PI-EMG: ENHANCING SURFACE ELECTROMYOGRAPHIC DECODING USING PRIVILEGED INFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The *emg2pose* benchmark facilitates training generic models for surface electromyography (sEMG) hand pose decoding, which can be used for novel intuitive human-computer interaction modalities. This benchmark defines two tasks: tracking, where hand pose is predicted from sEMG starting from a ground-truth initial pose, and regression, where the initial pose is not provided. After stabilizing position training, position decoding reduced tracking errors by 3.5% – 5.1% relative to matched velocity models. On the regression task, we find that initial pose conditioning provides valuable privileged information (PI), with multi-task tracking and regression training reducing regression task error by 5.3% – 8.0%. Other privileged information that is not available at test time, such as metadata, can also be used to enhance training. To take advantage of metadata such as user ID and sample ID, we implemented Metadata-conditioned Layers (McLayers), which stochastically use either metadata-conditioned mappings or generic mappings during training, but default to generic mappings at test-time. Our model, PI-EMG, incorporates architecture and hyperparameter optimization, multi-task training, and train-time metadata-conditioning, to reduce causal pose decoding error by 10.0% – 16.4% for tracking and 10.0% – 15.7% for regression on unseen users relative to the vemg2pose baseline. PI-EMG therefore achieves a new published state-of-the-art for causal sEMG decoding on the *emg2pose* benchmark. Likewise, for the *emg2qwerty* benchmark, which evaluates two-handed sEMG typing, adding user-wise McLayers to the prior published state-of-the-art model reduced beam search character error rates by 18% for unseen users.

1 INTRODUCTION

Surface electromyography (sEMG) is a promising input modality for hand-based spatial interactions in human-computer interaction and prosthetic control, as it directly represents muscle activations and therefore does not rely on cameras or finger-worn devices. Ideally, sEMG devices should perform well on new users without expensive calibration, which requires training from many users (Kaifosh et al., 2025). Recently, the *emg2pose* benchmark (Salter et al., 2024) was released with that goal, containing 370 hours of paired sEMG and joint angles from 193 users, the largest of its kind. *emg2pose* evaluates generalization for three conditions: held-out users, held-out movement sequences ("stages"), and held-out stages on held-out users. Within these conditions, it evaluates performance when ground-truth initial pose is provided (the tracking task) or omitted (the regression task). The tracking task reflects scenarios in which an initial pose can be established, such as by prompting or by observation with computer vision, whereas the regression task better reflects EMG-only cases where an initial pose is not prompted and computer vision is not available.

We first optimized the architecture and hyperparameters of Salter et al.'s models. We found that position models can perform the tracking task better than velocity integration, contrary to Salter et al.'s observations. A third approach, ΔPos, that predicts the total displacement from a fixed reference, performed better than both.

Next, we identified ways to incorporate additional information into training. The tracking and regression tasks are complementary, and differ only in the presence or absence of initial pose, re-

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.

spectively. This motivates simultaneously learning both tasks in a multi-task (MT) learning setting (Caruana, 1997), such that the model can glean network dynamics from tracking to benefit regression. From the perspective of the regression task, the tracking initial pose acts as privileged information (PI) available only during training (Vapnik & Vashist, 2009). Empirically, this improved performance by 5.3% – 8.0% for regression and did not improve tracking.

Variations in physiology and sensor placement are considered to be nuisance variables (Kaifosh et al., 2025; Li et al., 2026), such that the network must account for diverse input-output distributions across users and examples. We directly encode this variation into the training process using *Metadata-conditioned Layers* (McLayers) that modulate the electrode and feature spaces of the model on a per-user or per-sample basis. During training, features are stochastically routed through either generic or metadata-conditioned mappings, with generic mappings used during inference. This encourages the network to simultaneously learn features that represent each user and sample, along with a generic representation for all users. As a result, McLayers reduced error metrics on unseen users by 2% – 6% on *emg2pose* for tracking and regression tasks. McLayers also reduced the character error rate (CER) on a separate sEMG-based typing task, *emg2qwerty* (Sivakumar et al., 2024), by 15% – 18%.

2 Related Work

sEMG decoding. The *emg2pose* benchmark evaluates generalization of sEMG decoding to new users and movements (Salter et al., 2024). Other work instead applies the *emg2pose* dataset toward tendon-based control, tokenized pose prediction, and pose-informed representation learning with different tasks, data splits, or non-causal sEMG inputs (Verma, 2025; Lovecchio et al., 2025; Cui et al., 2025; 2026; Liu et al., 2026; Chen et al., 2026; Fasulo et al., 2025; Barmpas et al., 2026; Wang et al., 2026; Colot et al., 2026). Xie & Cheung (2026) implement gradient-free fine-tuning on *emg2pose* with per-user Feature-wise Linear Modulation (FiLM) layers (Perez et al., 2018) applied to extracted features. To our knowledge, Salter et al. (2024) remains the prior state-of-the-art for the standard *emg2pose* benchmark. The *emg2qwerty* benchmark is a similarly large dataset that evaluates generalization for two-handed sEMG typing (Sivakumar et al., 2024; Mehlman et al., 2025; Hadidi et al., 2026; Ortone et al., 2026).

Learning using privileged information. Learning using privileged information (LUPI) exploits *privileged information* (PI) during training which is unavailable at test time (Vapnik & Vashist, 2009), such as teacher-provided explanations. Providing the initial pose during training but not inference is one such example. TRAM and related methods use annotator-related and other metadata to improve learning when labels are noisy (Collier et al., 2022; Ortiz-Jimenez et al., 2023). One finding was that adding random PI can reduce overfitting by enabling the network to learn a shortcut for predicting incorrectly labeled examples (Ortiz-Jimenez et al., 2023). For McLayers, metadata-conditioning allows the network to learn shortcuts for modeling cross-user and cross-sample variability.

User conditioning in neural decoding. Several works use "Subject ID as metadata" in electroencephalography (EEG) decoding (Li et al., 2026), including adversarial methods to produce subject-invariant features (Özdenizci et al., 2020; Zhang et al., 2024; Hwang et al., 2020) and various contrastive learning methods (Shen et al., 2022; Kiyasseh et al., 2021; Wang et al., 2023; 2025; Zheng et al., 2026; Musellim et al., 2022). DResNet separates shared and subject-specific weights using adversarial learning, and uses shared weights for unseen subjects (Ma et al., 2019b;a). NeuroDyGait learns session-specific weights and uses an attention mechanism to combine them for out-of-domain prediction (Fu et al., 2026). Adversarial and contrastive approaches have similarly been applied to sEMG decoding (Yang et al., 2025; Rafiei & Mousavi, 2026; Kim et al., 2024). Other methods apply losses at the sample-level, such as the "contrast everything" approach (Wang et al., 2023; 2025). McLayers train generic and user-conditioned mappings by randomly dropping conditioning information, similar to classifier-free guidance for image generation (Ho & Salimans, 2022). This approach removes the need for an additional loss function while still ensuring that the generic mapping generalizes to all users.

3 METHODS

3.1 *emg2pose* DATASET

The *emg2pose* dataset (Salter et al., 2024) contains 370 hours of paired sEMG (16 bipolar channels) and hand pose labels (20-DoF finger joint angles) from 193 users, with sEMG sampled at 2 kHz and hand pose labels upsampled from 60 Hz to 2 kHz. Each user has an average of 3.9 distinct band placements on the distal forearm ("sessions"). Pose estimation is trained and evaluated under two conditions: tracking, where an initial pose $\mathbf{y}_0 \in \mathbb{R}^{20}$ is input to the decoder, and regression, where no initial pose is given. Decoding for left and right hands is evaluated separately.

3.2 POSITION AND VELOCITY EMG2POSE MODELS

The emg2pose and vemg2pose models are composed of a convolutional neural network (CNN) feature extractor and an output "decoder" module (Salter et al., 2024). This CNN is composed of standard temporal convolutions and time-depth separable (TDS) convolutions (Hannun et al., 2019), and has a causal receptive field of 0.895 s and output frequency of 25 Hz (A.4). These features are linearly upsampled to 50 Hz, then input to the decoder module.

Salter et al.'s emg2pose and vemg2pose architectures differ primarily in this decoder module: emg2pose uses a multi-layer perceptron (MLP) to directly predict position, whereas vemg2pose uses a long short-term memory (LSTM) whose velocity outputs are integrated starting from the true initial position (tracking) or a predicted position (regression). Outputs of both decoders are scaled by a scalar s. For scaled decoder output $\mathbf{o}_t \in \mathbb{R}^{20}$, the pose prediction $\hat{\mathbf{y}}_t$ at time t is represented by

$$\text{Pos:} \quad \hat{\mathbf{y}}_t = \mathbf{o}_t, \tag{1}$$

$$\text{Vel:} \quad \hat{\mathbf{y}}_t = \hat{\mathbf{y}}_{t-1} + \mathbf{o}_t. \tag{2}$$

The decoder module has an input dimension of 84: 64 dimensions for CNN features and 20 dimensions for initial pose and pose feedback inputs. On the first time step the decoder receives the initial pose (tracking) or a zero pose (regression), while on subsequent steps the decoder receives the last predicted position. For regression, vemg2pose has 20 additional output dimensions for directly predicting positions for the first 250ms, $\hat{\mathbf{y}}_t = \tilde{\mathbf{o}}_t$, with velocity integration beginning thereafter.

Pos and Vel models. Since the emg2pose MLP decoder lacks the stateful properties of an LSTM, we evaluated position models using the LSTM decoder module, such that both our position-based ("Pos") and velocity-based ("Vel") models use the same core architecture. For regression, Salter et al. (2024) use an initial pose of zero for decoder inputs, which we replace with an initial pose proxy parameter that is shared among all samples, and is not used for velocity integration.

Feedback ablation. By default, the most recent hand pose prediction is concatenated to the CNN features as auto-regressive inputs to the decoder LSTM. Although this feedback is useful to velocity models, as it provides a reference for the velocity at each time step, it is less useful for position models. We tested models without this feedback (feedback ablation, $-$FB), by replacing it on all non-initial steps with the initial pose proxy. For regression, feedback inputs are the same across time, and for tracking, the first feedback input reflects the true initial pose.

3.3 ΔPOS PREDICTION

Delta-position prediction (ΔPos) predicts the change in position from a reference pose $\mathbf{r}$:

$$\Delta\text{Pos:} \quad \hat{\mathbf{y}}_t = \mathbf{r} + \mathbf{o}_t. \tag{3}$$

For tracking, the reference pose is set to the initial position, whereas for regression we train a reference pose parameter shared among all samples, distinct from the initial pose proxy. This combines the initial pose grounding of velocity models with the resistance to error accumulation of position models.

3.4 EMG2POSE TRAINING

We trained all models using PyTorch's AdamW with default β and ϵ (Loshchilov & Hutter, 2017), with weight decay of 0.01 except on LayerNorm and McLayers. We adopt the learning rate schedule

of Sivakumar et al. (2024): linear warmup from 10^{-8} to 10^{-3} over 10 epochs, followed by cosine decay to 10^{-6} over 140 epochs. We used the same channel rotation data augmentation as Salter et al. (2024). Additional hyperparameters are listed in Table A.4.

3.5 Multi-task tracking and regression training

In Salter et al. (2024), models for the tracking and regression tasks were trained independently, i.e. in a single-task setting. Since tracking and regression are strongly related, we jointly optimized them in a multi-task learning setting (Caruana, 1997). The model used for regression decoding is thus able to benefit from the initial pose of the tracking task that would be privileged information for a single-task setting. Specifically, we applied a loss function

$$\mathcal{L}_{\text{multi}} = \lambda_{\text{tracking}}\mathcal{L}_{\text{tracking}} + \lambda_{\text{regression}}\mathcal{L}_{\text{regression}} \tag{4}$$

where $\lambda_{\text{tracking}}$ and $\lambda_{\text{regression}}$ control the relative importance of each task. In our case, $\mathcal{L}_{\text{tracking}}$ and $\mathcal{L}_{\text{regression}}$ were the same loss function as in Salter et al. (2024), namely the weighted sum of the joint angle mean absolute error (MAE) loss (radians) and fingertip distance loss (mm), averaged over time T:

$$\mathcal{L}_{\text{task}} = \frac{1}{T}\sum_{t=0}^{T-1}\left(||\hat{\mathbf{y}}_t - \mathbf{y}_t||_1 + \lambda_{\text{tip}} \cdot \frac{1}{5}\sum_i ||\text{FK}^i_{\text{tip}}(\hat{\mathbf{y}}_t) - \text{FK}^i_{\text{tip}}(\mathbf{y}_t)||_2\right), \lambda_{\text{tip}} = 0.01, \tag{5}$$

where $\mathbf{y}_t, \hat{\mathbf{y}}_t \in \mathbb{R}^{20}$ are the target and predicted joint angles at output time step t, respectively, and $\text{FK}^i_{\text{tip}}(\mathbf{y}_t) \in \mathbb{R}^3$ is a vector representing the forward kinematics from joint angles $\mathbf{y}_t$ to the ith fingertip position. Evaluation similarly defines the metrics

$$\text{AE} := \frac{1}{T}\sum_{t=0}^{T-1}||\hat{\mathbf{y}}_t - \mathbf{y}_t||_1, \quad \text{LD} := \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{17}\sum_j ||\text{FK}^j_{\text{L}}(\hat{\mathbf{y}}_t) - \text{FK}^j_{\text{L}}(\mathbf{y}_t)||_2, \tag{6}$$

for MAE angular error (AE, °) and landmark distance (LD, mm), where $\text{FK}^j_{\text{L}}(\mathbf{y}_t) \in \mathbb{R}^3$ is a vector representing the forward kinematics from joint angles $\mathbf{y}_t$ to the jth hand model landmark. Forward kinematics were temporally downsampled by a factor of 40 to reduce system memory usage, though we display formulas without downsampling for simplicity. We report results for $\lambda_{\text{tracking}} = 0.875$ and $\lambda_{\text{regression}} = 0.125$ for multi-task models.

