NEUROMOTOR HIERARCHY NETWORK: PHYSIOLOGICAL INDUCTIVE BIASES FOR ROBUST GENERALIZATION IN sEMG DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Surface electromyography (sEMG) provides a wearable, noninvasive interface to neuromuscular activity for movement decoding and human-computer interaction. Population-scale decoding remains difficult because the relationship between sEMG and neuromuscular activity varies across users and sessions, while task-relevant dynamics span channels and multiple timescales. Learning waveform-to-output mappings from task labels leaves the distinction between recording variability and coordinated motor activity implicit. We introduce the Neuromotor Hierarchy Network (NHN), which learns a compact latent neuromotor state from task supervision to represent task-relevant neuromuscular coordination. NHN constructs this latent state through a hierarchy inspired by neuromotor organization. It adapts recording statistics while preserving relative intensity. Its spatiotemporal encoder uses parameter-efficient channel interactions and modulates features with multi-timescale history. The resulting features yield candidate activations of learned motor primitives, which are temporally integrated and continuously weighted to form the state. Theoretical analysis characterizes the efficiency, temporal behavior, and optimization of NHN's core mechanisms. We evaluate the architecture for both continuous hand-pose estimation on *emg2pose* and touch-typing recognition on *emg2qwerty*. On *emg2pose*, NHN reduces user-averaged angular error by 0.52% to 2.84% across all three generalization splits in both Regression and Tracking relative to Hadidi et al.'s best task-specific variants, using 48.42% to 48.51% fewer parameters. On *emg2qwerty*, NHN reduces beam-search character error rate by 19.40% zero-shot and 30.42% after fine-tuning relative to SplashNet-Upscale, using 65.86% fewer parameters. Physiology-guided inference of a latent neuromotor state supports parameter-efficient sEMG decoding. **Code:** `https://anonymous.4open.science/r/nhn-review-artifact-B7C1/README.md`.

1 INTRODUCTION

Surface electromyography (sEMG) supports prosthetic control, gesture input, continuous hand-pose estimation, and touch-typing recognition without cameras or implanted sensors (Atzori et al., 2014; Farina et al., 2014; Kaifosh et al., 2025). Population-scale datasets pair wrist sEMG with diverse hand kinematics (Salter et al., 2024) or natural typing (Sivakumar et al., 2024) to support cross-user learning. Decoders must resolve fine-grained activity while limiting sensitivity to waveform variation across users and sessions. Resource-constrained wearables also require parameter efficiency.

Larger training populations and stronger temporal models have advanced pose and typing decoding (Kaifosh et al., 2025; Mehlman et al., 2025; Hadidi et al., 2026). Progress also comes from constraining what these models need to learn. SplashNet adapts signal statistics and shares weights between the two hands to improve typing efficiency (Hadidi et al., 2025), while EMBridge (Cui et al., 2026) transfers pose structure to electromyographic representations through cross-modal learning. These advances leave open how to organize a compact representation of task-relevant neuromuscular coordination while limiting sensitivity to recording-specific variation. The physiology of sEMG generation can guide what this representation retains and how it is organized.

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.

The neuromotor pathway links motor-neuron activity to coordinated muscle activation, giving rise to both movement and surface electrical signals (Figure 1). At the neural level, motor neurons integrate synaptic input over time, while recruitment and discharge-rate modulation shape graded motor output (Negro et al., 2016; Powers & Heckman, 2017; Heckman & Enoka, 2012). Their discharges evoke action potentials in muscle fibers, initiating contraction through excitation-contraction coupling (Heckman & Enoka, 2012). Across muscles, activation forms coordinated patterns that vary in strength and timing, producing forces that act through the musculoskeletal system to generate movement (d'Avella et al., 2003; Bizzi & Cheung, 2013). The same muscle-fiber action potentials also give rise to the electrical signals recorded at the skin, where tissue filtering and volume conduction shape the mixture observed by surface electrodes. Anatomy, electrode placement, contact, and gain affect the scale and spatial distribution of the recorded waveform (Farina et al., 2002; 2014; 2024).

Thus, sEMG and hand kinematics share a neuromotor origin, but their relationship varies with recording conditions, anatomy, and individual motor control (Farina et al., 2014; Kaifosh et al., 2025). Aligned latent representations support cross-session and cross-individual decoding in cortical and sEMG studies (Gallego et al., 2020; Safaie et al., 2023; Al-Mashhadani et al., 2025). We hypothesize that a compact latent neuromotor state can accommodate this variability by accounting for measurement effects and representing coordination through physiologically structured components (d'Avella et al., 2003). Their activation strength and timing can vary within a shared representation to accommodate individual motor patterns and reduce reliance on recording- or user-specific waveforms.

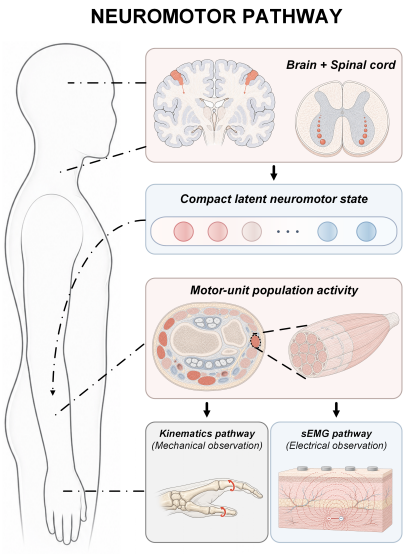


Figure 1: Physiological inspiration. Neuromotor coordination links hand motion and sEMG, motivating a compact latent neuromotor state.

The Neuromotor Hierarchy Network (NHN) implements this inference process (Figure 2). Its measurement adapter adjusts recording-dependent waveform statistics while retaining relative intensity. A causal spatiotemporal encoder combines temporal convolutions and parameter-efficient interactions, then modulates features using multi-timescale activity histories. A compact projection maps these features to non-negative candidate drives, which are causally integrated before Henneman-inspired graded allocation forms the compact latent neuromotor state. The state sequence and retained intensity cues support task-specific decoding. We provide theoretical support for these architectural choices (Appendix A).

We make three contributions.

- We introduce NHN, a physiology-guided sEMG decoding architecture that infers a compact, task-aligned latent neuromotor state to support generalization across users and sessions.
- NHN combines learnable causal adaptation that preserves relative intensity, parameter-efficient channel interactions, and feature regulation based on activity histories. Non-negative drives prevent cancellation during integration, while continuous graded allocation regulates each primitive's contribution.
- We evaluate a common NHN architecture with task-specific prediction heads on emg2pose and emg2qwerty, demonstrating its applicability to both continuous hand-pose estimation and discrete character-sequence recognition.

2 Related Work

Generalization in sEMG decoding. Population-scale training supports calibration-free interfaces (Kaifosh et al., 2025), while benchmarks (Yang et al., 2024) distinguish intersubject transfer from adaptation and broaden demographic coverage (Gowda et al., 2025). Target-domain adaptation uses voting-based pseudo-label refinement (Duan et al., 2023), memory-based labeling in spiking networks

(Guo et al., 2024), and class-level alignment (Zhong et al., 2025). Disentanglement preserves shared and target-specific features (Su et al., 2025a), while adapter expansion and rehearsal limit forgetting across new users (Chen et al., 2024b). Pretraining supports few-shot personalization through self-supervised task replay (Yoon et al., 2025). Source training learns cross-posture invariance through Fourier-phase distillation (Su et al., 2025b) and reusable patterns through quantized myoelectric codebooks (Huang et al., 2026). Architectures use short-window attention for noisy gesture decoding (Guo et al., 2025) and linear state-space models for typing (Ortone et al., 2026). We exclude the latter from quantitative comparisons because evaluation protocols differ. NHN organizes task-relevant coordination with physiological priors in a compact decoder.

Transferable representations for physiological signals. For brain recordings, geometry-aware encodings accommodate sensor layouts (El Ouahidi et al., 2025; Xiao et al., 2025), while factorized and cross-scale attention model spatiotemporal dependencies (Wang et al., 2025; Zhou et al., 2025). Temporal-spectral tokens (Ma et al., 2026), multiscale wavelets (Chen et al., 2025), and physiological frequency bands for ear-recorded signals (Yoon et al., 2026) structure time-frequency content. Pretraining combines masked reconstruction with latent alignment (Wang et al., 2024) or learns text-aligned tokens (Jiang et al., 2025). Optical and cardiac models exploit pulse morphology (Pillai et al., 2025) and electrical-to-peripheral timing (Zhou et al., 2026). Low-rank weight generation personalizes models using local data (Wu et al., 2025). For task-supervised sEMG, the question is how physiological priors can jointly guide measurement processing and activity representation. NHN unifies them in a neuromotor-inspired hierarchy.

Latent neuromotor states and motor primitives. Motor-state representations differ in scale and behavioral relevance. Simulation benchmarks (Mamidanna et al., 2025) and separation algorithms (Tacca et al., 2025) support decomposition into individual motor-unit discharge trains. Aligned population recordings reveal cross-animal regularities (Safaie et al., 2023), while neural trajectories exhibit dynamical constraints (Oby et al., 2025). Dynamical embeddings capture cross-recording variation (Vermani et al., 2025), input-driven models separate intrinsic dynamics from measured-input effects (Vahidi et al., 2025), and multi-region models distinguish shared and region-specific subspaces (Sristi et al., 2026). Behavior further constrains representations through movement-supervised neural embeddings (Chen et al., 2024a) and sparse-input dynamical models of larval zebrafish locomotion (Soares Mullen et al., 2024). Synergy-based control uses low-rank policies (Dong et al., 2022) or state-dependent synergies (He et al., 2024), while synergy-based neural manifolds link population activity to hand kinematics (Sun et al., 2025). For sEMG, the challenge is to infer a compact, task-relevant latent neuromotor state from mixed surface measurements. NHN organizes its non-negative components through temporal integration and graded allocation under task supervision.

3 Method: Neuromotor Hierarchy Network

Inspired by neuromotor organization, NHN learns a compact latent neuromotor state for sEMG decoding. Measurement adaptation and spatiotemporal encoding provide evidence of coordination, which the network organizes into graded primitives for task-specific prediction (Figure 2).

Let $\boldsymbol{x} \in \mathbb{R}^{C\times T_0}$ contain T_0 samples from C sEMG channels, and let κ index the task. Denote measurement adaptation by $\mathcal{A}$, spatiotemporal encoding by $\mathcal{E}$, latent neuromotor state construction by $\mathcal{B}$, and task-specific prediction by $\mathcal{H}_\kappa$. For pose estimation, the data flow is

$$(\tilde{\boldsymbol{x}}, \boldsymbol{q}) = \mathcal{A}(\boldsymbol{x}), \quad \boldsymbol{f} = \mathcal{E}(\tilde{\boldsymbol{x}}), \quad \boldsymbol{p} = \mathcal{B}(\boldsymbol{f}), \quad \hat{\boldsymbol{y}} = \mathcal{H}_\kappa(\boldsymbol{p}, \boldsymbol{q}^{\downarrow}). \tag{1}$$

The adapted waveform $\tilde{\boldsymbol{x}}$ and relative intensity $\boldsymbol{q}$ retain the input dimensions. Aligning $\boldsymbol{q}$ to the encoder's time grid gives $\boldsymbol{q}^{\downarrow} \in \mathbb{R}^{C\times L_\kappa}$. The encoded sequence $\boldsymbol{f} \in \mathbb{R}^{D\times L_\kappa}$ has width D, while the latent neuromotor state $\boldsymbol{p} \in \mathbb{R}_{\geq 0}^{N\times L_\kappa}$ contains N primitives over the same L_κ frames. Throughout, bold feature symbols denote sequences; a time subscript selects one column, and two subscripts select a scalar entry. We use the rectified linear unit (ReLU) and layer normalization (LN).

3.1 Measurement Adaptation with Intensity Preservation

Tissue filtering and electrode sampling motivate adjusting recording-dependent scale and covariance while retaining short-term intensity cues (De Luca & Merletti, 1988; Farina et al., 2014).

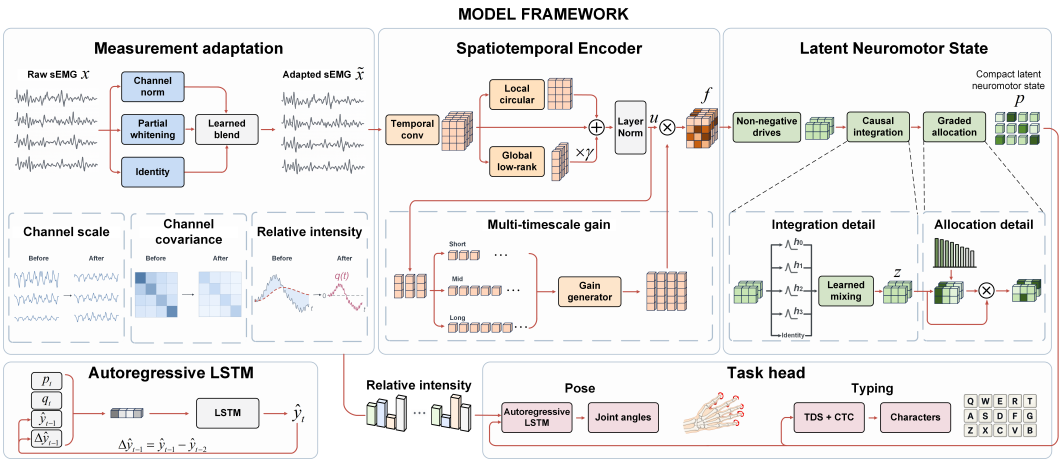


Figure 2: NHN architecture. Task-trained NHN adapts and encodes sEMG, then integrates and allocates candidate drives to form a compact latent neuromotor state for pose and typing decoding.

For input sample $n = 1, \ldots, T_0$, let $\ell_{c,n}$ be channel c's log standard deviation over a causal window ending at n. The intensity path centers it against its cumulative mean. Partial whitening $\mathcal{W}$ uses the channel mean $\boldsymbol{\mu}_b$ and covariance $\boldsymbol{\Sigma}_b$ of samples preceding each block b,

$$q_{c,n} = \ell_{c,n} - \frac{1}{n}\sum_{i=1}^{n} \ell_{c,i},$$
$$\mathcal{W}(\boldsymbol{x})_n = \left[(1-\alpha)\,\mathrm{Diag}(\boldsymbol{\Sigma}_b) + \alpha\boldsymbol{\Sigma}_b + \lambda_b \boldsymbol{I}_C\right]^{-1/2}(\boldsymbol{x}_n - \boldsymbol{\mu}_b), \quad n \in b. \tag{2}$$

Here Diag retains the covariance diagonal and $\boldsymbol{I}_C$ is the identity matrix. Fixed $\alpha \in [0, 1]$ interpolates between channel-wise normalization and joint whitening without a principal-component rotation. The regularizer $\lambda_b > 0$ is computed from mean channel variance. The first block uses cumulative diagonal normalization, with warmup confined to the input's left context. For pose, the waveform path blends $\mathcal{W}(\boldsymbol{x})$ with $\mathcal{D}(\boldsymbol{x})$, which normalizes using cumulative channel-wise means and standard deviations. Both operate on the original $\boldsymbol{x}$, with learned residual strengths $\eta_\mathrm{d}, \eta_\mathrm{w} \in \mathbb{R}$,

$$\tilde{\boldsymbol{x}} = (1-\eta_\mathrm{w})\left[\boldsymbol{x} + \eta_\mathrm{d}\big(s_0\mathcal{D}(\boldsymbol{x}) - \boldsymbol{x}\big)\right] + \eta_\mathrm{w} s_0 \mathcal{W}(\boldsymbol{x}). \tag{3}$$

The once-calibrated, fixed scale $s_0 > 0$ puts both candidates on the input's amplitude scale. $\boldsymbol{q}$ retains intensity for task-dependent fusion.

3.2 Spatiotemporal Encoding of Adapted sEMG

The encoder combines temporal convolution and local-global mixing in time-depth separable (TDS) blocks (Hannun et al., 2019), followed by multi-timescale adaptive gain to produce $\boldsymbol{f}$.

Temporal feature extraction. Within a block r of width D_r, $\boldsymbol{u}^\mathrm{in}$, $\boldsymbol{u}^\mathrm{mid}$, and $\boldsymbol{u}^\mathrm{out}$ denote its input, temporal features, and output, respectively. Convolutional input layers map $\tilde{\boldsymbol{x}}$ to the first block's $\boldsymbol{u}^\mathrm{in}$. Temporal convolution $\mathcal{T}_r$ produces

$$\boldsymbol{u}^\mathrm{mid} = \mathrm{LN}\big(\mathrm{crop}_r(\boldsymbol{u}^\mathrm{in}) + \mathrm{ReLU}\big(\mathcal{T}_r(\boldsymbol{u}^\mathrm{in})\big)\big). \tag{4}$$

crop_r retains trailing input frames to match the unpadded convolution's output length.

Local-global mixing of temporal features. Motivated by local correlations and distributed muscle coordination (Farina et al., 2014; d'Avella et al., 2003), local-global mixing combines shared local interactions with a low-rank path connecting distant groups. The global path uses the Gaussian error linear unit (GELU). At block frame t, the residual output is

$$\boldsymbol{u}_t^\mathrm{out} = \mathrm{LN}\left(\boldsymbol{u}_t^\mathrm{mid} + \mathcal{C}_k^{(r)}(\boldsymbol{u}_t^\mathrm{mid}) + \gamma_r \boldsymbol{W}_\mathrm{up}^{(r)}\,\mathrm{GELU}(\boldsymbol{W}_\mathrm{down}^{(r)}\boldsymbol{u}_t^\mathrm{mid})\right). \tag{5}$$

The local operator $\mathcal{C}_k^{(r)}$ applies two width-k circular convolutions separated by ReLU, sharing weights across G feature groups arranged on a ring, each containing D_r/G features. The global projections $\boldsymbol{W}_{\text{down}}^{(r)}$ and $\boldsymbol{W}_{\text{up}}^{(r)}$ project from width D_r to R and back, with a learned gate γ_r. Projection biases are omitted in Equation 5. Appendix A.1 characterizes this interaction structure and its parameter efficiency. Each block has independent parameters and passes $\boldsymbol{u}^{\text{out}}$ to the next block's $\boldsymbol{u}^{\text{in}}$, with convolutional projection when width or rate changes. A final projection, when needed, yields $\boldsymbol{u} \in \mathbb{R}^{D \times L_\kappa}$ for adaptive gain.

Multi-timescale contextualization. The mixed features $\boldsymbol{u}$ are reweighted using causal history at multiple timescales to capture brief and sustained activity, motivated by common drive and muscle coordination (Negro et al., 2016; d'Avella et al., 2003). Each of M context states accumulates the same input $\boldsymbol{u}_t$ with a different learned timescale,

$$\boldsymbol{s}_t^{(m)} = \left(1 - e^{-\Delta t/\tau_m}\right) \boldsymbol{u}_t + e^{-\Delta t/\tau_m} \boldsymbol{s}_{t-1}^{(m)}, \qquad m = 1, \ldots, M. \tag{6}$$

On the final encoder grid $t = 0, \ldots, L_\kappa - 1$, the context states $\boldsymbol{s}_t^{(m)} \in \mathbb{R}^D$ use frame interval Δt, learned $\tau_m \geq \Delta t$, and learned initial states $\boldsymbol{s}_{-1}^{(m)}$. A learned map $g_a : \mathbb{R}^{MD} \to \mathbb{R}^D$ converts the concatenated states into a residual gain on the current features,

$$\boldsymbol{f}_t = \boldsymbol{u}_t \odot \left[1 + \tanh\Big(g_a\Big([\boldsymbol{s}_t^{(1)}; \ldots; \boldsymbol{s}_t^{(M)}]\Big)\Big)\right], \tag{7}$$

where $\odot$ denotes elementwise multiplication.

3.3 Latent Neuromotor State Construction from Encoded Features

Through drive projection, temporal integration, and graded allocation, $\mathcal{B}$ maps contextualized features $\boldsymbol{f}$ into a compact latent neuromotor state $\boldsymbol{p}$, limiting the width passed to the prediction head.

