
EMG-GPT: Predictive Pretraining on Residual-Quantized EMG Tokens for Hand Pose Estimation

Ettore Magni *
Imperial College London

Rolandos Alexandros Potamias
Imperial College London

Stefanos Zafeiriou
Imperial College London

Konstantinos Barmpas
Imperial College London

Abstract

Surface electromyography (sEMG) is a low-power, cost-effective biosignal for hand-pose estimation and gesture classification. In this work, we examine whether self-supervised pretraining on sEMG can yield transferable representations for continuous hand-pose estimation. We introduce EMG-GPT, a causal transformer-based model that operates on discrete sEMG representations from a frozen residual vector quantization (RVQ) tokenizer and learns temporal dynamics through depth-autoregressive future-code prediction. The model combines within-frame integration with causal temporal modeling while preserving the geometry of the pretrained codebook. EMG-GPT shows competitive results in both Regression and Tracking tasks, supporting EMG-only pretraining as a viable approach for learning transferable sEMG representations.

1 Introduction

Hand pose estimation and gesture classification on wearable devices have attracted increasing interest due to their potential in a wide range of applications, particularly in human–computer interaction [Jarque-Bou et al., 2021]. Although camera-based approaches can achieve high accuracy [Pavlakos et al., 2023, Potamias et al., 2025, Si et al., 2026], they are susceptible to occlusions, which can compromise the stability and reliability of vision-based estimations and predictions. These limitations have motivated the exploration of alternative sensing modalities [Tchantchane et al., 2023]. Surface electromyography (sEMG) measures the electrical activity generated during muscle activation, providing a wearable and non-invasive signal for estimating hand movements.

Unlike gesture classification, continuous pose estimation must recover an evolving articulated state. Existing systems learn supervised mappings from sEMG to joint trajectories [Quivira et al., 2018, Liu et al., 2021, Simpetru et al., 2022]. The emg2pose benchmark [Salter et al., 2024] supports this task at unusual scale, with 370 hours of paired sEMG and motion capture from 193 participants and generalization tests across users, sensor placements and activity stages. Such supervised results establish task-specific decoding performance, but do not by themselves show whether unlabelled EMG supports reusable predictive representations. We therefore study whether temporal pretraining with EMG-derived targets can produce states that transfer to motor decoding without exposing the representation learner to kinematic labels. Downstream evaluation then tests what these states retain when pose-specific supervision is introduced.

Many biosignal pretraining approaches learn from masked or reconstructed signal segments [Yang et al., 2023, Cui et al., 2024, Jiang et al., 2024] while autoregressive pretraining has also been

*Conducted during an internship at Imperial.

explored for electroencephalography (EEG) [Yue et al., 2024]. NeuroRVQ [Barmpas et al., 2026] provides a complementary discrete interface: it decomposes biosignals into frequency-specific representations, encodes them with an ordered residual vector quantization (RVQ) hierarchy and explores masked-token modeling over the resulting codes. Its output representations are structured over electrodes, frequency branches and residual depth rather than drawn from one flat vocabulary, making preservation of that structure necessary and part of the modeling problem.

Building on the NeuroRVQ work, we introduce EMG-GPT, a causal predictor of structured EMG representation tokens. By keeping the pre-trained NeuroRVQ-EMG tokenizer and codebooks fixed, we study the temporal representation learned above this interface. Rather than flattening every electrode–branch–depth code into the temporal sequence, EMG-GPT forms one temporal state per EMG frame, keeps sequence length independent of the retained RVQ depth K and predicts future-frame codes in RVQ depth order against the corresponding frozen codebooks. Keeping the discrete interface fixed separates temporal pretraining from tokenizer adaptation.

