±0.063**
	Coarse-only	0.324±0.121	0.325±0.085	0.357±0.109	0.362±0.065
	Neural-only	0.310±0.038	0.299±0.060	0.315±0.042	0.290±0.040
$W \to U$	Full	0.371±0.072	0.387±0.067	**0.419±0.076**	**0.427±0.043**
	Coarse-only	**0.374±0.066**	**0.390±0.063**	0.395±0.081	0.399±0.065
	Neural-only	0.217±0.111	0.223±0.123	0.261±0.109	0.202±0.112

Table 10: Dual-source cross-dataset transfer under different target-data fractions. Results are trajectory PCC (mean±std); bold indicates the best result at each target-data fraction.

Dataset	Adaptation	10%	25%	50%	100%
Joystick-Track	Scratch	**0.730±0.087**	**0.702±0.104**	**0.733±0.096**	**0.742±0.109**
	Full-FT	0.717±0.101	0.674±0.118	0.721±0.097	0.729±0.110
	Stage-1-only	-0.064±0.140	0.035±0.248	0.073±0.241	0.245±0.243
WAY-EEG-GAL	Scratch	**0.757±0.094**	0.738±0.107	**0.787±0.083**	0.787±0.112
	Full-FT	0.717±0.092	0.743±0.113	0.780±0.090	0.775±0.097
	Stage-1-only	0.729±0.171	**0.789±0.102**	0.782±0.135	**0.816±0.079**
U-Limb	Scratch	0.137±0.049	0.152±0.080	0.163±0.120	0.192±0.146
	Full-FT	0.305±0.139	0.300±0.148	0.326±0.133	**0.363±0.059**
	Stage-1-only	**0.333±0.062**	**0.341±0.080**	**0.368±0.071**	0.349±0.062

partitions fixed. Checkpoints are selected on the pooled validation set using a dataset-size-weighted validation score. Let J denote the available training-set size of Joystick-Track, the smallest dataset. We construct balanced mixtures at four scales,

$$(J/4, J/4, J/4),\ (J/2, J/2, J/2),\ (3J/4, 3J/4, 3J/4),\ (J, J, J),$$

where each tuple gives the numbers of Joystick-Track, U-Limb, and WAY-EEG-GAL training trials. Thus, training data increase uniformly across datasets without changing their relative contribution. The optimization budget scales proportionally with data size, while the architecture and evaluation protocol remain fixed. We repeat each setting over 5 random seeds and report the aggregated results.

As shown in Table 11, increasing the balanced heterogeneous training set leads to an overall upward trend in decoding performance, with macro PCC increasing from 0.492 at 90 trials to 0.595 at 360 trials. Joystick-Track shows the clearest improvement with scale, while WAY-EEG-GAL improves at intermediate scales and U-Limb varies more across settings. Despite these dataset-specific fluctuations, the overall trend suggests that joint training may benefit from larger amounts of balanced heterogeneous neural data. These results suggest that scaling balanced heterogeneous neural data has the potential to improve shared continuous decoding.

Table 11: Trajectory reconstruction performance under balanced trial scaling. Values report the mean ± standard deviation of trajectory PCC over five common random seeds. The Macro column is the unweighted average across the three datasets.

Training trials	**U-Limb**	**WAY-EEG-GAL**	**Joystick**	**Macro**
90	0.359 ± 0.055	0.657 ± 0.172	0.459 ± 0.160	0.492 ± 0.082
180	0.306 ± 0.130	0.728 ± 0.051	0.629 ± 0.067	0.554 ± 0.069
270	0.344 ± 0.074	$\mathbf{0.747 \pm 0.006}$	0.650 ± 0.037	0.580 ± 0.033
360	$\mathbf{0.380 \pm 0.042}$	0.731 ± 0.030	$\mathbf{0.673 \pm 0.043}$	$\mathbf{0.595 \pm 0.029}$

E.3 Additional Model Analyses

E.3.1 Model Ablations

We further ablate the temporal encoder and action-chunk decoder under the `ctri-mix` protocol to isolate the contribution of the main architectural components. For the encoder, **w/o BiLSTM** removes the BiLSTM used to model temporal dependencies, while **w/o Multi-Horizon** removes the block that combines neural features from multiple temporal horizons. For the decoder, **Pooled Decoder** replaces the action-chunk decoder with a single globally pooled neural representation shared across all predicted future steps, removing step-specific access to the neural sequence.

Table 12: Model ablations under the `ctri-mix` protocol. Results are trajectory PCC (mean±std); bold indicates the best result.