Non-negative candidate drives. A learned pointwise affine map g_d, layer normalization, and Softplus produce N non-negative candidate drives per frame,

$$\boldsymbol{d}_t = \text{Softplus}(\text{LN}(g_d(\boldsymbol{f}_t))) \in \mathbb{R}_{\geq 0}^N. \tag{8}$$

Non-negativity prevents cancellation during integration. Low-dimensional coordination motivates the compact drive space (Tresch et al., 1999; d'Avella et al., 2003; Bizzi & Cheung, 2013).

Temporal integration of candidate drives. The candidate drives $\boldsymbol{d}$ are integrated before allocation, inspired by synaptic input integration (Powers & Heckman, 2017). Diverse motor-unit contraction times motivate heterogeneous temporal support (Burke et al., 1973; Heckman & Enoka, 2012), implemented with causal finite impulse response (FIR) paths. Combining M_f filtered versions of $\boldsymbol{d}$ with an identity path gives

$$\bar{\boldsymbol{d}}_t = \boldsymbol{\omega}_0 \odot \boldsymbol{d}_t + \sum_{m=1}^{M_f} \boldsymbol{\omega}_m \odot (\mathbf{k}_m *_{\text{c}} \boldsymbol{d})_t \,. \tag{9}$$

Each $\mathbf{k}_m$ contains a non-negative geometric kernel per primitive, with independently learned positive time constants. Numerical normalization gives kernel mass at most one (Appendix B.2). The causal operator $*_{\text{c}}$ filters each primitive separately, using left zero-padding to preserve length. Learned weights $\boldsymbol{\omega}_m \in \mathbb{R}^N$ are softmax-normalized per primitive across paths, including the identity path $m = 0$. These filters integrate candidate drives, whereas adaptive gain reweights encoder features. Appendix A.2 bounds both history aggregations.

Henneman-inspired allocation of integrated drives. Whereas muscle-synergy studies motivate the low-dimensional drive space, Henneman's size principle motivates heterogeneous participation priorities among the learned drive components (Henneman, 1957; Milner-Brown et al., 1973). Behavior-dependent motor-unit control motivates activity-dependent allocation (Marshall et al., 2022). NHN combines learned recruitment priorities $\boldsymbol{\beta} \in \mathbb{R}^N$ with the current drive at a drive-dependent temperature θ_t,

$$\theta_t = \max\{10^{-4}, \text{mean}(\text{Top}_K(\bar{\boldsymbol{d}}_t))\}, \qquad \boldsymbol{p}_t = N\bar{\boldsymbol{d}}_t \odot \text{softmax}\left(\frac{\bar{\boldsymbol{d}}_t + \boldsymbol{\beta}}{\theta_t}\right). \tag{10}$$

For fixed $K \in \{1, \ldots, N\}$, Top_K returns the K largest drive entries, whose mean calibrates temperature. Softmax weights all primitives. Multiplication by $\bar{\boldsymbol{d}}_t$ preserves continuous magnitudes. Collecting $\boldsymbol{p}_t$ across frames gives $\boldsymbol{p} = \mathcal{B}(\boldsymbol{f})$. Appendices A.3-A.4 analyze priority-controlled allocation and convergence of fixed-drive priority fitting.

3.4 Task-specific Prediction Heads

The prediction head $\mathcal{H}_\kappa$ maps the latent neuromotor state sequence $\boldsymbol{p}$ to joint-angle estimates $\hat{\boldsymbol{y}}$ using an autoregressive long short-term memory (LSTM) network, or to character probabilities using TDS with connectionist temporal classification (CTC) (Graves et al., 2006).

4 Experiments

We evaluate NHN on *emg2pose* and *emg2qwerty*, assessing generalization across users and recording conditions alongside inference efficiency. Both tasks use independently trained weights, encoder output width $D = 64$, and $N = 32$ primitives. The backbone takes $C = 16$ sEMG channels from a single wrist at 2 kHz. It processes one wrist for pose and each wrist separately for typing. The NHN backbone and prediction heads use causal operations. Appendix B.4 details architecture and temporal alignment.

Pose estimation on emg2pose. We use the data splits and evaluation protocol of Salter et al. (2024). The User, Stage, and User×Stage splits hold out users, movement stages, or both. Evaluation uses 5-s windows. Tracking receives a ground-truth initial pose, whereas Regression does not. An autoregressive LSTM head predicts joint angles at the 50-Hz rollout rate used by vemg2pose in Salter et al. (2024). We train separate Regression and Tracking models with joint supervision, adopting the task-weighting scheme of Hadidi et al. (2026). Metrics are angular error (AE) in degrees and landmark distance (LD) in millimeters. Dataset and evaluation protocols are detailed in Appendix C.1.

Typing recognition on emg2qwerty. We use emg2qwerty (Sivakumar et al., 2024) with SplashNet's data splits and evaluation settings (Hadidi et al., 2025). We pretrain NHN on pooled data from training users. On held-out users, we evaluate both the pretrained model without adaptation (zero-shot) and models obtained through subject-specific fine-tuning. Other-domain validation (ODV) uses separate validation users, whereas test-domain validation (TDV) and test-domain test (TDT) use validation and test sessions of held-out test users. Fine-tuned comparisons use TDV and TDT. We use a 100-Hz latent neuromotor state sequence to balance temporal resolution and computational cost, consistent with the feature rate used by Mehlman et al. (2025). NHN shares backbone weights between wrists and concatenates their state sequences for a TDS prediction head. Training uses CTC loss. We report character error rate (CER) under greedy decoding and beam search with a 6-gram language model (LM) trained on WikiText-103. Dataset and evaluation protocols are detailed in Appendix C.1.

Training and statistics. For pose training and typing pretraining, we train NHN from scratch using adaptive moment estimation with decoupled weight decay (AdamW), cosine learning-rate schedules, and five random seeds. We report the mean ± standard deviation (SD) across participants after averaging each participant's scores over seeds. Participant-level baseline statistics are reproduced as published. Training and evaluation details are provided in Appendices C.2 and C.3. Appendix D.8 reports per-seed results and across-seed SD.

Baselines and efficiency. The emg2pose baseline shares the dataset's name and directly predicts joint angles (Salter et al., 2024). Other pose baselines include vemg2pose, NeuroPose (Liu et al., 2021), Sensing Dynamics (Sîmpetru et al., 2022), the position (Pos) and velocity (Vel), single-task (ST) and multi-task (MT) variants of Hadidi et al. (2026), and Static. These pose comparisons use published results under matching evaluation protocols. We additionally evaluate available pretrained checkpoints at the session level. Typing comparisons use the TDS convolutional network (TDS ConvNet) (Sivakumar et al., 2024), SplashNet (Hadidi et al., 2025), and distilled Transformers (Mehlman et al., 2025). We report model parameters (Params.) and floating-point operations (FLOPs), expressed in billions (GFLOPs), for 5-s pose spans and 30-s typing inputs. These counts cover neural-network inference, including prediction heads, and exclude language-model and text-decoding costs. Baseline configurations and FLOPs accounting are detailed in Appendices C.4 and C.3.

Table 1: emg2pose Regression and Tracking. Split columns report user mean ± SD. Session columns weight sessions within each split, then splits equally (Appendix C.3). Bold marks column minima within each setting, excluding Static for efficiency. Dashes denote unreported values. Published split results: [a]Hadidi et al. (2026), [b,c,d]Salter et al. (2024).

Model	User		Stage		User×Stage		Session		Efficiency	
	AE↓ (°)	LD↓ (mm)	AE↓ (°)	LD↓ (mm)	AE↓ (°)	LD↓ (mm)	AE↓ (°)	LD↓ (mm)	Params. (10^6)↓	GFLOPs per 5 s↓
(a) Regression										
Static[a]	16.86±1.80	25.17±2.84	19.18±1.71	28.57±2.46	18.87±2.01	28.77±2.78	17.87	26.81	0	0
Sensing Dynamics[b]	15.50±1.40	21.80±2.10	18.80±1.60	26.60±2.00	18.70±1.60	27.20±2.00	–	–	–	–
NeuroPose[c]	13.20±1.10	17.50±1.30	17.20±1.70	24.00±2.10	17.50±1.50	24.90±1.70	15.52	21.61	6.35	**1.12**
emg2pose[d]	12.57±1.30	16.28±1.81	15.17±1.59	20.53±2.13	15.57±1.30	21.49±1.71	14.01	18.94	**2.95**	2.21
vemg2pose[d]	12.24±1.34	15.82±1.90	15.22±1.58	20.38±2.07	15.63±1.37	21.30±1.87	13.91	18.69	5.98	3.72
Pos-ST[a]	12.19±1.27	15.79±1.80	14.96±1.63	20.23±2.15	15.47±1.30	21.29±1.82	–	–	5.97	3.72
Vel-ST[a]	12.17±1.31	15.58±1.85	14.62±1.65	19.62±2.21	15.35±1.32	20.92±1.87	–	–	5.98	3.72
Pos-MT[a]	11.54±1.19	14.74±1.64	14.02±1.62	18.61±2.16	14.58±1.29	19.78±1.83	–	–	5.97	3.72
Vel-MT[a]	11.64±1.25	14.74±1.74	13.85±1.62	18.33±2.16	14.63±1.34	19.77±1.95	–	–	5.98	3.72
NHN (Ours)	**11.38**±1.13	**14.42**±1.53	**13.78**±1.57	**18.21**±2.06	**14.22**±1.26	**19.26**±1.88	**12.73**	**16.88**	3.08	2.40
(b) Tracking										
Static[a]	10.48±1.81	15.71±2.98	14.86±1.91	21.81±2.58	14.29±1.86	21.48±2.33	13.16	19.99	0	0
emg2pose[d]	11.33±1.10	14.75±1.53	15.39±1.50	20.88±1.92	15.05±1.22	20.69±1.39	13.44	18.23	**2.95**	**2.21**
vemg2pose[d]	7.70±0.98	10.22±1.54	11.20±1.43	15.18±1.81	11.03±1.01	15.35±1.46	9.84	13.56	5.97	3.72
Pos-ST[a]	7.35±0.92	9.67±1.48	10.21±1.32	13.73±1.69	10.25±0.93	14.18±1.35	–	–	5.97	3.72
Vel-ST[a]	7.64±0.94	10.02±1.52	10.76±1.34	14.43±1.71	10.75±0.93	14.82±1.40	–	–	5.97	3.72
Pos-MT[a]	7.36±0.90	9.68±1.46	10.28±1.33	13.81±1.70	10.29±0.97	14.22±1.37	–	–	5.97	3.72
Vel-MT[a]	7.64±0.95	10.00±1.53	10.77±1.36	14.44±1.75	10.75±0.97	14.81±1.41	–	–	5.97	3.72
NHN (Ours)	**7.30**±0.88	**9.55**±1.41	**10.12**±1.31	**13.59**±1.69	**10.16**±0.99	**14.03**±1.33	**9.08**	**12.39**	3.08	2.40

5 Results and Discussion

5.1 Main results

On *emg2pose*, NHN's advantage over vemg2pose is more pronounced when generalization includes unseen movement stages (Table 1). In both Regression and Tracking, absolute reductions in user-level AE and LD are larger on Stage and User×Stage than on User. For example, Regression AE decreases by $1.44°$ on Stage compared with $0.86°$ on User. Session-equal aggregation within each split preserves the advantage: AE decreases by 8.48% in Regression and 7.73% in Tracking, with corresponding LD improvements. Against the stronger variants of Hadidi et al. (2026), which retain the vemg2pose core architecture with revised decoding and training, NHN achieves lower user-level AE and LD with 48.42% to 48.51% fewer parameters and 35.55% to 35.64% less computation. Hadidi et al.'s best-performing variants differ between tasks (Vel-MT for Regression and Pos-ST for Tracking), whereas NHN uses a common architecture and joint-supervision formulation.

On *emg2qwerty*, NHN's gains are present before language-model-assisted decoding and persist after individual adaptation (Table 2). On TDT, greedy CER decreases by 6.16 percentage points relative to SplashNet-Upscale (Hadidi et al., 2025) in zero-shot evaluation and by 2.98 points relative to its Shared variant after fine-tuning. Relative beam-search CER reductions against these baselines are 19.40% and 30.42%, respectively. NHN outperforms TDS ConvNet and all SplashNet variants across the evaluated splits, settings, and decoding methods. These gains accompany 65.86% fewer parameters and 74.99% less neural-network computation than SplashNet-Upscale. The Transformer baselines achieve lower TDT greedy CER, but with larger parameter counts and greater computation.

Table 2: emg2qwerty CER (%). NHN reports mean ± SD across participants. Bold marks column minima within each setting. Dashes denote unreported results. ODV is not evaluated after fine-tuning. [a]Sivakumar et al. (2024) (ODV and efficiency from [b]). [b]Hadidi et al. (2025). [c]Mehlman et al. (2025).

Model	ODV		TDV		TDT		Efficiency	
	Greedy↓ (%)	Beam↓ (%)	Greedy↓ (%)	Beam↓ (%)	Greedy↓ (%)	Beam↓ (%)	Params.↓ (10^6)	GFLOPs↓ per 30 s
(a) Zero-shot								
TDS ConvNet[a]	72.44	72.07	55.57	52.10±5.54	55.38	51.78±4.61	5.29	61.61
SplashNet (Split-only)[b]	61.74	58.64	45.73	37.28±6.91	45.69	37.37±7.34	2.68	36.84
SplashNet-mini (Shared)[b]	61.07	58.20	45.33	36.46±7.09	45.26	36.41±7.30	1.38	36.84
SplashNet-Upscale[b]	60.16	56.95	44.79	35.49±7.56	44.78	35.67±6.79	2.58	71.38
Transformer-tiny[c]	–	–	–	–	**31.90**	–	2.20	∼9720
Transformer-small[c]	–	–	–	–	32.70	–	5.40	∼18720
NHN (Ours)	**56.93**±13.82	**51.73**±15.32	**39.03**±7.08	**29.29**±8.99	38.62±7.01	**28.75**±8.29	**0.88**	**17.85**
(b) Subject-specific fine-tuning								
TDS ConvNet[a]			11.39	8.31±3.19	11.28	6.95±3.61	5.29	61.61
SplashNet (Split-only)[b]			12.67	6.04±2.81	12.89	6.10±3.32	2.68	36.84
SplashNet-mini (Shared)[b]			13.01	6.13±2.96	13.22	5.87±3.04	1.38	36.84
SplashNet-mini (Unshared)[b]			12.68	5.85±2.83	13.07	5.96±3.28	2.68	36.84
SplashNet-Upscale (Shared)[b]			12.10	5.46±2.60	12.39	5.51±2.81	2.58	71.38
SplashNet-Upscale (Unshared)[b]			11.80	5.57±2.65	12.13	5.67±2.97	5.06	71.38
Transformer-tiny[c]			–	–	8.60	–	2.20	∼9720
Transformer-small[c]			–	–	**7.10**	–	5.40	∼18720
NHN (Ours)			**9.12**±3.76	**3.95**±1.62	9.41±3.51	**3.83**±1.51	**0.88**	**17.85**

5.2 Learned organization of the latent neuromotor state

Figure 3 shows how NHN's physiology-inspired hierarchy learns different patterns in the compact latent neuromotor state for pose estimation and typing. Its primitives (Section 3.3) are the individual state dimensions, labeled components in the figure. Relative activation is each primitive's share of the summed activation across primitives.

On emg2pose, relative primitive activations have overlapping associations with joint angles across fingers in the sampled validation data (Figure 3a). These associations suggest that hand posture is represented through the combined activity of multiple primitives, consistent with NHN's coordination-inspired design. Training increases all three adaptive-gain timescales but decreases mean integration lags, the kernel-weighted average age of candidate drives (Figure 3b). For continuous pose estimation, this suggests using longer context to reweight encoder features through adaptive gain, while emphasizing recent candidate drives when constructing the state. Allocation weights set the relative emphasis on integrated candidate drives (Equation 10). They remain spread across primitives, with the 12 largest accounting for about half the total allocation weight in the illustrated clip (Figure 3c).

On emg2qwerty, primitive-activation patterns across the sampled training windows show higher mean similarity for the same key than for different keys within participants, across all nine analyzed keys (Figure 3d). These recurring patterns suggest that repeated keystrokes reuse similar primitive combinations. Training decreases the short (S) and long (L) adaptive-gain timescales but increases the intermediate (M) one, while mean integration lags decrease (Figure 3e). For discrete typing events, this suggests using intermediate context for adaptive gain while emphasizing recent candidate drives during state construction, rather than extending history at every gain scale. Allocation is also spread across primitives with unequal weights in the illustrated typing clip (Figure 3f). Together, these patterns suggest that NHN learns to weigh recent and earlier activity differently for adaptive gain and primitive-wise drive integration, with task-specific preferences within the same architecture.

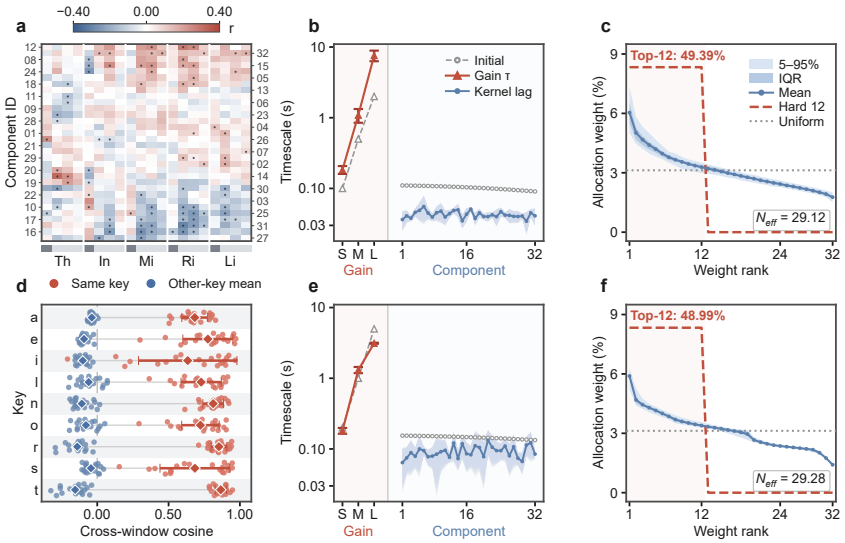


Figure 3: Learned organization of the compact latent neuromotor state. Pose Regression: (a) participant-mean Pearson correlations between relative primitive activations and joint angles. Dots mark $\geq$ 80% participant agreement with the mean sign. Dark/light strips indicate abduction-adduction/flexion-extension. Th/In/Mi/Ri/Li denote thumb/index/middle/ring/little fingers. (b) Initial and post-training adaptive-gain timescales (τ_m, Equation 6) and mean integration lags, kernel-weighted mean ages of candidate drives (Equation 9), with five-run mean±SD. S, M, and L denote short, intermediate, and long timescales. (c) Mean within-frame ranked allocation weights, with dark interquartile range (IQR) and light 5-95% frame bands. Hard-12/Uniform assign equal weights to 12/32 primitives. Typing: (d) within-participant cross-window key-profile cosine similarity. Circles denote participant-model observations, diamonds means, bars participant SD. (e) Initial and post-training adaptive-gain timescales and mean integration lags (five-run mean±SD). (f) Ranked allocation weights with bands and reference distributions as in (c). Details: Appendix D.2.

On the sampled validation data, uniform allocation, with equal allocation weights across primitives, increases pose AE by 3.94° and typing greedy CER by 42.12 percentage points compared with learned allocation (Appendix D.2). These increases occur at unchanged total state activation, indicating that the prediction heads rely on its distribution across primitives, not just its overall strength. These results complement the ablations and priority-allocation analysis (Appendices D.1 and A.3).

6 Limitations

NHN's learned components are task representations rather than genuine physiological sources. Generalization is evaluated on wrist sEMG. Physical-device deployment and transfer to other sensor layouts remain untested. At longer fixed tracking horizons, the reported NHN errors are higher, although its error remains below vemg2pose's and the gap between them is larger (Appendix D.3).

7 Conclusion

NHN uses a physiology-inspired hierarchy to infer a compact latent neuromotor state from wrist sEMG for hand-pose estimation and touch-typing recognition. The learned organization takes different forms: distributed joint-angle associations in pose and recurring key-specific combinations in typing, accompanied by distinct temporal preferences. Allocation interventions show that decoding uses relative activation patterns beyond total intensity. Theoretical analysis characterizes the underlying computational mechanisms. Together with the generalization and efficiency results, these findings support an architectural approach in which physiological organization guides representation structure while task supervision shapes its content.