However, token predictability is not equivalent to kinematic utility. A model may reduce predictive cross-entropy by exploiting token regularities unrelated to hand motion. We test movement relevance by transferring the learned states to the emg2pose Regression and Tracking tasks under held-out generalization conditions. Our contributions are:

1. EMG-GPT, a frame-level causal future-prediction formulation based on NeuroRVQ-EMG tokens whose temporal sequence length is independent of retained RVQ depth K
2. a depth-autoregressive prediction mechanism that conditions finer RVQ levels on coarser levels and scores them against their corresponding frozen codebooks
3. a controlled transfer study across retained depth and pretraining progress on continuous hand-pose Regression and Tracking tasks

2 Background

2.1 Frozen NeuroRVQ EMG representations

NeuroRVQ maps a multi-channel (C channels) EMG time window to B frequency branches and applies a branch-specific RVQ codebook stack (K layers) at every electrode [Barmpas et al., 2026]. Let $\mathbf{E}_{b,k} \in \mathbb{R}^{M \times d_e}$ be the codebook for branch b and depth k , where M denotes the number of tokens and d_e is the token dimension. The integer-valued code index $q_{t,c,b,k} \in \{0, \dots, M-1\}$ selects a row of $\mathbf{E}_{b,k}$ at frame t , electrode c , branch b , and depth k , and has no geometric or ordinal meaning. For retained depth K , the structured frame array is:

$$\mathbf{Q}_t = (q_{t,c,b,k})_{c,b,k} \in \{0, \dots, M-1\}^{C \times B \times K} \quad (1)$$

The reconstructed tokenizer vector at each electrode–branch position is:

$$\hat{\mathbf{z}}_{t,c,b}^{(K)} = \sum_{k=1}^K \mathbf{E}_{b,k}[q_{t,c,b,k}] \quad (2)$$

We refer the reader to Appendix A for a detailed analysis of the NeuroRVQ-EMG token representation.

2.2 Structured causal prediction

Equation (1) is not a one-dimensional language sequence: each time step contains electrode, branch, and residual-depth axes. A temporal predictor must therefore model causal dependencies across frames while accounting for the ordered residual structure within the frame being predicted. This creates a representation-learning problem with two distinct structures: (i) time-ordered context and (ii) a branch-specific coarse-to-fine code hierarchy.

3 Method

EMG-GPT uses the produced NeuroRVQ tokens and its codebook geometry to represent each EMG frame. A **bidirectional intra-frame encoder** integrates simultaneous electrode–frequency features into one frame state, while a **causal transformer** models their temporal evolution. EMG-only pretraining predicts future codes coarse-to-fine across RVQ depth. Tensor-level analysis is described in Appendix B.

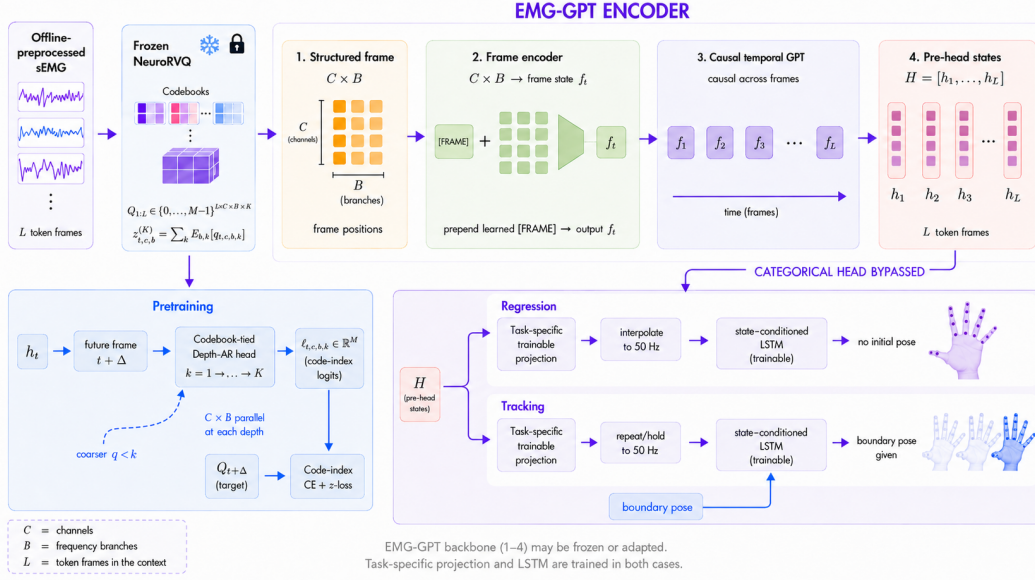


Figure 1: Overview of EMG-GPT. sEMG patches are mapped by frozen NeuroRVQ to structured time–electrode–branch–depth codes. Frozen vector sums are integrated within each frame and modeled causally across frames. The pre-head state either predicts a future frame coarse-to-fine with code-index cross-entropy, or bypasses the categorical head and transfers to parallel Regression and Tracking decoders under frozen or full-backbone adaptation. Regression interpolates features to the 50-Hz pose grid without an initial pose, whereas Tracking repeats features and receives the boundary pose. Only token-frame modeling is causal.