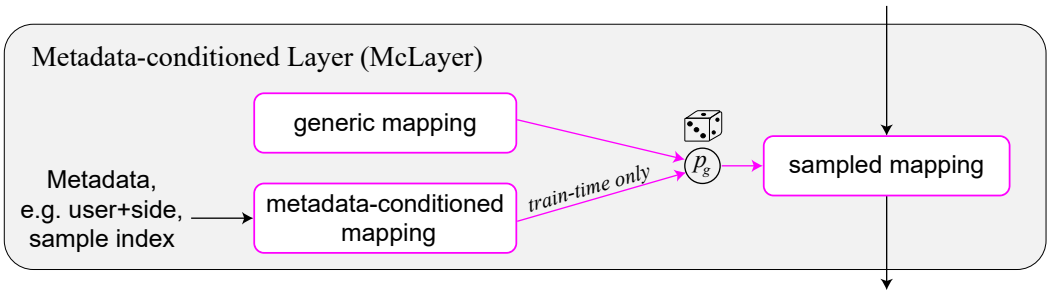

Figure 1: **Metadata-conditioned Layer (McLayer).** During training, features are randomly routed through a generic mapping with probability p_g and through a metadata-conditioned mapping with probability $1 - p_g$. At test time, features are only routed through the generic mapping.

3.6 Metadata-Conditioned Layer (McLayer)

Metadata is information that describes data, including the user ID and side (left vs. right hand), and can be considered privileged information if inference does not utilize this metadata. All hands have unique anatomy, and thus have different sEMG signal distributions for a given kinematics trajectory. It would therefore be reasonable to assume that the optimal network for each individual is also unique, else generalization to new users would be easily solved. Personalized parameters are a common approach, with special consideration needed for unseen users (Ma et al., 2019a; Fu et al., 2026; Lee et al., 2025).

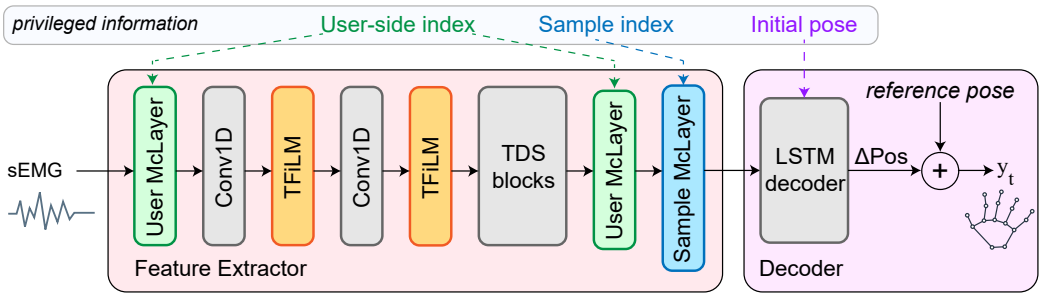

Figure 2: **PI-EMG architecture.** PI-EMG predicts hand pose y_t from sEMG, and follows the core architecture of emg2pose and vemg2pose (Salter et al., 2024), but adds metadata-conditioned layers (McLayers, Figure 1) and causal TFiLM layers. McLayers condition based on the user index and side information (i.e. per hand) or the sample index. PI-EMG additionally predicts the deviation from a reference pose rather than direct position or integrated velocity. This reference pose is the true initial pose when it is provided (tracking task) or a learned null parameter when it is not provided (regression task). This figure shows the PI-EMG-US variant, with both user-side McLayers and a sample McLayer.

In order to balance user variability and generic optimization, we created Metadata-conditioned Layers (McLayers, Figure 1), which stochastically route to a generic mapping with probability p_g or to metadata-conditioned mappings with probability $1 - p_g$, similar to the classifier-free guidance method (Ho & Salimans, 2022). During inference, we simply use the generic mapping. We set $p_g = 0.25$ after testing various p_g in preliminary experiments.

For discrete user ID and hand side metadata, we assign unique parameters for each user-side pair to represent the individual muscle-electrode transfer functions and individualized feature importance. These McLayers are placed before and after the CNN feature extractor, respectively. We also applied McLayers at the sample level, with each training sample having its own set of feature scale and bias parameters. All mappings within each McLayer were initialized to the same transformation, identity, and did not have weight decay applied to them. When the generic mapping is an identity transformation, this functions as a metadata-conditioned form of LayerDrop (Huang et al., 2016; Fan et al., 2019). Additionally, when the generic mapping is affine, it can often be merged with adjacent layers at inference to incur zero additional computational cost. Although we used affine mappings, McLayers can also be generalized to more complex transformations, or layers where conditioning information can be represented as an embedding rather than as a discrete index.

3.7 Causal TFiLM layers

Temporal Feature-wise Linear Modulation (TFiLM) (Birnbaum et al., 2019; Perez et al., 2018) layers apply an element-wise scale and bias at each time step. This allows the network to weight features in a time-varying manner. We implemented causal TFiLM layers by average-pooling features at 50Hz with causal left-padding, then inputting the pooled features to a one-layer GRU with hidden dimension of 128. GRU outputs were used to compute scale and bias arrays using a linear+sigmoid and a linear layer, respectively. The scale and bias were then upsampled to match the inputs without violating causality, and applied element-wise. When applicable, one TFiLM layer was placed after each of the first two convolution layers of the network (A.5).

3.8 PI-EMG model

We combine multi-task training, ΔPos prediction, McLayers, and causal TFiLM layers to produce the PI-EMG model, which uses privileged information to support sEMG decoder training (Figure 2). We report multiple variants: PI-EMG-U, which only has user-side McLayers, PI-EMG-S, which only has a sample-wise McLayer, and PI-EMG-US, which has both. We also added 20 decoder input dimensions (new total of 104 dimensions) to persistently input the ΔPos reference position, which helps support longer-horizon decoding.

3.9 *emg2qwerty* BENCHMARK

We evaluated the effect of McLayers on the *emg2qwerty* task (Sivakumar et al., 2024), building off Ortone et al.'s best model, S4D-large (Gu et al., 2022). We inserted an affine user-side-McLayer after the sEMG input and an element-wise affine user-McLayer after the low-frequency S4D module. Sample McLayers were inserted after the second user McLayer. This modulates the electrode and feature spaces of the S4D-large architecture, leaving the feature extractor itself unchanged.

4 RESULTS

4.1 OPTIMIZING POSITION-BASED EMG2POSE

We found that the position-decoding model from Salter et al. (2024), emg2pose, was suboptimal largely due to the choice of its decoder module and its hyperparameters. The MLP decoder cannot contextualize sEMG features, sometimes performing worse than a static decoder (A.5). Additionally, position models with an LSTM decoder often get stuck in low-movement solutions when the scaling hyperparameter s is too small, such as Salter et al.'s default of $s = 0.01$. After increasing s and tuning the optimizer (A.6, 3.4), position-predicting single-task Pos models for tracking and regression all achieved competitive performance (Table 1, Pos ST). Although, like Salter et al. (2024), we find that position-decoding models predict less smooth trajectories, we also find that a simple speed-adaptive filter is sufficient to achieve smoothness on par with velocity-decoding models without loss of accuracy.

4.2 POSITION AND ΔPOS DECODING REDUCE TRACKING DRIFT

After hyperparameter optimization, our position-based models outperformed velocity models for tracking (Table 1). Across generalization conditions, Pos ST reduced angular error by 3.8–5.1% relative to Vel ST, and position outperformed velocity under both single- and multi-task training in every condition (all $p \leq 1.91 \times 10^{-6}$, paired Wilcoxon signed-rank tests across users). Investigating average angular errors over time demonstrates this difference: position models begin with higher error, but accumulate less error as time progresses (Figure 3).

We speculate that velocity models are more prone to drift because the correct update depends on the preceding position estimate and must compensate for its errors. Position models instead target the true current pose directly, without requiring such error-dependent corrections. Consistent with this difference, removing autoregressive pose feedback significantly degraded velocity decoding but not position decoding (all $p < 0.01$ for Vel; all $p > 0.05$ for Pos; paired Wilcoxon signed-rank tests across users; Table A.5). Conversely, velocity models may have an advantage during the first few hundred milliseconds of tracking, when the pose likely remains close to its initial value. A near-zero velocity output preserves any pose, whereas position models must explicitly learn to reproduce each initial pose.

To capitalize on the advantages of both methods, we introduce ΔPos models. ΔPos prediction models achieve lower early tracking errors similar to those of velocity models, while maintaining the lower errors of position models for the rest of the trajectory (Figure 3). Persistently inputting the reference position also stabilizes long-horizon decoding (> 20 s), without which the decoder often forgets the tracking reference position (Figure A.3).

4.3 TRACKING FROM AN INITIAL POSE GROUNDS LEARNED REGRESSION DYNAMICS

Multi-task training substantially improves regression for both position and velocity decoding, with little effect on tracking performance (Tables 1 and A.5). Across generalization conditions, MT reduced regression angular error by 4.4–6.3% relative to the corresponding single-task models, with all paired comparisons significant ($p \leq 1.91 \times 10^{-6}$, paired Wilcoxon signed-rank tests across users). Although single-task and multi-task models start with similarly high amounts of error during regression, this error reduces more quickly for multi-task models, especially in the first 1 s (Figure 3). Multitask training improves not only accuracy but also smoothness on the regression task (Figures A.6 and A.8). We speculate that tracking facilitates learning hand-pose dynamics by providing the

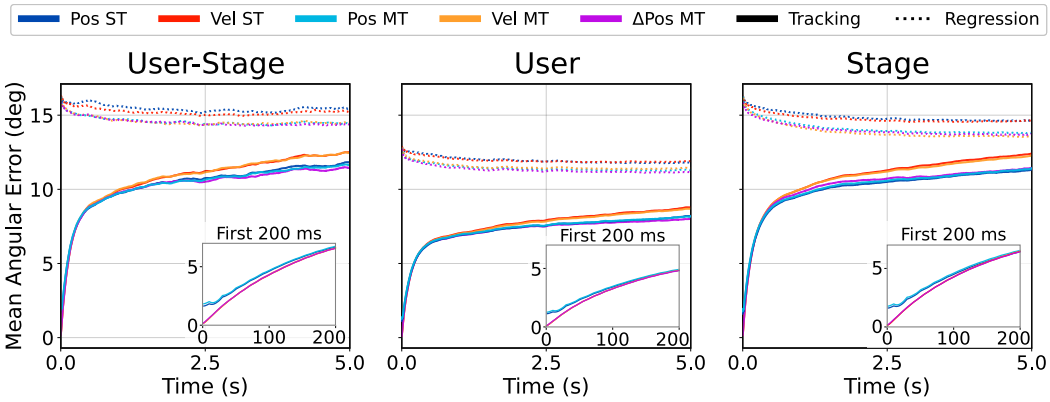

Figure 3: **Angular error at each relative time within the 5 s evaluation windows, averaged across examples.** Solid and dotted lines denote Tracking and Regression. Insets show the first 200 ms. These Pos, Vel, and ΔPos models do not contain McLayers.

Table 1: Tracking and regression task mean absolute joint angular error (AE, °) and mean landmark distance (LD, mm), mean ± sd across users, for single-task (ST), multi-task (MT), and feedback-ablated models (−FB). Models either predicted and integrated velocity (Vel), predicted position (Pos), or predicted the difference in position from the reference initial position (ΔPos). Feedback-ablation rows (−FB) are multi-task models trained without auto-regressive feedback.

	User-Stage		User		Stage	
Model	AE (°)	LD (mm)	AE (°)	LD (mm)	AE (°)	LD (mm)
Salter et al. (2024) †			***Tracking***			
emg2pose	15.05 ± 1.22	20.69 ± 1.39	11.33 ± 1.10	14.75 ± 1.53	15.39 ± 1.50	20.88 ± 1.92
vemg2pose	11.03 ± 1.01	15.35 ± 1.46	7.70 ± 0.98	10.22 ± 1.54	11.20 ± 1.43	15.18 ± 1.81
Ours						
Vel ST	10.75 ± 0.93	14.82 ± 1.40	7.64 ± 0.94	10.02 ± 1.52	10.76 ± 1.34	14.43 ± 1.71
Pos ST	10.25 ± 0.93	14.18 ± 1.35	7.35 ± 0.92	9.67 ± 1.48	10.21 ± 1.32	13.73 ± 1.69
+MT	10.29 ± 0.97	14.22 ± 1.37	7.36 ± 0.90	9.68 ± 1.46	10.28 ± 1.33	13.81 ± 1.70
−FB	10.26 ± 0.98	14.21 ± 1.38	7.34 ± 0.90	9.66 ± 1.45	10.28 ± 1.35	13.82 ± 1.73
ΔPos MT, −FB	**10.12** ± 0.98	**14.02** ± 1.39	**7.27** ± 0.91	**9.55** ± 1.47	**10.10** ± 1.36	**13.56** ± 1.75
Salter et al. (2024)			***Regression***			
emg2pose	15.57 ± 1.30	21.49 ± 1.71	12.57 ± 1.30	16.28 ± 1.81	15.17 ± 1.59	20.53 ± 2.13
vemg2pose	15.63 ± 1.37	21.30 ± 1.87	12.24 ± 1.34	15.82 ± 1.90	15.22 ± 1.58	20.38 ± 2.07
Ours						
Vel ST	15.35 ± 1.32	20.92 ± 1.87	12.17 ± 1.31	15.58 ± 1.85	14.62 ± 1.65	19.62 ± 2.21
Pos ST	15.47 ± 1.30	21.29 ± 1.82	12.19 ± 1.27	15.79 ± 1.80	14.96 ± 1.63	20.23 ± 2.15
+MT	14.58 ± 1.29	19.78 ± 1.83	11.54 ± 1.19	14.74 ± 1.64	14.02 ± 1.62	18.61 ± 2.16
−FB	**14.47** ± 1.25	19.69 ± 1.87	**11.51** ± 1.19	**14.66** ± 1.63	14.02 ± 1.61	18.64 ± 2.16
ΔPos MT, −FB	14.49 ± 1.29	**19.60** ± 1.94	11.57 ± 1.20	14.68 ± 1.60	**13.89** ± 1.63	**18.31** ± 2.16

ground-truth initial pose, rather than requiring models to infer both the initial pose and subsequent movement.