AI USE STATEMENT

We used large language models extensively for iterative manuscript revision, including language polishing and improvements to clarity and readability, as well as for assistance with code writing and debugging during implementation. We also used a generative image tool to produce an initial draft of the anatomical schematic in Figure 1, which we subsequently modified and refined manually to better reflect the relevant physiological structures. We reviewed the AI-assisted text for technical accuracy and reviewed and tested the AI-assisted code before using it in our experiments. The authors take responsibility for the final manuscript, figures, code, and reported results, including all content developed with AI assistance.

ETHICS STATEMENT

We use the publicly available emg2pose and emg2qwerty datasets and collected no additional human-participant data. The original studies report ethics approval and participant informed consent (Salter et al., 2024; Sivakumar et al., 2024). We use both datasets in accordance with their respective licenses.

REPRODUCIBILITY STATEMENT

Appendix A states the assumptions and provides proofs for the theoretical results. Architecture and implementation settings are specified in Appendix B, and dataset splits, training settings, evaluation procedures, and metric definitions in Appendix C. Per-run results and across-seed standard deviations are reported in Appendix D.8. The anonymous repository linked in the abstract provides inference and evaluation code for both pose tasks and for zero-shot and fine-tuned typing. It includes per-run configurations, five checkpoints each for pose Regression, pose Tracking, and pretrained typing, and 40 subject-specific fine-tuned typing checkpoints spanning five runs and eight participants. For fine-tuned typing, the evaluation code selects the checkpoint corresponding to each participant and training run. Full-session typing evaluation supports both greedy decoding and beam search with the character-level language model. Its README describes environment setup, dataset access, and checkpoint restoration. After the review process concludes, we will release the complete codebase used for training, subject-specific fine-tuning, and the analyses reported in this work.

REFERENCES

Zubaidah Al-Mashhadani, Trevor Overton, Di Wu, and Mohsen Rakhshan. Toward plug and play myoelectric control via one-shot latent representation alignment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 33:4409–4421, 2025. doi: 10.1109/TNSRE.2025.3626255.

Manfredo Atzori, Arjan Gijsberts, Claudio Castellini, Barbara Caputo, Anne-Gabrielle Mittaz Hager, Simone Elsig, Giorgio Giatsidis, Franco Bassetto, and Henning Müller. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific Data*, 1:140053, 2014. doi: 10.1038/sdata.2014.53.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113506–113543, 2025. doi: 10.52202/085713-3786. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4ca07aa108036f80cbb5b82285fd4b1-Paper-Conference.pdf.

Emilio Bizzi and Vincent C. K. Cheung. The neural origin of muscle synergies. *Frontiers in Computational Neuroscience*, 7:51, 2013. doi: 10.3389/fncom.2013.00051.

R. E. Burke, D. N. Levine, P. Tsairis, and F. E. Zajac, III. Physiological types and histochemical profiles in motor units of the cat gastrocnemius. *The Journal of Physiology*, 234(3):723–748, 1973. doi: 10.1113/jphysiol.1973.sp010369.

Chenggang Chen, Zhiyu Yang, and Xiaoqin Wang. Neural embeddings rank: Aligning 3D latent dynamics with movements. In *Advances in Neural Information Processing Systems*, volume 37,

pp. 141461–141489, 2024a. doi: 10.52202/079017-4492. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ffe78e2b3c80439e6dfd3f7f38cfa888-Paper-Conference.pdf.

Xingjian Chen, Weiyu Guo, Chuang Lin, Ning Jiang, and Jingyong Su. Cross-subject lifelong learning for continuous estimation from surface electromyographic signal. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 32:1965–1973, 2024b. doi: 10.1109/TNSRE.2024.3400535.

Yanlong Chen, Mattia Orlandi, Pierangelo Rapa, Simone Benatti, Luca Benini, and Yawei Li. PhysioWave: A multi-scale wavelet-transformer for physiological signal representation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 169357–169399, 2025. doi: 10.52202/085713-5639. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f7a7bb369e48f10e85fce85b67d8c516-Paper-Conference.pdf.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179.pdf.

Wenhui Cui, Christopher M. Sandino, Hadi Pouransari, Ran Liu, Juri Minxha, Ellen L. Zippi, Erdrin Azemi, and Behrooz Mahasseni. EMBridge: Enhancing gesture generalization from EMG signals through cross-modal representation learning. In *International Conference on Learning Representations*, pp. 149668–149684, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f216728558e6f93f815ea1552315c22d-Paper-Conference.pdf.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Andrea d'Avella, Philippe Saltiel, and Emilio Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience*, 6(3):300–308, 2003. doi: 10.1038/nn1010.

Peter Dayan, Geoffrey E. Hinton, Radford M. Neal, and Richard S. Zemel. The Helmholtz machine. *Neural Computation*, 7(5):889–904, 1995. doi: 10.1162/neco.1995.7.5.889.

Carlo J. De Luca and Roberto Merletti. Surface myoelectric signal cross-talk among muscles of the leg. *Electroencephalography and Clinical Neurophysiology*, 69(6):568–575, 1988. doi: 10.1016/0013-4694(88)90169-1.

Heng Dong, Tonghan Wang, Jiayuan Liu, and Chongjie Zhang. Low-rank modular reinforcement learning via muscle synergy. In *Advances in Neural Information Processing Systems*, volume 35, pp. 19861–19873, 2022. doi: 10.52202/068431-1444. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/7da6005a8d6942e8b328357da2872aed-Paper-Conference.pdf.

Di Duan, Huanqi Yang, Guohao Lan, Tianxing Li, Xiaohua Jia, and Weitao Xu. EMGSense: A low-effort self-supervised domain adaptation framework for EMG sensing. In *2023 IEEE International Conference on Pervasive Computing and Communications*, pp. 160–170. IEEE, 2023. doi: 10.1109/PERCOM56429.2023.10099164.

Yassine El Ouahidi, Jonathan Lys, Philipp Thölke, Nicolas Farrugia, Bastien Pasdeloup, Vincent Gripon, Karim Jerbi, and Giulia Lioi. REVE: A foundation model for EEG – adapting to any setup with large-scale pretraining on 25,000 subjects. In *Advances in Neural Information Processing Systems*, volume 38, pp. 22541–22577, 2025. doi: 10.52202/085713-0760. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/20a917f77773ac0fa8bea2bdd6606b66-Paper-Conference.pdf.

Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. DDSP: Differentiable digital signal processing. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=B1x1ma4tDr.

Dario Farina, Corrado Cescon, and Roberto Merletti. Influence of anatomical, physical, and detection-system parameters on surface EMG. *Biological Cybernetics*, 86(6):445–456, 2002. doi: 10.1007/s00422-002-0309-2.

Dario Farina, Roberto Merletti, and Roger M. Enoka. The extraction of neural strategies from the surface EMG: an update. *Journal of Applied Physiology*, 117(11):1215–1230, 2014. doi: 10.1152/japplphysiol.00162.2014.

Dario Farina, Roberto Merletti, and Roger M. Enoka. The extraction of neural strategies from the surface EMG: 2004–2024. *Journal of Applied Physiology*, 138(1):121–135, 2024. doi: 10.1152/japplphysiol.00453.2024.

Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, 1985. doi: 10.1523/JNEUROSCI.05-07-01688.1985.

Juan A. Gallego, Matthew G. Perich, Raeed H. Chowdhury, Sara A. Solla, and Lee E. Miller. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature Neuroscience*, 23(2):260–270, 2020. doi: 10.1038/s41593-019-0555-4.

Harshavardhana T. Gowda, Neha Kaul, Carlos Carrasco, Marcus Battraw, Safa Amer, Saniya Kotwal, Selena Lam, Zachary McNaughton, Ferdous Rahimi, Sana Shehabi, Jonathon Schofield, and Lee M. Miller. A database of upper limb surface electromyogram signals from demographically diverse individuals. *Scientific Data*, 12:517, 2025. doi: 10.1038/s41597-025-04825-z.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 369–376. ACM, 2006. doi: 10.1145/1143844.1143891.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/pdf?id=uYLFoz1vlAC.

Weiyu Guo, Ying Sun, Yijie Xu, Ziyue Qiao, Yongkui Yang, and Hui Xiong. SpGesture: Source-free domain-adaptive sEMG-based gesture recognition with jaccard attentive spiking neural network. In *Advances in Neural Information Processing Systems*, volume 37, pp. 36717–36747, 2024. doi: 10.52202/079017-1157. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/409334f42cbb57d07aa152f2d0433ec7-Paper-Conference.pdf.

Weiyu Guo, Ziyue Qiao, Ying Sun, Yijie Xu, and Hui Xiong. Revisiting noise resilience strategies in gesture recognition: Short-term enhancement in sEMG analysis. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20903–20920. PMLR, 2025. URL https://proceedings.mlr.press/v267/guo25h.html.

Nima Hadidi, Jason Chan, Ebrahim Feghhi, and Jonathan C. Kao. SplashNet: Split-and-share encoders for accurate and efficient typing with surface electromyography. In *Advances in Neural Information Processing Systems*, volume 38, pp. 166117–166141, 2025. doi: 10.52202/085713-5537. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f29af347aedbdaa316043aa8457a8fb6-Paper-Conference.pdf.

Nima Hadidi, Johannes Lee, Ebrahim Feghhi, Michael Yuan, and Jonathan C. Kao. Re-evaluating position and velocity decoding for hand pose estimation with surface electromyography. *arXiv preprint arXiv:2603.08212*, 2026. doi: 10.48550/arXiv.2603.08212. URL https://arxiv.org/abs/2603.08212.

Awni Hannun, Ann Lee, Qiantong Xu, and Ronan Collobert. Sequence-to-sequence speech recognition with time-depth separable convolutions. In *Proc. Interspeech 2019*, pp. 3785–3789, 2019. doi: 10.21437/Interspeech.2019-2460. URL https://www.isca-archive.org/interspeech_2019/hannun19_interspeech.pdf.

Kaibo He, Chenhui Zuo, Chengtian Ma, and Yanan Sui. DynSyn: Dynamical synergistic representation for efficient learning and control in overactuated embodied systems. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 18115–18132. PMLR, 2024. URL https://proceedings.mlr.press/v235/he24o.html.

C. J. Heckman and Roger M. Enoka. Motor unit. *Comprehensive Physiology*, 2(4):2629–2682, 2012. doi: 10.1002/j.2040-4603.2012.tb00465.x.

Elwood Henneman. Relation between size of neurons and their susceptibility to discharge. *Science*, 126(3287):1345–1347, 1957. doi: 10.1126/science.126.3287.1345.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Zhenghao Huang, Huilin Yao, Kaikai Wang, and Lin Shu. Reading your actions: Learning generalizable action representations via pre-training AEMG. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20108–20117, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Huang_Reading_Your_Actions_Learning_Generalizable_Action_Representations_via_Pre-training_AEMG_CVPR_2026_paper.pdf.

Wei-Bang Jiang, Yansen Wang, Bao-Liang Lu, and Dongsheng Li. NeuroLM: A universal multi-task foundation model for bridging the gap between language and EEG signals. In *International Conference on Learning Representations*, pp. 55436–55457, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/8b4add8b0aa8749d80a34ca5d941c355-Paper-Conference.pdf.

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. doi: 10.1038/s41586-025-09255-w.

Yilin Liu, Shijia Zhang, and Mahanth Gowda. Neuropose: 3D hand pose tracking using EMG wearables. In *Proceedings of the Web Conference 2021*, pp. 1471–1482. ACM, 2021. doi: 10.1145/3442381.3449890.

Jingying Ma, Feng Wu, Qika Lin, Yucheng Xing, Chenyu Liu, Ziyu Jia, and Mengling Feng. CodeBrain: Bridging decoupled tokenizer and multi-scale architecture for EEG foundation model. In *International Conference on Learning Representations*, pp. 83526–83574, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/86d1ab582afb247ccaa84bec4a7e24f7-Paper-Conference.pdf.

Pranav Mamidanna, Thomas Klotz, Dimitrios Halatsis, Agnese Grison, Irene Mendez-Guerra, Shihan Ma, Arnault H. Caillet, Simon Avrillon, Robin Rohlén, and Dario Farina. MUniverse: A simulation and benchmarking suite for motor unit decomposition. In *Advances in Neural Information Processing Systems*, volume 38, pp. 166708–166733, 2025. doi: 10.52202/085713-5020. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/dcce28990bf662c6b916c52cd7e975da-Paper-Datasets_and_Benchmarks_Track.pdf.

Najja J. Marshall, Joshua I. Glaser, Eric M. Trautmann, Elom A. Amematsro, Sean M. Perkins, Michael N. Shadlen, L. F. Abbott, John P. Cunningham, and Mark M. Churchland. Flexible neural control of motor units. *Nature Neuroscience*, 25(11):1492–1504, 2022. doi: 10.1038/s41593-022-01165-8.

Nicholas Mehlman, Jean-Christophe Gagnon-Audet, Michael Shvartsman, Kelvin Niu, Alexander H. Miller, and Shagun Sodhani. Scaling and distilling transformer models for sEMG. *Transactions on Machine Learning Research (TMLR)*, 2025. URL https://openreview.net/pdf?id=hFPWThwUiZ.

H. S. Milner-Brown, R. B. Stein, and R. Yemm. The orderly recruitment of human motor units during voluntary isometric contractions. *The Journal of Physiology*, 230(2):359–370, 1973. doi: 10.1113/jphysiol.1973.sp010192.

Francesco Negro, Utku Şükrü Yavuz, and Dario Farina. The human motor neuron pools receive a dominant slow-varying common synaptic input. *The Journal of Physiology*, 594(19):5491–5505, 2016. doi: 10.1113/JP271748.

Emily R. Oby, Alan D. Degenhart, Erinn M. Grigsby, Asma Motiwala, Nicole T. McClain, Patrick J. Marino, Byron M. Yu, and Aaron P. Batista. Dynamical constraints on neural population activity. *Nature Neuroscience*, 28(2):383–393, 2025. doi: 10.1038/s41593-024-01845-7.

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. doi: 10.1016/j.bspc.2026.111215.

Arvind Pillai, Dimitris Spathis, Fahim Kawsar, and Mohammad Malekzadeh. PaPaGei: Open foundation models for optical physiological signals. In *International Conference on Learning Representations*, pp. 48230–48261, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7871f51d332c8661cc729539d7987b21-Paper-Conference.pdf.

Randall K. Powers and Charles J. Heckman. Synaptic control of the shape of the motoneuron pool input-output function. *Journal of Neurophysiology*, 117(3):1171–1184, 2017. doi: 10.1152/jn.00850.2016.

Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999. doi: 10.1038/4580.

Mostafa Safaie, Joanna C. Chang, Junchol Park, Lee E. Miller, Joshua T. Dudman, Matthew G. Perich, and Juan A. Gallego. Preserved neural dynamics across animals performing similar behaviour. *Nature*, 623(7988):765–771, 2023. doi: 10.1038/s41586-023-06714-0.

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*, volume 37, pp. 55703–55728, 2024. doi: 10.52202/079017-1770. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/64f8884f6ba3d9ace5bb647c4f917896-Paper-Datasets_and_Benchmarks_Track.pdf.

Marco Santello, Martha Flanders, and John F. Soechting. Postural hand synergies for tool use. *The Journal of Neuroscience*, 18(23):10105–10115, 1998. doi: 10.1523/JNEUROSCI.18-23-10105.1998.

John P. Scholz and Gregor Schöner. The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental Brain Research*, 126(3):289–306, 1999. doi: 10.1007/s002210050738.

Raul C. Sîmpetru, Andreas Arkudas, Dominik I. Braun, Marius Osswald, Daniela Souza de Oliveira, Bjoern Eskofier, Thomas M. Kinfe, and Alessandro Del Vecchio. Sensing the full dynamics of the human hand with a neural interface and deep learning. *bioRxiv*, 2022. doi: 10.1101/2022.07.29.502064.

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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 91373–91389, 2024. doi: 10.52202/079017-2899. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a64d53074d011e49af1dfc72c332fe4b-Paper-Datasets_and_Benchmarks_Track.pdf.

Thomas Soares Mullen, Marine Schimel, Guillaume Hennequin, Christian K. Machens, Michael B. Orger, and Adrien Jouary. Learning interpretable control inputs and dynamics underlying animal locomotion. In *International Conference on Learning Representations*, pp. 24002–24024, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6882dbdc34bcd094e6f858c06ce30edb-Paper-Conference.pdf.

Ram Dyuthi Sristi, Sowmya Manojna Narasimha, Jingya Huang, Alice Despatin, Simon Musall, Vikash Gilja, and Gal Mishne. Coupled transformer autoencoder for disentangling multi-region neural latent dynamics. In *International Conference on Learning Representations*, pp. 71457–71488, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/747ef138fdb0f3d68c76e6b6b885594b-Paper-Conference.pdf.

Kejia Su, Kai Liu, Bo Wan, Hanbing Qiao, Jiayang Huang, Min Feng, and Jinhui Liu. Multisource adversarial feature disentanglement method for cross-subject gesture recognition using sEMG signals. *IEEE Transactions on Instrumentation and Measurement*, 74:1–12, 2025a. doi: 10.1109/TIM.2025.3557823.

Tanying Su, Xin Tan, Xinyu Jiang, Xiao Liu, Bo Hu, and Chenyun Dai. A dynamic balanced single-source domain generalization model for cross-posture myoelectric control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 33:255–265, 2025b. doi: 10.1109/TNSRE.2024.3521229.

Huaqin Sun, Zhengyi Wang, Yu Qi, and Yueming Wang. Decoding multi-joint hand movements from brain signals by learning a synergy-based neural manifold. *Patterns*, 6(11):101394, 2025. doi: 10.1016/j.patter.2025.101394.

Nicholas Tacca, Bryan R. Schlink, Jackson T. Levine, Mary K. Heimann, Collin Dunlap, Samuel C. Colachis, IV, Philip Putnam, Matthew A. Zeglen, Daniel J. Brobston, Austin M. Bollinger, José L. Pons, Lauren Wengerd, Eric C. Meyers, and David A. Friedenberg. Decomposing motor units through elimination for real-time intention driven assistive neurotechnology. In *Advances in Neural Information Processing Systems*, volume 38, pp. 122841–122860, 2025. doi: 10.52202/085713-4096. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b213d870740582dd6af77bbdaed900c9-Paper-Conference.pdf.

Emanuel Todorov and Michael I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, 2002. doi: 10.1038/nn963.

Matthew C. Tresch, Philippe Saltiel, and Emilio Bizzi. The construction of movement by the spinal cord. *Nature Neuroscience*, 2(2):162–167, 1999. doi: 10.1038/5721.

Parsa Vahidi, Omid G. Sani, and Maryam M. Shanechi. BRAID: Input-driven nonlinear dynamical modeling of neural-behavioral data. In *International Conference on Learning Representations*, pp. 2940–2969, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/082d3d795520c43214da5123e56a3a34-Paper-Conference.pdf.

Ayesha Vermani, Josue Nassar, Hyungju Jeon, Matthew Dowling, and Il Memming Park. Meta-dynamical state space models for integrative neural data analysis. In *International Conference on Learning Representations*, pp. 84623–84646, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d3222559698f41247261b7a6c2bbaedc-Paper-Conference.pdf.

Guagnyu Wang, Wenchao Liu, Yuhong He, Cong Xu, Lin Ma, and Haifeng Li. EEGPT: Pretrained transformer for universal and reliable representation of EEG signals. In *Advances in Neural Information Processing Systems*, volume 37, pp. 39249–39280, 2024. doi: 10.52202/079017-1239. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/4540d267eeec4e5dbd9dae9448f0b739-Paper-Conference.pdf.

Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Haiteng Jiang, Shijian Li, Tao Li, and Gang Pan. CBraMod: A criss-cross brain foundation model for EEG decoding. In *International Conference on Learning Representations*, pp. 75310–75346, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/bbbd6d915cb90be21c1254a82d45cedd-Paper-Conference.pdf.

Chenrui Wu, Haishuai Wang, Xiang Zhang, Chengqi Zhang, and Jiajun Bu. Efficient personalized adaptation for physiological signal foundation model. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67833–67851. PMLR, 2025. URL https://proceedings.mlr.press/v267/wu25ah.html.