3.1 Structured RVQ Frame Encoding

From Q_t in Eq. (1), we retain any prefix $1 \leq K \leq K_{\max}$, where $K_{\max}$ is the tokenizer’s full RVQ depth. Its frozen RVQ reconstruction is:

$$\hat{\mathbf{z}}_{t,c,b}^{(K)} = \sum_{k=1}^K \mathbf{E}_{b,k} [q_{t,c,b,k}], \quad (3)$$

$$\mathbf{x}_{t,c,b} = \mathbf{W}_{\text{in}} \text{LN} \left(\hat{\mathbf{z}}_{t,c,b}^{(K)} \right) + \mathbf{e}_c^{\text{ch}} + \mathbf{e}_b^{\text{br}} \quad (4)$$

Here d_{model} is the model hidden dimension, LN is a LayerNorm layer over the reconstructed $\hat{\mathbf{z}}_{t,c,b}^{(K)}$ vector and $\mathbf{W}_{\text{in}} \in \mathbb{R}^{d_{\text{model}} \times d_e}$ is a learned bias-free projection from the d_e dimension to the d_{model} dimension. Finally, $\mathbf{e}_c^{\text{ch}}$ and $\mathbf{e}_b^{\text{br}}$ are learned channel and frequency-branch embeddings. Residual vector summation follows the ordered RVQ construction [Zeghidour et al., 2022, Barmpas et al., 2026], whereas the normalization, projection, and structural embeddings define our model input.

We define the ordered frame sequence $\mathbf{X}_t$ as a channel-major and branch-minor sequence:

$$\mathbf{X}_t = [\mathbf{x}_{t,1,1}, \dots, \mathbf{x}_{t,1,B}, \mathbf{x}_{t,2,1}, \dots, \mathbf{x}_{t,C,B}] \in \mathbb{R}^{CB \times d_{\text{model}}} \quad (5)$$

Given a learned [FRAME] token and applying the intra-frame transformer gives:

$$\mathbf{f}_t = \left[\text{LN}_{\text{fr}} F_{\text{fr}}([\mathbf{e}^{\text{[FRAME]}}, \mathbf{X}_t]) \right]_{\text{[FRAME]}} \quad (6)$$

F_{fr} denotes the bidirectional intra-frame Transformer, LN_{fr} its final LayerNorm, $\mathbf{e}^{[\text{FRAME}]}$ the learned frame-summary token and $[\cdot]_{[\text{FRAME}]}$ selects its output, yielding $\mathbf{f}_t \in \mathbb{R}^{d_{\text{model}}}$ for frame t .

The CB positions are simultaneous electrode–frequency features, making the intra-frame attention *bidirectional*. Since RVQ depth has already been combined before this encoder, increasing K enriches each frame and adds prediction targets without increasing the CB positions or the temporal sequence length.

3.2 Hierarchical Predictive Pretraining

For a context of L token frames, learned absolute temporal embeddings are added to the frame states before a decoder-only transformer:

$$\mathbf{h}_{1:L} = \text{LN}_{\text{out}} F_{\text{GPT}}(\text{Drop}([\mathbf{f}_t + \mathbf{e}_t^{\text{time}}]_{t=1}^L)) \quad (7)$$

where $\mathbf{e}_t^{\text{time}}$, Drop , F_{GPT} and LN_{out} denote the temporal embedding, dropout, decoder-only temporal Transformer, and final LayerNorm, respectively, and $\mathbf{h}_t \in \mathbb{R}^{d_{\text{model}}}$ is the resulting frame state. The causal mask [Vaswani et al., 2017] restricts $\mathbf{h}_t$ to frame states through t . Training pairs are denoted with $\mathbf{Q}_{1:L}$ and $\mathbf{Q}_{1+\Delta:L+\Delta}$, where Δ is an offset in token frames. If adjacent frames are separated by δt , the forecast horizon is $\tau_{\text{forecast}} = \Delta \delta t$. We use a 25-Hz configuration (5, 40 ms), predicting 200 ms ahead. We therefore describe the temporal objective generically as causal future-frame prediction. Prediction follows RVQ depth because each finer code describes a residual left by coarser codes [Lee et al., 2022]. For target depth k , define the reconstruction from realized coarser codes:

$$\hat{\mathbf{z}}_{t+\Delta,c,b}^{(<k)} = \sum_{r=1}^{k-1} \mathbf{E}_{b,r}[q_{t+\Delta,c,b,r}], \quad \hat{\mathbf{z}}^{(<1)} = \mathbf{0} \quad (8)$$

Given $\mathbf{Q}_{\leq t}$ as the context, $\mathbf{Q}_{t+\Delta, :, :, <k}$ the earlier target depths and p_θ as the model distribution, the factorization is defined as:

$$p_\theta(\mathbf{Q}_{t+\Delta} \mid \mathbf{Q}_{\leq t}) = \prod_{k=1}^K \prod_{c=1}^C \prod_{b=1}^B p_\theta(q_{t+\Delta,c,b,k} \mid \mathbf{Q}_{\leq t}, \mathbf{Q}_{t+\Delta, :, :, <k}) \quad (9)$$

Bidirectional self-attention exchanges hidden information among same-depth queries before cross-attention to $\mathbf{h}_t$. Thus, the model is: (i) causal across time, (ii) autoregressive across RVQ depth and (iii) conditionally factorized across channel–branch targets within a depth. Training uses ground-truth coarser codes while generation uses previously predicted ones.

Each query is projected into codebook space and scored against its matching frozen branch–depth codebook by learned scaled *cosine similarity*: code indices provide categorical targets, while codebook vectors are the output prototypes. With logits $\ell_{n,t,c,b,k} \in \mathbb{R}^M$, the objective for N complete windows is:

$$\begin{aligned} \mathcal{L}_{\text{CE},k} &= -\frac{1}{NLCB} \sum_{n,t,c,b} \log \text{softmax}(\ell_{n,t,c,b,k})_{q_{n,t+\Delta,c,b,k}}, \\ \mathcal{L}_{\text{pre}} &= \frac{1}{K} \sum_{k=1}^K (\mathcal{L}_{\text{CE},k} + \lambda_z \mathcal{L}_{z\text{-loss}}) \end{aligned} \quad (10)$$

We use $\lambda_z = 10^{-4}$ for z-loss regularization term. We report the clean predictive metric $\mathcal{L}_{\text{CE}} = \frac{1}{K} \sum_k \mathcal{L}_{\text{CE},k}$, while optimization uses $\mathcal{L}_{\text{pre}}$. Complete shifted windows require no padding mask. No gradient reaches NeuroRVQ or its codebooks. Code-prediction loss trains the backbone, but pose transfer tests the pre-head states and bypasses this categorical objective.

3.3 Transfer to Hand Pose

Downstream computation uses the $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_L] \in \mathbb{R}^{N \times L \times d_{\text{model}}}$ as described in Section 3. A learned feature projection changes representation width, while temporal rate alignment is a separate operation. *Regression* linearly interpolates projected features to the 50-Hz pose grid, receives no ground-truth initial pose, predicts absolute joint angles for an initialization prefix, and then integrates predicted angular increments. *Tracking* causally repeats projected features on the 50-Hz grid, receives

the ground-truth pose at the scoring boundary, and integrates predicted increments from that state. They are parallel benchmark protocols and are reported separately [Salter et al., 2024].

Frozen transfer fixes the hidden-state backbone and trains only the feature projection and recurrent decoder, probing representation quality [Yang et al., 2021]. Full-backbone adaptation warm-starts the frozen pose model and additionally trains the EMG-GPT encoder. NeuroRVQ, its codebooks, and the bypassed categorical head remain fixed in every evaluation protocol.