Model	U-Limb	WAY-EEG-GAL	Joystick-Track
AMRT	**0.530±0.029**	**0.837±0.045**	**0.778±0.076**
w/o BiLSTM	0.364±0.038	0.831±0.018	0.725±0.024
w/o Multi-Horizon	0.330±0.037	0.768±0.095	0.720±0.031
Pooled Decoder	0.385±0.022	0.832±0.027	0.740±0.023

The encoder ablations show that both recurrent temporal modeling and multi-horizon aggregation contribute to decoding performance. Removing the BiLSTM reduces PCC most notably on U-Limb and Joystick-Track, while removing the multi-horizon block causes the largest overall degradation, particularly on U-Limb and WAY-EEG-GAL. Replacing the action-chunk decoder with temporal pooling also consistently reduces performance. These results indicate that AMRT benefits from both multi-scale temporal encoding and step-specific decoding of the motion chunk, rather than relying on a single globally pooled neural representation.

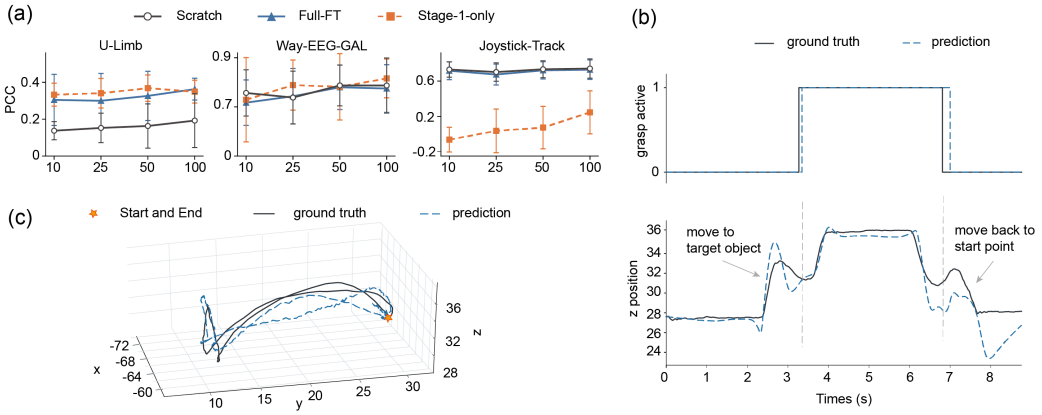

Figure 3: **Summary of supplementary experimental results. (a)** Dual-source cross-dataset transfer across target-data fractions. **(b)** Example of simultaneous trajectory and grasp-state decoding on WAY-EEG-GAL: ground-truth and predicted grasp state (top), and the corresponding z-axis trajectory (bottom). **(c)** Three-dimensional reconstruction of a representative WAY-EEG-GAL trajectory, comparing the prediction with the ground truth in the original movement space.

E.3.2 Grasp-State Decoding

WAY-EEG-GAL additionally provides grasp-state annotations, allowing us to examine whether the Stage-0 representation learned for continuous motion decoding also captures discrete motor-state information. We therefore evaluate binary grasp-state decoding (grasp vs. non-grasp) under the same `ctri`, `ctri-mix`, and `csub` protocols used for trajectory decoding. Specifically, the grasp classifier shares the neural encoder and Stage-0 action-chunk decoder with the motion prediction branch, and applies a lightweight binary classification head to the resulting action-chunk representation. The grasp head is optimized with binary cross-entropy loss. As shown in Table 13, grasp state remains consistently decodable across evaluation protocols, including the unseen-subject setting. This

suggests that the Stage-0 action-chunk representation retains grasp-related motor-state information beyond continuous kinematics. Figure 3(b) provides a qualitative example of joint continuous and discrete motor decoding. We show the z-axis trajectory because vertical motion is directly related to the grasp-and-lift phase of the task. The alignment between grasp transitions and vertical movement further illustrates that the shared Stage-0 representation captures coordinated information about both motor state and continuous kinematics.

Table 13: Grasp-state decoding performance on WAY-EEG-GAL under different evaluation protocols. We report accuracy and F1 score as mean±std.

Protocol	Accuracy	F1
`ctri`	0.905±0.030	0.848±0.066
`ctri-mix`	0.868±0.040	0.794±0.056
`csub`	0.883±0.031	0.844±0.047

E.3.3 CASE STUDY

Figure 3(c) provides a three-dimensional view of the representative trial analyzed along individual coordinate axes in Figure 2(e). While the axis-wise plots show how Stage 1 corrects errors in each coordinate, the 3D visualization further shows that these corrections form a coherent spatial trajectory. The reconstructed path closely follows the overall geometry of the ground truth, including the major changes in movement direction and elevation. Residual deviations mainly occur around sharper local transitions, where the prediction remains smoother than the recorded trajectory.