MARC: Asymmetric Memory Coupling for Myoelectric Hand Motion Decoding

Anonymous authors
Paper under double-blind review

Abstract

Streaming hand motion decoding from surface electromyography (EMG) benefits from motion history, but feeding predicted poses back into memory can also propagate their errors. We study how an explicit motion predictor should interact with persistent sensor memory. MARC combines joint-conditioned EMG encoding, observation-driven memory updates, and pose-conditioned motion prediction. Predictions contribute to the decoded pose without writing back into observation memory, giving the two state pathways an asymmetric dependency. Across emg2pose, MANUS, and NinaPro DB8, MARC improves mean reconstruction accuracy over reproduced TDS+LSTM and EMGFormer-L baselines. On emg2pose, it reduces angular error by 9.5–19.3% and fingertip error by 13.1–21.6% relative to TDS+LSTM, while improving displacement accuracy and motion direction. Matched feedback controls and complementary state interventions reveal a trade-off: restricting pose feedback reduces the output response to an injected pose error, but slows recovery of the unperturbed output after observation-memory erasure. Together, these results establish the practical value of MARC and identify prediction-to-memory feedback as a testable design choice in continuous myoelectric decoding.

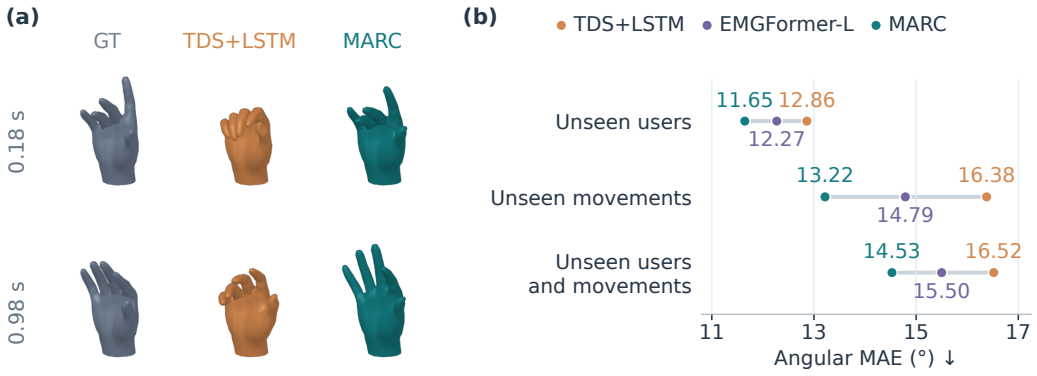

Figure 1: **Recovering hand configuration from EMG.** (a) A selected test configuration-recovery example: MARC more closely recovers the reference configuration, while TDS+LSTM retains excessive finger flexion. Poses share the same hand geometry, camera, and scale. (b) Angular MAE on the complete emg2pose test axes; lower is better. Appendix H provides the full example and selection rule.

1 Introduction

Hand motion provides a natural interface for interacting with digital and physical environments. Wrist electromyography offers access to this motion without requiring a camera view of the hand, and recent systems demonstrate its promise for continuous pose decoding and population-level interaction (Liu et al., 2021; Kaifosh et al., 2025; Salter et al., 2024). The underlying inference problem

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.

remains difficult: muscle activity is an indirect measurement of hand configuration. A decoder must infer where the fingers are, how they are moving, and how to update that estimate as new signals arrive.

Current and recent EMG provide evidence about pose, while movement history constrains plausible changes. Strong sequence decoders learn both implicitly (Salter et al., 2024; Xi et al., 2026); learned motion representations provide a complementary way to organize hand kinematics (Lovecchio et al., 2025; Verma, 2025). Making motion prediction explicit raises a question about how it should interact with sensing: *should a decoder's own pose estimates also update the memory used to interpret subsequent observations?*

Using a prediction for the current output and writing it into memory create different dependencies. A stored pose can supply context after observation memory is disrupted, but feeding it into that memory also opens a path for pose errors to affect later sensor updates. Learned filtering and selective memory updates provide foundations for combining prediction with new evidence (Fraccaro et al., 2017; Becker et al., 2019; Gu & Dao, 2024; Yang et al., 2025). Here we examine the write access given to predicted poses within a decoder that explicitly maintains both sensor memory and a motion estimate.

We introduce MARC, a joint-structured decoder with **asymmetric memory coupling**. Joint-conditioned EMG observations update persistent memory. A separate motion branch reads that memory and the preceding pose estimate to predict motion; its result contributes to the output without being written back into observation memory. The two operations share joint-attention parameters, and a joint-wise readout combines their estimates. This interface lets the decoder use an explicit motion prediction while preserving an observation-memory update that does not depend on its preceding pose estimate.

We evaluate both the complete decoder and its memory organization. Across emg2pose, MANUS, and NinaPro DB8, MARC improves mean reconstruction accuracy over the reproduced TDS+LSTM and EMGFormer-L baselines on every evaluated test setting. The emg2pose gains extend from joint angles to fingertip geometry and motion reconstruction. Matched feedback controls then test the memory interface within the same architecture. Pose perturbation and observation-memory erasure produce opposite responses: feedback prolongs the influence of an erroneous pose, yet helps restore the unperturbed output after memory loss. These complementary experiments establish decoding performance and expose the consequences of the feedback route.

Our contributions are:

- **A structured interface between sensing and prediction.** MARC combines joint-conditioned sensor memory with an explicit motion predictor, allowing predicted poses to affect the output while restricting their access to persistent observation memory.
- **Mechanistic evidence about prediction feedback.** Matched input-source controls, complementary state interventions, and a dependency analysis distinguish pose-error propagation from recovery after observation-memory loss.
- **Decoding gains across recording conditions.** Comparisons on three public datasets establish whole-model accuracy gains; fingertip, displacement, direction, and participant- and joint-resolved analyses identify their spatial and temporal extent.

2 Related work

Continuous myoelectric decoding. NeuroPose and emg2pose establish continuous hand-pose estimation from wearable EMG, while population-level neuromotor interfaces demonstrate its broader potential (Liu et al., 2021; Salter et al., 2024; Kaifosh et al., 2025). Sequence decoders learn the signal-to-motion relationship through temporal convolution and recurrence, as in TDS+LSTM, or attention, as in EMGFormer (Hannun et al., 2019; Xi et al., 2026). A complementary line organizes the output space: VQ-MyoPose learns discrete movement representations, and emg2tendon uses generative decoding with musculoskeletal structure (Lovecchio et al., 2025; Verma, 2025). MARC combines configuration evidence and learned motion transitions within a continuous estimator, evaluated through angular accuracy, hand geometry, and reconstructed movement.

Memory and prediction in learned sequence models. Prediction followed by observation-based correction is a longstanding principle of state estimation (Kalman, 1960). Learned state-space models extend it to nonlinear representations (Karl et al., 2017; Krishnan et al., 2017). KVAE separates recognition from dynamics, RKN structures filtering in learned feature spaces, and KalmanNet learns filter components under partially specified dynamics (Fraccaro et al., 2017; Becker et al., 2019; Revach et al., 2022). Selective state-space models and gated associative memories study how input-dependent updates retain and revise history (Gu et al., 2022; Gu & Dao, 2024; Yang et al., 2025); recent recurrent-network analyses connect memory demand to representation geometry (Sharma et al., 2026). Our focus is the interface between an explicit pose predictor and persistent sensor memory. Matched controls change the predictor's access to that memory, and complementary interventions distinguish the propagation of pose errors from recovery after memory loss.

3 SENSOR-GROUNDED MEMORY

MARC assigns different write access to sensor observations and pose estimates. EMG-derived observations update persistent memory; a motion predictor reads that memory and the preceding pose, but does not write its prediction back into the memory. Joint-wise correction combines the two estimates. Figure 2 shows the resulting asymmetric dependencies.

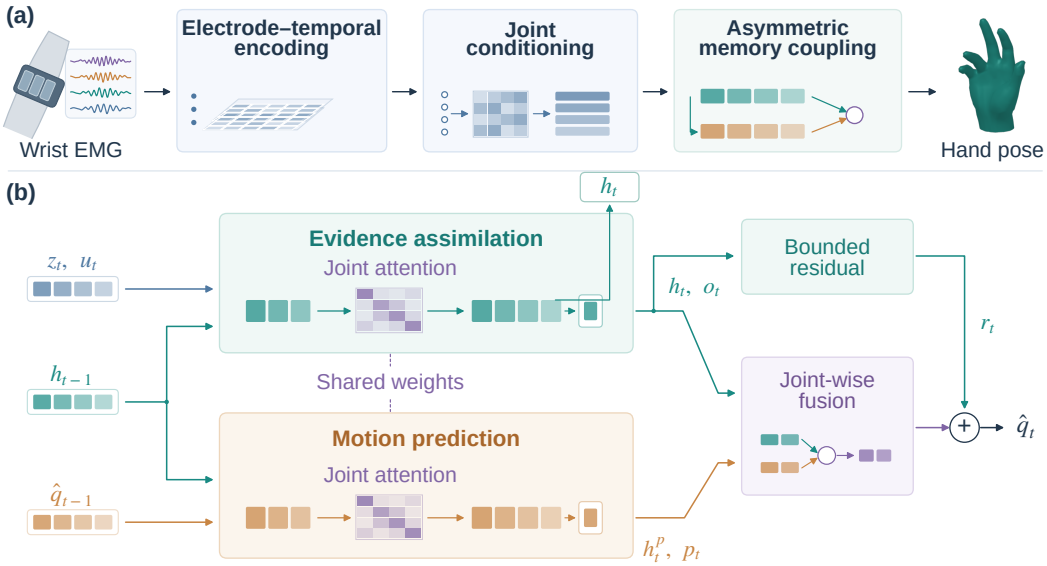

Figure 2: **MARC architecture.** (a) Electrode–temporal encoding and learned joint queries provide joint-conditioned EMG features and an absolute pose observation. (b) Evidence assimilation and motion prediction share joint-attention parameters but receive distinct inputs. Only prediction reads the preceding estimated pose. The observation memory h_t and emitted pose $\hat{q}_t$ are retained, whereas h_t^p is temporary. Joint-wise fusion is supplemented by a bounded pose residual driven by observation features. Feature blocks, attention matrices, and the hand rendering are schematic.

3.1 JOINT-CONDITIONED SENSOR ENCODING

Let X_t denote the EMG patch ending at time t and $y_t \in \mathbb{R}^J$ the corresponding hand target. MARC estimates pose from arrived EMG alone, without an initial reference pose. Its recurrent interface is

$$(\hat{q}_t, \mathcal{S}_t) = F_\theta(X_t, \mathcal{S}_{t-1}), \qquad \mathcal{S}_t = (\mathcal{K}_t, h_t, \hat{q}_t), \tag{1}$$

where $\mathcal{K}_t$ stores encoder context and h_t stores observation memory. Completed patches define a 20 ms update interval.

Electrode–temporal encoding. Waveform embeddings preserve local signal shape, and a parallel log-energy projection retains amplitude information. Electrode indices identify the channels. Al-

ternating electrode attention, masked temporal attention, and gated feed-forward transformations integrate activity across sensors and time (Vaswani et al., 2017). This factorization preserves the electrode axis until features are assigned to joints.

Joint conditioning. Learned joint queries attend to the encoded electrodes to form $z_t \in \mathbb{R}^{J \times d_h}$. A pooled pose readout and joint-specific corrections produce an absolute pose observation u_t. This direct configuration estimate complements the motion transition: pose can be corrected from EMG evidence without relying exclusively on accumulated increments. Appendix A specifies dimensions and attention operations.