4.4 Early feature adaptation with Causal TFiLM layers improves decoding

Causal TFiLM layers increase the ability of the feature extractor to adapt to inputs in a time-varying manner. These layers moderately improved performance across all conditions (Table 2). To understand whether this improvement was due to architectural difference or simply due to increased computation, we conducted an experiment such that the resulting model had a similar FLOP count to the

Table 2: Tracking and regression MAE angular error (AE, °) and mean landmark distance (LD, mm) for various ΔPos models (3.3). Stage results use generic parameters. Results for Salter et al. (2024) are their best scores (A.3). All PI-EMG models achieved significantly better scores than ΔPos, TFiLM (paired Wilcoxon signed-rank test, $p < 4.9 \times 10^{-4}$ for all comparisons). The ΔPos, TFiLM and PI-EMG models use persistent reference inputs; the plain ΔPos baseline does not (Table A.8).

	User-Stage		User		Stage	
Model	AE (°)	LD (mm)	AE (°)	LD (mm)	AE (°)	LD (mm)
			Tracking			
Salter et al.	11.03 ± 1.01	15.35 ± 1.46	7.70 ± 0.98	10.22 ± 1.54	11.20 ± 1.43	15.18 ± 1.81
ΔPos	10.12 ± 0.98	14.02 ± 1.39	7.27 ± 0.91	9.55 ± 1.47	10.10 ± 1.36	13.56 ± 1.75
ΔPos, TFiLM	9.97 ± 1.00	13.77 ± 1.38	7.20 ± 0.92	9.45 ± 1.49	9.95 ± 1.34	13.32 ± 1.70
PI-EMG						
PI-EMG-U	9.53 ± 0.90	13.07 ± 1.27	6.99 ± 0.86	9.10 ± 1.40	9.65 ± 1.30	12.86 ± 1.65
PI-EMG-S	9.68 ± 0.93	13.28 ± 1.32	7.02 ± 0.86	9.14 ± 1.41	9.61 ± 1.29	12.77 ± 1.65
PI-EMG-US	**9.42** ± 0.90	**12.83** ± 1.26	**6.93** ± 0.84	**8.99** ± 1.36	**9.59** ± 1.30	**12.71** ± 1.63
			Regression			
Salter et al.	15.57 ± 1.30	21.30 ± 1.87	12.24 ± 1.34	15.82 ± 1.90	15.17 ± 1.59	20.38 ± 2.07
ΔPos	14.49 ± 1.29	19.60 ± 1.94	11.57 ± 1.20	14.68 ± 1.60	13.89 ± 1.63	18.31 ± 2.16
ΔPos, TFiLM	14.29 ± 1.39	19.24 ± 2.05	11.38 ± 1.18	14.37 ± 1.58	13.69 ± 1.63	18.00 ± 2.15
PI-EMG						
PI-EMG-U	**13.49** ± 1.32	**17.95** ± 1.88	11.03 ± 1.10	**13.76** ± 1.48	13.28 ± 1.61	17.28 ± 2.10
PI-EMG-S	13.75 ± 1.36	18.41 ± 2.04	11.11 ± 1.15	13.93 ± 1.52	**13.16** ± 1.57	**17.12** ± 2.07
PI-EMG-US	13.54 ± 1.29	18.09 ± 1.93	**11.01** ± 1.02	13.81 ± 1.37	13.51 ± 1.63	17.64 ± 2.10

model without TFiLM layers by reducing the decoder LSTM hidden size (Table A.9). The FLOPs-adjusted model did not perform significantly differently from the unadjusted model ($p > 0.06$ for all comparisons, paired Wilcoxon signed rank test), indicating that the early-feature contextual adaptation of the TFiLM layers more efficiently reduces error than placing that computational capacity in the LSTM decoder.

4.5 Metadata-Conditioned Layers improve generalization

McLayers improved generalization both on models with TFiLM layers (Table 2) and without (Table A.7). For seen users (stage generalization), we report results using generic mappings; we additionally display results using personalized mappings in the Appendix (Table A.11). McLayers applied on inputs and on CNN features both contribute toward this improvement (Table A.7). While both user-side and sample McLayers significantly improved performance, models with user-side McLayers usually performed better. This suggests that the conditioning with structured metadata may be more beneficial than conditioning over the variability of all training samples.

As a further demonstration, we tested McLayers on the *emg2qwerty* sEMG typing task (Sivakumar et al., 2024). Rather than continuous hand pose estimation, *emg2qwerty* tests models' ability to predict the characters in typed words using sEMG. The output of each model is then decoded in a greedy manner or using beam search with a 6-gram character language model (LM). Adding McLayers to Ortone et al.'s S4D-large model substantially improved decoding for unseen users, decreasing greedy decoding error by 15%, and by 18% when using the LM to perform beam search (Table 3). This demonstrates the transferability of McLayers to multiple sEMG decoding frameworks.

5 Discussion

In this work, we present several key findings for sEMG decoding. First, position decoding can outperform velocity decoding for sEMG hand pose estimation, as it is more resistant to accumulated errors. Delta-position decoding unifies the advantages of both approaches. Second, exploiting priv-

Table 3: Character error rates (CER) on unseen users (zero-shot) with greedy decoding and with a 6-gram character language model (LM). We modified Ortone et al.'s S4D-large model by adding User (+U) and Sample (+S) McLayers. Note that the model from Mehlman et al. (2025) requires roughly $900\times$ the computational cost of the S4D-large model from Ortone et al. (2026). Standard deviations measure variation across seeds *.

Model	Greedy zero-shot CER	6-gram LM zero-shot CER
TDS ConvNet (Sivakumar et al., 2024)	55.38	51.78
SplashNet (Hadidi et al., 2026)	44.78	35.67
Distilled Transformer tiny (Mehlman et al., 2025)	$\mathbf{31.9 \pm 0.4}$	—
S4D-large (Ortone et al., 2026)	42.7 ± 0.6	34.0 ± 0.7
S4D-large +S	40.7 ± 1.2	32.0 ± 1.6
S4D-large +U +S	37.3 ± 0.7	28.3 ± 0.9
S4D-large +U	36.4 ± 0.6	$\mathbf{27.8 \pm 0.3}$

ileged information in task structure can benefit lower-information tasks such as regression by joint training with higher-information tasks such as tracking. Finally, stochastic metadata-conditioning improves generalization across both hand pose decoding and two-handed sEMG typing without increasing model complexity during inference.

Initial pose information grounds regression dynamics. Successful decoding depends on learning hand pose dynamics, which describe state transitions from each time step to the next. During tracking, the model is able to learn these dynamics without first recovering the initial state $\mathbf{y}_0$. However, regression must instead learn these dynamics without knowing this initial state. Introducing this conditioning information during training encourages the LSTM to mostly reflect the initial-pose-conditioned dynamics learned from tracking, producing smoother and more accurate trajectories during regression (Figures A.6 and A.8).

McLayers provide shortcuts for sample variability. McLayers allow shared weights to be optimized with mappings specialized to individual users or examples, while requiring the shared weights to remain useful with generic mappings. One possible benefit is that specialized transformations accommodate variation that the shared model would otherwise have to fit entirely within its common representation. Consistent with this interpretation, user identity was less linearly decodable from generic McLayer features than from an otherwise matched model without McLayers (26.6% vs. 42.3% balanced accuracy), while activating the user-specific mappings increased user decodability to 84.0% (Figure A.4). The generic route then prevents training from depending exclusively on those transformations. This is supported empirically, as a low default probability of $p_g = 0.01$ performed worse than our default $p_g = 0.25$ even for user-conditioned mappings (Table A.11).

The fact that sample mappings can help without a user-level grouping suggests that McLayers can still help when user-identifying metadata is unavailable. At the same time, the stronger user-only typing result shows that finer conditioning is not always preferable. Grouping several examples under a shared mapping imposes a different constraint from giving each its own parameters, and its usefulness depends on the prediction problem.

5.1 Limitations

The evaluation in our work is limited to offline benchmarks, and we did not evaluate in any closed-loop settings. Future work should include practical deployment to new users, as well as evaluating the relationship between performance and the amount of calibration data. Additionally, the train-time parameter cost of McLayers scales with the number of metadata classes, potentially requiring solutions such as memory-mapped parameters.

AI USE STATEMENT

In this work, we used generative AI to implement author-developed ideas by generating code for experiments and several figures. We also used generative AI to provide feedback on research methodology, and for brainstorming, identifying relevant literature, and summarizing existing literature. We did not use generative AI to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, or interpret results. Generative AI was used to edit the paper and produce alternative drafts. The authors take full responsibility for the final content of this work, including all code, figures, claims, and results.

REPRODUCIBILITY STATEMENT

The *emg2pose* dataset is publicly available per its original paper. Code used to produce these results is available as supplementary materials, and will be uploaded to GitHub upon acceptance. Models were trained on a server with 8 NVIDIA RTX 6000 PRO Blackwell GPUs, where each training run using 1 GPU typically completed in less than 1 day.

AUTHOR CONTRIBUTIONS

redacted

ACKNOWLEDGMENTS

redacted

REFERENCES

Konstantinos Barmpas, Na Lee, Dimitrios Chalatsis, William Raftery, Yannis Panagakis, Dimitrios A. Adamos, Nikolaos Laskaris, Alexandros Koliousis, Dario Farina, and Stefanos Zafeiriou. Neurorvq: Multi-scale biosignal tokenization for generative foundation models, 2026. URL https://arxiv.org/abs/2510.13068.

Sawyer Birnbaum, Volodymyr Kuleshov, Zayd Enam, Pang Wei W Koh, and Stefano Ermon. Temporal film: Capturing long-range sequence dependencies with feature-wise modulations. *Advances in Neural Information Processing Systems*, 32, 2019.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on human factors in computing systems*, pp. 2527–2530, 2012.

Yanlong Chen, Mattia Orlandi, Pierangelo Rapa, Simone Benatti, Luca Benini, and Yawei Li. Physiowave: A multi-scale wavelet-transformer for physiological signal representation. *Advances in Neural Information Processing Systems*, 38:169357–169399, 2026.

Mark Collier, Rodolphe Jenatton, Effrosyni Kokiopoulou, and Jesse Berent. Transfer and marginalize: Explaining away label noise with privileged information. In *International conference on machine learning*, pp. 4219–4237. PMLR, 2022.

Martin Colot, Cédric Simar, Guy Cheron, Ana Maria Cebolla Alvarez, and Gianluca Bontempi. Decoding high-dimensional finger motion from emg using riemannian features and rnns. *arXiv preprint arXiv:2604.22499*, 2026.

Wenhui Cui, Christopher Sandino, Hadi Pouransari, Ran Liu, Juri Minxha, Ellen Zippi, Aman Verma, Anna Sedlackova, Erdrin Azemi, and Behrooz Mahasseni. Cpep: Contrastive pose-emg pre-training enhances gesture generalization on emg signals. *arXiv preprint arXiv:2509.04699*, 2025.

Wenhui Cui, Christopher M. Sandino, Hadi Pouransari, Ran Liu, Juri Minxha, Ellen Zippi, Erdrin Azemi, and Behrooz Mahasseni. Embridge: Enhancing gesture generalization from emg signals through cross-modal representation learning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 149668–149684, 2026.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019.

Matteo Fasulo, Giusy Spacone, Thorir Mar Ingolfsson, Yawei Li, Luca Benini, and Andrea Cossettini. Tinymyo: a tiny foundation model for flexible emg signal processing at the edge. *arXiv preprint arXiv:2512.15729*, 2025.

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, 2026.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in neural information processing systems*, 35:35971–35983, 2022.

Nima Hadidi, Jason Chan, Ebrahim Feghhi, and Jonathan Kao. Splashnet: Split-and-share encoders for accurate and efficient typing with surface electromyography. *Advances in Neural Information Processing Systems*, 38:166117–166141, 2026.

Awni Hannun, Ann Lee, Qiantong Xu, and Ronan Collobert. Sequence-to-sequence speech recognition with time-depth separable convolutions. *arXiv preprint arXiv:1904.02619*, 2019.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pp. 646–661. Springer, 2016.

Sunhee Hwang, Minsong Ki, Kibeom Hong, and Hyeran Byun. Subject-independent eeg-based emotion recognition using adversarial learning. In *2020 8th international winter conference on brain-computer interface (BCI)*, pp. 1–4. IEEE, 2020.

Patrick Kaifosh, Thomas R Reardon, and CTRL-labs at Reality Labs. A generic non-invasive neuromotor interface for human-computer interaction. *Nature*, 645(8081):702–711, 2025.