Qinfan Xiao, Ziyun Cui, Chi Zhang, Siqi Chen, Wen Wu, Andrew Thwaites, Alexandra Woolgar, Bowen Zhou, and Chao Zhang. BrainOmni: A brain foundation model for unified EEG and MEG signals. In *Advances in Neural Information Processing Systems*, volume 38, pp. 41179–41212, 2025. doi: 10.52202/085713-1375. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/3aef4a18c2646b9d11d9ab4d9bec72c8-Paper-Conference.pdf.

Jehan Yang, Maxwell Soh, Vivianna Lieu, Douglas J. Weber, and Zackory Erickson. EMGBench: Benchmarking out-of-distribution generalization and adaptation for electromyography. In *Advances in Neural Information Processing Systems*, volume 37, pp. 50313–50342, 2024. doi: 10.52202/079017-1593. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/59fe60482e2e5faf557c37d121994663-Paper-Datasets_and_Benchmarks_Track.pdf.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, pp. 29687–29707, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/4904fad153f6434a7bcf04465d4be2cc-Paper-Conference.pdf.

Hyungjun Yoon, Jaehyun Kwak, Biniyam Aschalew Tolera, Gaole Dai, Mo Li, Taesik Gong, Kimin Lee, and Sung-Ju Lee. SelfReplay: Adapting self-supervised sensory models via adaptive meta-task replay. In *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*, pp. 226–239. ACM, 2025. doi: 10.1145/3715014.3722066.

Hyungjun Yoon, Seungjoo Lee, Yu Yvonne Wu, Xiaomeng Chen, Taiting Lu, Freddy Yifei Liu, Taeckyung Lee, Hyeongheon Cha, Haochen Zhao, Gaoteng Zhao, Dongyao Chen, Cecilia Mascolo, Sung-Ju Lee, and Lili Qiu. Beyond hearing: Learning task-agnostic ExG representations from earphones via physiology-informed tokenization. In *International Conference on Learning Representations*, pp. 34511–34534, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/3a5abb38fcb597bff82f481c8f1fedca-Paper-Conference.pdf.

Xiao-Cong Zhong, Qisong Wang, Dan Liu, Xuefu Wang, Rui Li, Yunfei Wang, Meiyan Zhang, and Jinwei Sun. Plug-and-play sEMG-driven hand gesture recognition with subdomain adaptation for exoskeleton rehabilitation gloves. *IEEE Transactions on Instrumentation and Measurement*, 74: 1–10, 2025. doi: 10.1109/TIM.2024.3502881.

Hao Zhou, Simon A. Lee, Cyrus Tanade, Keum San Chun, Juhyeon Lee, Migyeong Gwak, Megha Thukral, Justin Sung, Eugene Hwang, Mehrab Bin Morshed, Li Zhu, Viswam Nathan, Md Mahbubur Rahman, Subramaniam Venkatraman, and Sharanya Arcot Desai. Physiology-aware masked cross-modal reconstruction for biosignal representation learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64405.

Yuchen Zhou, Jiamin Wu, Zichen Ren, Zhouheng Yao, Weiheng Lu, Kunyu Peng, Qihao Zheng, Chunfeng Song, Wanli Ouyang, and Chao Gou. CSBrain: A cross-scale spatiotemporal brain foundation model for EEG decoding. In *Advances in Neural Information Processing Systems*, volume 38, pp. 87150–87195, 2025. doi: 10.52202/085713-2915. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/7e199ad8ae40eb19b2980f61f659cb07-Paper-Conference.pdf.

APPENDIX

TABLE OF CONTENTS

A Mathematical Properties of the Architecture **18**
A.1 Parameter-efficient local-global interactions . . . 18
A.2 Bounded temporal integration and residual initialization . . . 18
A.3 Priority-regulated continuous allocation . . . 19
A.4 Convergence of relative-priority fitting . . . 20

B Implementation Details **21**
B.1 Task-supervised construction of the latent neuromotor state . . . 21
B.2 Measurement conditioning and discrete temporal operators . . . 21
B.3 Recurrent pose prediction and character emissions . . . 22
B.4 Architecture and implementation settings . . . 22

C Detailed Experimental Protocol **24**
C.1 Datasets . . . 24
C.2 Training . . . 24
C.3 Metrics and decoding . . . 26
C.4 Baselines . . . 27

D Additional Results and Ablations **27**
D.1 Architecture ablations and sensitivity . . . 27
D.1.1 Measurement adaptation and spatiotemporal encoding . . . 28
D.1.2 Latent neuromotor state construction and width . . . 28
D.1.3 Backbone and prediction-head capacity . . . 30
D.1.4 Task supervision, conditioning, and augmentation . . . 30
D.2 Figure 3: sampling and aggregation . . . 31
D.3 Long-horizon tracking without re-anchoring . . . 33
D.4 Qualitative tracking visualization . . . 35
D.5 CTC emission behavior . . . 35
D.6 Replacing the character language model with GPT-2 . . . 37
D.7 Session-weighted pose evaluation . . . 38
D.8 Performance across training seeds . . . 39

E Numerical Constants Quick Reference **39**

F Complementary Perspectives on Representation Learning **41**
F.1 Task objectives and motor redundancy . . . 41
F.2 Temporal coverage and selective memory . . . 41
F.3 Structured generation and differentiable synthesis . . . 41

A Mathematical Properties of the Architecture

The Neuromotor Hierarchy Network (NHN) learns a task-supervised compact latent neuromotor state through a physiology-inspired hierarchy. We analyze the parameter cost of mixing, bounded history aggregation, continuous allocation, and convergence of fixed-drive priority fitting. Notation follows Section 3.

A.1 Parameter-efficient local-global interactions

For the local-global mixer in Equation 5, we quantify parameter cost and interaction structure.

Proposition: parameter budget and differential structure. For a block of width D_r, let $d_g = D_r/G$ be the number of features per group. The two local convolutions and the global projection pair contain

$$P_{\text{mix}} = 2kd_g^2 + 2RD_r \tag{11}$$

weight parameters, excluding biases, normalization parameters, and the scalar gate. Two dense $D_r \times D_r$ layers contain $2D_r^2$ weights. Before residual addition and output layer normalization, the local correction has cyclic group support of at most $\min(G, 2k-1)$ offsets for an odd contiguous kernel of width k. At every differentiable input, the Jacobian of the global correction has rank at most R.

Proof. Each local convolution uses k matrices of size $d_g \times d_g$, shared across all G groups. The global down- and up-projections each contain RD_r weights, proving Equation 11. The nonlinearities are the rectified linear unit (ReLU) and Gaussian error linear unit (GELU). For a frame $\mathbf{v} = \boldsymbol{u}_t^{\text{mid}}$, write the local correction as $\mathbf{A}_2\,\mathrm{ReLU}(\mathbf{A}_1\mathbf{v} + \mathbf{b}_1) + \mathbf{b}_2$, where $\mathbf{A}_1, \mathbf{A}_2$ are the block-circulant matrices induced by the convolutions. On an open ReLU activation region, its Jacobian is

$$J_{\text{local}}(\mathbf{v}) = \mathbf{A}_2\,\mathrm{Diag}(\mathbf{1}_{\mathbf{A}_1\mathbf{v}+\mathbf{b}_1>0})\,\mathbf{A}_1. \tag{12}$$

The two kernel supports add, giving at most $\min(G, 2k-1)$ cyclic offsets. The activation mask can vary across groups, making the Jacobian input-dependent while preserving this support.

The global correction has Jacobian

$$J_{\text{global}}(\mathbf{v}) = \gamma_r \boldsymbol{W}_{\text{up}}^{(r)}\,\mathrm{Diag}\Big(\mathrm{GELU}'(\boldsymbol{W}_{\text{down}}^{(r)}\mathbf{v} + \mathbf{b}_{\text{down}})\Big)\,\boldsymbol{W}_{\text{down}}^{(r)}, \tag{13}$$

which factors through $\mathbb{R}^R$ and therefore has rank at most R. The implementation's elementwise clipping adds a diagonal derivative factor wherever differentiable and preserves this rank bound. □

The global path connects features beyond the local support. For $G = 16$, $k = 5$, and $R = 32$, the weight counts are 18,944 versus 131,072 at $D_r = 256$, and 4,256 versus 8,192 at $D_r = 64$, reducing mixing weights by approximately $6.92\times$ and $1.92\times$ at unchanged block widths.

A.2 Bounded temporal integration and residual initialization

In real arithmetic, NHN's context recurrence and finite impulse response (FIR) integration give bounded aggregates of spatiotemporal encoder features and non-negative candidate drives, respectively.

Proposition: causal averaging. For fixed, finite model parameters and positive adaptive-gain timescales, let $a_m = e^{-\Delta t/\tau_m} \in (0,1)$. The context recurrence in Equation 6 has the exact expansion

$$\boldsymbol{s}_t^{(m)} = a_m^{t+1}\boldsymbol{s}_{-1}^{(m)} + (1-a_m)\sum_{j=0}^{t} a_m^j \boldsymbol{u}_{t-j}. \tag{14}$$

For any vector norm,

$$\|\boldsymbol{s}_t^{(m)}\| \le \max\Big(\|\boldsymbol{s}_{-1}^{(m)}\|, \max_{0\le j\le t}\|\boldsymbol{u}_j\|\Big). \tag{15}$$

For non-negative drives extended by zero to negative times, let $h_i[j]$ be primitive i's effective causal kernel, including its identity path. Non-negative path kernels with sums at most one and softmax path weights give

$$h_i[j] \geq 0, \qquad \sum_{j\geq 0} h_i[j] \leq 1, \qquad 0 \leq \bar{d}_{i,t} \leq \max_{0\leq j\leq t} d_{i,j}. \tag{16}$$

For fixed kernels and the same left-padding convention, integration also satisfies $\|\bar{\boldsymbol{d}} - \bar{\boldsymbol{d}}'\|_\infty \leq \|\boldsymbol{d} - \boldsymbol{d}'\|_\infty$, with the maximum taken over components and frames.

Proof. Unrolling the recurrence gives Equation 14, whose non-negative coefficients sum to $a_m^{t+1} + (1 - a_m)\sum_{j=0}^{t} a_m^j = 1$. Convexity gives the context-state norm bound. For drive integration, $h_i[j] = \omega_{0,i}\mathbf{1}_{j=0} + \sum_{m=1}^{M_f} \omega_{m,i} k_{m,i}[j]$. The path weights are non-negative and sum to one, so $\sum_j h_i[j] = \omega_{0,i} + \sum_m \omega_{m,i} \sum_j k_{m,i}[j] \leq 1$. Non-negative drives and zero padding then give Equation 16. Applying the same kernel to $\boldsymbol{d} - \boldsymbol{d}'$ and using the triangle inequality proves non-expansion in the sequence maximum norm. □

The additive FIR stabilizer reduces kernel mass from one to below one (Appendix B.2). Equation 14 also gives recurrent-convolutional equivalence for convolution over the full available history with the same initial-state term. Learned adaptive-gain timescales, FIR time constants, and path weights determine temporal weighting while FIR tap counts remain fixed.

Residual gain and initialization. Each multiplier $1 + \tanh(\cdot)$ lies in $(0, 2)$, bounding each feature-wise gain by two. With typing's preceding intensity modulation by g_q, this gives $|f_{i,t}| \leq 4|u_{i,t}|$, relative to features before both modulations. Zero final affine layers in g_a and g_q initialize these factors to one. Likewise, $\gamma_r = 0$ removes the global correction, and $\boldsymbol{\beta} = 0$ removes additive priority offsets while retaining drive-dependent soft allocation.

For integration-path logits initialized to 2 for identity and 0 for each of $M_f = 4$ filtered paths,

$$\omega_{0,i} = \frac{e^2}{e^2 + 4} \approx 0.65, \qquad |\bar{d}_{i,t} - d_{i,t}| \leq (1 - \omega_{0,i}) \max_{0\leq j\leq t} d_{i,j}. \tag{17}$$

The filtered values and current drive share a non-negative interval, proving the bound. Thus initialization combines unit feature gains with an identity-dominant drive path.

A.3 Priority-regulated continuous allocation

Allocation combines integrated drives with learned priorities to control the magnitude and participation pattern of the latent neuromotor state. Fix a frame and write $\mathbf{a} = \bar{\boldsymbol{d}}_t \geq 0$, $\epsilon_\theta = 10^{-4}$, and

$$\theta(\mathbf{a}) = \max\left\{\epsilon_\theta, \frac{1}{K}\sum_{i\in I_K(\mathbf{a})} a_i\right\}, \quad \mathbf{w} = \operatorname{softmax}\left(\frac{\mathbf{a} + \boldsymbol{\beta}}{\theta(\mathbf{a})}\right), \quad \boldsymbol{p} = N\mathbf{a} \odot \mathbf{w}. \tag{18}$$

Here $I_K(\mathbf{a})$ indexes the K largest entries. Their sum is the maximum over all size-K subset sums, hence continuous even at ties. The positive temperature floor makes $\mathbf{w}$ and $\boldsymbol{p}$ continuous in drives and priorities. N sets the latent neuromotor state width. K calibrates temperature through the top-K mean, while allocation weights span all primitives.

Proposition: magnitude and scale behavior. For finite logits, every w_i is positive and

$$N \min_i a_i \leq \|\boldsymbol{p}\|_1 = N\sum_i a_i w_i \leq N \max_i a_i. \tag{19}$$

If $c > 0$ and the temperature floor is inactive for both $\mathbf{a}$ and $c\mathbf{a}$, then

$$\mathbf{w}(c\mathbf{a}, \boldsymbol{\beta}) = \mathbf{w}(\mathbf{a}, \boldsymbol{\beta}/c), \qquad \boldsymbol{p}(c\mathbf{a}, \boldsymbol{\beta}) = c\boldsymbol{p}(\mathbf{a}, \boldsymbol{\beta}/c). \tag{20}$$

Under these floor conditions, $\boldsymbol{\beta} = 0$ gives scale-invariant weights and a scale-equivariant latent neuromotor state.

Proof. The total magnitude is a weighted average of Na_i. Positive scaling preserves the drive ordering and, with inactive floors, gives $\theta(c\mathbf{a}) = c\theta(\mathbf{a})$. Substitution proves both identities. □

For positive drives, relative state participation is $p_i/\|\boldsymbol{p}\|_1 = a_i w_i / \sum_j a_j w_j$. Priorities can affect both this pattern and total intensity, motivating the intensity-matched interventions in Section 5.2.

Proposition: bounded priority sensitivity. At fixed $\mathbf{a}$, for any two priority vectors,

$$\|\mathbf{w}(\mathbf{a}, \boldsymbol{\beta}') - \mathbf{w}(\mathbf{a}, \boldsymbol{\beta})\|_2 \le \frac{\|\boldsymbol{\beta}' - \boldsymbol{\beta}\|_2}{2\theta(\mathbf{a})}, \tag{21}$$

$$\|\boldsymbol{p}(\mathbf{a}, \boldsymbol{\beta}') - \boldsymbol{p}(\mathbf{a}, \boldsymbol{\beta})\|_2 \le \frac{N\|\mathbf{a}\|_\infty}{2\theta(\mathbf{a})}\|\boldsymbol{\beta}' - \boldsymbol{\beta}\|_2 \le \frac{NK}{2}\|\boldsymbol{\beta}' - \boldsymbol{\beta}\|_2. \tag{22}$$

The final bound is independent of the drive amplitude.

Proof. Let $\mathbf{S} = \mathrm{Diag}(\mathbf{w}) - \mathbf{w}\mathbf{w}^\top$. The absolute row sum of $\mathbf{S}$ is $2w_i(1 - w_i) \le 1/2$. Since $\mathbf{S}$ is symmetric, $\|\mathbf{S}\|_2 \le 1/2$. Differentiating with respect to priorities gives

$$J_{\boldsymbol{\beta}}\mathbf{w} = \frac{\mathbf{S}}{\theta}, \qquad J_{\boldsymbol{\beta}}\boldsymbol{p} = \frac{N}{\theta}\,\mathrm{Diag}(\mathbf{a})\mathbf{S}. \tag{23}$$

Integrating along the segment between priorities gives the first bounds. For nonzero drives, $\theta \ge \|\mathbf{a}\|_\infty / K$ gives the last inequality. When $\mathbf{a} = 0$, $\boldsymbol{p} = 0$ directly. □

The priority differences act explicitly on log odds, $\log(w_i/w_j) = (a_i - a_j + \beta_i - \beta_j)/\theta$. For $\boldsymbol{\delta} = \boldsymbol{\beta}' - \boldsymbol{\beta}$, define $r_\delta = (\max_i \delta_i - \min_i \delta_i)/\theta$. The identity $w_i' = w_i e^{\delta_i/\theta} / \sum_j w_j e^{\delta_j/\theta}$ gives $e^{-r_\delta} \le w_i'/w_i \le e^{r_\delta}$. Consequently, the effective participation used in Figure 3, $n_{\mathrm{eff}}(\mathbf{w}) = 1/\sum_i w_i^2$, satisfies

$$\left|\log \frac{n_{\mathrm{eff}}(\mathbf{w}')}{n_{\mathrm{eff}}(\mathbf{w})}\right| \le 2r_\delta. \tag{24}$$

Squaring and summing the ratio bounds proves this bound on changes in effective participation.

A.4 Convergence of relative-priority fitting

We analyze gradient descent for priority fitting with fixed integrated drives. Finite Softplus inputs and the positive identity-path weight in Equation 9 give strictly positive drives. Allocation is invariant to a common priority shift: $\boldsymbol{p}(\mathbf{a}, \boldsymbol{\beta} + c\mathbf{1}) = \boldsymbol{p}(\mathbf{a}, \boldsymbol{\beta})$ for every scalar c. We therefore use the centered subspace $V = \{\mathbf{v} \in \mathbb{R}^N : \mathbf{1}^\top \mathbf{v} = 0\}$.

Lemma: identifiable relative priorities. For a fixed strictly positive drive vector, the map $\boldsymbol{\beta} \mapsto \boldsymbol{p}(\mathbf{a}, \boldsymbol{\beta})$ is injective on V. Its Jacobian has rank $N - 1$. For any $\mathbf{v} \in V$,

$$\|J_{\boldsymbol{\beta}}\boldsymbol{p}\,\mathbf{v}\|_2 \ge \frac{Na_{\min}w_{\min}}{\theta}\|\mathbf{v}\|_2, \qquad a_{\min} = \min_i a_i, \quad w_{\min} = \min_i w_i. \tag{25}$$

Proof. At positive drives, outputs determine every priority difference through $\beta_i - \beta_j = \theta \log(p_i a_j/(p_j a_i)) - (a_i - a_j)$. Centering removes the remaining common shift. Moreover,

$$\mathbf{v}^\top \mathbf{S}\mathbf{v} = \sum_i w_i (v_i - \mathbf{w}^\top \mathbf{v})^2 \ge w_{\min} \sum_i (v_i - \mathbf{w}^\top \mathbf{v})^2 \ge w_{\min}\|\mathbf{v}\|_2^2 \tag{26}$$

The last inequality uses $\sum_i v_i = 0$. Cauchy-Schwarz gives $\|\mathbf{S}\mathbf{v}\|_2 \ge w_{\min}\|\mathbf{v}\|_2$. Equation 23 then proves the lower bound and, since $\mathbf{S}\mathbf{1} = 0$, rank $N - 1$. □

Proposition: convergence rates for priority fitting. For fixed positive drives $\mathbf{a}_1, \ldots, \mathbf{a}_Q$ and finite targets $\mathbf{z}_1, \ldots, \mathbf{z}_Q$, define the allocation-fitting objective

$$F(\boldsymbol{\beta}) = \frac{1}{2Q}\sum_{s=1}^{Q} \|\boldsymbol{p}(\mathbf{a}_s, \boldsymbol{\beta}) - \mathbf{z}_s\|_2^2. \tag{27}$$

This auxiliary objective has a globally Lipschitz gradient with a finite constant $L_F > 0$. For a centered initial point and $0 < \eta \le 1/L_F$, gradient descent satisfies

$$\min_{0\le t<T} \|\nabla F(\boldsymbol{\beta}_t)\|_2^2 \le \frac{2\big(F(\boldsymbol{\beta}_0) - F_{\inf}\big)}{\eta T}, \qquad F_{\inf} = \inf_{\boldsymbol{\beta}\in V} F(\boldsymbol{\beta}). \tag{28}$$

If one $\boldsymbol{\beta}^\star \in V$ realizes all targets, gradient descent with $0 < \eta \le 1/L$ converges linearly from sufficiently nearby centered initialization, as in Equation 30. Here L bounds the local Hessian norm.