3.2 Asymmetric memory coupling

Let $\mu, s \in \mathbb{R}^J$ denote fixed training-set pose statistics and $\bar{q} = (q - \mu)/s$. Evidence assimilation updates observation memory and refines its pose readout; the predictive transition continues the preceding estimate. Joint-wise correction then forms the output:

$$\begin{aligned} h_t &= U_\theta(h_{t-1}, z_t, \bar{u}_t, \Delta t), \qquad o_t = u_t + s \odot C_\theta(h_t), \\ (p_t, h_t^p) &= P_\theta(h_{t-1}, \hat{q}_{t-1}, \Delta t), \\ r_t &= s \odot \tanh R_\theta(h_t, \bar{o}_t, \Delta t), \quad \hat{q}_t = o_t + g_t \odot (p_t - o_t) + r_t. \end{aligned} \tag{2}$$

U_θ assimilates the new joint observations, and C_θ corrects their absolute readout. P_θ uses the preceding estimate and observation memory to predict an increment, which is added to the preceding pose to form p_t. The two operations share a joint-attention block, with distinct input projections and operation embeddings. The shared operator captures interactions among joints, while the separate inputs determine which information can enter each route.

The state dependencies are asymmetric: observation memory informs prediction, while predicted poses enter only the predictive route. The temporary prediction memory h_t^p contributes to the current correction and is then discarded. A sigmoid gate g_t uses both memories, normalized observation and prediction, their discrepancy, and the time interval to weight the prediction for each joint. The observation-conditioned residual permits a bounded adjustment beyond their convex combination, with $|r_{t,j}| < s_j$. At initialization, observation memory is zero and prior fusion is disabled.

This organization yields a testable property. With the EMG trajectory, encoder cache, and initial observation memory fixed, a perturbation to stored pose can propagate through prediction and output correction, but cannot enter subsequent observation-memory updates. Adding pose feedback opens that route. We therefore evaluate pose perturbation and memory reset as complementary tests of error propagation and recovery after lost context. Appendix D gives the local sensitivity analysis.

3.3 Learning pose and motion

The model is trained on pose error and movement consistency:

$$\mathcal{L}_{\text{train}} = \frac{\mathcal{L}_q + \lambda_o \mathcal{L}_o + \lambda_\Delta \mathcal{L}_\Delta + \lambda_p(e)\mathcal{L}_{\text{pred}} + \lambda_{\text{FK}} \mathcal{L}_{\text{tip}}}{\bar{s}}, \quad \bar{s} = J^{-1} \sum_j s_j. \tag{3}$$

$\mathcal{L}_q$ and $\mathcal{L}_o$ supervise the final pose and corrected observation. $\mathcal{L}_\Delta$ matches pose changes over multiple intervals; $\mathcal{L}_{\text{pred}}$ provides auxiliary future-pose supervision during training.

On emg2pose, $\mathcal{L}_{\text{tip}}$ supervises five fingertip positions through the generic-hand forward-kinematics mapping (Salter et al., 2024; Han et al., 2022). This term is omitted for datasets without a compatible mapping. Normalization by $\bar{s}$ sets a common training scale. Section 4.1 gives every loss coefficient; Appendix B defines the losses and optimization schedule.

4 Experiments

We evaluate three questions: whether the complete decoder improves accuracy across recording settings; whether its gains extend to hand geometry and movement; and how predicted-pose access to observation memory changes error propagation and recovery.

4.1 EVALUATION PROTOCOL

All methods decode continuous hand targets from EMG alone, without a ground-truth initial pose. We train each model separately on the three datasets in Table 1. emg2pose follows the official user/movement assignments and uses complete five-second scored windows. MANUS uses a fixed user/gesture partition of the authors' recommended 15-person cohort, evaluated on complete trials (Graf & Barthet, 2026). The MANUS training, validation, and test partitions contain 10/2/3 users and 14/3/4 gestures, respectively. NinaPro DB8 uses the acquisition split in Table 1 on ten intact participants (Krasoulis et al., 2019; NinaPro, n.d.). Appendix C gives the partitions and alignment.

We reproduce TDS+LSTM and the published large EMGFormer configuration, EMGFormer-L (Xi et al., 2026), with identical partitions, causal input availability, scored timestamps, and metric aggregation. Each benchmark configuration is trained with three random seeds; values are mean $\pm$ sample SD. Memory-access controls use one matched initialization. Validation error selects checkpoints and supplies any plateau-based learning-rate or early-stopping decisions, using the mean participant-macro error across validation settings. Each run uses one selected model for all test settings. Mean absolute error (MAE) averages valid frames and target channels within each participant, then weights participants equally. emg2pose and MANUS measure joint angles in degrees. DB8 measures reconstruction of 18 calibrated CyberGlove sensor channels on their released numerical scale, without converting them to angles. Controlled retraining is specified in Section 4.4. Confidence intervals (CI) use paired-participant bootstrap resampling (Agarwal et al., 2021).

Table 1: **Three recording settings for continuous hand decoding.** Each model is trained separately. EMG entries give channel count and input sampling rate; DB8 is supplied after resampling to 2 kHz. Targets are joint angles on emg2pose/MANUS and calibrated CyberGlove measurements on DB8.

Dataset	EMG	Targets	Evaluation setting
emg2pose	16 / 2 kHz	20 angles	Official user/movement assignments; complete scored windows
MANUS	8 / 200 Hz	20 angles	Fixed user/gesture partition of the recommended 15-person cohort; full trials
NinaPro DB8	16 / 2 kHz	18 calibrated sensor channels	First acquisition: training; second: validation; third: testing

Architecture and learning. The shared model uses six electrode–temporal blocks of width 256 and joint memories of width 64. In Eq. 3, $\lambda_o = 0.5$, $\lambda_\Delta = 0.1$, and $\lambda_p(e) = 0.05 \min(1, e/3)$, where e is the training epoch. The fingertip coefficient is $\lambda_{\mathrm{FK}} = 0.01(180/\pi)$ on emg2pose and zero on MANUS and DB8. The combined objective is divided by the training-set scale $\bar{s}$. Displacement intervals are 20, 40, 50, 60, 80, 100, 150, and 200 ms. AdamW uses effective batches of 128/64/32 and peak learning rates of $4.243/3.000/2.121 \times 10^{-4}$ on emg2pose/MANUS/DB8; Appendix B gives the complete schedule.

4.2 GENERALIZATION ACROSS RECORDING CONDITIONS

Accuracy improves across users, movements, and recording systems. MARC improves over both baseline families on every test setting (Table 2). On emg2pose, reductions relative to TDS+LSTM are 9.5% for unseen users, 19.3% for unseen movements, and 12.1% when both are unseen; the corresponding gains over EMGFormer-L are 5.1%, 10.6%, and 6.3%. Participant-level error is lower than TDS+LSTM for all 20, 158, and 20 participants in these settings, respectively, and lower than EMGFormer-L for 19, 157, and 19. The improvement therefore extends across participants.

On MANUS, the mean improvements over EMGFormer-L are 0.303, 0.517, and 0.426 degrees for unseen users, unseen gestures, and their combination. On DB8, CyberGlove reconstruction error decreases by 15.5% relative to TDS+LSTM and 8.2% relative to EMGFormer-L. The same core architecture thus applies across different sensor systems and target representations; unfamiliar gestures remain the harder MANUS setting.

Table 2: **Generalization across users, movements, and acquisitions.** Complete-test-set participant-macro MAE, reported as mean ± SD; lower is better and bold marks the lowest mean. emg2pose and MANUS use degrees; DB8 uses the released scale of 18 calibrated CyberGlove channels. Unseen means absent from training.

Dataset	Test setting	TDS+LSTM	EMGFormer-L	MARC
emg2pose	Unseen users	12.861 ± 0.052	12.272 ± 0.005	$\mathbf{11.645} \pm 0.044$
	Unseen movements	16.380 ± 0.050	14.788 ± 0.332	$\mathbf{13.217} \pm 0.243$
	Unseen users and movements	16.519 ± 0.067	15.500 ± 0.098	$\mathbf{14.527} \pm 0.104$
MANUS	Unseen users	11.812 ± 0.091	10.987 ± 0.157	$\mathbf{10.684} \pm 0.454$
	Unseen gestures	14.499 ± 0.052	14.597 ± 0.423	$\mathbf{14.080} \pm 0.685$
	Unseen users and gestures	16.004 ± 0.217	15.239 ± 0.217	$\mathbf{14.813} \pm 0.392$
NinaPro DB8	Third acquisition	7.800 ± 0.027	7.179 ± 0.034	$\mathbf{6.593} \pm 0.095$

4.3 Spatial and Temporal Reconstruction

Angular improvements translate to fingertip geometry. Using the same generic-hand forward kinematics for predictions and targets, MARC achieves five-fingertip errors of 24.26, 29.77, and 33.08 mm for unseen users, unseen movements, and both on emg2pose. These are reductions of 13.1%, 21.6%, and 15.4% from TDS+LSTM. The spatial improvement is largest on unseen movements, consistent with the angular result. Figure 3 resolves this result alongside temporal reconstruction; Appendix E reports individual fingers and joints.

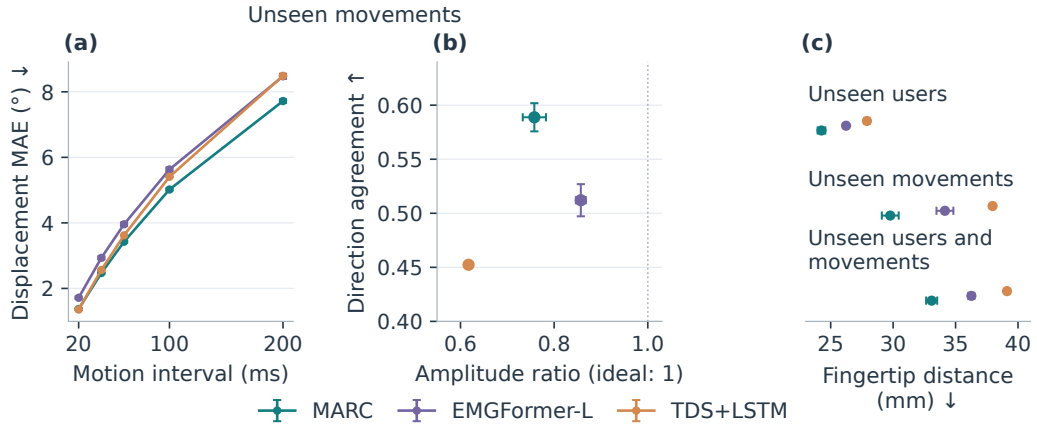

Figure 3: **Reconstruction quality in space and time.** All three models are evaluated on the complete emg2pose test sets. (a) Absolute error of joint-angle changes over 20–200 ms for unseen movements; lower is better. (b) The same setting at 100 ms: direction agreement is displacement-vector cosine (one means aligned); amplitude ratio compares predicted and reference motion magnitude (one is ideal, below one means attenuation). (c) Euclidean fingertip distance from generic-hand forward kinematics; lower is better. Error bars show SD.

Movement direction improves, while amplitude remains compressed. For unseen movements at 100 ms, displacement MAE improves from 5.631 ± 0.061 degrees for EMGFormer-L to 5.018 ± 0.034, and directional agreement rises from 0.512 ± 0.015 to 0.589 ± 0.013. The corresponding amplitude ratios are 0.857 ± 0.011 and 0.758 ± 0.025. MARC thus reconstructs displacement more accurately while underestimating its magnitude. Together, these measures reveal the trade-off between directional agreement and amplitude attenuation.