Hyeonji Kim, Jaehoon Kim, and Seoung Bum Kim. Cross-subject generalizable representation learning with class-subject dual labels for biosignals. *Knowledge-Based Systems*, 295:111855, 2024.

Dani Kiyasseh, Tingting Zhu, and David A Clifton. Clocs: Contrastive learning of cardiac signals across space, time, and patients. In *International conference on machine learning*, pp. 5606–5615. PMLR, 2021.

Hunmin Lee, Ming Jiang, Zhi Yang, and Qi Zhao. Fedemg: Achieving generalization, personalization, and resource efficiency in emg-based upper-limb rehabilitation through federated prototype learning. *IEEE Transactions on Biomedical Engineering*, 73(2):803–817, 2025.

Taida Li, Yujun Yan, Fei Dou, Wenzhan Song, and Xiang Zhang. Cross-subject generalization for eeg decoding: a survey of deep learning methods. *Progress in Biomedical Engineering*, 8(2): 022013, 2026.

Ruofan Liu, Yichen Peng, Takanori Oku, Chen-Chieh Liao, Erwin Wu, Shinichi Furuya, and Hideki Koike. From pose to muscle: Multimodal learning for piano hand muscle electromyography. *Advances in Neural Information Processing Systems*, 38:115558–115587, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Rossana Lovecchio, Pranav Mamidanna, Bart Jansen, Dario Farina, and Tom Verstraten. Vq-myopose: Movement tokenization improves decoding of hand kinematics from surface emg wristbands. In *International Conference on Computer Vision Hands Workshop*, 2025.

Bo-Qun Ma, He Li, Yun Luo, and Bao-Liang Lu. Depersonalized cross-subject vigilance estimation with adversarial domain generalization. In *IJCNN*, pp. 1–8, 2019a.

Bo-Qun Ma, He Li, Wei-Long Zheng, and Bao-Liang Lu. Reducing the subject variability of eeg signals with adversarial domain generalization. In *International Conference on Neural Information Processing*, pp. 30–42. Springer, 2019b.

Nicholas Mehlman, Jean-Christophe Gagnon-Audet, Michael Shvartsman, Kelvin Niu, Alexander H Miller, and Shagun Sodhani. Scaling and distilling transformer models for semg. *arXiv preprint arXiv:2507.22094*, 2025.

Serkan Musellim, Dong-Kyun Han, Ji-Hoon Jeong, and Seong-Whan Lee. Prototype-based domain generalization framework for subject-independent brain-computer interfaces. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 711–714. IEEE, 2022.

Guillermo Ortiz-Jimenez, Mark Collier, Anant Nawalgaria, Alexander Nicholas D'Amour, Jesse Berent, Rodolphe Jenatton, and Efi Kokiopoulou. When does privileged information explain away label noise? In *International Conference on Machine Learning*, pp. 26646–26669. PMLR, 2023.

Andrea Ortone, Maxime Fabre, Riccardo Massa, Alberto Mazzoni, Giuseppe Desoli, Calogero Maria Oddo, and Emre Neftci. Efficient semg decoding with hybrid, multi-timescale neural networks. *Biomedical Signal Processing and Control*, 128:111215, 2026.

Ozan Özdenizci, Ye Wang, Toshiaki Koike-Akino, and Deniz Erdoğmuş. Learning invariant representations from eeg via adversarial inference. *IEEE access*, 8:27074–27085, 2020.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Hamed Rafiei and Ali Mousavi. Conservative subject invariant emg-based gesture recognition. *arXiv preprint arXiv:2607.03783*, 2026.

Sasha Salter, Richard Warren, Collin Schlager, Adrian Spurr, Shangchen Han, Rohin Bhasin, Yujun Cai, Peter Walkington, Anuoluwapo Bolarinwa, Robert Wang, et al. emg2pose: A large and diverse benchmark for surface electromyographic hand pose estimation. *Advances in Neural Information Processing Systems*, 37:55703–55728, 2024.

Xinke Shen, Xianggen Liu, Xin Hu, Dan Zhang, and Sen Song. Contrastive learning of subject-invariant eeg representations for cross-subject emotion recognition. *IEEE Transactions on Affective Computing*, 14(3):2496–2511, 2022.

Viswanath Sivakumar, Jeffrey Seely, Alan Du, Sean R Bittner, Adam Berenzweig, Anuoluwapo Bolarinwa, Alexandre Gramfort, and Michael I Mandel. emg2qwerty: A large dataset with baselines for touch typing using surface electromyography. *Advances in Neural Information Processing Systems*, 37:91373–91389, 2024.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.

Sagar Verma. emg2tendon: From sEMG Signals to Tendon Control in Musculoskeletal Hands. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.135.

Nana Wang, Suli Wang, Gen Li, Pengfei Ren, and Hao Su. New synthetic goldmine: Hand joint angle-driven emg data generation framework for micro-gesture recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 17787–17795, 2026.

Yihe Wang, Yu Han, Haishuai Wang, and Xiang Zhang. Contrast everything: A hierarchical contrastive framework for medical time-series. *Advances in Neural Information Processing Systems*, 36:55694–55717, 2023.

Yihe Wang, Nan Huang, Nadia Mammone, Marco Cecchi, and Xiang Zhang. Lead: An eeg foundation model for alzheimer's disease detection. *arXiv preprint arXiv:2502.01678*, 2025.

Eric Xie and Hei Shing Cheung. React: A conditioning framework for user-adaptive semg hand pose estimation. *arXiv preprint arXiv:2605.30127*, 2026.

Yuanbo Yang, Hiu Hong Teo, Yeong Jin King, Yao Zhang, Gang Wang, and Xiangxu Qu. Calibration-free semg intention recognition via self-supervised pretraining and adversarial domain alignment for upper-limb rehabilitation. *Scientific Reports*, 15(1):44850, 2025.

Zongpeng Zhang, Taoyun Ji, Mingqing Xiao, Wen Wang, Guojing Yu, Tong Lin, Yuwu Jiang, Xiaohua Zhou, and Zhouchen Lin. Cross-patient automatic epileptic seizure detection using patient-adversarial neural networks with spatio-temporal eeg augmentation. *Biomedical Signal Processing and Control*, 89:105664, 2024.

Ruizhe Zheng, Lingyan Mao, Dingding Han, Tian Luo, Yi Wang, Jing Ding, and Yuguo Yu. Fapex: fractional amplitude-phase expressor for robust cross-subject seizure prediction. *Advances in Neural Information Processing Systems*, 38:126404–126424, 2026.

A Appendix

A.1 Glossary of abbreviations

Table A.1: Glossary of abbreviations.

Abbreviation	Term	Reference
sEMG	surface electromyography	
EEG	electroencephalography	
PI	privileged information	Vapnik & Vashist (2009)
ST	single-task training	
MT	multi-task training	
ΔPos	Delta position, i.e. outputting change in position from a reference position	
$\mathcal{L}$	loss	
$\mathbf{y} \in \mathbb{R}^{20}$	20-DoF finger joint angles	
$\mathbf{y}' \in \mathbb{R}^{20}$	temporal derivative of joint angles	
$\mathbf{x} \in \mathbb{R}^{T \times 16}$	input sEMG	
$\mathbf{o} \in \mathbb{R}^{20}$	Scaled decoder module output	
$\tilde{\mathbf{o}} \in \mathbb{R}^{20}$	Second scaled decoder module output, only used for the first 250 ms of velocity models	
p_g	the probability of using the generic mapping in a McLayer	
FK	forward kinematics	
MAE	mean absolute error	
AE	angular error (degrees), usually averaged across 20 DoF joints	
LD	landmark distance (mm), usually averaged across 17 landmark locations	
FB	output feedback. $-$FB: without output feedback	
CNN	convolutional neural network	
LSTM	long short-term memory	
GRU	gated recurrent unit	
MLP	multi-layer perceptron	
TDS	Time-Depth Separable Convolution	Hannun et al. (2019)
FiLM	Feature-wise Linear Modulation	Perez et al. (2018)
TFiLM	Temporal Feature-wise Linear Modulation	Birnbaum et al. (2019)
FLOPs	FLoating point OPerations	
S4D	Structured State Space sequence diagonal model	Gu et al. (2022)

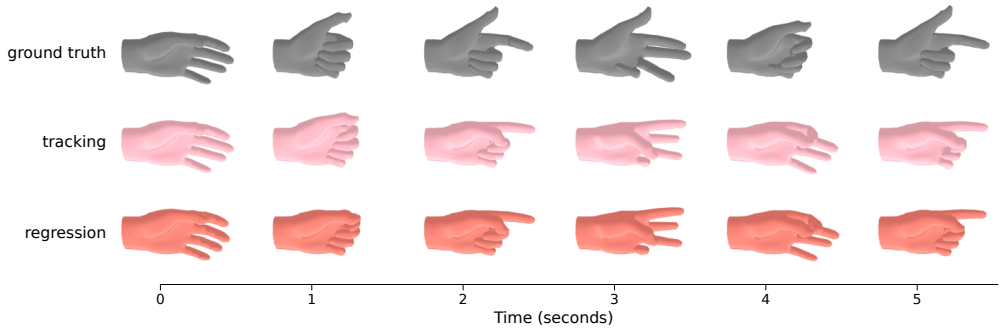

Figure A.1: Example hand for tracking and regression for a PI-EMG-U checkpoint for the held-out Counting1 stage, which consists of "Fingers counting up and down (5)". Data was taken from a window without inverse kinematics failures from an arbitrary test file from an unseen user.

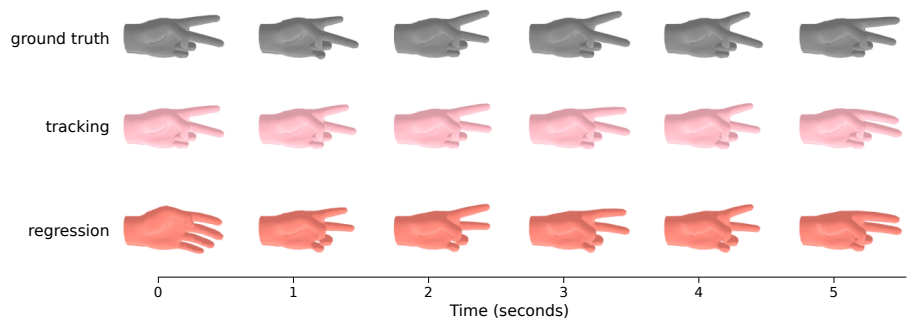

Figure A.2: Example hand for tracking and regression for a PI-EMG-U checkpoint for the held-out Gesture3 stage, which consists of "Extend thumb and pinky, curl middle fingers", "Vulcan salute", and "Peace sign". Data was taken from a window without inverse kinematics failures from an arbitrary test file from an unseen user.

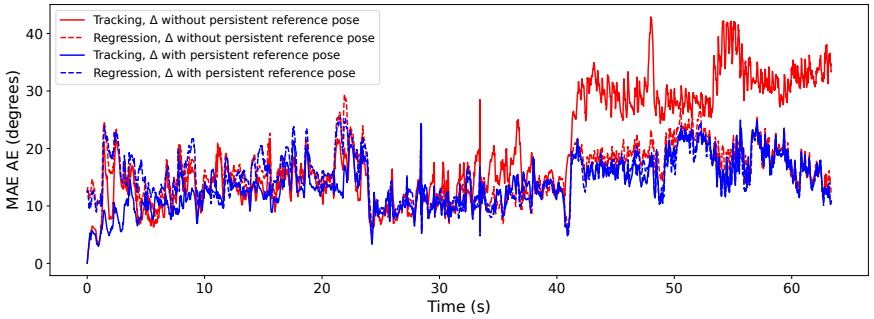

Figure A.3: Example trajectory for PI-EMG-U decoders with and without persistent reference pose inputs. Despite achieving similar performance on 5 s windows (the *emg2pose* default), predictions for PI-EMG-U begin to drift after 20–40 s if the reference position is not persistently input during tracking.

A.2 McLayers reduce user-specific representations in the generic backbone

We hypothesize that McLayers accommodate user-specific variation in dedicated parameters, allowing the generic backbone to focus on features shared across users. We examine whether the learned representations are consistent with this division by asking how readily user identity can be decoded under generic and personalized inference, relative to a model trained without McLayers.

We use data from 130 training-seen users performing two movement stages held out from model training, with both hands and two recording sessions per user. All conditions use identical input windows and sampled time points, balanced across users, sessions, stages, and hands. Figure A.4(a–c) shows independent t-SNE projections of the same observations from a subset of 20 users, with consistent user colors across panels.

We quantify user ID decodability using a regularized multinomial logistic-regression probe on the original 64-dimensional features of all 130 users. The probe is trained on one session per user and tested on the other, then the sessions are exchanged. Standardization and regularization selection use only the training data, with grouped cross-validation keeping recordings and paired hands together. Figure A.4(d) reports balanced accuracy over the held-out predictions, measuring user ID decodability across recording sessions.