Proof. At fixed drives, allocation outputs and their first two priority derivatives are polynomials in softmax weights with fixed finite coefficients. Since the weights lie in the compact unit simplex, the Hessian of F is globally bounded. The descent inequality $F(\boldsymbol{\beta}_{t+1}) \le F(\boldsymbol{\beta}_t) - \frac{\eta}{2}\|\nabla F(\boldsymbol{\beta}_t)\|_2^2$ then yields Equation 28 by summing over T steps and using $F \ge F_{\inf} \ge 0$.

For the local rate, write $\boldsymbol{p}_s = \boldsymbol{p}(\mathbf{a}_s, \boldsymbol{\beta})$ and $J_s = J_{\boldsymbol{\beta}}\boldsymbol{p}_s$. The Gauss-Newton matrix $\mathbf{G} = Q^{-1}\sum_s J_s^\top J_s$ is positive definite on V: Equation 25 gives $\mathbf{v}^\top \mathbf{G}\mathbf{v} = Q^{-1}\sum_s \|J_s\mathbf{v}\|_2^2 > 0$ for every nonzero $\mathbf{v} \in V$. The full Hessian also contains a residual term,

$$\nabla^2 F = \mathbf{G} + \mathbf{R}, \qquad \mathbf{R} = \frac{1}{Q}\sum_{s,i}(p_{s,i} - z_{s,i})\nabla^2_{\boldsymbol{\beta}} p_{s,i}. \tag{29}$$

At the realizable solution, $\nabla F(\boldsymbol{\beta}^\star) = 0$ and $\mathbf{R}(\boldsymbol{\beta}^\star) = 0$, so the Hessian inherits the positive curvature of $\mathbf{G}$ on V. Continuity gives a sufficiently small closed ball in V around $\boldsymbol{\beta}^\star$ with $0 < \mu I_V \preceq \nabla^2 F|_V \preceq L I_V$, where I_V is the identity on V. Gradient descent initialized in this ball with $0 < \eta \le 1/L$ obeys

$$\|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}^\star\|_2 \le (1 - \eta\mu)\|\boldsymbol{\beta}_t - \boldsymbol{\beta}^\star\|_2. \tag{30}$$

This follows by integrating the Hessian along the segment to $\boldsymbol{\beta}^\star$. Shift invariance gives $\nabla F \in V$, and contraction keeps iterates inside the ball, proving local linear convergence for centered initialization. □

Thus arbitrary finite targets admit an $O(1/T)$ stationarity bound, while jointly realizable targets yield local linear convergence of relative priorities.

B Implementation Details

This appendix specifies the computation of the hierarchy in Section 3. Dimensions, sampling rates, and configuration values are given in Appendix B.4.

B.1 Task-supervised construction of the latent neuromotor state

For surface electromyography (sEMG) decoding, NHN learns a compact latent neuromotor state from hand-pose or character supervision, with physiological organization guiding the maps in Equation 1. The N primitives are the individual components of $\boldsymbol{p}$, with $p_i(t)$ denoting the activation of primitive i at frame t. Latent neuromotor state construction $\mathcal{B}$ and task prediction follow $\boldsymbol{f} \longmapsto \boldsymbol{d} \longmapsto \bar{\boldsymbol{d}} \longmapsto \boldsymbol{p} \longmapsto \hat{\boldsymbol{y}}$. Projection supplies non-negative candidate drives, integration combines their histories, and continuous allocation determines their participation in $\boldsymbol{p}$. The relative intensity stream $\boldsymbol{q}^{\downarrow}$ enters at the task-dependent point specified in Appendix B.4.

Operator properties, reduced-data design comparisons, and fixed-model analyses appear in Appendices A and D.1, and Section 5.2, respectively.

B.2 Measurement conditioning and discrete temporal operators

The following conditioning and temporal conventions connect the implemented operators to the bounds in Appendix A and the quantities in Figure 3.

Regularized measurement conditioning. The waveform transform combines channel-wise scale adjustment with cross-channel covariance conditioning. For a fixed block's historical statistics, define

$$\mathbf{Q}_b = (1 - \alpha)\,\mathrm{Diag}(\boldsymbol{\Sigma}_b) + \alpha\boldsymbol{\Sigma}_b + \lambda_b \boldsymbol{I}_C. \tag{31}$$

If $\mathbf{\Sigma}_b \succeq 0$, $\alpha \in [0, 1]$, and $\lambda_b > 0$, both covariance terms are positive semidefinite, giving $\mathbf{Q}_b \succeq \lambda_b \boldsymbol{I}_C$. Its symmetric inverse square root therefore exists and satisfies $\|\mathbf{Q}_b^{-1/2}\|_2 \leq \lambda_b^{-1/2}$. For two samples transformed with the same block statistics,

$$\|\mathcal{W}(\boldsymbol{x})_n - \mathcal{W}(\boldsymbol{x}')_n\|_2 \leq \lambda_b^{-1/2} \|\boldsymbol{x}_n - \boldsymbol{x}'_n\|_2. \tag{32}$$

The regularizer thus bounds amplification along low-variance directions for the frozen block transform. The transformed current-block covariance is $\mathbf{Q}_b^{-1/2} \mathbf{\Sigma}_{\text{current}} \mathbf{Q}_b^{-1/2}$, which need not be the identity because the transform uses regularized historical statistics.

In the pose adapter, Equation 3 has coefficients $(1 - \eta_{\text{w}})(1 - \eta_{\text{d}})$, $(1 - \eta_{\text{w}})\eta_{\text{d}}$, and η_{w} on $\boldsymbol{x}$, $s_0 \mathcal{D}(\boldsymbol{x})$, and $s_0 \mathcal{W}(\boldsymbol{x})$, respectively. These unconstrained learned real coefficients sum to one, making the blend affine rather than necessarily convex. Appendix B.4 specifies typing's specialization and both intensity-fusion interfaces.

Finite integration kernels and lag convention. Mean integration lag is the kernel-weighted mean age of candidate drives. It depends on the learned FIR time constant, coefficient order, finite support, and identity-path weight. For a filtered path with L_m taps and primitive-specific time constant $\tau_{m,i}^{\text{FIR}} > 0$, let $r_{m,i} = \exp(-\Delta t / \tau_{m,i}^{\text{FIR}})$. The coefficients in array order are

$$c_{m,i}[n] = \frac{(1 - r_{m,i}) r_{m,i}^n}{\sum_{v=0}^{L_m - 1} (1 - r_{m,i}) r_{m,i}^v + \epsilon_f}, \quad n = 0, \ldots, L_m - 1, \qquad \epsilon_f = 10^{-7}. \tag{33}$$

With left padding and one-dimensional cross-correlation, coefficient n multiplies $d_{i,t-(L_m-1)+n}$. Thus the kernel indexed by past lag is $k_{m,i}[j] = c_{m,i}[L_m - 1 - j]$, for $0 \leq j < L_m$. Combining all filtered paths with identity gives

$$h_i[j] = \omega_{0,i} \mathbf{1}_{j=0} + \sum_{m=1}^{M_f} \omega_{m,i} k_{m,i}[j], \qquad \ell_i^{\text{eff}} = \Delta t \, \frac{\sum_{j \geq 0} j h_i[j]}{\sum_{j \geq 0} h_i[j]}, \tag{34}$$

where ℓ_i^{eff} is the mean integration lag and path kernels are zero outside their supports. The identity path contributes mass at zero lag. The denominator accounts for the finite-kernel stabilizer, retained in the positivity and boundedness results of Appendix A.2.

Adaptive gain instead summarizes encoder features through recurrences with learned timescales τ_m and decay $e^{-\Delta t / \tau_m}$ on their feature grid (Equation 6), including the learned initial state (Equation 14). Figure 3 compares initial and learned adaptive-gain timescales and mean integration lags, both expressed in seconds across the tasks' different frame rates.

B.3 Recurrent pose prediction and character emissions

The pose head uses a long short-term memory (LSTM) network to model output-space dependencies, with memory distinct from the compact latent neuromotor state $\boldsymbol{p}$. At each prediction step, it receives the resampled state, relative intensity, previous predicted pose, and its one-step difference. A learned projection maps the second layer's hidden output directly to joint angles. Layer widths, initial pose, output scaling, and resampling are specified in Appendix B.4.

For typing, the two wrist-specific latent neuromotor state sequences are concatenated and mapped to framewise character distributions $\boldsymbol{\pi}_t$ by the time-depth separable (TDS) prediction head. Connectionist temporal classification (CTC) trains these distributions against target strings by marginalizing over compatible blank-augmented alignments (Graves et al., 2006). Greedy and language-model-assisted beam decoding operate on these distributions, as detailed in Appendix C.3.

B.4 Architecture and implementation settings

The shared hierarchy uses task-specific temporal resolutions, intensity-fusion points, and prediction heads.

Common architecture. Both tasks use the NHN hierarchy with independent weights, encoder width $D = 64$, and $N = 32$ primitives. Each wrist supplies $C = 16$ channels at 2 kHz. The common convolutional stem maps $16 \to 256 \to 256$ channels with kernel widths 11 and 5 and strides 5 and 2. Two TDS stages each contain two sub-blocks. Local-global mixing uses $G = 16$ feature groups, circular kernel width $k = 5$, and global rank $R = 32$ in all four sub-blocks. All encoder temporal convolutions use valid padding and unit dilation.

Pose instantiation. NHN retains vemg2pose's encoder stride of $5 \times 2 \times 4 \times 2 = 80$ (Salter et al., 2024), mapping one forearm's 2-kHz input to a 25-Hz feature sequence. A final pointwise projection maps 256 stage features to $D = 64$. Both TDS stages use 16 temporal-convolution channels with 16 feature positions each, giving width 256. Their entry convolutions have kernel widths 17 and 9 and strides 4 and 2, while their internal temporal kernels have widths 9 and 5. The adapter learns η_{d} and η_{w} from initial values 1 and 0, using s_0 calibrated once as the ratio of mean absolute amplitudes before and after diagonal normalization. The intensity stream is sampled at each encoder frame's input endpoint and concatenated with the latent neuromotor state. Linear interpolation with aligned endpoints resamples these 48 feature channels to 50 Hz before a two-layer autoregressive LSTM prediction head with 256 hidden units per layer. This retains the baseline's 25-Hz encoding and 50-Hz prediction rates (Salter et al., 2024). At each step, the head's 88-dimensional input contains the 48 features, the previous 20 joint-angle estimates, and their one-step differences as pose-difference feedback. This feedback is zero at initialization and subsequently uses predicted angles. Tracking starts from the ground-truth pose, whereas Regression uses a learned, zero-initialized pose. The prediction head's final linear output uses fixed scales of 0.10 for Tracking and 1 for Regression, following Hadidi et al. (2026). Input weights for intensity and pose-difference feedback start at zero, as do the recurrent hidden and cell states. A leaky rectified linear activation with slope 0.01 on negative inputs and a $256 \to 20$ linear projection produce joint angles at 50 Hz, resampled to the 2 kHz evaluation grid.

Typing instantiation. The typing specialization selects the partial-whitening branch of Equation 3 with $\eta_{\mathrm{w}} = s_0 = 1$, giving $\tilde{\boldsymbol{x}} = \mathcal{W}(\boldsymbol{x})$, and fuses relative intensity within the encoder. Its TDS stages use 16 and 4 temporal-convolution channels, each with 16 feature positions, giving widths 256 and 64. Entry kernels have width 5 with strides 2 and 1, and internal temporal kernels have width 9. The total stride $5 \times 2 \times 2 \times 1 = 20$ yields a 100-Hz state sequence from the 2-kHz input. Block averaging over 20 input samples and right alignment produce $\boldsymbol{q}^{\downarrow}$ at 100 Hz. Before adaptive gain, a zero-initialized affine map $g_q : \mathbb{R}^C \to \mathbb{R}^D$ applies the multiplicative factor $1 + \tanh(g_q(\boldsymbol{q}_t^{\downarrow}))$ to $\boldsymbol{u}_t$. A shared backbone encodes the synchronized left- and right-wrist inputs separately. The two 32-dimensional latent neuromotor state sequences are concatenated into 64 channels and decoded jointly by four causal TDS blocks and a 99-class CTC head, including the blank class. Each prediction-head block has width 64 and a stride-one depthwise temporal kernel of width 31 with left padding 30, retaining a 10-ms emission grid for CTC alignment. Pose inputs contain past context and the scored samples, without added right context or explicit input advance. For typing, we allow limited future context, as in the emg2qwerty and SplashNet setups (Sivakumar et al., 2024; Hadidi et al., 2025), and implement this through a 120 ms input advance, distinct from pose's rate resampling.

Adaptation and temporal settings. Both tasks fix $\alpha = 0.50$, the covariance block length at 2,000 input samples, and the intensity window and normalization warmup at 128 samples. The variance-dependent ridge λ_b uses a multiplier of 10^{-3} and an additive floor of 10^{-6}. Slow-varying common input (Negro et al., 2016) motivates smoothed context, and heterogeneous contraction times (Burke et al., 1973) motivate multiple integration paths. We use $M = 3$ context states for adaptive gain and a $192 \to 16 \to 64$ gain perceptron with ReLU. The adaptive-gain timescales start at $(0.10, 0.50, 2)$ s for pose and $(0.20, 1, 5)$ s for typing, with a lower bound of one encoder interval. Drive integration uses $M_f = 4$ filtered paths plus an identity path, with 3/7/11/15 taps at 25 Hz for pose and 11/25/50/100 taps at 100 Hz for typing. Each filtered path initializes its primitive-wise time constants geometrically between 20 and 100 ms, with typing additionally capping them at the corresponding filter's support duration. Tap counts are fixed, while FIR time constants and path weights are learned. Allocation temperature uses $K = 12$ and a floor of 10^{-4}. Gradients propagate through the drive-dependent allocation temperature. Primitive count, K, group count, kernel widths, and rank are architectural hyperparameters. Sensitivity analyses of selected settings on emg2pose are reported in Appendix D.1.

Initialization and data-driven quantities. The drive projection uses Xavier-uniform weights and zero bias, while global mixing gates, initial context states, gain output layers, and recruitment priorities $\boldsymbol{\beta}$ start at zero. Integration-path logits start at 2 for the identity path and 0 for each filtered path. These quantities, the pose adapter coefficients, time constants, and remaining network weights are learned jointly with the prediction head. In contrast, running means, covariance, relative intensity, λ_b, and allocation temperature are computed from the input. The pose reference scale s_0 is data-calibrated and then held fixed.

Numerical settings. Candidate-drive layer normalization includes learnable scale and bias for each primitive. Input normalization uses an additive stabilizer of 10^{-6}, while whitening eigenvalues are floored at 10^{-12} for pose and 10^{-6} for typing. Pose floors the square-root factor at 10^{-4} in the backward pass. Layer normalization uses PyTorch's default stabilizer of 10^{-5}. Finite-filter normalization adds 10^{-7} to the weight sum. Softplus also uses the framework defaults of unit slope and linear evaluation above 20. Global-path activations are clipped to $[-50, 50]$. Encoder, primitive, and prediction-head dropout rates are zero. Task objectives and training protocols are given in Appendix C.

C Detailed Experimental Protocol

The two benchmarks test the same NHN hierarchy under different output requirements. Pose estimation produces continuous joint trajectories, whereas typing recognition maps a long sEMG sequence to a discrete string. This section specifies the data partitions, task-specific optimization, and aggregation rules used in Section 4. Architecture and temporal alignment are given in Appendix B.4.

C.1 Datasets

emg2pose. The official *emg2pose* release (Salter et al., 2024) contains 193 users and 751 sessions covering 29 movement stages. Its 370 hours count left- and right-hand recordings separately. Each wrist supplies 16 sEMG channels at 2 kHz. Hand motion is captured at 60 Hz, and the resulting joint-angle trajectories are linearly interpolated to the sEMG time grid. Our prediction and angular-error evaluation use 20 finger-related joint angles.

The three generalization splits separate changes in the person from changes in the movement repertoire. User holds out users but retains training stages, Stage holds out stages but retains training users, and User$\times$Stage holds out both. Each split is evaluated with 5-s windows under two tasks. Tracking supplies the ground-truth pose at the start of each window. Regression supplies no initial pose and requires the entire trajectory to be inferred from sEMG. The extended Tracking evaluation in Appendix D.3 changes the interval between pose anchors while retaining these test partitions.

emg2qwerty. We use *emg2qwerty* (Sivakumar et al., 2024): 108 users, 1,135 sessions, 346 hours total. Each session contains synchronized left- and right-wristband sEMG (2×16 differential channels at 2 kHz) paired with ground-truth keystroke sequences annotated with per-character onset timestamps. We pretrain on pooled data from the training users. Zero-shot evaluation applies the pretrained model to held-out users without fine-tuning. For personalized evaluation, we fine-tune it on each user's training sessions (Sivakumar et al., 2024).

We use the evaluation partitions of Hadidi et al. (2025). Other-domain validation (ODV) contains four separate validation users. Test-domain validation (TDV) and test-domain test (TDT) contain validation and test sessions from eight held-out test users. The pretrained model is evaluated on ODV, TDV, and TDT, and fine-tuned models on their respective users' TDV and TDT sessions. Pretraining uses 4-s training and validation windows, following the original emg2qwerty protocol (Sivakumar et al., 2024). Fine-tuning uses 4-s training windows. Following SplashNet (Hadidi et al., 2025), fine-tuning validation and all reported zero-shot and fine-tuned evaluations use complete sessions. Table 3 summarizes the two benchmarks.

C.2 Training

Shared training principles. For pose training and typing pretraining, we train the complete NHN from scratch using adaptive moment estimation with decoupled weight decay (AdamW), with a

Table 3: Datasets and evaluation settings. emg2pose duration sums left- and right-hand recordings.

Property	emg2pose (Salter et al., 2024)	emg2qwerty (Sivakumar et al., 2024)
Users / sessions	193 / 751	108 / 1,135
Total duration	370 hours	346 hours
sEMG input	16 channels, one wrist, 2 kHz	2×16 channels, both wrists, 2 kHz
Ground truth	Finger angles from 60-Hz motion capture	keystroke sequence with onset stamps
Task type	Continuous regression / tracking	Discrete CTC-style symbol decoding
Evaluation span	5 s (Tracking and Regression)	Full session
Evaluation splits	User, Stage, User×Stage	ODV, TDV, TDT
Mean rate per typist	Not applicable	130 to 439 keystrokes/min

10-epoch warmup followed by cosine decay. Channel-axis circular shifts augment the sEMG in accordance with the wristband's ring topology (Hadidi et al., 2025). Learning rates, weight decay, and later personalization schedules are task-specific, as detailed below.

emg2pose objectives and optimization. We train separate NHN models with independent weights for the Regression and Tracking evaluations. Within each training run, the same network and prediction head process each batch twice, once for a Tracking rollout with the ground-truth initial pose and once for a Regression rollout without it. The two rollout losses are weighted by $7/8$ and $1/8$, respectively. Each rollout loss combines joint-angle mean absolute error (MAE) in radians with 0.01 times the mean Euclidean distance between predicted and target fingertips after forward kinematics (FK), in millimeters (Salter et al., 2024). The Regression rollout uses a learned initial pose in the Regression model and a zero initial pose in the Tracking model. At evaluation, each model uses its corresponding rollout mode. Both models use the same architecture. The final head's output scale is 1.00 for the Regression configuration and 0.10 for Tracking, following the task-specific settings of Hadidi et al. (2026). Weight decay is 10^{-5} for measurement adaptation, the spatiotemporal encoder, and latent neuromotor state construction, and 10^{-2} for the prediction head and learned initial pose, without normalization-layer exemptions. Two accumulated batches of 128 give an effective batch size of 256. Gradient clipping is 1.00 for Regression and 0.50 for Tracking. The 300-epoch schedule starts at 10^{-4}, warms up to 10^{-3}, and then follows cosine decay toward 10^{-6}. Training windows contain 11,790 samples, including 1,790 samples of left context and 10,000 scored samples. Augmentations comprise channel shifts in $\{-1, 0, 1\}$, time warping by a factor in $[0.85, 1.18]$, and independent per-channel amplitude scaling in $[0.80, 1.20]$ both before and after measurement adaptation.

emg2qwerty pretraining. The typing model is pretrained end-to-end with a 4-block TDS-CTC head (kernel width 31) and sequence-level CTC supervision. Pretraining uses batch size 128, weight decay 10^{-4} on a single optimizer parameter group, and gradient clipping at 1.00. The learning rate warms from 2×10^{-9} to 2×10^{-4} over 10 epochs, then decays by a cosine schedule toward 10^{-6}. Pretraining runs for at most 200 epochs with early-stopping patience 50.