Gains are distributed across the articulated hand. Figure 4 uses the same two public baselines throughout. Improvements cover fingers and generalization settings, with the largest reductions on unseen movements. A few channels remain worse than EMGFormer-L.

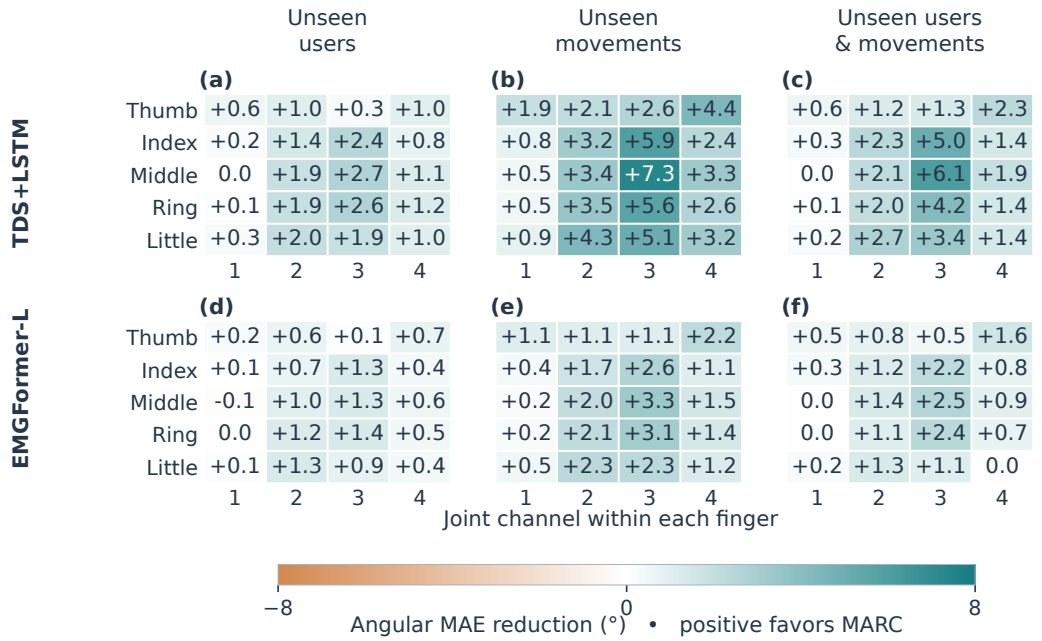

Figure 4: **Accuracy gains across the articulated hand.** Complete emg2pose test results. Top row: TDS+LSTM minus MARC; bottom row: EMGFormer-L minus MARC. Positive values mean lower MARC error, averaged equally over participants and then across training runs. Columns show the indicated unseen-user/movement settings. Within each finger, channels 1–4 follow the released anatomical order; their full joint and movement definitions appear in Appendix E.3. All panels share one degree scale.

Configuration recovery and movement reconstruction. The visual examples focus on the official TDS+LSTM baseline, complementing the two-baseline quantitative comparisons. In Figure 1a, MARC reduces the persistent flexion error of TDS+LSTM: full-window angular and fingertip errors fall from 15.87 to 6.45 degrees and from 43.85 to 17.81 mm. A separate dynamic example (Figure 5) shows more faithful opening and closing, with angular error falling from 22.52 to 11.94 degrees and fingertip error from 57.06 to 27.48 mm. Reference motion determines the event frames and plotted joint. These selected advantage examples are complemented by median-gain and failure cases in Appendix H, which specifies the selection rule.

4.4 Ablations: memory access

We ask whether predicted poses should write observation memory. Table 3 compares controlled modifications of the same MARC decoder, with matched initialization, data, selection rule, and training budget.

Memory-write controls. Pose feedback adds a projection of the preceding normalized pose to the observation-memory update. The observation-projection control adds the same number of parameters, but projects the current EMG-derived pose observation instead. Comparing these two controls holds added capacity fixed: on unseen movements, using the current observation yields 0.222-degree lower error than using the preceding predicted pose. MARC's original memory route has the lowest error in all three settings.

Error propagation versus context recovery. We intervene on the three memory-write models in Table 3. On 256 uniformly sampled windows per test setting, we perturb the stored pose by alternating ± 5 degrees or reset observation memory at two seconds; the EMG stream and encoder cache are unchanged. These interventions test whether the same feedback route that carries useful context also carries errors.

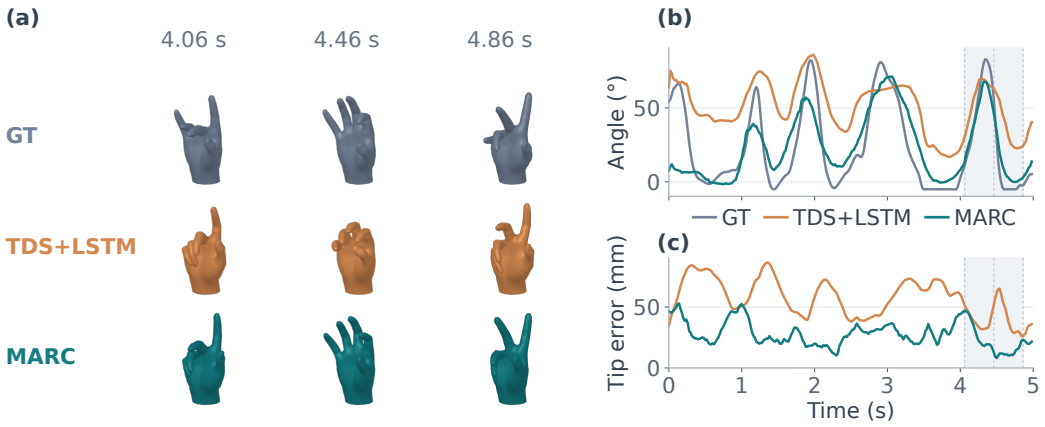

Figure 5: **Reconstructing a rapid hand transition.** (a) Aligned hand configurations around the largest short-time change in reference motion. (b) Index-finger proximal interphalangeal flexion and (c) mean fingertip error over the complete five-second window; shading marks the displayed interval. The full trajectory shows both improved tracking and remaining amplitude errors. This is a different sequence from Figure 1a.

Table 3: **Memory-access ablations.** Angular MAE (degrees) on the complete emg2pose test splits, averaged equally over participants; lower is better. Models share initialization, data, selection rule, and training budget. The two controls add equal-capacity projections of different inputs. Appendix F also reports fingertip distance and controlled state interventions.

Model change	Unseen users	Unseen movements	Unseen users and movements
MARC reference	**11.623**	**13.640**	**14.625**
Add observation projection	11.766	13.742	14.773
Add pose feedback	11.780	13.964	14.831

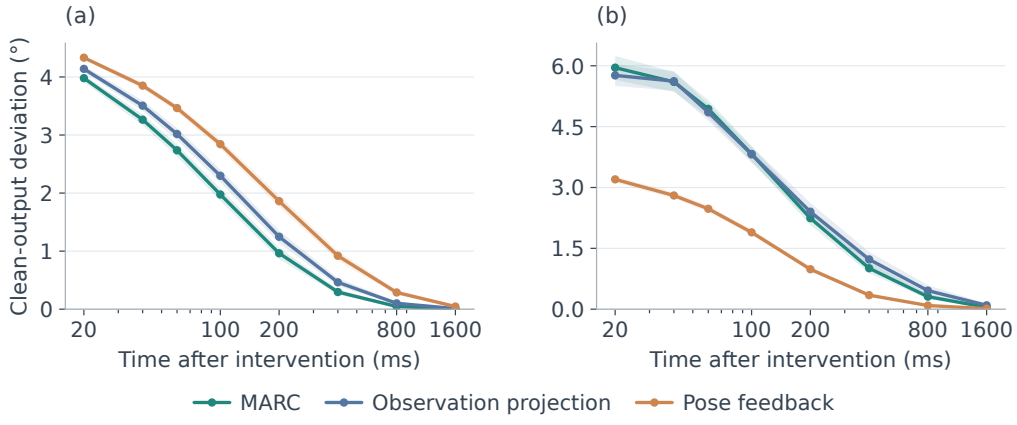

Figure 6: **Pose feedback changes how estimation errors propagate.** (a) Response to a pose-state perturbation. (b) Response to erasing observation memory. Curves show angular deviation from each model's unperturbed output on unseen movements; lower means faster recovery of that trajectory, not necessarily lower error against the reference pose. The observation-projection control matches the feedback variant's added capacity. Shading denotes 95% CI. Feedback carries erroneous pose information in (a), but helps recover the unperturbed output after memory loss in (b).

At 200 ms after pose perturbation, the clean-output deviations are 0.963 degrees for MARC, 1.861 for pose feedback, and 1.248 for the observation-projection control. After memory reset, they are

2.243, 0.985, and 2.403 degrees. The feedback-minus-MARC differences have paired-user 95% bootstrap intervals of [0.834, 0.959] and $[-1.431, -1.100]$, respectively. Feedback carries information that helps recover the clean output after memory loss, but can also feed an erroneous pose into subsequent observation updates. The opposite responses expose this trade-off more clearly than a single ablation score. Appendix D derives the state dependence; Appendix F reports excess target error alongside clean-output recovery.

5 Discussion and conclusion

MARC treats the interface between motion prediction and observation memory as an architectural choice. Its asymmetric coupling allows pose estimates to guide motion prediction while reserving observation-memory updates for EMG-derived information. The complete decoder improves reconstruction over two public baseline families across three datasets. Within the architecture, matched controls and state interventions expose a feedback trade-off: predicted poses help restore the unperturbed output after observation-memory loss, but also carry pose errors into subsequent observation updates.

The benchmark comparisons establish the decoder's practical value; the controlled experiments explain consequences of a particular information route. Restricting that route does not optimize both sides of the trade-off simultaneously. Motion amplitude also remains attenuated despite better displacement accuracy, and fingertip geometry is assessed with a generic hand. The broader research question is how a streaming sensor decoder should use its own predictions without making later observation updates unnecessarily dependent on their errors. MARC provides a concrete implementation and evidence from myoelectric decoding; other sensing tasks can test how the trade-off depends on observation ambiguity and available history.

Reproducibility statement

Appendices A–C specify the architecture, streaming state, loss coefficients, optimization, dataset partitions, temporal alignment, and metric aggregation. Appendix D states the assumptions and derivation underlying the memory analysis; Appendices E–F detail the reconstruction diagnostics and matched controls. Appendix H specifies the qualitative selection procedure.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Philipp Becker, Harit Pandya, Gregor Gebhardt, Cheng Zhao, C. James Taylor, and Gerhard Neumann. Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces. In *International Conference on Machine Learning*, pp. 544–552, 2019.

Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Max Graf and Mathieu Barthet. sEMG-MANUS dataset: synchronised myo semg and manus finger joint recordings for hand pose estimation and xr musical interaction, 2026.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling*, 2024.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Shangchen Han, Po-chen Wu, Yubo Zhang, Beibei Liu, Linguang Zhang, Zheng Wang, Weiguang Si, Peizhao Zhang, Yujun Cai, Tomas Hodan, Randi Cabezas, Luan Tran, Muzaffer Akbay, Tsz-Ho Yu, Cem Keskin, and Robert Wang. UmeTrack: Unified multi-view end-to-end hand tracking for VR. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. doi: 10.1145/3550469.3555378.