User ID decoding accuracy falls from 42.3% for the baseline to 26.6% under generic McLayer inference, and rises to 84.0% when the personalized mappings are activated (Figure A.4(d)). The personalized path receives known user IDs, so its higher accuracy reflects the effect of that con-

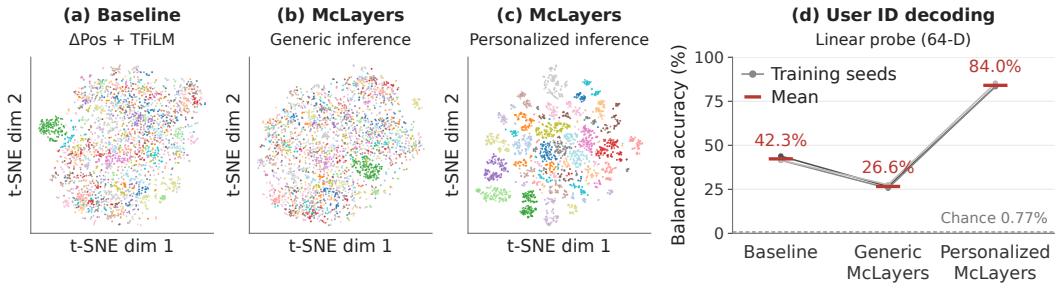

Figure A.4: **User identity decodability in generic and personalized encoder representations.** We compare otherwise matched ΔPos + TFiLM models without McLayers (a) and with McLayers under generic (b) and personalized (c) inference. Generic inference uses the learned default mappings; personalized inference uses the known user and hand-side identity to select both the input and output mappings. The latter two conditions share the same checkpoints and backbone weights. Features are frozen 64-dimensional encoder outputs, after the output McLayer when present and before decoder processing.

ditioning on the representation. Together, these results are consistent with McLayers providing user-specific representational capacity during training, reducing the burden on the generic backbone to encode user-dependent variation and preserving capacity for features shared across users. The analysis measures linear user ID decodability and supports this capacity interpretation indirectly.

A.3 Evaluation

We use the same data splits as those provided by Salter et al. (2024). 15 users are held-out as a validation set, and 20 users are held-out for testing. Additionally, 23 stages (gesture sequences, with each containing multiple gestures) were used for training, with 6 withheld for testing.

† Salter et al.'s code differs slightly from their description of statistical analysis, and is therefore ambiguous. As a result, some values in our tables are lower than those in their manuscript because we recalculated metrics according to the manuscript's description. The ambiguity lies in whether metrics should be averaged across windows within each stage before averaging across stages, or whether windows should be pooled across all stages and directly averaged, when calculating results for Stage generalization. For this work, following their description, we follow the latter, and averaged metrics equally across windows within each of 198 user-generalization pairs (20 User-Stage, 20 User, and 158 Stage).

We averaged metrics across seeds within each user-generalization pair, then report the mean and standard deviation across users. Where reported, statistical comparisons use paired Wilcoxon signed-rank tests on user-level scores after averaging across seeds ($n = 20$ for User and User-Stage; $n = 158$ for Stage). Training for Pos and Vel models used 9 seeds each, and ΔPos models used 3 seeds each due to computational constraints. Metrics for each window were averaged across its corresponding timesteps. We also recomputed metrics for Salter et al. (2024) with this method using their provided model checkpoints.

* For *emg2qwerty* models, S4D-large+U was trained with 4 seeds, and S4D-large+S and S4D-large+U+S were trained with 3 seeds each due to computational constraints. Ortone et al. (2026) train their models with 4 seeds each. For each predicted–target sequence pair, the character error rate (CER) is their Levenshtein distance divided by the length of the target sequence.

Distilled Transformer tiny was evaluated on 4-second sequences whereas Ortone et al.'s S4D was evaluated on 80-second sequences. All other models, including ours, evaluate each session as a single sequence (2 test sessions for each of 8 held-out users). S4D-large +U achieves a greedy CER of 36.9 ± 0.6 and a 6-gram CER of 27.8 ± 0.3 for 80-second sequences.

Table A.2: Architecture of Pos, Vel, and ΔPos models, similar to Salter et al. (2024). Indented entries are part of their respective parent module, with entries at the same level applied in series.

Layer	Output channels	Kernel size	Stride
sEMG	16	—	2000 Hz
Conv1dBlock	256	11	5
Conv1dBlock	256	5	2
TDSStage			
Conv1dBlock	256	17	4
TDSConvEncoder			
TDSConv2dBlock	16×16	(1, 9)	(1, 1)
TDSFullyConnectedBlock	256	—	—
TDSConv2dBlock	16×16	(1, 9)	(1, 1)
TDSFullyConnectedBlock	256	—	—
TDSStage			
Conv1dBlock	256	9	2
TDSConvEncoder			
TDSConv2dBlock	$16 \times 16, W = 16$	(1, 5)	(1, 1)
TDSFullyConnectedBlock	256	—	—
TDSConv2dBlock	$16 \times 16, W = 16$	(1, 5)	(1, 1)
TDSFullyConnectedBlock	256	—	—
Linear	64	—	—
Linear upsample	—	0.5	—
SequentialLSTM			
LSTM Layer	512	—	—
LSTM Layer	512	—	—
LeakyReLU	—	—	—
Linear	20 (40 for Vel)	—	—

A.4 EMG2POSE ARCHITECTURE

The sEMG wristband (sEMG-RD) used in the dataset has 16 bipolar channels approximately evenly spaced around the wrist (Kaifosh et al., 2025). Following Salter et al. (2024), we apply rotation augmentation, with channels circularly shifted to the left or to the right by one index with a probability of $1/3$ each. Models were trained with an effective batch size of 256 and input windows of 2.895 s (5,790 samples = 1,790 samples causal context + 4,000 samples prediction). Models were evaluated on input windows of 5.895 s, including the same causal context length.

For convenience, we use the layer names from Salter et al.'s code. Each Conv1dBlock consists of a 1d convolution, ReLU activation, unused dropout ($p_{\text{dropout}} = 0$), and LayerNorm applied on convolutional channels. Each TDSStage consists of a Conv1dBlock, a TDSConvEncoder, and an optional Linear output layer (Hannun et al., 2019). The TDSConvEncoder consists of a TDSConv2dBlock, which first reshapes inputs from shape $(C_{\text{in}}, T_{\text{in}})$ to $(C_{\text{in}}/W, W, T_{\text{in}})$, applies a 2D convolution with kernel size $(1, K)$, then reshapes the output to $(C_{\text{out}}, T_{\text{out}})$, for channels C, width hyperparameter W, number of time steps T, and kernel width K. The TDSFullyConnectedBlock is an MLP with a hidden dimension of 256, ReLU activation, and LayerNorm.

A.5 PI-EMG ARCHITECTURE

The PI-EMG architecture is shown in Table A.3, while novel layers are explained in the Methods.

A.6 OPTIMIZING EMG2POSE HYPERPARAMETERS

Previously, Salter et al. (2024) reported that velocity decoding outperformed position decoding on the tracking task. However, two aspects of their implementation indicate that the position-decoding model emg2pose was trained in an unstable regime. First, they selected an MLP for position decoding but an LSTM for velocity decoding. Second, their reported position-decoding tracking accuracy

Table A.3: Modified architecture of ΔPos model for PI-EMG. We only insert layers into the ΔPos model (Table A.2), without modifying individual layers from the base architecture, except for certain control experiments.

Layer	Output channels	Kernel size	Stride
sEMG	16	—	2000 Hz
User-side McLayer	16	—	—
Conv1dBlock	256	11	5
Causal TFiLM	256	8	1 = 8/8
Conv1dBlock	256	5	2
Causal TFiLM	256	8	1 = 4/4
TDSStage	256	81	4
TDSStage	64	25	2
User-side McLayer	64	—	—
Sample McLayer (diagonal weights)	64	—	—
Linear upsample	—	0.5	—
SequentialLSTM	20	—	—

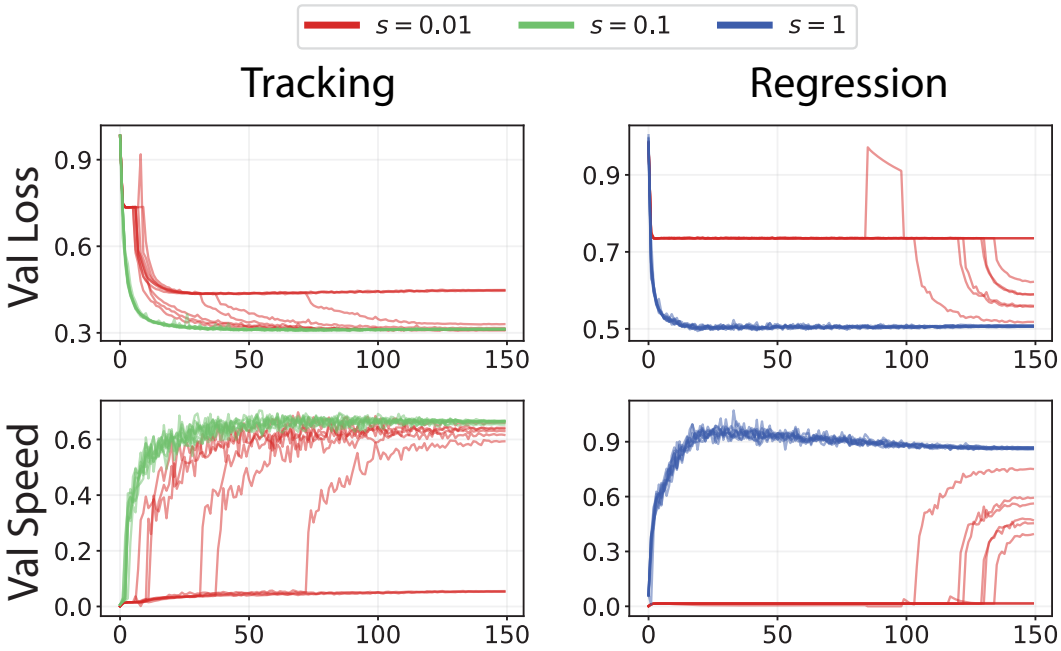

Figure A.5: **Validation trajectories reveal that inappropriate output scaling can trap position decoding models in low-movement solutions.** Validation loss (top) and mean predicted speed (bottom) across epochs for position decoding models trained on Tracking (left) and Regression (right) with different decoder output scalars s. Thin lines show individual seeds. With the default scalar used by Salter et al. (2024) ($s = 0.01$; red), training is often unstable and frequently converges to low-movement solutions, reflected in persistently low predicted speed. Increasing the scalar yields stable convergence for Tracking at $s = 0.1$ and for Regression at $s = 1$, indicating that position decoding performance is highly sensitive to output scaling.

is very close to a trivial Static baseline that simply holds the ground truth initial pose throughout the window (Table A.5).

We trained multiple seeds of single-task Pos models, which have an LSTM decoder module, using AdamW and Sivakumar et al.'s learning rate schedule (3.4). Although some seeds converged when using the default decoder output scalar $s = 0.01$, many collapsed to low-movement solutions (Figure A.5, red lines). Increasing this scalar to $s = 0.1$ for tracking and $s = 1$ for regression produced

consistent non-trivial learning, and allowed position decoding models to achieve competitive results (Table 1, Pos ST, Pos MT).

A.7 Other hyperparameters

We found that the multi-task loss weights $\lambda_{\text{tracking}} = 0.875$ and $\lambda_{\text{regression}} = 0.125$ consistently outperformed $\lambda_{\text{tracking}} = 0.75$ and $\lambda_{\text{regression}} = 0.25$ during validation. This places a relative importance of 7:1 for tracking vs. regression, such that tracking dominates the total training loss. We evaluated with separate tracking and regression checkpoints based on the respective validation losses on the 15 held-out validation users.

We use gradient clipping with a threshold of 1.0 for all models.

We list the LSTM scale hyperparameter values for Tables 1 and A.5 in Table A.4.

Table A.4: Hyperparameters of Pos, Vel, and ΔPos models with an LSTM decoder module (Tables 1, A.5). We remark that the LSTM scale could instead be learnable, thus reducing sensitivity of results to this hyperparameter, though we do not show it here.

Model	task	LSTM scale
Pos ST	tracking	0.1
Pos ST	regression	1.0
Pos MT	tracking, regression	0.1
Pos MT −FB	tracking, regression	0.1
Vel ST	tracking	0.01
Vel ST	regression	1.0
Vel MT	tracking, regression	0.01
Vel MT −FB	tracking, regression	0.01
ΔPos (all variations)	tracking, regression	0.1
PI-EMG (all variations)	tracking, regression	0.1

A.8 *emg2qwerty* training

We trained *emg2qwerty* decoding models by modifying Ortone et al.'s codebase. We additionally added a global gradient norm clip threshold of 1.0 to encourage stable training.

A.9 Prediction error across time and frequency

These analyses use the Pos/Vel ST and MT models, without McLayers or TFiLM. They quantify the errors underlying the benchmark scores rather than introduce a separate model-selection criterion.

Trajectories and within-window error The two rows of Figure A.6 use the same underlying recordings. The top row compares Pos ST and Vel ST on Tracking; the bottom compares Pos ST and Pos MT on Regression. Ground truth is the same, but the predictions differ because the tasks supply different initial-state information. Tracking Pos predictions show more rapid local fluctuations than Vel. Regression Pos MT has lower error and fewer local fluctuations than Pos ST in these examples.

Within-window AE is computed at each relative time in the existing 5 s evaluation windows, then aggregated across examples. It is not computed from new sliding windows. Tracking error generally rises with time from initialization, whereas Regression error decreases as more sEMG context becomes available. MT reduces Regression error more rapidly (Figure 3).

Residual spectra For each window and joint, define the residual $e_{n,j} = y_{n,j} - \hat{y}_{n,j}$. We take the magnitude of a one-sided real Fourier transform along the 5 s time dimension and average these magnitudes over windows and joints. If $C_A(f)$ is the resulting mean residual magnitude for model A, a pairwise comparison plots

$$D_{A,B}(f) = C_A(f) - C_B(f). \tag{7}$$

Positive values indicate larger residual magnitude for the first-named model. Figure A.7 shows the 0–10 Hz range. This compares residual amplitudes, rather than power or prediction smoothness.