Subject-specific typing fine-tuning. Each held-out user's model starts from the corresponding pretrained model. Fine-tuning uses learning rate 5×10^{-4}, zero weight decay, and a 10-epoch warmup. A ReduceLROnPlateau schedule monitors validation character error rate (CER) with patience 20 and minimum learning rate 2×10^{-5}. The maximum budget is 450 epochs, with early-stopping patience 50. Fine-tuning uses batch size 128.

Hardware and software stack. All training runs used a single NVIDIA RTX 5090 graphics processing unit (GPU) with 32 GB of memory. The software environment comprised Python 3.10.13, PyTorch $2.10.0 + cu130$, and PyTorch Lightning 2.6.0. For the Pose task, Regression and Tracking training each required approximately 96 hours per run. For the Typing task, an emg2qwerty pretraining run required approximately 20 hours.

C.3 METRICS AND DECODING

emg2pose. Following Salter et al. (2024) we report two primary metrics, both per-frame: **angular error (AE)** in degrees, the mean L_1 distance over the 20 predicted joint angles, and **landmark distance (LD)** in millimeters, the mean Euclidean distance over the benchmark's 17 evaluated hand landmarks after forward kinematics. This differs from the five-fingertip distance used in the training loss. For NHN's user-level columns in Table 1, window-level errors are averaged within each user and seed, then across seeds. We report the mean and sample standard deviation (SD) of these user scores separately for User (20 users), Stage (158), and User×Stage (20). Each window scores 5.00 seconds, excluding left context. Tracking receives one initial-pose anchor per window, whereas Regression receives none. The user-weighted overall AE and LD weight the three split means equally rather than pooling all users or windows. Across seeds, these Regression scores are $13.12 \pm 0.04^\circ$ AE and 17.30 ± 0.04 mm LD, where these two SD values describe seed variability, not user variability. For long-horizon analysis we additionally report Tracking AE at window lengths of 5 s (native), 15 s, 30 s, 60 s, and full-session, to probe accumulated-error behavior under extended rollouts.

Session-weighted pose evaluation. The Session columns in Table 1 instead give equal weight to acquisition sessions within each split, followed by equal weight to the three splits. We use the official `EMG2PoseEvaluation` with grouping conditions `[generalization, session]`, 5 s scored windows, and `skip_ik_failures=True`. A session is the acquisition-session identifier in the dataset metadata. Multiple stage recording files can share one acquisition-session identifier. All runs use the same 751 non-empty split-session groups: 78 User, 597 Stage, and 76 User×Stage. Three Stage groups have no eligible scored windows and are excluded from every run. Sessions appearing in different splits remain separate evaluation groups. For each NHN seed, we first compute the three session means and their equal-weight average, then report the mean and sample SD of the five resulting overall scores. The main table displays only the means; Appendix D.7 reports seed variability and split-specific results. Learned baselines use one released checkpoint each. Static uses the task-specific fixed poses described in Appendix C.4. These session-weighted scores complement the user-weighted results above without extending the tracking horizon.

emg2qwerty. We report CER on the ODV, TDV, and TDT partitions defined above, using TDV and TDT for fine-tuned models. We use two decoders: **greedy** (argmax over blank-augmented posterior at each frame, followed by CTC collapse) and **beam search**, which adds a 6-gram character-level language model (LM) trained on WikiText-103 with beam size 50. The beam decoder uses language-model weight 2.00, insertion bonus 2.00, and up to 10 candidate labels per frame. Blank bias is zero and the posterior temperature is 1.00. Each participant's evaluated sessions are pooled by character count before computing CER (Sivakumar et al., 2024; Hadidi et al., 2025). For NHN, participant-level CER is first averaged over five random seeds, after which we report the mean and population SD across participants: eight for TDV/TDT and four for ODV. This SD measures participant heterogeneity. Seed variability is computed as the sample SD of the participant-macro CER across the five seeds.

Inference efficiency. For emg2pose Regression, batch-one floating-point operations (FLOPs) cover the input context as well as the 5.00-s scored span. Causal TDS models receive 11,790 samples at 2 kHz, comprising 1,790 samples of left context and 10,000 scored samples. Their prediction-head input feature sequences are interpolated from 125 frames at 25 Hz to 250 autoregressive prediction-head steps at 50 Hz. We count convolution, linear, and matrix multiplication operations from PyTorch dispatch and add the matrix operations of fused recurrent layers analytically. One multiply-add counts as two FLOPs. Elementwise nonlinearities, normalization, interpolation, indexing, concatenation, cumulative scalar statistics, and the 16×16 whitening eigendecomposition are excluded. Whitening covariance and application matrix multiplications are included.

For emg2qwerty, we report model FLOPs for a batch-one, 30-s dual-wrist input sampled at 2 kHz, counting one multiply-add as two floating-point operations. The forward pass includes the neural prediction head and excludes language-model and text-decoding costs. Training uses the parallel convolutional implementation of multi-timescale adaptive gain, which is also the default evaluation path. Reported typing FLOPs use the recurrent implementation of Equation 6 with the same learned initial states. The two formulations are mathematically equivalent, with differences arising from floating-point accumulation order. We express counts in billions of FLOPs (GFLOPs). Under this

counting convention, the recurrent path gives 17.85 GFLOPs per 30 s input. On 16 TDT sessions, the same checkpoint produced identical greedy CER and edit counts with both implementations.

C.4 Baselines

emg2pose. The original benchmark provides **vemg2pose** (Salter et al., 2024), a TDS convolutional encoder with a state-conditioned LSTM velocity decoder (5.97 million parameters for Tracking and 5.98 million in the official Regression checkpoint) that defines the original emg2pose benchmark, and the smaller **emg2pose** model ($\sim$2.95 million parameters), which directly predicts joint angles. Hadidi et al. (2026) retain the vemg2pose core architecture while revising decoding and training settings. Their four reported variants combine position (Pos) or velocity (Vel) prediction with single-task (ST) or multi-task (MT) training: **Pos-ST**, **Vel-ST**, **Pos-MT**, and **Vel-MT**. These are one baseline family, with the best-performing variant depending on the evaluated task. We additionally compare against two alternative published wrist-sEMG-to-pose architectures, **NeuroPose** (Liu et al., 2021) and **Sensing Dynamics** (Sîmpetru et al., 2022), and a **Static** baseline (Hadidi et al., 2026). Static predicts the ground-truth initial pose throughout each Tracking window. For Regression, it instead predicts the training-set median angle of each joint at every frame, without access to the test window's initial pose.

emg2qwerty. The published benchmark backbone is the TDS convolutional network (**TDS ConvNet**) architecture of Sivakumar et al. (2024). Our primary baseline is **SplashNet** (Hadidi et al., 2025), a split-and-share TDS-CTC encoder that extends this backbone. We report three published architectural variants. **Split-only** uses independent per-hand encoders (2.68 million parameters, 36.84 GFLOPs/30 s), **SplashNet-mini (Shared)** shares their weights (1.38 million parameters, 36.84 GFLOPs/30 s), and **SplashNet-Upscale** widens the shared model (2.58 million parameters, 71.38 GFLOPs/30 s). We also report the published fine-tuning variants with unshared weights, which replace shared per-hand weights with user-specific copies at identical FLOPs. The TDT beam-search comparisons use SplashNet-Upscale in zero-shot evaluation and its Shared variant after fine-tuning. We additionally benchmark against the distilled **Transformer** models of Mehlman et al. (2025) as attention-based comparators (**Transformer-tiny**, **Transformer-small**). These models provide the lowest reported TDT greedy CER among the compared baselines, so greedy and beam-search comparisons are kept separate. Baseline accuracy and variability are reproduced as published.

D Additional Results and Ablations

D.1 Architecture ablations and sensitivity

Physiological organization motivates NHN's hierarchy, and model development determines how to implement it for decoding. We use the emg2pose official-small setting (Salter et al., 2024) to study these choices before adopting the resulting architecture in the full-data experiments. We examine whether preserving intensity, using history, and constructing a compact latent neuromotor state improve decoding beyond increasing network capacity. This state is learned from task supervision, with physiological principles guiding its organization rather than specifying its learned primitives.

The comparisons progress from measurement adaptation and spatiotemporal encoding to latent neuromotor state construction, prediction-head capacity, and training objectives. Component ablations test the contribution of an operation, while sensitivity analyses compare ways of instantiating it. These experiments cover Regression and Tracking, with task and seed coverage specified for each comparison in the tables. Comparisons across these settings test whether the design choices support both pose-decoding requirements.

The official-small split is specified by 857 training, 98 validation, and 308 test metadata rows, from which input windows are constructed. We train configurations from scratch with a 300-epoch budget and no early stopping. AdamW uses a learning rate of 10^{-3}, a 10-epoch linear warmup, and cosine decay toward 10^{-6}. The batch size is 128, with gradients accumulated over two batches. Inputs contain 11,790 samples at 2 kHz, including 1,790 samples of left context and a 5 s scored interval. Unless the training objective is ablated, Tracking and Regression losses are weighted by $7/8$ and $1/8$, and each combines joint-angle MAE with fingertip-distance loss at weights 1 and 0.01. Variants inherit their task-specific control settings apart from the declared architectural or training change.

Appendix C.2 specifies the task-dependent initial-pose, output-scale, and gradient-clipping settings used by the controls.

We repeat key architectural and training-objective ablations with three random seeds to assess the consistency of their effects. To cover a broader range of hyperparameters and supplementary configurations, the remaining comparisons use a single seed. Three-seed AE and LD results are reported as mean ± sample standard deviation across seeds. Evaluation uses $2{,}155$ windows of 5 s across the User, Stage, and User×Stage test splits, with 18, 102, and 13 users, respectively. The scorer averages windows within each user, then users equally within each split, and finally the three splits equally for each seed. Single-seed results are reported with their matching full-model controls. ΔAE is the variant minus its same-seed control, averaged across seeds where applicable. Bold marks the lowest displayed means and parameter counts (Params.) within each task-and-seed block. Matching full NHN control runs are reused across tables within each task and seed setting. These experiments use the official-small setting to compare design choices. For the evaluation reported in the main text, we train NHN on the full dataset with five random seeds.

D.1.1 Measurement adaptation and spatiotemporal encoding

Anatomical and electrode properties shape the measured sEMG (Farina et al., 2002). Adapting waveform statistics addresses this measurement dependence, but the associated normalization also motivates retaining an explicit intensity path. Slow-varying common input to human motor neuron pools (Negro et al., 2016) motivates historical context. The component comparisons test waveform conditioning and intensity preservation separately, then examine structured mixing and adaptive gain (Table 4). The Tracking reference uses partial-whitening coefficient $\alpha = 0.50$, global-mixing rank $R = 32$, circular kernel width $k = 5$, and adaptive-gain timescales initialized at $(0.10, 0.50, 2)$ s. Fixing the waveform-conditioning weights at zero retains the relative-intensity channels, whereas the intensity-channel ablation removes them. These two comparisons distinguish the use of intensity to condition the waveform from its availability as a decoding cue. The TDS comparison replaces local-global mixing within NHN, and the gain ablation removes history-dependent feature reweighting by adaptive gain. Appendix D.1.3 separately compares complete backbones.

Both ways of using intensity contribute to decoding. Fixing the waveform-conditioning weights at zero or removing relative-intensity channels increases average AE by 0.61° or 0.69°, respectively. The corresponding single-seed Regression increases are 0.67° and 0.46°. This agreement across pose settings supports retaining waveform conditioning and an explicit intensity path as distinct parts of the design.

Replacing local-global mixing with conventional TDS mixing incurs 1.11° higher AE and 1.98 mm higher LD despite using more parameters. Removing adaptive gain increases AE by 0.82° and LD by 1.40 mm. All four Tracking component ablations increase AE and LD in each of the three seeds. The mixing result supports structured feature interactions with fewer parameters, while the gain result supports history-dependent feature reweighting by adaptive gain.

The single-seed comparisons then examine mixing capacity, gain initialization, and whitening strength. Both tested alternatives for each setting have higher AE, supporting $R = 32$, $k = 5$, $\alpha = 0.50$, and the reference gain initialization as computational choices for implementing this organization. The gain settings in this sweep are initial adaptive-gain timescales that remain learnable, whereas R, k, and α define the tested configurations. This sweep selects a working capacity and initial temporal range for the same architecture.

D.1.2 Latent neuromotor state construction and width

Human hand coordination exhibits dominant postural synergies while finer components retain object-specific information (Santello et al., 1998). This motivates a compact latent neuromotor state with sufficient capacity for task-specific detail. Orderly motor-unit recruitment (Milner-Brown et al., 1973) motivates learned priorities for graded allocation. We examine the capacity, allocation, and temporal integration of this learned representation (Table 5). The reference uses $N = 32$ learned primitives, $K = 12$ drive entries for temperature calibration, and FIR supports $(3, 7, 11, 15)$. Here N sets representational capacity, while K calibrates the concentration of continuous allocation across all N primitives. We vary K at fixed N and compare widths at a fixed K/N ratio to separate these two configuration questions. Temporal controls replace FIR integration with the identity, change its

Table 4: **Measurement adaptation and spatiotemporal encoding.** Results on emg2pose official-small. Task and seed coverage are stated in each block.

Variant	Params.↓ (10^6)	Mean AE↓ (°)	Mean LD↓ (mm)	ΔAE↓ (°)
(a) Tracking, three seeds				
Full NHN	3.08	$\mathbf{12.36} \pm 0.10$	$\mathbf{16.65} \pm 0.13$	**0.00**
Conditioning weights fixed at zero	3.08	12.97 ± 0.13	17.69 ± 0.19	$+0.61$
No relative-intensity channels	**3.06**	13.05 ± 0.13	17.93 ± 0.12	$+0.69$
Conventional TDS mixing	3.53	13.47 ± 0.04	18.63 ± 0.09	$+1.11$
No adaptive gain	3.07	13.18 ± 0.31	18.05 ± 0.58	$+0.82$
(b) Regression, single seed				
Full NHN, LSTM-256	3.08	**16.24**	**22.26**	**0.00**
Conditioning weights fixed at zero	3.08	16.91	22.45	$+0.67$
No relative-intensity channels	**3.06**	16.70	22.76	$+0.46$
(c) Tracking sensitivity, single seed				
Full NHN	3.08	**12.45**	**16.78**	**0.00**
Global-mixing rank $R = 16$	**3.05**	13.39	18.67	$+0.94$
Global-mixing rank $R = 64$	3.14	13.21	17.79	$+0.76$
Circular kernel width $k = 3$	3.08	12.86	17.57	$+0.42$
Circular kernel width $k = 7$	3.08	13.54	18.63	$+1.09$
Initial gain timescales $(0.05, 0.25, 1)$ s	3.08	13.47	18.80	$+1.02$
Initial gain timescales $(0.20, 1, 4)$ s	3.08	13.15	17.82	$+0.71$
Partial whitening $\alpha = 0.25$	3.08	12.84	17.81	$+0.39$
Partial whitening $\alpha = 0.75$	3.08	12.98	17.82	$+0.53$

support lengths, or move it after allocation. The allocation controls either bypass both grading and priority or retain grading while removing the learned priority.

Table 5: **Latent neuromotor state construction and width.** Results on emg2pose official-small. Task and seed coverage are stated in each block.

Variant	Params.↓ (10^6)	Mean AE↓ (°)	Mean LD↓ (mm)	ΔAE↓ (°)
(a) Tracking, three seeds				
Full NHN	**3.08**	$\mathbf{12.36} \pm 0.10$	$\mathbf{16.65} \pm 0.13$	**0.00**
Identity instead of FIR integration	**3.08**	13.14 ± 0.19	17.81 ± 0.41	$+0.79$
No graded allocation or priority	**3.08**	13.14 ± 0.23	18.18 ± 0.60	$+0.78$
(b) Tracking sensitivity and ordering, single seed				
Full NHN	3.08	**12.45**	**16.78**	**0.00**
Temperature-calibration count $K = 8$	3.08	12.98	17.42	$+0.53$
Temperature-calibration count $K = 16$	3.08	13.12	17.78	$+0.68$
FIR supports $(3, 5, 7, 9)$	3.08	13.69	18.96	$+1.24$
FIR supports $(5, 9, 15, 21)$	3.08	13.33	17.91	$+0.89$
Latent neuromotor state width $N = 24$, $K = 9$	**3.07**	13.61	18.52	$+1.16$
Latent neuromotor state width $N = 48$, $K = 18$	3.10	13.17	17.78	$+0.72$
FIR integration after allocation	3.08	13.12	18.01	$+0.67$
No priority, retain graded allocation	3.08	12.75	17.31	$+0.30$

Replacing FIR integration with an identity increases Tracking AE by $0.79°$ and LD by 1.16 mm across three seeds. Jointly bypassing graded allocation and learned priority increases these errors by $0.78°$ and 1.52 mm. Both changes increase AE and LD in every seed, with identical reported

parameter counts across the three configurations. The single-seed controls further examine how these operations are organized. Moving FIR integration after allocation increases AE by $0.67°$, while removing priority with graded allocation retained increases it by $0.30°$. The ordering comparison favors integrating candidate drives before allocating their participation. The priority comparison tests the learned relative preference among primitives while retaining graded allocation.

At fixed K/N, reducing the latent neuromotor state width from 32 to 24 increases AE by $1.16°$, while expanding it to 48 increases AE by $0.72°$. Neither width alternative improves decoding. At $N = 32$, changing K from 12 to 8 or 16 increases AE by $0.53°$ or $0.68°$. The shorter and longer FIR-support sets also increase AE, by $1.24°$ and $0.89°$, respectively. These comparisons support integrating drives before allocation and using $N = 32$, $K = 12$ to instantiate a compact, graded representation.

D.1.3 Backbone and prediction-head capacity

We separate the effects of backbone organization and prediction-head capacity by crossing backbone choice with head width (Table 6). The Regression comparison crosses NHN and conventional TDS (Hannun et al., 2019) with LSTM (Hochreiter & Schmidhuber, 1997) hidden widths of 256 and 512. The common head widths allow comparison of backbone organization at each recurrent hidden dimension. Varying the width within each backbone tests whether a larger head improves decoding. A gated recurrent unit (GRU) head with width 296 (Cho et al., 2014) and Mamba2-256 (Dao & Gu, 2024) provide additional prediction-head comparisons at approximately the reference parameter count, testing the selected interface with other sequence-model families.

Table 6: **Backbone and prediction-head capacity.** Results on emg2pose official-small. Task and seed coverage are stated in each block.

Variant	Params.↓ (10^6)	Mean AE↓ (°)	Mean LD↓ (mm)	ΔAE↓ (°)
(a) Regression: backbone and LSTM width, three seeds				
Full NHN, LSTM-256	**3.08**	$\mathbf{16.20} \pm 0.06$	$\mathbf{22.19} \pm 0.07$	**0.00**
NHN with LSTM-512	5.54	16.33 ± 0.12	22.26 ± 0.08	$+0.13$
TDS with LSTM-256	3.54	16.74 ± 0.09	22.77 ± 0.25	$+0.54$
TDS with LSTM-512	6.01	16.65 ± 0.07	22.60 ± 0.06	$+0.45$
(b) Regression: prediction-head family, single seed				
Full NHN, LSTM-256	3.08	**16.24**	**22.26**	**0.00**
NHN with GRU-296	3.07	16.41	22.62	$+0.17$
NHN with Mamba2-256	**3.04**	16.94	22.91	$+0.70$

With the same LSTM hidden width, NHN reduces average Regression AE relative to TDS by $0.54°$ at width 256 and $0.32°$ at width 512, using 3.08 versus 3.54 million parameters and 5.54 versus 6.01 million parameters, respectively. All three seeds favor NHN at each width. Increasing the LSTM width from 256 to 512 reduces mean AE by $0.09°$ for TDS but increases it by $0.13°$ for NHN. The NHN width effect varies across seeds, so the mean comparison supports the smaller prediction head in this setting.