Awni Hannun, Ann Lee, Qiantong Xu, and Ronan Collobert. Sequence-to-sequence speech recognition with time-depth separable convolutions. In *Interspeech*, pp. 3785–3789, 2019. doi: 10.21437/Interspeech.2019-2460.

Patrick Kaifosh, Thomas R. Reardon, and CTRL-labs at Reality Labs. A generic non-invasive neuromotor interface for human-computer interaction. *Nature*, 645:702–711, 2025. doi: 10.1038/s41586-025-09255-w.

Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. In *International Conference on Learning Representations*, 2017.

Agamemnon Krasoulis, Sethu Vijayakumar, and Kianoush Nazarpour. Effect of user practice on prosthetic finger control with an intuitive myoelectric decoder. *Frontiers in Neuroscience*, 13: 891, 2019. doi: 10.3389/fnins.2019.00891.

Rahul G. Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. doi: 10.1609/aaai.v31i1.10779.

Yilin Liu, Shijia Zhang, and Mahanth Gowda. NeuroPose: 3d hand pose tracking using EMG wearables. In *Proceedings of the Web Conference 2021*, 2021. doi: 10.1145/3442381.3449890.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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 *HANDS Workshop*, 2025. URL `https://hands-workshop.org/files/2025/VQ-MyoPose.pdf`. Extended abstract.

NinaPro. Ninapro database 8: Official dataset instructions. Official dataset documentation, n.d. URL `https://ninapro.hevs.ch/instructions/DB8.html`. Accessed 26 September 2026.

Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. Kalmannet: Neural network aided kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 2022. doi: 10.1109/TSP.2022.3158588.

Sasha Salter, Richard Warren, Collin Schlager, Adrian Spurr, Shangchen Han, Rohin Bhasin, Yujun Cai, Peter Walkington, Anuoluwapo Bolarinwa, Robert Wang, Nathan Danielson, Josh Merel, Eftychios Pnevmatikakis, and Jesse Marshall. emg2pose: A large and diverse benchmark for surface electromyographic hand pose estimation. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024.

Pratyaksh Sharma, Alexandra M. Proca, Lucas Prieto, and Pedro A. M. Mediano. Temporal superposition and feature geometry of rnns under memory demands. In *International Conference on Learning Representations*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Sagar Verma. emg2tendon: From sEMG signals to tendon control in musculoskeletal hands. In *Robotics: Science and Systems XXI*, 2025.

Ziheng Xi, Jiayi Yu, Yitao Wang, Yanbo Duan, Jianjiang Feng, and Jie Zhou. Egoemg: A multimodal egocentric dataset with bilateral emg and vision for hand pose estimation. *arXiv preprint arXiv:2605.05712*, 2026. URL `https://arxiv.org/abs/2605.05712`.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, 2025.

A Architecture and streaming inference

This appendix specifies inference, clocks, objectives, and evaluation. Each dataset uses an independently trained instance of the same core architecture.

A.1 Signals, targets, and persistent state

Let $X_t \in \mathbb{R}^{B\times C\times P}$ be a completed EMG patch: B is batch size, C is the electrode count, and P is the number of samples per electrode. The endpoint of the patch defines time t. Only samples that have arrived by that endpoint may enter X_t. The network predicts J continuous target channels. Its persistent state is

$$\mathcal{S}_t = (\mathcal{K}_t, h_t, \hat{q}_t), \qquad h_t \in \mathbb{R}^{B\times J\times d_h}, \quad \hat{q}_t \in \mathbb{R}^{B\times J}, \tag{4}$$

where $\mathcal{K}_t$ contains the encoder key/value memory and the absolute token index. Inference uses EMG and the model's own history.

Identical input prefixes and initial states must yield identical output prefixes. This contract covers attention, preprocessing, training normalization, and sequence boundaries. Targets and validity masks enter only losses and evaluation.

Table 4: State and feature dimensions for one update. The encoder width is $d_e = 256$, state width $d_h = 64$, and encoder depth $L = 6$.

Quantity	Shape	Interpretation
Completed patch X_t	$B \times C \times P$	training-normalized arrived EMG samples
Electrode tokens E_t	$B \times C \times d_e$	Electrode identity retained through all L blocks
Pooled representation c_t	$B \times d_e$	Attention-weighted electrode summary
Joint features z_t	$B \times J \times d_h$	Learned joint-query readout of E_t
Absolute readout u_t	$B \times J$	Pooled prediction plus joint-specific correction
Observation memory h_t	$B \times J \times d_h$	Persistent state updated from EMG-derived features
Prediction memory h_t^p	$B \times J \times d_h$	Temporary state used to propose p_t
$o_t, p_t, g_t, \hat{q}_t$	$B \times J$ each	Corrected observation, prior, gate, output
Per-layer cached K, V	$BC \times 4 \times \ell \times 64$	$0 \le \ell \le 99$ previous tokens per electrode

For emg2pose, $(C, P, J) = (16, 40, 20)$ at 2 kHz; for MANUS, $(8, 4, 20)$ at nominal 200 Hz; for DB8, $(16, 40, 18)$ at 2 kHz. All produce native 20 ms updates. The emg2pose instance has 5,735,833 trainable parameters. Dataset-specific input/output dimensions retain the same observation–prediction dependencies.

A.2 Electrode–temporal encoder

Each electrode patch is embedded by a two-layer MLP, $P \to 128 \to 256$, with GELU and output LayerNorm. A separate amplitude path computes

$$a_{t,c} = \tfrac{1}{2} \log \left(P^{-1} \sum_{p=1}^{P} X_{t,c,p}^2 + 10^{-8} \right). \tag{5}$$

A linear projection of $a_{t,c}$ is added to the waveform embedding, preserving an explicit amplitude path alongside the normalized waveform features.

The encoder adds a learned projection of 16 fixed Fourier coordinates: sine and cosine of eight harmonics of $2\pi c/C$. These encode channel index, not measured electrode locations. No circular channel augmentation is used. This encoding does not assert a verified anatomical sensor geometry on DB8.

Each of the six blocks applies three residual operations. First, pre-normalized four-head attention mixes electrodes at the same arrived time. Second, attention mixes time independently for each electrode. Rotary position encoding uses the global token index, including across streaming chunks. A query at index t can access keys k satisfying $0 \le t - k < 100$; thus the attention span includes the current token and at most 99 past tokens. Third, a gated feed-forward layer maps 256 features

to two 512-dimensional branches, multiplies one branch by a SiLU-transformed gate, and projects back to 256. Layer normalization acts on feature dimensions; there is no batch normalization that pools future sequence samples.

For a training chunk of T tokens, electrode attention acts on BT independent sequences of length C; temporal attention acts on BC sequences along time. Cached keys and values are reused, rather than re-encoding all previous raw samples at every step.

The cache contains at most $2LC(99)d_e$ scalar values per stream: 4,866,048 for emg2pose. Precision determines its byte size; measured peak allocation additionally includes parameters and workspaces. Stacked layers and recurrent state can propagate information beyond the per-layer 100-token span. This count includes persistent key/value memory only; temporary attention activations depend on the processed chunk and are excluded.

A.3 Joint readout and the two state operations

After the final encoder block, a normalized score and softmax over electrodes produce c_t. An MLP $256 \to 256 \to J$ yields a normalized global pose estimate. Separately, J learned 64-dimensional queries attend to projected electrode tokens using four-head cross-attention. A query residual and LayerNorm yield z_t. A zero-initialized scalar head maps each joint feature to a pose correction. The two readouts combine as

$$u_t = \mu + s \odot \big(H(c_t) + H_J(z_t)\big), \tag{6}$$

where $\mu, s \in \mathbb{R}^J$ are fixed training-set pose location and scale. On emg2pose and DB8, $s_j = \max(\sigma_j, 5)$ in the dataset's target units; on MANUS, $s_j = \sqrt{\max(\sigma_j^2, 10^{-6})}$. Here σ_j is the training-set standard deviation. The joint queries are free learned parameters. Attention between joints is not a hand-crafted graph, and observed attention weights are not automatically identifiable physiological synergies.

The shared state block receives a previous hidden tensor h, a projected input v, and an operation index $r \in \{0, 1\}$. With operation embedding ϕ_r, it computes

$$b = h + v + \phi_r, \tag{7}$$

$$a = b + \mathrm{MHA}(\mathrm{LN}_1(b), \mathrm{LN}_1(b), \mathrm{LN}_1(b)), \tag{8}$$

$$S(h, v, r) = \mathrm{LN}_2\big(a + W_2\,\mathrm{SiLU}(W_1 a + b_1) + b_2\big). \tag{9}$$

Attention operates across joints with four 16-dimensional heads and feed-forward hidden width 128. It is evaluated separately at each arrived update.

Writing $\bar{q} = (q - \mu)/s$ and $\delta_t = \Delta t / 0.02$, the inference calls are

$$h_t^p = S\left(h_{t-1}, \widetilde{W}_p[\bar{q}_{t-1}, \delta_t] + b_p, 0\right), \tag{10}$$

$$p_t = \hat{q}_{t-1} + s \odot H_p(h_t^p), \tag{11}$$

$$h_t = S\left(h_{t-1}, z_t + \widetilde{W}_o[\bar{u}_t, \delta_t] + b_o, 1\right), \tag{12}$$

$$o_t = u_t + s \odot H_o(h_t). \tag{13}$$

The linear input maps are written after combining identical inputs and omitting constant-zero coordinates. Prediction and observation share the attention and feed-forward parameters, with distinct projections and operation embeddings. Shared parameters allow both learning signals to shape the joint representation.

The fusion head is an MLP $132 \to 64 \to 1$ shared across joints. It reads $[h_t, h_t^p, \bar{o}_t, \bar{p}_t, (o_t - p_t)/s, \delta_t]$. Its sigmoid output weights the prior contribution. The observation-side residual uses the same input width but zeroes the prior hidden features, normalized prior, and innovation slots:

$$r_t = s \odot \tanh R\left([h_t, 0, \bar{o}_t, 0, 0, \delta_t]\right), \qquad \hat{q}_t = o_t + g_t \odot (p_t - o_t) + r_t. \tag{14}$$

Thus $|r_{t,j}| < s_j$ for finite activations, without clipping the complete pose. The gate is a mixing coefficient, not calibrated confidence. At cold start, $h_{t-1} = 0$, $p_t = u_t$, and the effective prior weight is zero; observation correction and residual remain available.

Table 5: Streaming update of MARC. The ordered steps expose the persistent state and the dependencies required to emit one pose.

Step	Operation
1	Receive a completed normalized patch X_t and its interval Δt. Read only the existing state $\mathcal{S}_{t-1}$.
2	Encode X_t with the cached keys/values. Update each layer's bounded cache and global token index.
3	Produce electrode summary c_t, joint features z_t, and absolute readout u_t.
4	If a preceding state exists, compute (h_t^p, p_t) from $h_{t-1}, \hat{q}_{t-1}$. Otherwise initialize the cold-start quantities above.
5	Compute h_t and o_t from h_{t-1}, z_t, u_t. This operation does not read p_t, h_t^p, or $\hat{q}_{t-1}$.
6	Compute the fusion gate and bounded observation residual; emit $\hat{q}_t$.
7	Persist $(\mathcal{K}_t, h_t, \hat{q}_t)$; discard the temporary prediction memory h_t^p. Reset all state at the declared sequence boundary.