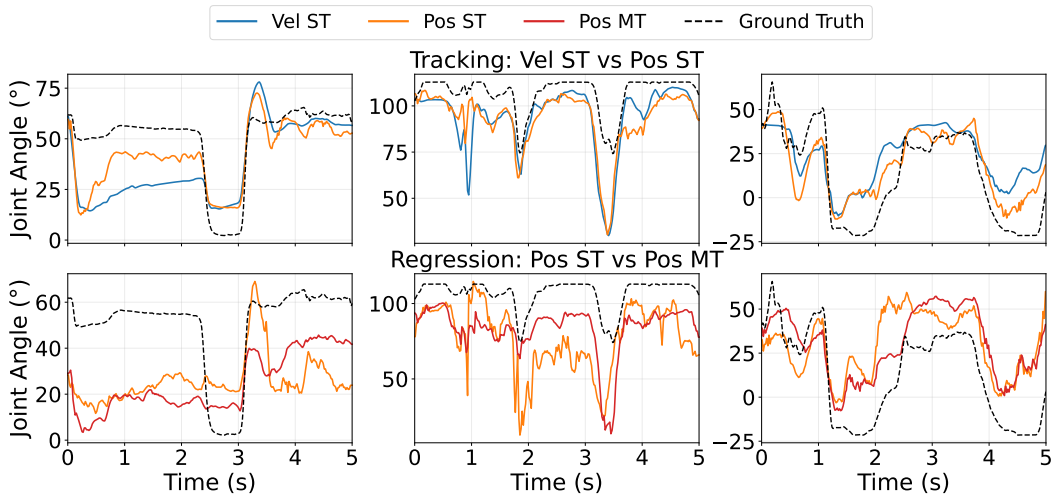

Figure A.6: Predictions on the same three example recordings. Top: Vel ST and Pos ST on Tracking. Bottom: Pos ST and Pos MT on Regression. Dashed black curves are ground truth. Only the models named for each row are plotted in that row.

Position decoding reduces low-frequency Tracking residual magnitudes while increasing higher-frequency residuals relative to velocity decoding. MT reduces Regression residual magnitudes over a broader frequency range. These observations help explain how a model can have lower overall error while retaining more high-frequency fluctuation.

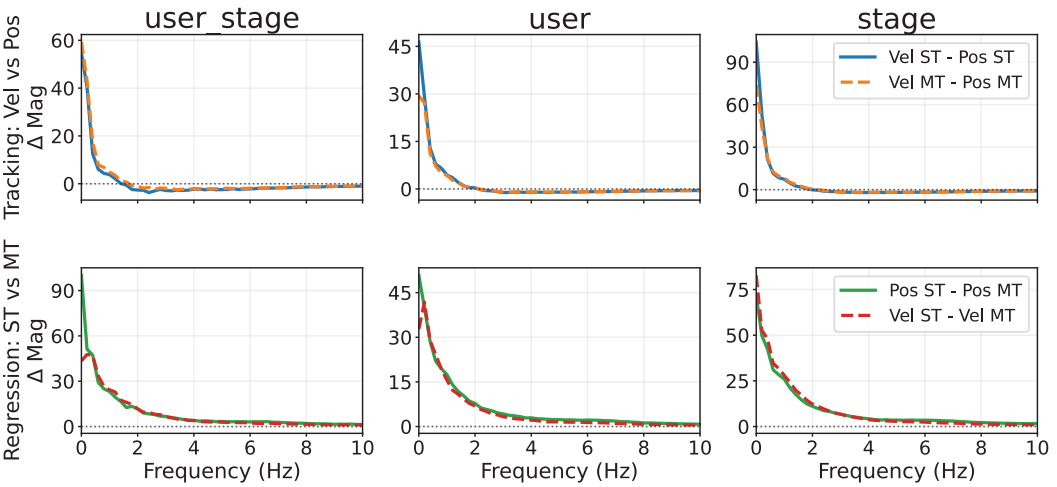

Figure A.7: Differences between mean residual-magnitude spectra, as defined in Equation 7. A positive difference favors the second-named model. The upper spectral row compares Vel with Pos on Tracking; the lower row compares ST with MT on Regression.

A.10 Adaptive filtering and the mean-speed–accuracy comparison

Filter and evaluation procedure For each joint, the filter combines the current unfiltered prediction x_t with its preceding filtered value $\tilde{x}_{t-1}$:

$$\tilde{x}_t = \alpha_t x_t + (1 - \alpha_t)\tilde{x}_{t-1}, \tag{8}$$

$$v_t = \frac{|x_t - \tilde{x}_{t-1}|}{T_e}, \qquad \alpha_t = \frac{2\pi\beta v_t T_e}{1 + 2\pi\beta v_t T_e}. \tag{9}$$

Here T_e is the sampling period and v_t is a speed-like discrepancy. The recurrence is a first-order low-pass filter whose coefficient changes over time. Setting cutoff $f_c = \beta v_t$ in $\alpha = T_e/(T_e + (2\pi f_c)^{-1})$ gives Equation 9. The update uses no future prediction. It is a simplified One Euro filter (Casiez et al., 2012).

We sweep $\beta \in \{0.1, 0.2, 0.3, 0.5, 1, 1.5, 2, 3, 4, 5, 7.5, 10\}$ for each model. Mean predicted speed is the mean absolute angular velocity over joints and time. Lower values reflect less predicted movement, which may result from reduced jitter or suppression of actual motion. We therefore report it together with AE and LD. The same β can produce different speeds for different models. Plotting error against achieved mean speed permits comparison at matched values of this proxy. It does not measure response delay or closed-loop usability.

Tracking and Regression results Unfiltered Pos trajectories have higher mean speed than Vel trajectories on Tracking. Filtering reduces mean speed, and Pos retains lower AE and LD over the overlapping speed range (Figure A.8, top). The raw smoothness difference therefore does not require choosing the less accurate decoder in this comparison.

On Regression, MT improves accuracy and lowers mean predicted speed for both output representations. Filtering ST predictions reduces their speed but does not recover the MT accuracy gain. At matched speed, MT Pos and Vel are close and both outperform ST models (Figure A.8, bottom). These results are specific to the Pos/Vel models tested; they are not measurements of filtered ΔPos or PI-EMG performance.

A.11 Additional results

We present several additional tables containing extended versions of tables and other related results.

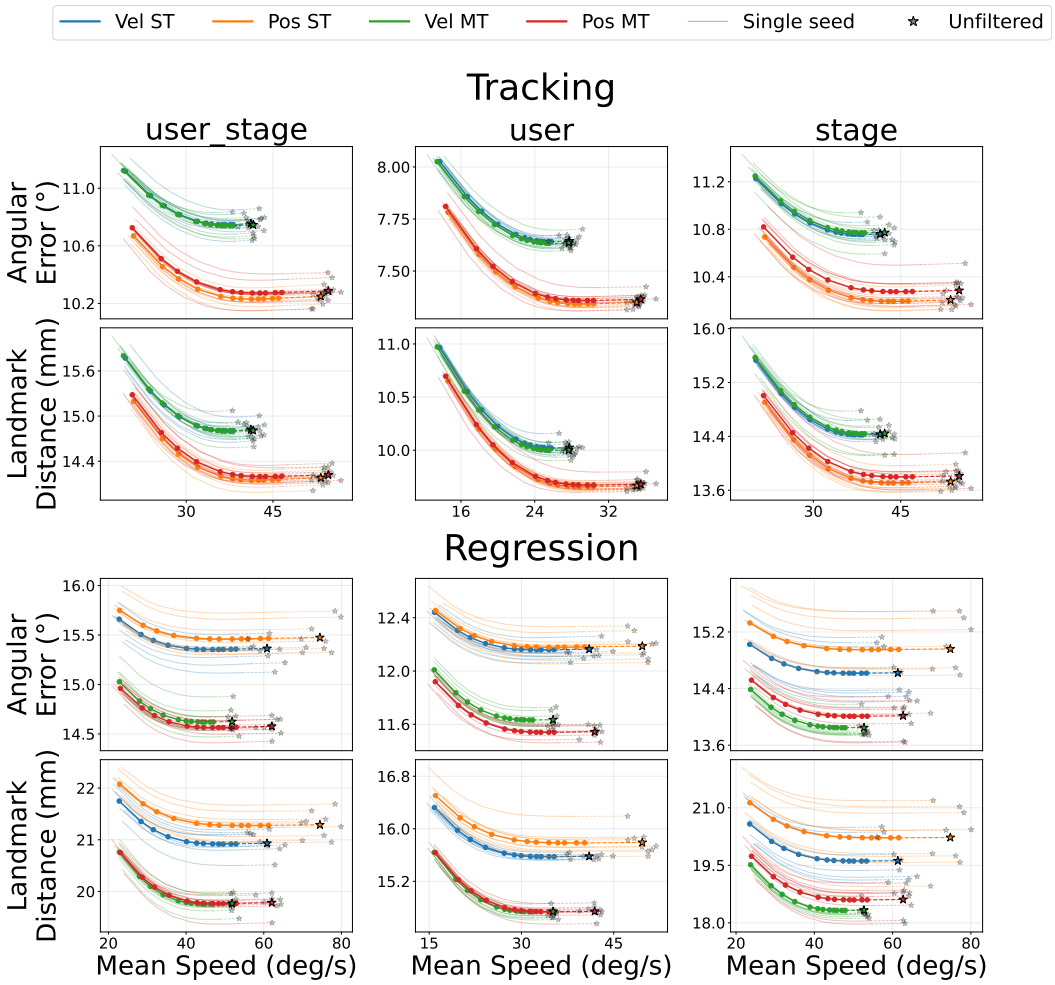

Figure A.8: AE and LD against mean predicted speed across the adaptive-filter sweep. Stars mark unfiltered predictions; thin lines show individual seeds. Top two rows: Tracking. Bottom two rows: Regression. Each point uses one value of β, but models are compared at achieved speed rather than at an identical filter parameter.

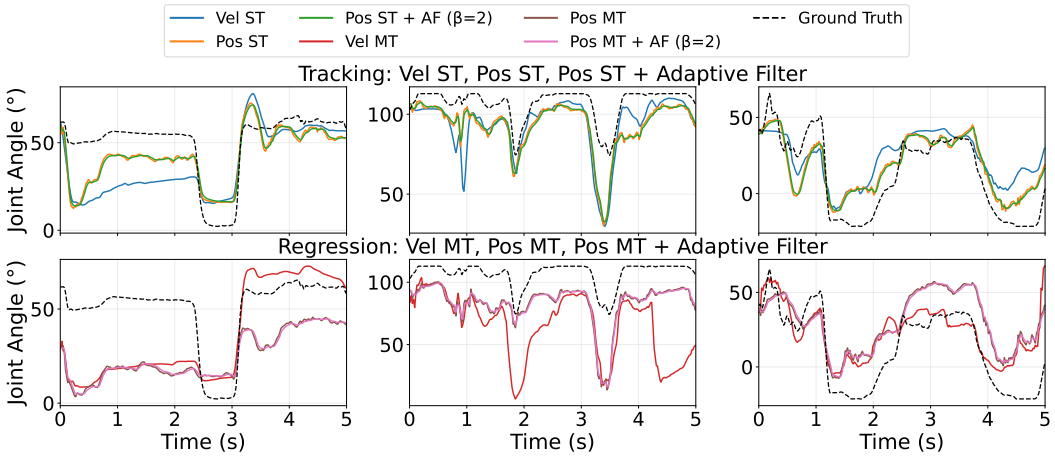

Figure A.9: Example trajectories before and after filtering with $\beta = 2$. Top: Tracking Vel ST, Pos ST, and filtered Pos ST. Bottom: Regression Vel MT, Pos MT, and filtered Pos MT. Dashed black curves are ground truth.

Table A.5: Extended version of Table 1. Tracking and regression task mean absolute joint angular error (AE, °) and landmark distance (LD, mm), mean ± sd across users, for single-task (ST), multi-task (MT), and feedback-ablated models (−FB). Feedback-ablation rows are multi-task models trained without auto-regressive feedback. ΔPos models were evaluated when the feedback was the output position (+FB) or the delta position (+FB-Δ), or with feedback ablation (−FB), and do not have persistent reference pose inputs. Static models use the initial pose (tracking) or per-joint median pose in the training set (regression).