In the single-seed comparison, replacing LSTM-256 with GRU-296 or Mamba2-256 increases AE by $0.17°$ or $0.70°$ and LD by 0.36 or 0.65 mm. These comparisons support pairing NHN's structured representation with LSTM-256 for the full-data pose experiments. This choice combines the lowest mean error in the tested comparisons with fewer parameters than either LSTM-512 configuration.

D.1.4 Task supervision, conditioning, and augmentation

The compact latent neuromotor state is learned through task supervision. The training objective therefore determines which aspects of the signal the architecture must preserve. Joint Regression and Tracking supervision exposes the decoder to pose estimation both without and with an initial pose anchor. To assess whether these requirements provide complementary supervision, we compare each single-task objective with its control trained with both objectives across three seeds (Table 7).

The single-seed controls examine context and variation. Removing one-step pose-difference feedback tests the usefulness of changes between successive predictions while retaining the preceding pose. The augmentations vary temporal structure through time warping and signal magnitude through post-conditioning amplitude scaling. We also compare alternative relative task weights to assess the balance of joint supervision.

Table 7: **Task supervision, conditioning, and augmentation.** Results on emg2pose official-small. Task and seed coverage are stated in each block.

Variant	Params.↓ (10^6)	Mean AE↓ (°)	Mean LD↓ (mm)	ΔAE↓ (°)
(a) Tracking objective, three seeds				
Full NHN	**3.08**	$\mathbf{12.36} \pm 0.10$	$\mathbf{16.65} \pm 0.13$	**0.00**
Tracking-only training	**3.08**	13.18 ± 0.15	18.05 ± 0.46	$+0.82$
(b) Regression objective, three seeds				
Full NHN, LSTM-256	**3.08**	$\mathbf{16.20} \pm 0.06$	$\mathbf{22.19} \pm 0.07$	**0.00**
Regression-only training	**3.08**	17.10 ± 0.04	23.23 ± 0.18	$+0.90$
(c) Tracking conditioning and augmentation, single seed				
Full NHN	3.08	**12.45**	**16.78**	**0.00**
No one-step pose-difference feedback	**3.06**	13.66	18.74	$+1.21$
No time-warp augmentation	3.08	13.18	17.60	$+0.73$
No post-conditioning amplitude scaling	3.08	12.95	17.69	$+0.51$
(d) Regression conditioning and task weights, single seed				
Full NHN, LSTM-256	3.08	**16.24**	**22.26**	**0.00**
No one-step pose-difference feedback	**3.06**	16.74	22.91	$+0.50$
Tracking/Regression weights $(1/2,\ 1/2)$	3.08	16.68	22.84	$+0.44$
Tracking/Regression weights $(1/8, 7/8)$	3.08	16.80	23.29	$+0.56$

The three-seed objective ablations show increases of $0.82°$ AE and 1.40 mm LD for Tracking-only training, and $0.90°$ AE and 1.04 mm LD for Regression-only training, relative to the respective multitask controls. Both tasks therefore benefit from retaining the other task's objective under the evaluated joint-supervision setup. In the single-seed comparisons, removing one-step pose-difference feedback increases AE by $1.21°$ for Tracking and $0.50°$ for Regression. Removing time-warp augmentation or post-conditioning amplitude scaling increases Tracking AE by $0.73°$ or $0.51°$. The two alternative task-weight settings increase Regression AE by $0.44°$ and $0.56°$. These findings support retaining multitask supervision, one-step pose-difference feedback, and the tested augmentations alongside the architectural choices above. They complement the architectural comparisons by identifying useful prediction context and training variation.

Across the four groups, the development experiments support a consistent configuration: adapted waveforms with retained intensity cues, local-global mixing with adaptive gain, temporal integration before continuous allocation, and a compact LSTM prediction head. Physiological principles motivate this organization, while the ablations identify useful operations and the sensitivity comparisons guide their numerical settings. The full-data experiments then evaluate the adopted configuration at benchmark scale.

D.2 Figure 3: sampling and aggregation

Figure 3 compares task-linked combinations, temporal preferences, and continuous allocation learned under the two task objectives. The analyses use task-trained models at each task's native temporal resolution. Correlations are aggregated across participants, temporal parameters across trained models, and allocation weights across frames.

Task-linked combinations (a,d). For pose, panel (a) examines how learned primitive activations relate to hand posture using one trained model and 32 five-second validation windows. From an

existing pool of $1{,}024$ validation windows, a fixed identifier-based rule selects 16 windows from each available validation group, User and User$\times$Stage. The selected windows cover 13 participants and contain 125 state frames per window at 25 Hz. We compute window-wise Pearson correlations between relative primitive activation, $p_i(t)/\sum_j p_j(t)$, and 20 finger-related degrees of freedom (DoFs). Valid correlations are averaged within each participant and then equally across participants. Estimates with insufficient variance are omitted from these averages, leaving 12 or 13 participants per cell. All 32 primitives and 20 angles are retained. A dot marks a cell when at least 80% of its available participant coefficients share the displayed mean's sign, with at least 10 participants available. This marks directional agreement across the sampled activities rather than statistical significance or repetition of a matched action. Positive correlations mean that a primitive's activation share tends to increase as the joint angle increases, while negative correlations mean that it tends to decrease. For example, primitive 12 is positively associated with middle- and ring-finger flexion-extension, whereas primitives 16 and 31 show negative associations with several of these coordinates. Primitive 14 is positively associated with thumb flexion-extension. The overlapping associations suggest that multiple primitives jointly represent hand posture, consistent with NHN's coordination-inspired design.

Rows are ordered by average-linkage Euclidean clustering of their mean correlation profiles. Their original primitive identifiers, from top to bottom, are 12, 32, 8, 15, 24, 5, 18, 13, 11, 6, 9, 23, 28, 4, 1, 26, 21, 7, 29, 2, 20, 14, 19, 30, 22, 3, 10, 25, 17, 31, 16, 27. All primitive identifiers are labeled on alternating sides of the matrix. The finger groups Th, In, Mi, Ri, and Li denote thumb, index, middle, ring, and little fingers, respectively. Within each finger, the dark strip marks the abduction-adduction angle and the light strip marks three flexion-extension angles. For the thumb, these are carpometacarpal abduction-adduction followed by carpometacarpal, metacarpophalangeal and interphalangeal flexion-extension. The other fingers use metacarpophalangeal abduction-adduction followed by metacarpophalangeal, proximal interphalangeal and distal interphalangeal flexion-extension. The common color scale is linear from -0.40 to 0.40.

For typing, we take the first four training participants in participant/session identifier order and use their first sessions in panel (d). We sample 16 windows at evenly spaced positions in each session and compare centered key-specific primitive-activation profiles between the first eight and last eight windows. The nine keys are `a, e, i, l, n, o, r, s, t`. The profiles use 611 unique events from four participants, evaluated with five independently trained models on the same windows and events. Each event's cached mean state is first L_1-normalized separately for the two wrists. Event profiles are averaged for each key within each window group, then centered by subtracting that group's equally weighted mean over the nine keys. Cosine similarity compares the same key across window groups, while each other-key value averages the eight alternative key matches within the same participant and model. Circles retain all 180 participant-model same-key observations and their 180 other-key averages. Diamonds show means. Red bars show sample standard deviations across participants after averaging the five models within each participant. The comparison therefore tests within-participant repeatability of task-linked patterns, complementing the continuous associations in pose.

Learned temporal preferences (b,e). The temporal panels compare configured initial values with values after training in seconds. Adaptive-gain timescales τ_m describe the history used to reweight encoder features in Equations 6 and 7. A mean integration lag is the kernel-weighted mean age of candidate drives relative to the current integration frame, including the zero-lag identity path (Appendix B.2). It summarizes temporal weighting, not the full kernel support or end-to-end latency. Both panels summarize five independently trained models per task. Trained curves show arithmetic means, with sample standard deviations indicated by red gain bars and blue mean-integration-lag bands. Statistics are computed in seconds before logarithmic display. Adaptive-gain timescales retain their short (S), intermediate (M), and long (L) identities. At each of the 32 original primitive indices, mean-integration-lag statistics summarize the corresponding parameter position across models, without assuming identical learned functions. Initialization is identical across the five configurations within each task and has zero across-model SD. Initial adaptive-gain timescales are 0.10, 0.50, and 2 s for pose, and 0.20, 1, and 5 s for typing. Integration uses 25-Hz and 100-Hz grids, respectively, with task-specific kernel supports. Temporal adaptation is therefore assessed relative to each task's own initialization. At the individual-model level, all 32 pose mean integration lags decrease in every model, whereas 26 to 29 typing lags decrease per model and the remainder increase. For pose,

the increased adaptive-gain timescales and decreased mean integration lags suggest longer context for reweighting encoder features and greater emphasis on recent candidate drives. Typing instead decreases the S and L timescales but increases M, suggesting task-specific temporal preferences within the same architecture.

Continuous allocation (c,f). Panels (c,f) illustrate allocation within one fixed validation clip per task. For pose, we use the first eligible 5-s window in the original validation manifest (125 frames). For typing, we use a 12-s left-wrist clip (1,200 frames) from the first ODV session in participant/session identifier order. It begins one second before the first keypress occurring at least five seconds into the recording. Within each clip, the softmax allocation weights in Equation 10 are ranked independently at each frame. Curves show the mean at each rank. Dark and light bands show the interquartile and 5% to 95% ranges across frames, respectively. The Hard-12 reference assigns weight $1/12$ to 12 ranks and zero to the remainder, while the uniform reference assigns $1/32$ to every primitive. These fixed-subset and equal-allocation distributions serve as analytical references for the learned weights. The labels report the frame-mean top-12 weight mass and effective allocation participation, computed per frame as $1/\sum_i w_{i,t}^2$ and then averaged. For pose and typing, respectively, the top-12 masses are 49.39% and 48.99%, and mean effective participation is 29.12 and 29.28 out of 32. Allocation weights and latent neuromotor state amplitudes are distinct because the latter also include the integrated candidate drives. The weights therefore describe a different quantity from the relative primitive activations used in panel (a). Legends in (b,c) also apply to (e,f).

Amplitude-matched allocation intervention. The allocation spectra describe learned distributions. To test whether the prediction head uses their organization beyond total activation intensity, we replace the learned weights by uniform weights in a frozen model. Because $\boldsymbol{p}_t = N\boldsymbol{w}_t \odot \bar{\boldsymbol{d}}_t$, the unscaled uniform control reduces to $\bar{\boldsymbol{d}}_t$. The amplitude-matched control rescales this vector by $\|\boldsymbol{p}_t\|_1/\|\bar{\boldsymbol{d}}_t\|_1$ separately at every frame and for each wrist. It therefore retains the original total state intensity while removing the learned relative allocation. Both conditions are passed through the original prediction head.

The pose intervention uses the same 32 validation windows from 13 participants as panel (a), across User and User×Stage. Errors are averaged within split-participant pairs, then equally across participants within each split and across the two splits. The amplitude-matched intervention increases AE by 3.94°. For typing, we use the first two ODV participants in participant/session identifier order, with one complete validation session each. Replaying the frozen head increases mean greedy CER by 42.12 percentage points. These intervention results are aggregated over the stated samples, while panels (c,f) illustrate individual clips. The paired changes show that the trained heads use the allocation pattern in addition to overall intensity. They complement the architecture-design experiments in Appendix D.1, which train different configurations rather than intervening on one fixed model.

D.3 Long-horizon tracking without re-anchoring

The standard 5 s Tracking evaluation supplies a new ground-truth pose at the start of each window. We extend the uninterrupted tracking horizon to test whether NHN's accuracy advantage persists as prediction continues farther from the initial pose anchor. The preceding sEMG context duration remains fixed while the duration without re-anchoring increases.

Models and controlled comparison. We compare NHN Tracking models from five independent training runs with the official emg2pose and vemg2pose Tracking checkpoints of Salter et al. (2024), together with a Static predictor that holds the initial pose. Static measures the error from retaining the anchor without following subsequent motion. We evaluate the existing checkpoints without retraining for the longer horizons, using the same checkpoint for each NHN run across all durations. Each learned model receives the ground-truth pose once at the start of a segment and 1,790 samples (895 ms) of preceding sEMG context. Models use the same segments, initial poses, and evaluation timestamps, with no further pose correction. Each horizon thus gives a paired comparison of complete decoders.

Horizons and test coverage. We construct non-overlapping 5, 15, 30, and 60 s windows from the full test partitions, requiring each window to lie within a contiguous inverse-kinematics-valid

Table 8: **Long-horizon Tracking on emg2pose.** Split-equal errors, with NHN mean ± sample SD across five training runs. Bold marks the lowest error. Gain is vemg2pose minus NHN, computed before rounding. Full-session uses variable-duration valid intervals.

	Fixed tracking horizon				
Model	5 s	15 s	30 s	60 s	Full-session
Angular error (°)↓					
Static	13.21	15.34	16.31	17.27	17.66
emg2pose	13.91	14.27	14.08	14.07	14.87
vemg2pose	9.97	11.43	11.96	13.30	12.36
NHN (ours)	**9.20 ±0.03**	**10.25 ±0.04**	**10.59 ±0.03**	**11.22 ±0.15**	**11.09 ±0.02**
Gain over vemg2pose↑	0.78	1.18	1.37	2.07	1.27
Landmark distance (mm)↓					
Static	19.67	22.69	23.82	25.08	25.54
emg2pose	18.76	19.58	19.41	19.69	20.23
vemg2pose	13.59	15.46	16.16	17.85	16.37
NHN (ours)	**12.39 ±0.05**	**13.82 ±0.08**	**14.37 ±0.08**	**15.09 ±0.18**	**14.67 ±0.11**
Gain over vemg2pose↑	1.20	1.65	1.79	2.76	1.70
Fingertip distance (mm)↓					
Static	32.88	38.12	40.07	42.60	42.87
emg2pose	31.61	32.99	32.61	33.19	34.06
vemg2pose	22.57	25.76	26.99	29.94	27.25
NHN (ours)	**20.43 ±0.10**	**22.83 ±0.17**	**23.75 ±0.17**	**25.12 ±0.37**	**24.18 ±0.21**
Gain over vemg2pose↑	2.14	2.93	3.24	4.82	3.07

interval. The Full-session condition instead evaluates the longest contiguous valid interval in each recording. These intervals range from 0.07 to 237.88 s, with mean duration 28.67 s. This condition retains recordings whose valid intervals are too short for a long fixed window and also includes intervals longer than 60 s. We report it separately from the fixed-horizon sequence.

Metrics and aggregation. We score each complete evaluation interval using angular error, landmark distance, and fingertip distance. Angular error measures pose accuracy in joint coordinates, while the spatial distances describe the resulting hand and fingertip positions. Errors are averaged over windows within each user, then equally over users within each generalization split and over the three splits. The larger Stage split therefore does not dominate the overall result. Longer fixed windows admit fewer recordings and users, as detailed in Table 9. All methods are compared on the same eligible subset at each horizon. We then report the mean and sample standard deviation across the five NHN runs. This standard deviation describes training variation, whereas the split columns of the main results table report variation across users. Each official baseline uses one fixed checkpoint. This user-weighted aggregation differs from the Session columns in Table 1, which weight sessions equally within each split (Appendix C.3).

Long-horizon tracking performance. NHN maintains lower errors at longer tracking horizons (Table 8). Its AE advantage over vemg2pose grows from 0.78° at 5 s to 1.18°, 1.37°, and 2.07° at 15, 30, and 60 s. The relative AE reduction increases from 7.78% to 15.60%. LD and fingertip distance show the same widening absolute advantage, from 1.20 to 2.76 mm and from 2.14 to 4.82 mm, respectively. NHN has the lowest error on all three metrics at every evaluated horizon. All five NHN runs outperform vemg2pose across every split, horizon, and metric. The adopted architecture retains its advantage during longer tracking without an updated ground-truth pose, complementing the component ablations with a whole-model evaluation.

Table 9: **Long-horizon test coverage and NHN split results.** User counts are reported separately for each split. Window counts cover all three splits and are identical across models. AE values are mean ± sample SD across the five NHN runs. Full-session has variable duration.

(a) Test coverage

	Users			
Horizon	User	Stage	User×Stage	Windows
5 s	20	158	20	45,321
15 s	20	158	20	9,977
30 s	20	156	19	2,818
60 s	11	58	10	268
Full-session	20	158	20	6,114

(b) NHN angular error (°)↓

Horizon	User	Stage	User×Stage
5 s	7.30±0.02	10.12±0.04	10.16±0.07
15 s	7.84±0.04	11.36±0.02	11.55±0.10
30 s	7.74±0.07	11.88±0.05	12.14±0.09
60 s	8.62±0.22	11.97±0.14	13.07±0.36
Full-session	8.91±0.02	11.99±0.04	12.38±0.05

Coverage and variable-duration tracking. The eligible test population changes with the fixed horizon. The 5 and 15 s conditions retain all $20/158/20$ user keys across User, Stage, and User×Stage, whereas 60 s retains $11/58/10$ user keys and 268 windows. Thus, the reported error increases describe aggregate evaluation trends under increasingly long valid-window requirements, rather than the growth of error along a common set of trajectories. Table 9 gives the split-level NHN errors alongside this coverage. At every evaluated horizon, mean AE is lower on User than on the settings involving unseen stages.

Full-session restores the complete $20/158/20$ user-key coverage while allowing each recording to contribute its longest valid interval. NHN also has the lowest errors in this condition, with advantages of $1.27°$ AE, 1.70 mm LD, and 3.07 mm fingertip distance over vemg2pose. Its mean AE of $11.09°$ summarizes a mixture of short and long intervals, not a fifth fixed horizon beyond 60 s. The fixed-horizon and Full-session results together show that NHN's relative accuracy advantage extends beyond the standard short-window setting, both on eligible long segments and with full user coverage under variable-duration tracking.

D.4 Qualitative tracking visualization

The aggregate results establish accuracy across users and training runs. We additionally inspect how predicted hand configurations and errors evolve within individual test windows. Figure 4 uses one final Tracking model and the first 5 s window in the frozen evaluation manifest for each of User, Stage, and User×Stage. These three windows are fixed before inference, without ranking prediction errors. Each receives one ground-truth initial pose and the same context as the quantitative evaluation. All 20 finger-related angles vary by more than $10°$ within each window.

The paired sequences show repeated opening and closing in the User and Stage examples, while retaining visible finger-specific discrepancies. For example, the User×Stage prediction at 5 s extends the thumb more than the ground truth. The traces report frame-wise mean angular error over the complete window. Their shaded ranges summarize differences among angle coordinates. These single-model trajectories complement the five-run aggregate results in Table 8.