A.4 Streaming update

Training uses BF16 mixed precision and no dropout. Warmup observation updates are excluded from backpropagation; scored updates remain differentiable. The encoder processes each causal chunk jointly, retaining gradients from scored features.

B Training objectives, clocks, and optimization

B.1 Pose and displacement objectives

For a tensor of per-joint nonnegative errors $e_{b,t,j}$ and a target-validity mask $m_{b,t}$, define

$$\mathcal{A}(e, m) = \frac{\sum_{b,t} m_{b,t} J^{-1} \sum_j e_{b,t,j}}{\max(1, \sum_{b,t} m_{b,t})}. \tag{15}$$

The final-pose and observation terms are $\mathcal{L}_q = \mathcal{A}(|\hat{q} - y|, m)$ and $\mathcal{L}_o = \mathcal{A}(|o - y|, m)$ on the native 50 Hz output grid. Targets retain their dataset units: degrees for emg2pose and the exported MANUS angular channels, and the released CyberGlove measurement scale for DB8. The two absolute-pose losses do not supervise a velocity integral from a known initial pose.

The training displacement objective uses the 100 Hz target view. Let $\widetilde{q}_{2t} = \widetilde{q}_{2t+1} = \hat{q}_t$ denote causal zero-order hold of the native output. For $\tau \in \mathcal{H} = \{20, 40, 50, 60, 80, 100, 150, 200\}$ ms, set $k_\tau = \tau/(10\,\mathrm{ms})$. Then

$$\mathcal{L}_\Delta = \frac{1}{|\mathcal{H}|} \sum_{\tau \in \mathcal{H}} \mathcal{A}\left(\left|(\widetilde{q}_{n+k_\tau} - \widetilde{q}_n) - (y_{n+k_\tau} - y_n)\right|, m_{n+k_\tau} m_n\right). \tag{16}$$

The denser target clock repeats native predictions through causal hold. Main motion diagnostics therefore use the native 20 ms output grid, separately from this displacement-training view.

The auxiliary term $\mathcal{L}_{\mathrm{pred}}$ averages absolute prediction errors over $\mathcal{H}$. For model-generated future predictions $\widehat{y}^p_{t \to t+\tau}$, it is

$$\mathcal{L}_{\mathrm{pred}} = \frac{1}{|\mathcal{H}|} \sum_{\tau \in \mathcal{H}} \mathcal{A}\left(\left|\widehat{y}^p_{t \to t+\tau} - y_{t+\tau}\right|, m_t m_{t+\tau}\right). \tag{17}$$

B.2 Geometry supervision and total scale

On emg2pose, the released generic-hand forward-kinematics implementation maps the 20 predicted angles to 21 landmarks. Angles are converted from degrees to radians before the mapping. We average Euclidean distance over the five fingertip landmarks, whose verified indices are 0–4, obtaining $\mathcal{L}_{\mathrm{tip}}$ in the model's nominal millimetres. Prediction and reference use the same generic hand. The reference tip coordinates are derived from angular labels and are not independent optical fingertip ground truth.

The actual scalar passed to backpropagation is

$$\mathcal{L}_{\text{train}} = \frac{1}{\bar{s}}\left[\mathcal{L}_q + 0.5\mathcal{L}_o + 0.1\mathcal{L}_\Delta + 0.05\min(1, e/3)\mathcal{L}_{\text{pred}} + \lambda_{\text{FK}}\mathcal{L}_{\text{tip}}\right], \quad \bar{s} = J^{-1}\sum_j s_j, \tag{18}$$

Here e is epoch and $\lambda_{\text{FK}} = 0.01(180/\pi)$ on emg2pose. Geometry is omitted on MANUS/DB8 because their target-to-hand mapping is unverified. All emg2pose scored frames are valid, so the fingertip average includes no padding. Division by mean training pose scale is a common scalar normalization, not per-joint reweighting.

Table 6: Optimization settings for the shared architecture, trained independently on each dataset.

Setting	emg2pose	MANUS	NinaPro DB8
Effective batch	128	64	32
Peak learning rate	4.243×10^{-4}	3.0×10^{-4}	2.121×10^{-4}
Maximum epochs	64	160	160
Minimum epochs before early stop	24	32	32
Early-stop patience	20	20	20
Plateau LR patience / multiplier	5 / 0.5	5 / 0.5	5 / 0.5
Minimum learning rate	10^{-6}	10^{-6}	10^{-6}
Geometry term	Five fingertips	None	None

AdamW (Loshchilov & Hutter, 2019) applies weight decay 0.01 to parameters with at least two dimensions and zero decay to lower-dimensional parameters. Gradient norm is clipped to one. The first epoch increases learning rate from approximately 5% to 100% of its peak; the peak follows $3\times10^{-4}\sqrt{B/64}$. Thereafter ReduceLROnPlateau follows the selection criterion in Section 4.1. An improvement must exceed 10^{-4}; one selected model supplies every test setting within a run.

B.3 Public baseline implementations

TDS+LSTM uses the emg2pose convolutional encoder and recurrent decoder for EMG-only regression. EMGFormer-L uses model width 384, twelve attention heads, eight layers, feed-forward width 1536, and a 512-unit readout. Masked temporal attention and prefix-only channel pooling restrict its inputs to the available signal history. On emg2pose these implementations contain 5,980,328 and 16,342,308 parameters, respectively, compared with 5,735,833 for MARC. Input and output projections adapt to the other datasets; shared evaluation support and user-macro aggregation are unchanged.

MANUS TDS+LSTM uses batch size four and FP32. A sixteen-epoch cosine-decay phase is followed by plateau-based learning-rate reduction from the retained optimizer state, starting at 10^{-5}. The extension uses a maximum of 80 total epochs, a minimum of 32, early-stop patience 20, and the selection criterion in Section 4.1. The learning rate halves after five non-improving epochs, down to 10^{-6}; checkpoint improvement requires 10^{-4}. MARC uses the minimum duration and plateau stopping rule in the optimization table above. DB8 TDS+LSTM uses batch size 32; its alternative recipe is not mixed into the main comparison. Baseline and MARC objectives are trained independently; no model receives reference pose at inference.

C Datasets and evaluation

C.1 emg2pose: official assignment and fixed-window realization

We preserve official user/movement partition assignments (Salter et al., 2024) and evaluate fixed complete windows. Training contains 114,202 windows; Table 7 defines held-out support. The decoder consumes 294 patches and emits 250 scored native outputs after 44 warmup updates. Nine normalized zero samples prepend the raw sequence. Complete patch endpoints are $30 + 40t$ in the zero-based raw-sample coordinates used by the evaluator. The first scored endpoint is therefore $30 + 40 \times 44 = 1790$. A 500-frame, 100 Hz target view covers the five-second scored interval; every second target is paired with a native model output. This endpoint definition, rather than merely specifying "50 Hz", is needed to reproduce alignment exactly.

Table 7: emg2pose evaluation support. Counts describe the complete scored test windows and represented participants.

Evaluation set	Generalization setting	Windows	Participants
Test	Unseen users	14,337	20
Test	Unseen movements	27,422	158
Test	Unseen users and movements	3,562	20

EMG mean and standard deviation are fixed from training. Evaluation does not improve temporal correspondence by shifting predictions against test targets. Native motion metrics are computed from the scored output sequence directly; denser held-output metrics remain separately identified.

C.2 MANUS: COHORT, ASSIGNMENT, AND WHOLE-TRIAL STATE

The experiment uses the authors' recommended 15-participant cohort (Graf & Barthet, 2026). A fixed user/gesture assignment is defined before window construction. It is our benchmark assignment on that cohort, not an author-supplied official list. Training users are u_3, u_8, u_10, u_11, u_12, u_16, u_7, u_4, u_5, and u_6; test users are u_9, u_18, and u_13.

The 14 training gesture categories are flexext_pinky, flexext_middle, flexext_index, flexext_thumb, point_index, finger_tap_surface, ab_add_index_middle, ab_add_all, flexext_ring, tap_index, tap_ring, pinched_tap_thumb_index, tap_middle, and tap_thumb. Test gestures are ab_add_thumb_index, flexext_fist, pinched_tap_index_middle, and pinched_tap_middle_ring. The melody condition is excluded. User and gesture groups are disjoint; held-out identity combinations are not reassigned to training.

Table 8: MANUS evaluation support under the fixed cohort partition. "Known" denotes membership in the training partition.

Setting	Held-out population	Test trials
Unseen users	New users, training gestures	377
Unseen gestures	Training users, new gestures	360
Both populations unseen	New users, new gestures	108

One unseen-user test trial contains no eligible scored window and is excluded, leaving 377 trials. Training uses segments with 99 warmup patches and up to 100 scored outputs. Testing instead carries state across every complete retained trial, resets it at the next trial boundary, and masks only padded batch tails. The first scored prediction corresponds to source index 399, after consuming 100 four-sample patches. Subsequent outputs correspond to indices $399 + 4t$; the denser target view uses stride two.

Inputs use fixed training electrode RMS. The 20 targets retain their exported order, without an emg2pose FK remapping. The nominal 200 Hz interface lacks original sensor timestamps for independent synchronization reconstruction. Whole-trial evaluation complements, rather than controls the length of, the emg2pose task.

C.3 NINAPRO DB8: ACQUISITION TRANSFER

We use intact participants S1–S10 and the A1 and A3 recordings for training and testing, respectively (NinaPro, n.d.). This is acquisition transfer within the same participant cohort. It is neither held-out-user evaluation nor a three-axis counterpart of emg2pose. All 16 EMG channels enter the model; the 18 calibrated CyberGlove measurement targets enter only supervision and evaluation. Their released units are retained, rather than reported as degrees. The contralateral measurement arrangement and sensor placement also differ from wristband hand decoding.

Each indexed segment includes 199 patches of 40 samples. The first 99 updates supply warmup and the remaining up to 100 outputs are scored. In zero-based source coordinates, the first scored target is at segment start plus 3999; subsequent native targets advance by 40 samples. The 100 Hz target

view advances by 20 samples. Inputs use fixed training electrode RMS statistics. Nonfinite input or target values are replaced numerically only to keep tensors finite; the corresponding support is excluded by the validity mask, so replacement values are not accepted as valid labels. Tail padding likewise never increases the scored frame count.

C.4 Metric aggregation

For a participant u, let N_u be the number of valid scored native frames across the retained windows or trials. The participant-level angular or channel MAE is

$$E_u = \frac{1}{JN_u} \sum_{(w,t)\in\mathcal{T}_u} \sum_{j=1}^{J} |\hat{q}_{w,t,j} - y_{w,t,j}|. \tag{19}$$

The reported user-macro score is $|\mathcal{U}|^{-1} \sum_u E_u$. It first pools valid frames within each participant and then weights participants equally. It does not average per-window MAEs equally when window supports differ. The frame-micro score is $\sum_u N_u E_u / \sum_u N_u$ and is separately named. Aggregation across models follows Section 4.1.

Table 9: From frame errors to user-macro test scores.

Step	Procedure
1	Use the fixed dataset partition, training-set normalization, and selected model.
2	For each declared window or trial, reset or carry state exactly as the dataset protocol specifies. Run EMG-only inference.
3	Align the native output with its target endpoint; accumulate per-joint absolute error, squared error, and valid count by user. Accumulate geometry only where its mapping is verified.
4	Check unique sample identities, expected support counts, target units, and finite predictions. Divide each user's accumulated errors by its own valid support.
5	Average users equally within each test setting.
6	Report all test axes from that same checkpoint and participant-level errors. Aggregate scores as specified in Section 4.1.