	User-Stage		User		Stage	
Model	AE (°)	LD (mm)	AE (°)	LD (mm)	AE (°)	LD (mm)
Tracking						
Static	14.29 ± 1.86	21.48 ± 2.33	10.48 ± 1.81	15.71 ± 2.98	14.86 ± 1.91	21.81 ± 2.58
Salter et al.						
emg2pose	15.05 ± 1.22	20.69 ± 1.39	11.33 ± 1.10	14.75 ± 1.53	15.39 ± 1.50	20.88 ± 1.92
vemg2pose	11.03 ± 1.01	15.35 ± 1.46	7.70 ± 0.98	10.22 ± 1.54	11.20 ± 1.43	15.18 ± 1.81
Ours						
Vel ST	10.75 ± 0.93	14.82 ± 1.40	7.64 ± 0.94	10.02 ± 1.52	10.76 ± 1.34	14.43 ± 1.71
+MT	10.75 ± 0.97	14.81 ± 1.41	7.64 ± 0.95	10.00 ± 1.53	10.77 ± 1.36	14.44 ± 1.75
−FB	10.89 ± 0.97	14.97 ± 1.38	7.71 ± 0.93	10.09 ± 1.51	10.85 ± 1.34	14.49 ± 1.71
Pos ST	10.25 ± 0.93	14.18 ± 1.35	7.35 ± 0.92	9.67 ± 1.48	10.21 ± 1.32	13.73 ± 1.69
+MT	10.29 ± 0.97	14.22 ± 1.37	7.36 ± 0.90	9.68 ± 1.46	10.28 ± 1.33	13.81 ± 1.70
−FB	10.26 ± 0.98	14.21 ± 1.38	7.34 ± 0.90	9.66 ± 1.45	10.28 ± 1.35	13.82 ± 1.73
ΔPos MT, −FB	10.12 ± 0.98	14.02 ± 1.39	**7.27** ± 0.91	**9.55** ± 1.47	**10.10** ± 1.36	**13.56** ± 1.75
+FB	10.20 ± 1.02	14.15 ± 1.41	**7.27** ± 0.92	9.60 ± 1.51	10.25 ± 1.38	13.81 ± 1.77
+FB-Δ	**10.11** ± 1.00	**13.99** ± 1.34	7.28 ± 0.91	9.60 ± 1.47	10.18 ± 1.36	13.67 ± 1.73
Regression						
Static	18.87 ± 2.01	28.77 ± 2.78	16.86 ± 1.80	25.17 ± 2.84	19.18 ± 1.71	28.57 ± 2.46
Salter et al.						
emg2pose	15.57 ± 1.30	21.49 ± 1.71	12.57 ± 1.30	16.28 ± 1.81	15.17 ± 1.59	20.53 ± 2.13
vemg2pose	15.63 ± 1.37	21.30 ± 1.87	12.24 ± 1.34	15.82 ± 1.90	15.22 ± 1.58	20.38 ± 2.07
Ours						
Vel ST	15.35 ± 1.32	20.92 ± 1.87	12.17 ± 1.31	15.58 ± 1.85	14.62 ± 1.65	19.62 ± 2.21
+MT	14.63 ± 1.34	19.77 ± 1.95	11.64 ± 1.25	14.74 ± 1.74	13.85 ± 1.62	18.33 ± 2.16
−FB	14.98 ± 1.30	20.16 ± 1.95	11.91 ± 1.19	15.07 ± 1.64	14.28 ± 1.56	18.78 ± 2.05
Pos ST	15.47 ± 1.30	21.29 ± 1.82	12.19 ± 1.27	15.79 ± 1.80	14.96 ± 1.63	20.23 ± 2.15
+MT	14.58 ± 1.29	19.78 ± 1.83	11.54 ± 1.19	14.74 ± 1.64	14.02 ± 1.62	18.61 ± 2.16
−FB	**14.47** ± 1.25	19.69 ± 1.87	**11.51** ± 1.19	**14.66** ± 1.63	14.02 ± 1.61	18.64 ± 2.16
ΔPos MT, −FB	14.49 ± 1.29	**19.60** ± 1.94	11.57 ± 1.20	14.68 ± 1.60	**13.89** ± 1.63	**18.31** ± 2.16
+FB	14.55 ± 1.29	19.74 ± 1.90	11.61 ± 1.22	14.75 ± 1.65	13.91 ± 1.67	18.43 ± 2.22
+FB-Δ	14.64 ± 1.39	19.89 ± 2.00	11.60 ± 1.24	14.84 ± 1.67	14.14 ± 1.64	18.76 ± 2.17

Table A.6: Results for multi-task (MT) models that predict the delta-position from initial pose (ΔPos) for varying forms of LSTM feedback (FB). FB inputs the predicted position, i.e. initial position + Δ, as feedback. FB-Δ inputs the most recently predicted Δ value itself. The $-$FB model, i.e. feedback ablation, inputs the learned position embedding on non-initial time-steps. These models do not use persistent reference pose inputs.

	User-Stage		User		Stage	
Model	AE (°)	LD (mm)	AE (°)	LD (mm)	AE (°)	LD (mm)
			Tracking			
ΔPos MT, FB	$10.20_{\pm 1.02}$	$14.15_{\pm 1.41}$	**7.27**$_{\pm 0.92}$	$9.60_{\pm 1.51}$	$10.25_{\pm 1.38}$	$13.81_{\pm 1.77}$
ΔPos MT, FB-Δ	**10.11**$_{\pm 1.00}$	**13.99**$_{\pm 1.34}$	$7.28_{\pm 0.91}$	$9.60_{\pm 1.47}$	$10.18_{\pm 1.36}$	$13.67_{\pm 1.73}$
ΔPos MT, $-$FB	$10.12_{\pm 0.98}$	$14.02_{\pm 1.39}$	**7.27**$_{\pm 0.91}$	**9.55**$_{\pm 1.47}$	**10.10**$_{\pm 1.36}$	**13.56**$_{\pm 1.75}$
			Regression			
ΔPos MT, FB	$14.55_{\pm 1.29}$	$19.74_{\pm 1.90}$	$11.61_{\pm 1.22}$	$14.75_{\pm 1.65}$	$13.91_{\pm 1.67}$	$18.43_{\pm 2.22}$
ΔPos MT, FB-Δ	$14.64_{\pm 1.39}$	$19.89_{\pm 2.00}$	$11.60_{\pm 1.24}$	$14.84_{\pm 1.67}$	$14.14_{\pm 1.64}$	$18.76_{\pm 2.17}$
ΔPos MT, $-$FB	**14.49**$_{\pm 1.29}$	**19.60**$_{\pm 1.94}$	**11.57**$_{\pm 1.20}$	**14.68**$_{\pm 1.60}$	**13.89**$_{\pm 1.63}$	**18.31**$_{\pm 2.16}$

Table A.7: Tracking and regression MAE angular error (AE, °) and mean landmark distance (LD, mm) for ΔPos models without persistent reference pose inputs. User-side McLayers were placed after the inputs, applied to electrode signals (+eU), after the CNN features (+fU), or at both locations (+U). Sample McLayers were placed after the CNN or after the +fU layer (+S). The Salter et al. (2024) results display their best performance (A.3).

	User-Stage		User		Stage	
Model	AE (°)	LD (mm)	AE (°)	LD (mm)	AE (°)	LD (mm)
			Tracking			
Salter et al.	$11.03_{\pm 1.01}$	$15.35_{\pm 1.46}$	$7.70_{\pm 0.98}$	$10.22_{\pm 1.54}$	$11.20_{\pm 1.43}$	$15.18_{\pm 1.81}$
ΔPos	$10.12_{\pm 0.98}$	$14.02_{\pm 1.39}$	$7.27_{\pm 0.91}$	$9.55_{\pm 1.47}$	$10.10_{\pm 1.36}$	$13.56_{\pm 1.75}$
+eU	$9.98_{\pm 0.98}$	$13.77_{\pm 1.33}$	$7.21_{\pm 0.90}$	$9.47_{\pm 1.46}$	$10.08_{\pm 1.35}$	$13.54_{\pm 1.72}$
+fU	$9.84_{\pm 0.97}$	$13.58_{\pm 1.32}$	$7.11_{\pm 0.88}$	$9.32_{\pm 1.40}$	$9.98_{\pm 1.34}$	$13.37_{\pm 1.72}$
+U	$9.75_{\pm 0.97}$	$13.41_{\pm 1.28}$	$7.08_{\pm 0.86}$	$9.25_{\pm 1.38}$	$9.95_{\pm 1.33}$	$13.34_{\pm 1.70}$
+S	$9.83_{\pm 0.94}$	$13.54_{\pm 1.28}$	$7.11_{\pm 0.89}$	$9.29_{\pm 1.44}$	$9.81_{\pm 1.31}$	$13.11_{\pm 1.66}$
+U +S	$9.72_{\pm 0.92}$	$13.44_{\pm 1.24}$	$7.08_{\pm 0.85}$	$9.28_{\pm 1.40}$	$9.96_{\pm 1.34}$	$13.41_{\pm 1.70}$
			Regression			
Salter et al.	$15.57_{\pm 1.30}$	$21.30_{\pm 1.87}$	$12.24_{\pm 1.34}$	$15.82_{\pm 1.90}$	$15.17_{\pm 1.59}$	$20.38_{\pm 2.07}$
ΔPos	$14.49_{\pm 1.29}$	$19.60_{\pm 1.94}$	$11.57_{\pm 1.20}$	$14.68_{\pm 1.60}$	$13.89_{\pm 1.63}$	$18.31_{\pm 2.16}$
+eU	$14.23_{\pm 1.28}$	$19.15_{\pm 1.83}$	$11.44_{\pm 1.22}$	$14.43_{\pm 1.69}$	$13.61_{\pm 1.65}$	$17.93_{\pm 2.21}$
+fU	$14.14_{\pm 1.32}$	$19.10_{\pm 1.93}$	$11.31_{\pm 1.15}$	$14.32_{\pm 1.54}$	$13.85_{\pm 1.62}$	$18.31_{\pm 2.14}$
+U	$13.93_{\pm 1.31}$	$18.67_{\pm 1.77}$	$11.26_{\pm 1.14}$	$14.19_{\pm 1.50}$	$13.84_{\pm 1.65}$	$18.19_{\pm 2.22}$
+S	$14.16_{\pm 1.32}$	$19.16_{\pm 1.87}$	$11.35_{\pm 1.18}$	$14.39_{\pm 1.55}$	$13.79_{\pm 1.57}$	$18.17_{\pm 2.08}$
+U +S	$14.13_{\pm 1.28}$	$19.08_{\pm 1.81}$	$11.31_{\pm 1.01}$	$14.32_{\pm 1.39}$	$14.20_{\pm 1.74}$	$18.71_{\pm 2.24}$

Table A.8: Comparison of MAE angular error (AE, °) and mean landmark distance (LD, mm) for PI-EMG models when the reference pose is persistently input (standard PI-EMG) or omitted. The persistent reference pose input improves longer horizon decoding (> 20 s), which is not tested by the standard *emg2pose* task (Figure A.3). The PI-EMG-Session model uses McLayers conditioned on session ID, where session McLayers replace the user-side McLayers of PI-EMG-U.

	User-Stage		User		Stage	
Model	AE (°)	LD (mm)	AE (°)	LD (mm)	AE (°)	LD (mm)
Tracking						
PI-EMG (without persistent reference pose)						
ΔPos	$10.12_{\pm 0.98}$	$14.02_{\pm 1.39}$	$7.27_{\pm 0.91}$	$9.55_{\pm 1.47}$	$10.10_{\pm 1.36}$	$13.56_{\pm 1.75}$
ΔPos, TFiLM	$9.92_{\pm 0.97}$	$13.66_{\pm 1.34}$	$7.21_{\pm 0.91}$	$9.44_{\pm 1.51}$	$9.89_{\pm 1.35}$	$13.18_{\pm 1.74}$
PI-EMG-U	$9.57_{\pm 0.90}$	$13.10_{\pm 1.24}$	$7.00_{\pm 0.85}$	$9.10_{\pm 1.40}$	$9.75_{\pm 1.31}$	$13.00_{\pm 1.68}$
PI-EMG-S	$9.63_{\pm 0.93}$	$13.22_{\pm 1.30}$	$7.02_{\pm 0.87}$	$9.15_{\pm 1.43}$	$\mathbf{9.62}_{\pm 1.31}$	$\mathbf{12.82}_{\pm 1.68}$
PI-EMG-US	$\mathbf{9.45}_{\pm 0.90}$	$\mathbf{12.95}_{\pm 1.28}$	$\mathbf{6.95}_{\pm 0.83}$	$\mathbf{9.05}_{\pm 1.35}$	$9.66_{\pm 1.30}$	$12.86_{\pm 1.64}$
PI-EMG (with persistent reference pose)						
ΔPos	$10.17_{\pm 1.01}$	$14.08_{\pm 1.38}$	$7.30_{\pm 0.91}$	$9.63_{\pm 1.49}$	$10.13_{\pm 1.34}$	$13.61_{\pm 1.71}$
ΔPos, TFiLM	$9.97_{\pm 1.00}$	$13.77_{\pm 1.38}$	$7.20_{\pm 0.92}$	$9.45_{\pm 1.49}$	$9.95_{\pm 1.34}$	$13.32_{\pm 1.70}$
PI-EMG-Session	$9.57_{\pm 0.87}$	$13.10_{\pm 1.24}$	$7.01_{\pm 0.87}$	$9.12_{\pm 1.42}$	$9.69_{\pm 1.30}$	$12.90_{\pm 1.64}$
PI-EMG-U	$9.53_{\pm 0.90}$	$13.07_{\pm 1.27}$	$6.99_{\pm 0.86}$	$9.10_{\pm 1.40}$	$9.65_{\pm 1.30}$	$12.86_{\pm 1.65}$
PI-EMG-S	$9.68_{\pm 0.93}$	$13.28_{\pm 1.32}$	$7.02_{\pm 0.86}$	$9.14_{\pm 1.41}$	$9.61_{\pm 1.29}$	$12.77_{\pm 1.65}$
PI-EMG-US	$\mathbf{9.42}_{\pm 0.90}$	$\mathbf{12.83}_{\pm 1.26}$	$\mathbf{6.93}_{\pm 0.84}$	$\mathbf{8.99}_{\pm 1.36}$	$\mathbf{9.59}_{\pm 1.30}$	$\mathbf{12.71}_{\pm 1.63}$
Regression						
PI-EMG (without persistent reference pose)						
ΔPos	$14.49_{\pm 1.29}$	$19.60_{\pm 1.94}$	$11.57_{\pm 1.20}$	$14.68_{\pm 1.60}$	$13.89_{\pm 1.63}$	$18.31_{\pm 2.16}$
ΔPos, TFiLM	$14.22_{\pm 1.35}$	$19.11_{\pm 2.00}$	$11.36_{\pm 1.20}$	$14.40_{\pm 1.68}$	$13.76_{\pm 1.67}$	$18.00_{\pm 2.20}$
PI-EMG-U	$\mathbf{13.57}_{\pm 1.25}$	$\mathbf{18.02}_{\pm 1.79}$	$11.11_{\pm 1.08}$	$\mathbf{13.87}_{\pm 1.50}$	$13.42_{\pm 1.59}$	$17.48_{\pm 2.10}$
PI-EMG-S	$13.83_{\pm 1.29}$	$18.56_{\pm 1.96}$	$11.12_{\pm 1.14}$	$14.02_{\pm 1.60}$	$\mathbf{13.38}_{\pm 1.58}$	$\mathbf{17.43}_{\pm 2.06}$
PI-EMG-US	$13.67_{\pm 1.24}$	$18.30_{\pm 1.86}$	$\mathbf{11.07}_{\pm 1.02}$	$13.90_{\pm 1.40}$	$13.67_{\pm 1.64}$	$17.85_{\pm 2.10}$
PI-EMG (with persistent reference pose)						
ΔPos	$14.56_{\pm 1.39}$	$19.71_{\pm 2.11}$	$11.63_{\pm 1.28}$	$14.77_{\pm 1.73}$	$13.86_{\pm 1.67}$	$18.30_{\pm 2.22}$
ΔPos, TFiLM	$14.29_{\pm 1.39}$	$19.24_{\pm 2.05}$	$11.38_{\pm 1.18}$	$14.37_{\pm 1.58}$	$13.69_{\pm 1.63}$	$18.00_{\pm 2.15}$
PI-EMG-Session	$13.53_{\pm 1.26}$	$18.00_{\pm 1.89}$	$11.06_{\pm 1.08}$	$13.83_{\pm 1.46}$	$13.35_{\pm 1.57}$	$17.37_{\pm 2.05}$
PI-EMG-U	$\mathbf{13.49}_{\pm 1.32}$	$\mathbf{17.95}_{\pm 1.88}$	$11.03_{\pm 1.10}$	$\mathbf{13.76}_{\pm 1.48}$	$13.28_{\pm 1.61}$	$17.28_{\pm 2.10}$
PI-EMG-S	$13.75_{\pm 1.36}$	$18.41_{\pm 2.04}$	$11.11_{\pm 1.15}$	$13.93_{\pm 1.52}$	$\mathbf{13.16}_{\pm 1.57}$	$\mathbf{17.12}_{\pm 2.07}$
PI-EMG-US	$13.54_{\pm 1.29}$	$18.09_{\pm 1.93}$	$\mathbf{11.01}_{\pm 1.02}$	$13.81_{\pm 1.37}$	$13.51_{\pm 1.63}$	$17.64_{\pm 2.10}$