D.5 CTC emission behavior

CTC converts NHN's frame-wise character evidence into a variable-length keystroke sequence. Figure 5 illustrates successful decoding using cached full-session outputs from one final fine-tuned

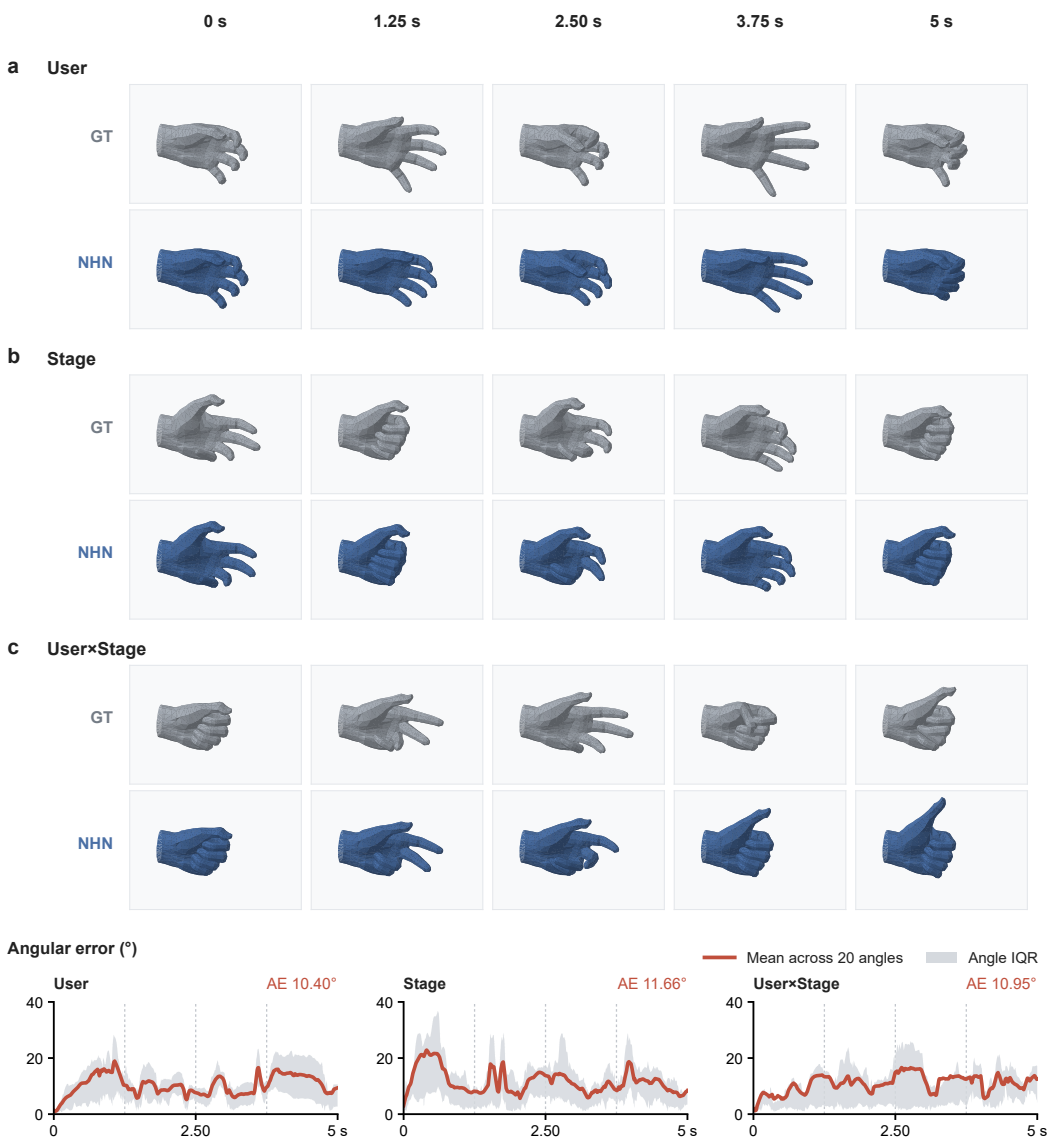


Figure 4: **Final-model Tracking examples.** (a to c) One fixed test window per generalization split. Meshes compare ground truth (GT, gray) with NHN (blue) at five evenly spaced times from 0 to 5 s, using the last available frame for 5 s. All meshes share a generic hand model, fixed wrist pose, orthographic camera, scale and equal-sized frame. Right hands are mirrored to the common display orientation. Red curves show frame-wise mean absolute error over 20 finger-related angles, with their interquartile range (IQR) in gray. Numbers report each complete window's AE. Shading describes coordinate variation, not uncertainty across users or runs.

model on TDV. From the first participant's first TDV session, we select a 3 s excerpt with exact agreement between greedy decoding and the recorded text. The excerpt begins 0.25 s before a word onset and contains `fox jumps over`, including both spaces. The model retains its full-session context. Output times follow the recorded input clock at the latest sample required by the model. Emission timestamps remain unchanged.

Colored curves show individual character probabilities on a logarithmic axis, with a dashed black curve for blank. Matching character labels mark the first frame of each nonblank argmax run. We collapse the complete session's argmax path before cropping, merging consecutive repeats and

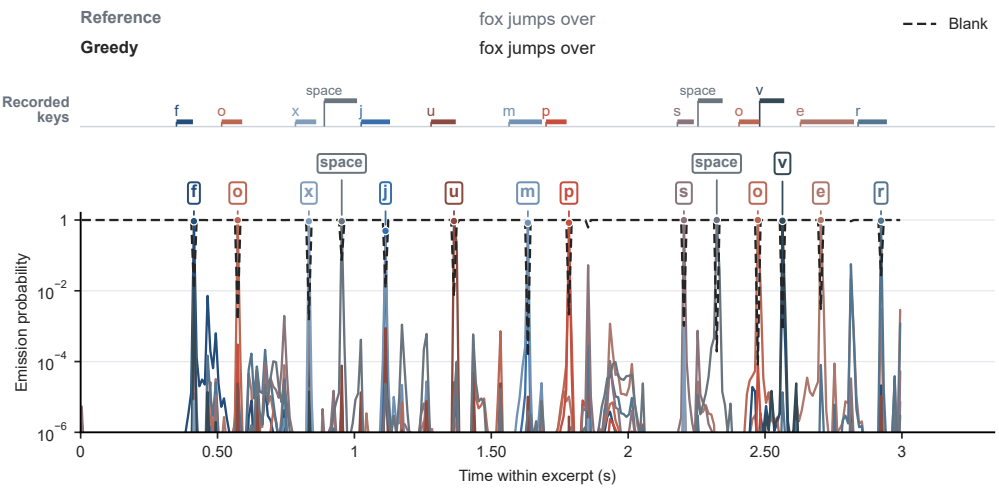


Figure 5: **Correct CTC decoding of a typing excerpt.** A 3 s TDV excerpt selected for exact greedy/reference agreement. Colored curves show individual character probabilities and dashed black denotes blank. Boxed characters mark unchanged greedy emission times. Probabilities are unsmoothed, with display floor 10^{-6}. Emission times follow input availability.

removing blanks. The example uses unmodified network probabilities, without smoothing, a language model or blank-score adjustment.

Brief character emissions are separated by blank-dominated frames, producing the correct character sequence and word boundaries in this example. Competing character probabilities do not necessarily produce additional symbols because decoding follows the argmax path. This illustration complements the aggregate CER in Table 2. The following section examines whether additional linguistic context can improve decoding from the same network outputs.

D.6 Replacing the character language model with GPT-2

Motivation. Typing accuracy depends on both the character evidence inferred from sEMG and the decoder's use of linguistic context. Our main results use beam search with a character-level 6-gram language model, which scores local character sequences. To test whether a longer-context language model improves decoding, we replace KenLM with a frozen GPT-2 model during beam search. NHN checkpoints and their output probabilities remain fixed, so the comparison measures a change in decoding rather than NHN training.

GPT-2 integration. We use the pretrained Hugging Face GPT-2 checkpoint (124 million parameters) without fine-tuning the language model. The decoder retains beam width 50 and at most 10 acoustic labels per frame. At each step, candidate keystroke prefixes h are ranked by

$$s_t(h) = \log P_{\mathrm{CTC},t}(h) + \lambda S_{\text{GPT-2}}(T(h)) + 2\ell(h),$$

where $P_{\mathrm{CTC},t}(h)$ is the CTC prefix probability accumulated through frame t, $T(h)$ renders the keystrokes as text, and $S_{\text{GPT-2}}$ sums its token log probabilities. The insertion count $\ell(h)$ increases for each non-Backspace key and decreases for Backspace, with a floor of zero, retaining the baseline insertion bonus. GPT-2 contributes before beam pruning, rather than reranking a final candidate list. No KenLM score enters this search.

GPT-2 scores the rendered text of each candidate, while CER follows the original keystroke-based evaluation.

GPT-2 scores every text token once, using an initial end-of-text token as context and overlapping 1024-token windows with stride 512. The score uses natural logarithms, without length normalization

Table 10: TDT CER (%, ↓) with GPT-2 replacing KenLM during beam search. All decoders use the same NHN run and the same 16 complete sessions per regime. Values are mean ± SD across eight participants. Δ is GPT-2 minus KenLM in percentage points (pp), computed before rounding. Bold marks the lowest displayed CER within each row.

Regime	Greedy	KenLM	GPT-2	Δ (pp)↓
Zero-shot	38.11 ± 6.80	$\mathbf{27.90} \pm 7.93$	30.73 ± 6.40	$+2.83$
Fine-tuned	9.20 ± 3.42	3.72 ± 1.40	$\mathbf{3.53} \pm 1.59$	-0.20

Table 11: Fine-tuned TDT CER (%) per participant for one NHN run, pooling edit counts over two complete sessions per participant. Δ is GPT-2 minus KenLM in percentage points, computed before rounding. Negative values indicate improvement.

Participant	KenLM ↓	GPT-2 ↓	Δ (pp)↓
1	5.92	5.94	$+0.02$
2	3.04	2.72	-0.32
3	3.75	3.38	-0.37
4	2.47	1.94	-0.54
5	3.14	2.81	-0.33
6	1.50	1.23	-0.26
7	4.57	4.61	$+0.05$
8	5.41	5.59	$+0.17$
Mean	3.72	3.53	-0.20

or a terminal end-of-text score. Both regimes use a fixed coefficient $\lambda = 2/\ln 10 \approx 0.87$, matching the logarithm units of the baseline coefficient 2.00 on KenLM's base-10 scores. This conversion does not calibrate the two models' score distributions. The coefficient is fixed before TDT decoding, with no weight search.

Evaluation. We evaluate one NHN training run in both the zero-shot and fine-tuned regimes on all TDT sessions. Each regime covers eight participants with two complete sessions each, giving 32 paired model–session evaluations. Greedy, KenLM, and GPT-2 use the same NHN outputs and reference keystrokes. We pool edit counts across each participant's two sessions, then report the mean CER and population standard deviation across the eight participants. These are single-run results, distinct from the five-run averages in the main tables.

Results. Both language models reduce mean CER relative to greedy decoding (Table 10), but replacing KenLM has different effects in the two regimes. Zero-shot CER rises from 27.90% to 30.73%, an increase of 2.83 pp, with higher error for seven of the eight participants. After fine-tuning, mean CER falls from 3.72% to 3.53%, a reduction of 0.20 pp computed before rounding. Five participants improve and three worsen (Table 11). The largest reduction is 0.54 pp for Participant 4, while the largest increase is 0.17 pp for Participant 8.

Interpretation. The benefit of replacing KenLM depends on the decoding regime. GPT-2 lowers average CER after fine-tuning but raises it in zero-shot decoding, where NHN's character predictions are less accurate. It can change which prefixes survive throughout search, so the comparison is not restricted to the final KenLM candidates. The observed gains and losses reflect how the language-model scores interact with NHN evidence at the fixed decoding weights. We therefore retain KenLM for the main evaluations. GPT-2 remains an external decoding model, excluded from the NHN-only parameter and computation figures.

D.7 Session-weighted pose evaluation

User-weighted evaluation gives each participant equal weight, regardless of how many sessions they contribute. Session-weighted evaluation instead characterizes accuracy across acquisition sessions by

Table 12: Session-weighted emg2pose results. NHN reports mean ± sample SD across five seeds. Learned baselines use one released checkpoint. Bold marks the lowest error within each setting.

	User		Stage		User×Stage	
Model	AE (°)↓	LD (mm)↓	AE (°)↓	LD (mm)↓	AE (°)↓	LD (mm)↓
(a) Regression						
Static	16.93	25.18	18.60	27.78	18.07	27.47
NeuroPose	13.22	17.48	16.54	23.26	16.80	24.10
emg2pose	12.64	16.22	14.57	19.91	14.80	20.70
vemg2pose	12.33	15.78	14.58	19.77	14.84	20.51
NHN	**11.49** ±0.06	**14.43** ±0.06	**13.18** ±0.03	**17.62** ±0.05	**13.53** ±0.05	**18.58** ±0.04
(b) Tracking						
Static	10.44	15.50	14.96	22.55	14.08	21.93
emg2pose	11.35	14.67	14.74	20.19	14.22	19.83
vemg2pose	7.70	10.12	11.02	15.19	10.80	15.35
NHN	**7.33** ±0.01	**9.49** ±0.02	**9.96** ±0.02	**13.62** ±0.05	**9.95** ±0.06	**14.06** ±0.08

using sessions as the averaging unit within each setting. The same three generalization settings receive equal weight (Appendix C.3). We evaluate the five NHN models for each task and the available pretrained baseline checkpoints on the same non-empty split-session groups, retaining the standard 5 s scoring horizon. We also evaluate Static under its task-specific definitions (Appendix C.4). For Regression, the median pose is estimated from fully valid official training windows and held fixed throughout evaluation. Table 12 provides the split-specific results underlying the main table's Session columns.

Across the three settings, NHN achieves $12.73 \pm 0.03°$ AE and 16.88 ± 0.04 mm LD on Regression, and $9.08 \pm 0.03°$ AE and 12.39 ± 0.03 mm LD on Tracking. These standard deviations are computed from the five overall seed scores, not from the variation between sessions. Relative to vemg2pose, AE and LD decrease by 8.48% and 9.69% on Regression and by 7.73% and 8.58% on Tracking. NHN also has lower AE and LD in each individual split, with larger absolute reductions on Stage and User×Stage than on User in both tasks. The advantage therefore holds under both user-weighted and session-weighted evaluation, including splits that contain unseen stages. Both aggregations use the same models and differ in how they weight the test data.

D.8 Performance across training seeds

Table 13 summarizes the five training seeds for both tasks. For pose, each seed's score averages users or sessions equally within each split, then averages the three splits equally. For typing, each seed's CER averages participants equally after pooling session edit counts within each participant (Appendix C.3).

E Numerical Constants Quick Reference

Accuracy. NHN attains a session-weighted AE of $12.73°$ on emg2pose Regression and $9.08°$ on Tracking (Tables 1(a), 1(b)). The vemg2pose Salter checkpoints yield $13.91°$ and $9.84°$, respectively, under the same session-weighted evaluation. These scores average sessions within each split, then weight the three splits equally (Appendix C.3). Averaged over five emg2qwerty seeds, zero-shot TDT CER is $38.62 \pm 7.01\%$ with greedy decoding and $28.75 \pm 8.29\%$ with beam search. Fine-tuning gives $9.41 \pm 3.51\%$ and $3.83 \pm 1.51\%$, respectively. Each ± denotes SD across participants (Tables 2(a) and 2(b)).

Table 13: NHN performance across five training seeds. The final column reports the mean and sample SD across seeds, computed before rounding. AE is in degrees, LD in millimeters, and CER in percent.

Setting, metric	Run 1	Run 2	Run 3	Run 4	Run 5	Mean $\pm$ SD
(a) emg2pose: user-weighted						
Regression, AE	13.07	13.10	13.18	13.13	13.14	13.12 ± 0.04
Regression, LD	17.25	17.29	17.36	17.28	17.30	17.30 ± 0.04
Tracking, AE	9.14	9.22	9.22	9.20	9.20	9.20 ± 0.03
Tracking, LD	12.34	12.47	12.38	12.40	12.35	12.39 ± 0.05
(b) emg2pose: session-weighted						
Regression, AE	12.68	12.72	12.78	12.74	12.75	12.73 ± 0.03
Regression, LD	16.82	16.90	16.92	16.85	16.89	16.88 ± 0.04
Tracking, AE	9.03	9.08	9.10	9.08	9.10	9.08 ± 0.03
Tracking, LD	12.36	12.43	12.39	12.42	12.36	12.39 ± 0.03
(c) emg2qwerty: zero-shot CER						
ODV, greedy	56.01	56.50	57.26	57.68	57.20	56.93 ± 0.67
ODV, beam	50.55	50.65	52.27	52.76	52.41	51.73 ± 1.05
TDV, greedy	38.57	38.95	39.52	39.89	38.22	39.03 ± 0.68
TDV, beam	28.44	29.37	29.24	30.49	28.88	29.29 ± 0.76
TDT, greedy	38.11	38.79	38.95	39.25	38.00	38.62 ± 0.54
TDT, beam	27.90	28.89	29.00	29.52	28.44	28.75 ± 0.61
(d) emg2qwerty: fine-tuned CER						
TDV, greedy	8.82	9.19	9.24	9.08	9.27	9.12 ± 0.18
TDV, beam	3.76	4.00	3.90	4.00	4.08	3.95 ± 0.12
TDT, greedy	9.20	9.52	9.51	9.38	9.45	9.41 ± 0.13
TDT, beam	3.72	3.89	3.83	3.70	4.02	3.83 ± 0.13

Model size and compute. NHN Regression uses 3.08 million parameters, compared with 5.98 million in the official vemg2pose Regression checkpoint. The Tracking configurations use 3.08 and 5.97 million parameters, respectively. For both settings, NHN and vemg2pose require 2.40 and 3.72 GFLOPs per 5 s scored window, respectively (Table 1). emg2qwerty uses 0.88 million parameters and 17.85 GFLOPs per 30 s input (approximately 0.59 GFLOPs/s), against SplashNet-Upscale's 2.58 million parameters and 2.38 GFLOPs/s (Table 2). These are neural-operation counts, excluding language-model decoding, under the accounting convention in Appendix C.3. One local-global mixing block at $D_r = 256$, $G = 16$, $k = 5$, and $R = 32$ uses $18{,}944$ convolution and projection weights. Two dense mixers use $131{,}072$ weights ($6.92\times$ more), excluding biases and normalization in both counts.

Architecture constants. Appendix B.4 gives the shared and task-specific configurations, including adaptation, temporal settings, initialization, and numerical constants.

Theoretical quantities. Appendix A.3 gives the allocation-weight sensitivity $1/(2\theta)$ and latent neuromotor state sensitivity $N\|\bar{\boldsymbol{d}}_t\|_\infty/(2\theta) \le NK/2$ to priority perturbations. For $N = 32$ and $K = 12$, the latter upper bound is 192. This bounds worst-case sensitivity to priority perturbations at fixed drives. Appendix A.4 gives an $O(1/T)$ squared-gradient stationarity bound for fixed-drive priority fitting and local linear convergence near realizable solutions, supported by relative-priority identifiability.

F Complementary Perspectives on Representation Learning

Representational structure can arise from the task being performed, the organization of memory, or assumptions about how observations are generated.

F.1 Task objectives and motor redundancy

Accounts of coordination can focus on preferred movement trajectories or on variability that preserves task performance. Minimum-jerk models explain characteristic planar reaching trajectories by minimizing the integrated squared rate of change of hand acceleration (Flash & Hogan, 1985). The uncontrolled-manifold framework distinguishes joint variation that preserves a hypothesized task variable from variation that changes it (Scholz & Schöner, 1999). Optimal feedback control explains how selective correction of task-relevant deviations can maintain performance while permitting variability in redundant directions (Todorov & Jordan, 2002). These accounts connect coordination to the objective being stabilized. A variation tolerated by one behavioral objective may remain informative for another, so useful invariance depends on the target variable rather than dimensional reduction alone.

F.2 Temporal coverage and selective memory

Long-range sequence models address both how far history extends and which information survives in memory. Structured state-space sequence models use a linear dynamical core that admits recurrent and convolutional evaluations (Gu et al., 2022). Its learned kernel is independent of the particular input sequence. Content-dependent memory uses a different mechanism: Gated Delta Networks combine forgetting gates with targeted delta-rule updates to stored associations (Yang et al., 2025). Titans places part of the history in neural-memory parameters updated at test time, alongside attention to recent context (Behrouz et al., 2025). These approaches separate a memory's temporal reach from its storage and update rules. Increasing the available context therefore addresses a different question from deciding which events to retain, revise, or forget.

F.3 Structured generation and differentiable synthesis

Generative models constrain representations through the observations they must explain. Helmholtz machines pair a stochastic generative model with a recognition pathway that approximates inference over hidden causes (Dayan et al., 1995). Hierarchical predictive coding couples predictions and residual errors (Rao & Ballard, 1999). Higher levels predict activity below them, while residual errors pass forward. Differentiable digital signal processing (DDSP) introduces structure directly into audio synthesis through differentiable oscillators, filters, and reverberation (Engel et al., 2020). Its predicted controls acquire meaning through the operations that generate sound, rather than through an unconstrained waveform output. In each case, generative assumptions constrain learned inference. The form of the synthesizer can organize a latent space even when its controls are learned from observations.