All test axes share one checkpoint, selected as specified in Section 4.1. Test comparisons also informed architecture development.

D Understanding memory access through controlled interventions

This analysis supports the comparison in Section 4.4: should a previous pose estimate be allowed to modify the memory used to interpret the next EMG observation? We first identify the affected dependency, then derive a testable consequence, and finally specify the two interventions and their measurements.

D.1 The dependency changed by pose feedback

Let e_t denote the current encoder features, h_t the persistent observation memory, and $\hat{q}_t$ the emitted pose. After collecting the intermediate operations, MARC can be written as

$$h_t = U_\theta(h_{t-1}, e_t), \qquad \hat{q}_t = F_\theta(h_{t-1}, \hat{q}_{t-1}, e_t). \tag{20}$$

Thus a stored pose can influence the next output but cannot directly write observation memory. In the pose-feedback control, the first equation becomes $h_t = \widetilde{U}_\theta(h_{t-1}, \hat{q}_{t-1}, e_t)$. A zero-initialized, bias-free scalar-to-64 projection introduces this additional input. A second control projects the current EMG-derived absolute pose instead; it adds the same 64 parameters and separates information source from added capacity.

A consequence that can be checked exactly. Take two deterministic MARC executions with identical parameters, encoder cache and observation memory at time t_0, and the same subsequent

EMG inputs, but different stored poses. Their subsequent encoder features, observation memories, and observation readouts are identical. At the next step, both executions supply the same (h_{t_0}, e_{t_0+1}) to U_θ; repeating this argument proves the statement by induction. Their final poses can still differ through F_θ. The claim concerns the inference dependency at fixed parameters, not output stability or statistical independence.

D.2 Why two interventions are needed

For local perturbations, order the state as $(h, \hat{q})$. The transition Jacobians of MARC and the feedback control have the forms

$$J_t = \begin{bmatrix} A_t & 0 \\ C_t & D_t \end{bmatrix}, \qquad \widetilde{J}_t = \begin{bmatrix} \widetilde{A}_t & B_t \\ \widetilde{C}_t & \widetilde{D}_t \end{bmatrix}. \tag{21}$$

Here A describes memory-to-memory sensitivity, B pose-to-memory sensitivity, C memory-to-output sensitivity, and D pose-to-output sensitivity. Tildes denote derivatives of the separately trained feedback model. Its two-step pose response contains an additional route,

$$(\widetilde{J}_t \widetilde{J}_{t-1})_{qq} = \widetilde{C}_t B_{t-1} + \widetilde{D}_t \widetilde{D}_{t-1}. \tag{22}$$

The first term carries a pose change through memory and back to the output; the second carries it through consecutive pose predictions. Either term may amplify or offset the other. Removing the first route therefore does not prove contraction or universally better accuracy. It predicts a useful contrast: pose feedback may preserve an erroneous pose after a pose intervention, but may also restore useful context after memory is erased. We test both possibilities.

D.3 Intervention protocol and interpretation

The three models share initialization of common parameters, training data, training duration, and evaluation windows. Each test setting uses 256 windows sampled uniformly, without ranking by model error. At two seconds into the scored trajectory, the main intervention comparison applies one of the following changes; the encoder cache and incoming EMG are retained.

Table 10: Two complementary interventions on the trained decoder. Each changes one stored quantity while retaining the other; neither changes the participant's movement or the target trajectory.

Intervention	Changed quantity	Question
Pose perturbation	Alternating $+5^\circ$ and -5° offsets across stored joint poses; retain observation memory	Does an erroneous pose enter later observation updates?
Memory reset	Set observation memory to zero; retain the stored pose	Can stored pose help restore the original output after context loss?

For each participant u, let $\mathcal{W}_u$ contain its sampled windows. At a delay τ, output deviation is

$$R(\tau) = \frac{1}{|\mathcal{U}|} \sum_u \frac{1}{|\mathcal{W}_u| J} \sum_{w \in \mathcal{W}_u, j} \left| \hat{q}^{\text{pert}}_{w, t_0+\tau, j} - \hat{q}^{\text{clean}}_{w, t_0+\tau, j} \right|. \tag{23}$$

This quantity is zero when the perturbed decoder returns to its own unperturbed trajectory. It does not measure distance to the true hand pose. We therefore also report *excess angular MAE*: perturbed minus unperturbed error against the reference target. A negative excess means that the perturbation incidentally reduced target error. Appendix F shows both measurements, together with observation-readout deviation.

Responses are sampled at 20, 40, 60, 100, 200, 400, 800, and 1,600 ms. Paired confidence intervals resample represented participants 2,000 times, using the same participant draw for both models. They quantify participant variation for these trained models and sampled windows. The interventions diagnose a computational information route; ordinary decoding gains are assessed separately by the matched retraining results.

E RECONSTRUCTION DIAGNOSTICS

The main table reports aggregate performance on emg2pose, MANUS, and NinaPro DB8. Here we resolve the emg2pose result into hand geometry, motion, participants, and individual fingers. Every comparison uses the same scored frames and participant weighting.

E.1 WHAT THE DIAGNOSTICS MEASURE

Angular accuracy, spatial accuracy, and motion fidelity answer different questions. A hand can remain offset while reproducing movement direction, or appear smooth because movement amplitude has been suppressed. Table 11 separates these properties.

Table 11: Interpretation of the reconstruction diagnostics. MAE denotes mean absolute error. Each metric is computed within a participant and then averaged equally over participants. For motion, $\Delta_\tau q_t = q_t - q_{t-\tau}$ and the corresponding reference displacement use the same time interval τ; sums below include all scored time–joint pairs for one participant.

Measure	Quantity and interpretation	Preferred value
Angular MAE	Mean absolute difference between decoded and reference joint angles.	Lower; degrees
Fingertip distance	Mean Euclidean distance between five reconstructed and reference fingertips under the same hand geometry.	Lower; millimetres
Displacement MAE	Mean of $\lvert\Delta_\tau q - \Delta_\tau y\rvert$: accuracy of the amount and direction of movement.	Lower; degrees
Amplitude ratio	$\sum \lvert\Delta_\tau q\rvert / \sum \lvert\Delta_\tau y\rvert$: reconstructed movement magnitude relative to the reference.	One; below one indicates attenuation
Direction cosine	$\langle \Delta_\tau q, \Delta_\tau y \rangle / (\lVert\Delta_\tau q\rVert_2 \lVert\Delta_\tau y\rVert_2)$ after concatenating time–joint pairs.	Higher, up to one; zero indicates no alignment

Table 12: Five-fingertip reconstruction error on the complete emg2pose test splits (millimetres; lower is better). At each scored frame, Euclidean distance is averaged over five fingertips, then over frames within each participant and equally over participants. Prediction and reference use the same generic-hand forward kinematics. Values are mean ± SD; these distances do not include subject-specific hand-size error.

Model	Unseen users	Unseen movements	Unseen users and movements
TDS+LSTM	27.904 ± 0.211	37.963 ± 0.104	39.119 ± 0.117
EMGFormer-L	26.231 ± 0.107	34.148 ± 0.677	36.266 ± 0.240
MARC	24.262 ± 0.308	29.769 ± 0.676	33.085 ± 0.442

The spatial result uses identical reference angles and a common generic hand. It therefore isolates how decoding errors affect articulated geometry, rather than combining them with participant-specific hand-size estimates or independent tracking error. MARC improves fingertip reconstruction on all three generalization settings, with the largest absolute reduction on unseen movements.

E.2 MOTION ACROSS TIME INTERVALS

Direction and amplitude reveal a distinction that angular error alone cannot. MARC consistently preserves direction better than either baseline, while its motion amplitude lies between those of TDS+LSTM and EMGFormer-L. On unseen movements at 100 ms, its displacement MAE is 5.018 ± 0.034 degrees, compared with 5.407 ± 0.011 for TDS+LSTM and 5.631 ± 0.061 for EMGFormer-L. Its amplitude ratio of 0.758 ± 0.025 nevertheless remains below one. This combination supports improved movement reconstruction while retaining measurable attenuation.

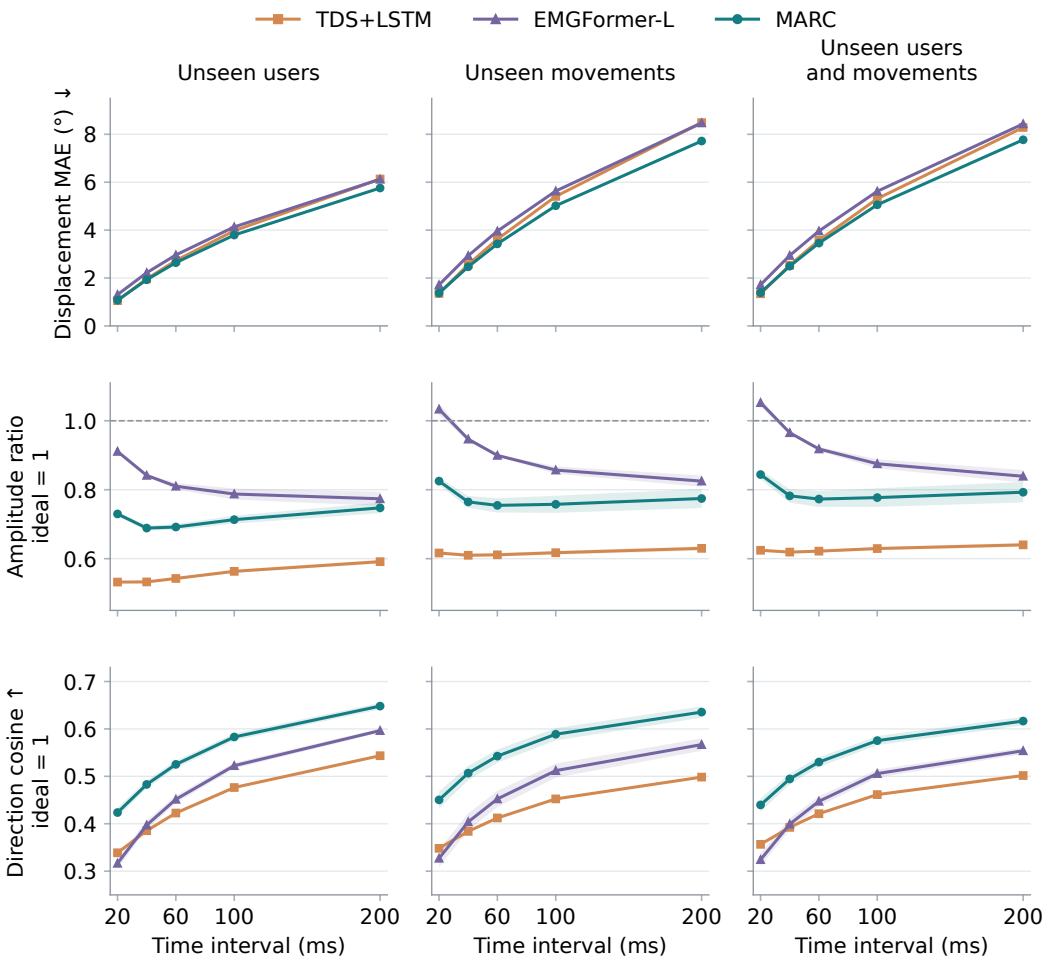

Figure 7: **Motion fidelity across the complete emg2pose test splits.** Columns identify the held-out populations and movements. Top: displacement MAE, where lower is better. Middle: the ratio of decoded to reference absolute displacement; the dashed line marks the ideal value one. Bottom: displacement direction cosine, where higher is better. All three models are evaluated at the native 20 ms output clock using intervals of 20, 40, 60, 100, and 200 ms. Lines show means; shaded bands show SD. Both endpoints use the observations available at their own times, so these are reconstruction metrics rather than future-prediction errors.