Table A.9: Performance for ΔPos models with TFiLM layers (+T) whose user McLayer generic mappings were set to identity (+U-identity), and when the hidden dimension of the decoder LSTM was set to 476, such that its FLOP count is reduced by the FLOP count of the two GRU TFiLM layers (+$d_{\mathrm{h}} = 476$). Each GRU TFiLM layer has $\approx 6.6\%$ the floating point operation (FLOP) count of the 512-dimensional two-layer decoder LSTM. Results in this table do not use persistent reference pose inputs. The +$d_{\mathrm{h}} = 476$ conditions did not perform significantly differently from their ΔPos +U +TFiLM and ΔPos +U +S +TFiLM counterparts ($p > 0.06$ for all comparisons, paired Wilcoxon signed rank test).

	User-Stage		User		Stage	
Model	AE (°)	LD (mm)	AE (°)	LD (mm)	AE (°)	LD (mm)
Tracking						
ΔPos +U +T	$9.57_{\pm 0.90}$	$13.10_{\pm 1.24}$	$7.00_{\pm 0.85}$	$9.10_{\pm 1.40}$	$9.75_{\pm 1.31}$	$13.00_{\pm 1.68}$
+U-identity	$9.57_{\pm 0.89}$	$13.09_{\pm 1.24}$	$6.98_{\pm 0.87}$	$9.07_{\pm 1.41}$	$9.72_{\pm 1.31}$	$12.94_{\pm 1.67}$
+$d_{\mathrm{h}} = 476$	$9.55_{\pm 0.89}$	$13.08_{\pm 1.26}$	$6.99_{\pm 0.84}$	$9.09_{\pm 1.37}$	$9.73_{\pm 1.33}$	$12.98_{\pm 1.70}$
ΔPos +U +S +T	$9.45_{\pm 0.90}$	$12.95_{\pm 1.28}$	$6.95_{\pm 0.83}$	$9.05_{\pm 1.35}$	$9.66_{\pm 1.30}$	$12.86_{\pm 1.64}$
+U-identity	$9.47_{\pm 0.88}$	$12.92_{\pm 1.28}$	$6.96_{\pm 0.83}$	$9.02_{\pm 1.35}$	$9.66_{\pm 1.31}$	$12.85_{\pm 1.66}$
+$d_{\mathrm{h}} = 476$	$9.46_{\pm 0.91}$	$12.91_{\pm 1.32}$	$6.96_{\pm 0.84}$	$9.03_{\pm 1.38}$	$9.66_{\pm 1.31}$	$12.84_{\pm 1.66}$
Regression						
ΔPos +U +T	$13.57_{\pm 1.25}$	$18.02_{\pm 1.79}$	$11.11_{\pm 1.08}$	$13.87_{\pm 1.50}$	$13.42_{\pm 1.59}$	$17.48_{\pm 2.10}$
+U-identity	$13.54_{\pm 1.29}$	$18.02_{\pm 1.84}$	$11.05_{\pm 1.08}$	$13.83_{\pm 1.53}$	$13.42_{\pm 1.63}$	$17.48_{\pm 2.16}$
+$d_{\mathrm{h}} = 476$	$13.60_{\pm 1.28}$	$18.05_{\pm 1.85}$	$11.09_{\pm 1.07}$	$13.84_{\pm 1.42}$	$13.42_{\pm 1.61}$	$17.49_{\pm 2.11}$
ΔPos +U +S +T	$13.67_{\pm 1.24}$	$18.30_{\pm 1.86}$	$11.07_{\pm 1.02}$	$13.90_{\pm 1.40}$	$13.67_{\pm 1.64}$	$17.85_{\pm 2.10}$
+U-identity	$13.65_{\pm 1.27}$	$18.24_{\pm 1.89}$	$11.07_{\pm 1.02}$	$13.89_{\pm 1.35}$	$13.66_{\pm 1.64}$	$17.90_{\pm 2.12}$
+$d_{\mathrm{h}} = 476$	$13.65_{\pm 1.19}$	$18.26_{\pm 1.77}$	$11.03_{\pm 1.01}$	$13.85_{\pm 1.38}$	$13.66_{\pm 1.64}$	$17.83_{\pm 2.13}$

Table A.10: Performance for PI-EMG-U without persistent reference pose inputs when the generic probability p_g is varied. Our main results use $p_g = 0.25$.

	User-Stage		User		Stage	
Model	AE (°)	LD (mm)	AE (°)	LD (mm)	AE (°)	LD (mm)
Tracking						
$p_g = 0.01$	$10.00_{\pm 0.96}$	$13.77_{\pm 1.34}$	$7.25_{\pm 0.87}$	$9.54_{\pm 1.40}$	$10.59_{\pm 1.53}$	$14.28_{\pm 1.94}$
$p_g = 0.125$	$\mathbf{9.55}_{\pm 0.90}$	$13.11_{\pm 1.24}$	$7.01_{\pm 0.85}$	$9.15_{\pm 1.39}$	$9.86_{\pm 1.34}$	$13.20_{\pm 1.72}$
$p_g = 0.25$	$9.57_{\pm 0.90}$	$\mathbf{13.10}_{\pm 1.24}$	$\mathbf{7.00}_{\pm 0.85}$	$\mathbf{9.10}_{\pm 1.40}$	$9.75_{\pm 1.31}$	$13.00_{\pm 1.68}$
$p_g = 0.5$	$9.68_{\pm 0.93}$	$13.27_{\pm 1.31}$	$7.06_{\pm 0.87}$	$9.20_{\pm 1.43}$	$\mathbf{9.71}_{\pm 1.31}$	$\mathbf{12.98}_{\pm 1.69}$
Regression						
$p_g = 0.01$	$14.22_{\pm 1.43}$	$19.23_{\pm 2.16}$	$11.51_{\pm 1.03}$	$14.65_{\pm 1.39}$	$14.86_{\pm 2.31}$	$19.70_{\pm 2.83}$
$p_g = 0.125$	$13.73_{\pm 1.26}$	$18.39_{\pm 1.84}$	$11.15_{\pm 1.01}$	$14.07_{\pm 1.40}$	$13.85_{\pm 1.69}$	$18.20_{\pm 2.22}$
$p_g = 0.25$	$\mathbf{13.57}_{\pm 1.25}$	$\mathbf{18.02}_{\pm 1.79}$	$\mathbf{11.11}_{\pm 1.08}$	$\mathbf{13.87}_{\pm 1.50}$	$13.42_{\pm 1.59}$	$17.48_{\pm 2.10}$
$p_g = 0.5$	$13.78_{\pm 1.31}$	$18.34_{\pm 1.93}$	$11.17_{\pm 1.15}$	$13.96_{\pm 1.59}$	$\mathbf{13.28}_{\pm 1.60}$	$\mathbf{17.31}_{\pm 2.12}$

Table A.11: Stage-generalization results for ΔPos models using user-conditioned parameters when user-hand McLayers are applied as electrode mappings (+eU), feature mappings (+fU), or both (+U). Additional configurations use sample-conditioned McLayers during training (+S) and causal temporal FiLM layers (+TFiLM). Results in this table do not use persistent reference pose inputs.

	Stage (generic)		Stage (personalized)	
Model	AE (°)	LD (mm)	AE (°)	LD (mm)
Tracking				
ΔPos	$10.10_{\pm 1.36}$	$13.56_{\pm 1.75}$	—	—
+eU	$10.08_{\pm 1.35}$	$13.54_{\pm 1.72}$	$9.65_{\pm 1.29}$	$12.84_{\pm 1.65}$
+fU	$9.98_{\pm 1.34}$	$13.37_{\pm 1.72}$	$9.94_{\pm 1.41}$	$13.37_{\pm 1.90}$
+U	$9.95_{\pm 1.33}$	$13.34_{\pm 1.70}$	$9.85_{\pm 1.40}$	$13.25_{\pm 1.92}$
+U: $p_g = 0.01$, +TFiLM	$10.59_{\pm 1.53}$	$14.28_{\pm 1.94}$	$10.18_{\pm 1.46}$	$13.71_{\pm 2.01}$
+U: $p_g = 0.125$, +TFiLM	$9.86_{\pm 1.34}$	$13.20_{\pm 1.72}$	$9.98_{\pm 1.43}$	$13.42_{\pm 1.94}$
+U, +TFiLM ($p_g = 0.25$)	$9.75_{\pm 1.31}$	$13.00_{\pm 1.68}$	$9.72_{\pm 1.39}$	$13.05_{\pm 1.89}$
+U: $p_g = 0.5$, +TFiLM	$9.71_{\pm 1.31}$	$12.98_{\pm 1.69}$	$9.64_{\pm 1.36}$	$12.96_{\pm 1.83}$
+S	$9.81_{\pm 1.31}$	$13.11_{\pm 1.66}$	—	—
+U, +S	$9.96_{\pm 1.34}$	$13.41_{\pm 1.70}$	$9.43_{\pm 1.34}$	$12.69_{\pm 1.82}$
+U, +S, +TFiLM	$\mathbf{9.66}_{\pm 1.30}$	$\mathbf{12.86}_{\pm 1.64}$	$\mathbf{9.20}_{\pm 1.33}$	$\mathbf{12.33}_{\pm 1.80}$
Regression				
ΔPos	$13.89_{\pm 1.63}$	$18.31_{\pm 2.16}$	—	—
+eU	$13.61_{\pm 1.65}$	$17.93_{\pm 2.21}$	$12.60_{\pm 1.62}$	$16.47_{\pm 2.22}$
+fU	$13.85_{\pm 1.62}$	$18.31_{\pm 2.14}$	$12.39_{\pm 1.77}$	$16.58_{\pm 2.45}$
+U	$13.84_{\pm 1.65}$	$18.19_{\pm 2.22}$	$12.13_{\pm 1.74}$	$16.19_{\pm 2.42}$
+U: $p_g = 0.01$, +TFiLM	$14.86_{\pm 2.31}$	$19.70_{\pm 2.83}$	$12.24_{\pm 1.77}$	$16.36_{\pm 2.44}$
+U: $p_g = 0.125$, +TFiLM	$13.85_{\pm 1.69}$	$18.20_{\pm 2.22}$	$12.17_{\pm 1.77}$	$16.28_{\pm 2.43}$
+U, +TFiLM ($p_g = 0.25$)	$13.42_{\pm 1.59}$	$17.48_{\pm 2.10}$	$11.98_{\pm 1.73}$	$15.96_{\pm 2.40}$
+U: $p_g = 0.5$, +TFiLM	$\mathbf{13.28}_{\pm 1.60}$	$\mathbf{17.31}_{\pm 2.12}$	$12.08_{\pm 1.72}$	$16.07_{\pm 2.37}$
+S	$13.79_{\pm 1.57}$	$18.17_{\pm 2.08}$	—	—
+U, +S	$14.20_{\pm 1.74}$	$18.71_{\pm 2.24}$	$12.02_{\pm 1.71}$	$16.03_{\pm 2.35}$
+U, +S, +TFiLM	$13.67_{\pm 1.64}$	$17.85_{\pm 2.10}$	$\mathbf{11.81}_{\pm 1.69}$	$\mathbf{15.72}_{\pm 2.31}$