At the shortest interval, TDS+LSTM can achieve a slightly lower displacement MAE with substantially less movement amplitude. The direction and amplitude panels prevent interpreting this local error ranking as uniformly better dynamics. Increasing the interval includes larger movements and more accumulated endpoint error, so displacement errors across different intervals should not be interpreted as the same prediction task.

E.3 Which fingers improve?

The improvement extends across all five fingers and all three generalization settings, rather than arising from a single easy joint group. The little finger remains the least accurate group for every model. A finger average can conceal opposing changes among its constituent joints; the joint-level analysis in the main text resolves this finer structure. Label range, movement frequency, target noise, and representation quality can each contribute to these differences.

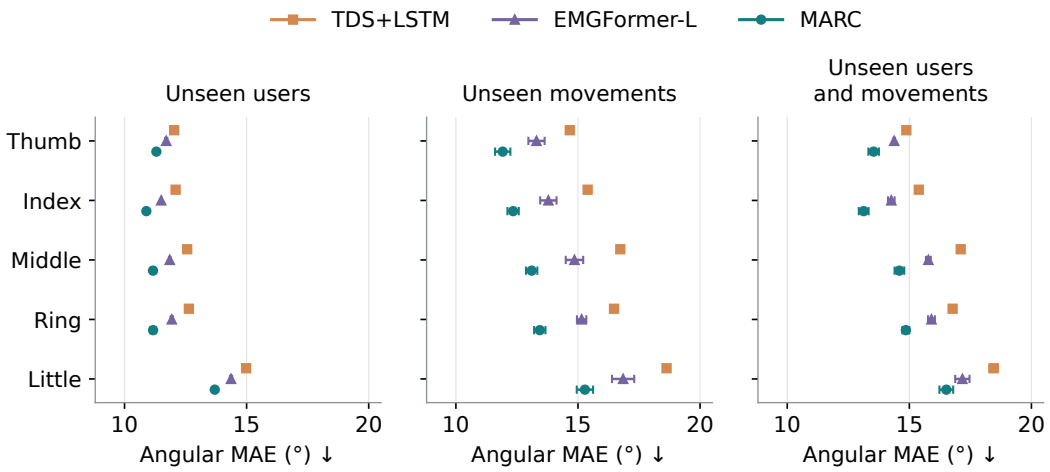

Figure 8: **Finger-resolved reconstruction on the complete emg2pose test splits.** Each point averages angular MAE over a finger's four native joint channels, then equally over participants; lower values indicate better reconstruction. Error bars show SD. The same three models, participant sets, and evaluation frames are used in every panel. This decomposition measures where accuracy improves without attributing a finger-level difference to a particular muscle.

Table 13: Anatomical meaning of joint channels 1–4 in the joint-level figures. The channel number resets within each finger and follows the dataset's native ordering. Flexion–extension describes bending and straightening; abduction–adduction describes lateral spreading and closing.

Channel	Thumb	Index, middle, ring, and little fingers
1	Carpometacarpal flexion–extension: bending at the thumb base.	Metacarpophalangeal abduction–adduction: spreading at the knuckle.
2	Carpometacarpal abduction–adduction: lateral motion at the thumb base.	Metacarpophalangeal flexion–extension: bending at the knuckle.
3	Metacarpophalangeal flexion–extension: bending at the thumb knuckle.	Proximal interphalangeal flexion–extension: bending at the middle joint of the finger.
4	Interphalangeal flexion–extension: bending at the thumb's distal joint.	Distal interphalangeal flexion–extension: bending at the joint closest to the fingertip.

E.4 How broadly do participants benefit?

Every participant improves over TDS+LSTM on each evaluated split. Against EMGFormer-L, 19 of 20 unseen participants improve in each of the two unseen-user settings, and 157 of 158 participants improve on unseen movements. The remaining small negative differences are retained in Figure 9. The wider intervals for unseen users reflect the smaller participant sets and heterogeneous gains; additional windows from the same users would not replace additional participants.

E.5 Accuracy during holds and faster movement

MARC retains lower angular error than both baselines in every speed bin. The low-speed bins also show why accurate dynamics and accurate pose are complementary: very small displacement error can coexist with substantial posture error. TDS+LSTM has smaller short-interval displacement error in several bins but substantially attenuated movement overall, as shown in Figure 7. Cross-bin differences remain descriptive because each bin contains a different mixture of movements and valid participants.

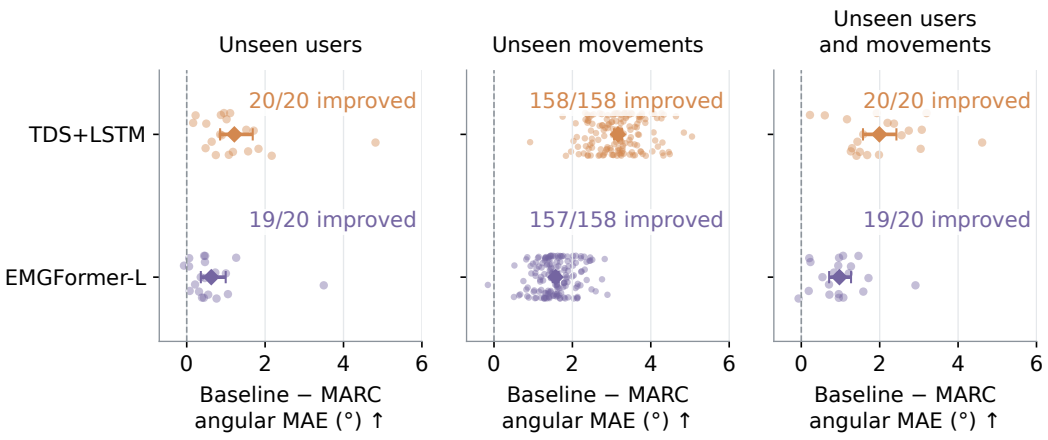

Figure 9: **Participant-level improvement over both public baselines on emg2pose test data.** Each dot is one participant's baseline angular MAE minus MARC angular MAE, after averaging the independent training runs for each method. Positive values favor MARC; the dashed line marks equal accuracy. Diamonds show mean improvement and horizontal bars show paired-participant 95% bootstrap confidence intervals from 10,000 resamples. Counts identify participants with positive improvement. Dot height is only a spacing aid.

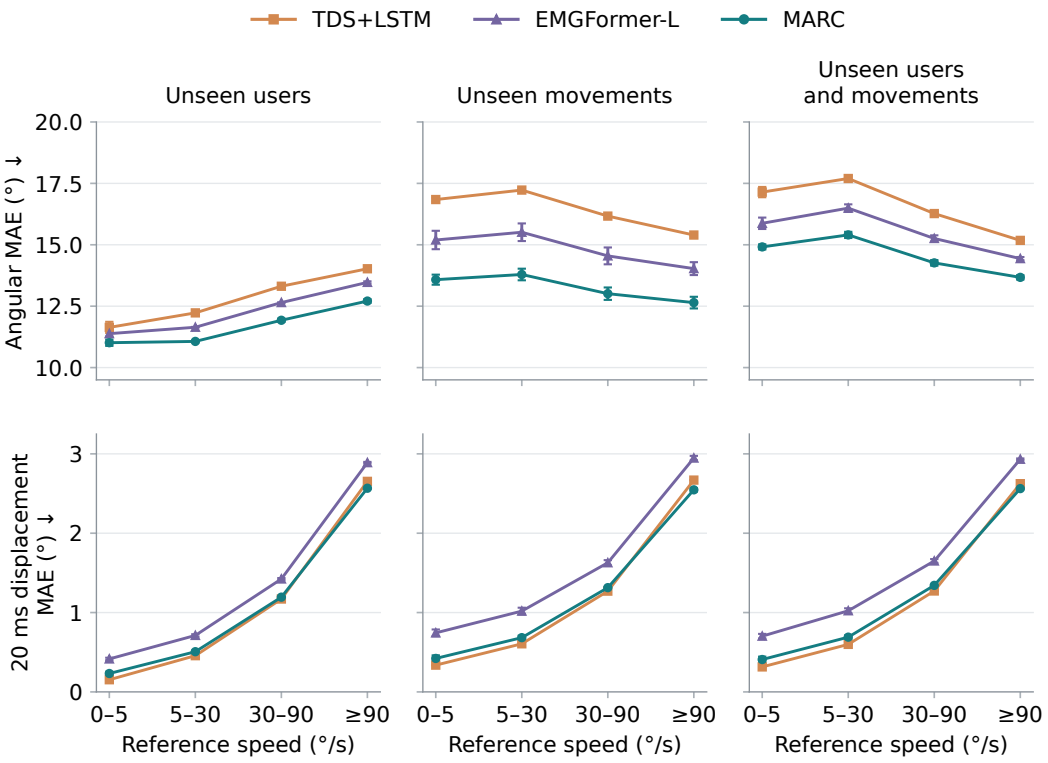

Figure 10: **Pose and movement reconstruction by reference speed on emg2pose test data.** Speed is the mean absolute reference angular change across 20 joints divided by the native 20 ms interval. Bins are $[0, 5)$, $[5, 30)$, $[30, 90)$, and $[90, \infty)$ degrees per second. Top: angular MAE; bottom: 20 ms displacement MAE. Lower is better for both rows. Within each bin, errors are averaged over valid frames per participant and then equally over participants; points show means and error bars show SD. Reference speed is used only to define evaluation strata.

F MEMORY-ACCESS CONTROLS AND INTERVENTIONS

These experiments examine memory access through matched retraining and error propagation through controlled interventions. Retrained controls share initialization and training budget; interventions alter stored quantities in those same trained models.

F.1 SHOULD PREDICTED POSE WRITE INTO OBSERVATION MEMORY?

Table 14: Memory-routing controls on the complete emg2pose test splits. The observation-projection control adds the current EMG-derived pose estimate to memory through a learned scalar-to-64 projection; the pose-feedback control adds the preceding predicted pose through the same-sized projection. Both projections start at zero. Models share initialization, training data, a common 18-epoch budget, and evaluation support. The comparison reports participant-averaged angular MAE and five-fingertip distance; lower is better.

Memory route	Angular MAE (°)	Fingertip distance (mm)
Unseen users		
MARC	11.623	24.325
Added observation projection	11.766	24.756
Added predicted-pose feedback	11.780	24.748
Unseen movements		
MARC	13.640	31.065
Added observation projection	13.742	31.289
Added predicted-pose feedback	13.964	31.818
Unseen users and movements		
MARC	14.625	33.599
Added observation projection	14.773	34.097
Added predicted-pose feedback	14.831	34.190

The additional observation projection controls for the added mapping capacity of the pose-feedback route. All initial shared computations are identical because the added projections start at zero. MARC has lower angular and fingertip errors in all three settings. On unseen movements, the angular gap from predicted-pose feedback is 0.324 degrees, with a paired-participant 95% interval of [0.272, 0.380]. The same trained models supply the state interventions below.

F.2 WHICH ERRORS RE-ENTER MEMORY?

Pose feedback has opposite effects under the two interventions. On unseen movements it increases output deviation after pose perturbation by 0.898 degrees, but reduces deviation after memory reset by 1.258 degrees. Their paired-participant 95% intervals are [0.834, 0.959] and $[-1.431, -1.100]$, respectively, across the 124 participants represented in the fixed diagnostic subset. The results distinguish a route that can recover missing memory from one that can also return pose errors to observation updates.

For MARC, zero observation deviation following pose-only perturbation verifies the intended dependency separation. It does not establish global stability or universal robustness. The reference-error column also prevents confusing recovery of a model's clean output with recovery of the true pose. Artificial memory resets are targeted interventions, rather than a simulation of electrode slip, fatigue, or changing muscle recruitment.

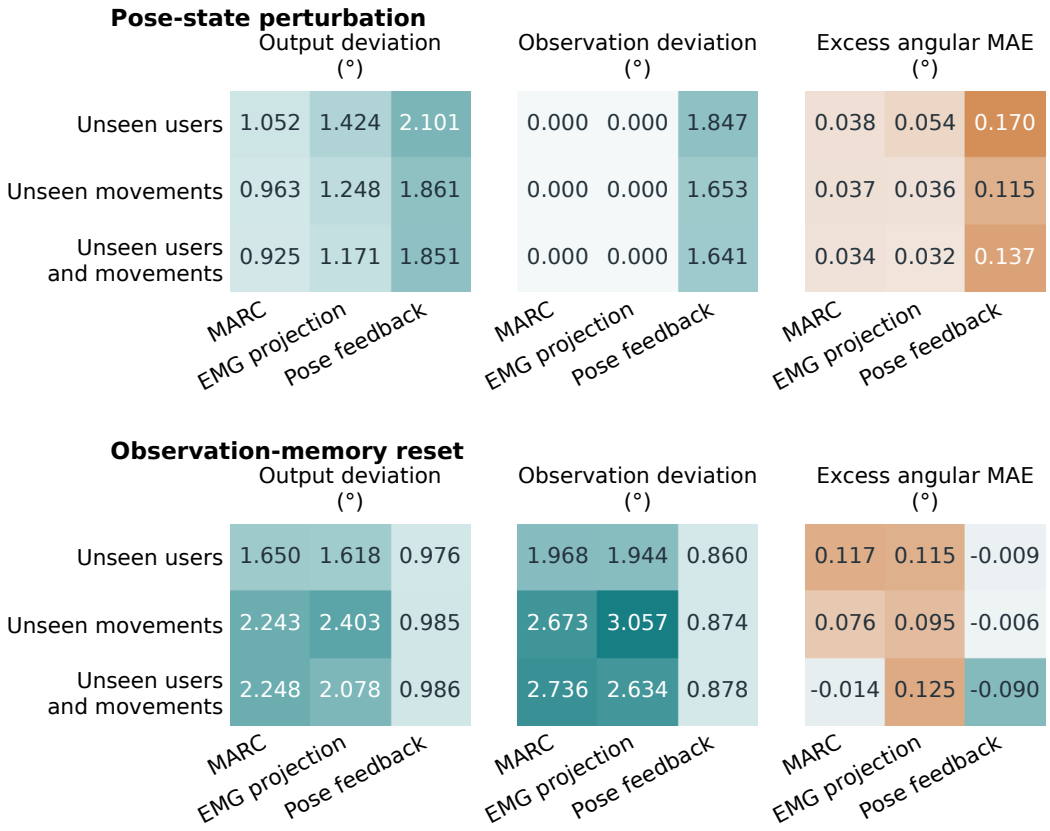

Figure 11: **Controlled interventions separate pose-error propagation from memory recovery.** The three memory routes are trained under the matched setting in Table 14. At two seconds, we either add alternating $\pm 5^\circ$ to the predicted-pose state (top) or erase observation memory (bottom), preserving EMG and encoder history. Values are measured 200 ms later on the same 256 uniformly sampled windows per emg2pose test split. Output and observation deviations are mean absolute differences from each model's own unperturbed trajectory; zero means recovery of that trajectory. Excess angular MAE is perturbed minus unperturbed error against the reference pose; negative values indicate an incidental reduction in reference error. All quantities are in degrees and are averaged equally over participants. Colour intensity encodes the printed values, with the same scale across the two interventions for each metric.

G Computational cost

The architecture contains 5.74 million parameters. On an RTX 5090, a single model update takes 6.07 ms on average, with a peak allocated model-and-tensor memory of 70.5 MiB. Measurements use batch size one, eager FP32 execution, and disabled TF32 and autocast. We warm up for 128 updates and use three repetitions of 200 synchronized updates. The repetition-level 95th-percentile times range from 6.06 to 10.64 ms. Inputs already reside on the GPU; acquisition, host transfer, and visualization are excluded. These measurements characterize model-update cost and do not establish an end-to-end latency or a speedup over the public baselines.

H Qualitative reconstruction and case selection

Two favorable windows illustrate configuration recovery and movement reconstruction. The pool contains 256 uniformly spaced windows from the emg2pose test split with unseen users and movements, decoded at 50 Hz. Both examples have reference 100 ms motion between the pool's 50th and 95th percentiles. For the configuration example, we maximize the smaller of the full-window relative improvements in angular and fingertip errors, requiring both to be positive. For the dynamic

example, both improvements must reach 40%; among the four qualifying windows, we select the one with the greatest reference motion.

Each event is the largest centered 80 ms reference angular change, with snapshots 400 ms before and after it. For the opening illustration, both angular and fingertip errors must improve at each displayed snapshot; among admissible pairs of these three event frames, Figure 1a uses the largest reference configuration change. The complete three-frame sequence remains below. The plotted joint has the largest reference angular range within the displayed interval. All meshes use the same generic hand, camera, and scale.

For the median-gain and failure examples, reference motion is restricted to the pool's 60th–90th percentiles. The median-gain window is nearest the median angular improvement among the 77 retained windows; the failure is nearest the median among the 17 negative-gain windows. These selections illustrate behavior and do not estimate its prevalence.

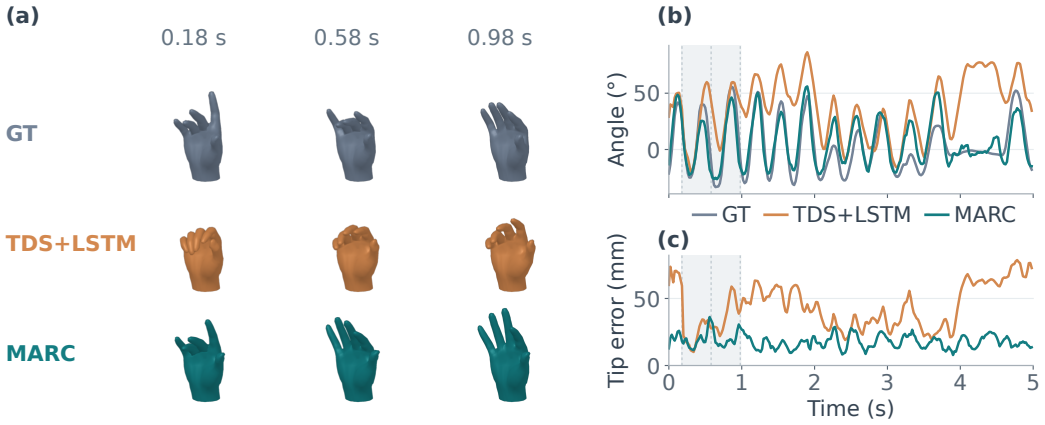

Figure 12: **Full trajectory of the opening example.** (a) Three reference-motion-selected configurations from the sequence in Figure 1a. (b) The joint with the largest reference excursion in the displayed interval and (c) mean five-fingertip error across the full window. The complete sequence exposes the persistent TDS+LSTM flexion error and the remaining MARC errors outside the two opening snapshots.

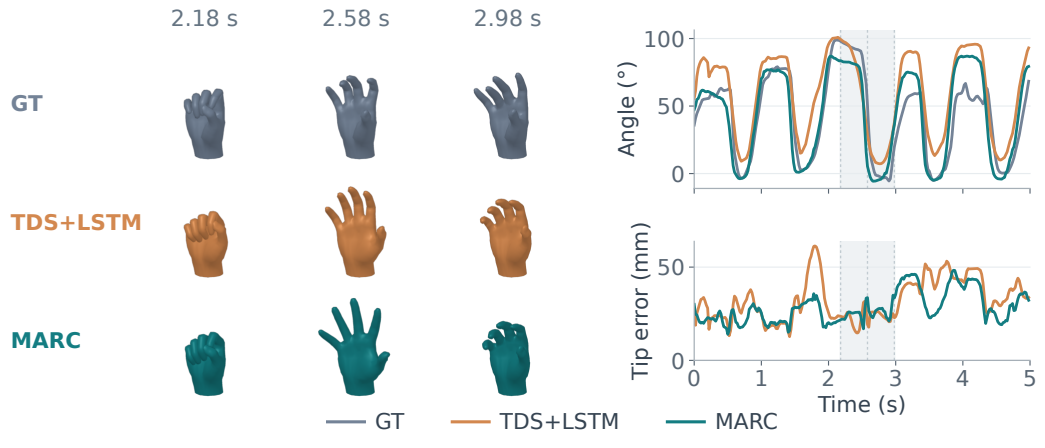

Figure 13: **Median-gain case.** Compared with TDS+LSTM, MARC reduces full-window angular MAE from 13.74 to 12.22 degrees and fingertip error from 32.36 to 28.23 mm. Left: synchronized reference and decoded poses. Right: ring-finger proximal interphalangeal flexion and mean five-fingertip error over the complete sequence.

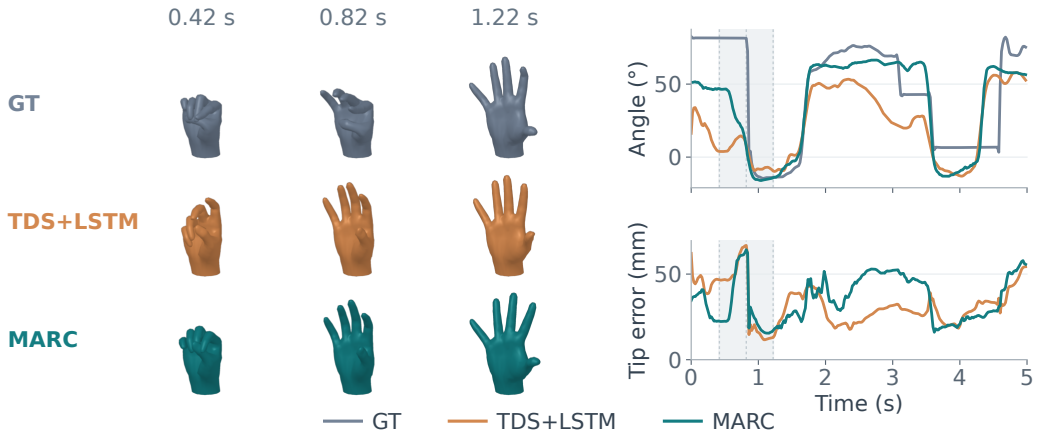

Figure 14: **Failure case.** Compared with TDS+LSTM, MARC increases full-window angular MAE from 16.44 to 17.50 degrees and fingertip error from 32.29 to 35.48 mm. Left: synchronized reference and decoded poses. Right: index-finger metacarpophalangeal flexion and mean five-fingertip error over the complete sequence. Selection and rendering match Figure 13.