4 Evaluation

We preserve the emg2pose partitions and report Regression separately from Tracking, which receives a ground-truth boundary pose [Salter et al., 2024]. *User* holds out participants, *Stage* holds out kinematic categories, and *User, Stage* holds out both. During evaluation we use $K = 4$, 25-Hz representation and full-backbone adaptation (see Appendix C). Tracking adds velocity supervision. Selection details are described in Appendices D. Appendix E reports details evaluation results.

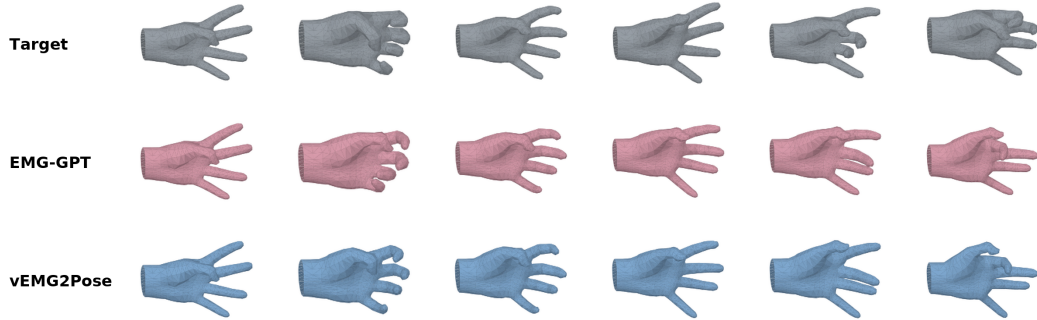


Figure 2: Tracking poses for EMG-GPT and vEMG2Pose. Top: motion capture; middle: EMG-GPT Tracking; bottom: vEMG2Pose [Salter et al., 2024].

Table 1: Regression test results. Published baselines are from Salter et al. [2024]; EMG-GPT is a fully adapted model. Angular and landmark values are mean $\pm$ sample SD across users within each generalization condition.

Test set	Model	Angular MAE ($^{\circ}$) $\downarrow$	Landmark (mm) $\downarrow$
User	SensingDynamics	15.5 ± 1.4	21.8 ± 2.1
	NeuroPose	13.2 ± 1.1	17.5 ± 1.3
	vEMG2Pose	12.2 ± 1.3	15.8 ± 1.9
	EMG-GPT	12.8 ± 1.1	16.3 ± 1.3
Stage	SensingDynamics	18.8 ± 1.6	26.6 ± 2.0
	NeuroPose	17.2 ± 1.7	24.0 ± 2.1
	vEMG2Pose	15.2 ± 1.6	20.4 ± 2.2
	EMG-GPT	16.3 ± 1.6	22.0 ± 1.9
User, Stage	SensingDynamics	18.7 ± 1.6	27.2 ± 2.0
	NeuroPose	17.5 ± 1.5	24.9 ± 1.7
	vEMG2Pose	15.8 ± 1.4	21.6 ± 2.0
	EMG-GPT	16.6 ± 1.3	22.7 ± 1.4

Across all three conditions, Regression has lower angular and landmark errors than SensingDynamics and NeuroPose, while vEMG2Pose remains marginally stronger. Tracking displays good results and remains only marginally behind vEMG2Pose (see Appendix E). Because the external results differ in frontend, augmentation, temporal alignment, and scoring grid, these tables support descriptive comparison rather than a significance or benchmark-superiority claim. We refer the reader to Appendix E for a matching evaluation protocol.

5 Conclusions

In this work, we introduce EMG-GPT, an transformer-based model that combines frozen NeuroRVQ geometry, bidirectional within-frame integration, causal temporal modeling and depth-autoregressive future-code prediction. A series of experiments support transfer from EMG-only pretraining. Future work will investigate joint tokenizer–predictor adaptation, motion-preserving pretraining objectives, and causal streaming inference. More broadly, scaling pretraining across users, devices, and datasets could support a transferable sEMG foundation model for pose estimation, gesture recognition, and human–machine interaction.

A Frozen NeuroRVQ EMG Tokenization Interface

This section describes the NeuroRVQ-EMG tokenizer formulation [Barmpas et al., 2026]. It provides the residual-quantization formulation and codebook notation required by our EMG-GPT as described in Section 3.

A.1 Codebook assignment and residual recurrence

For a fixed frame t , electrode c and branch b , the transformer encoder in NeuroRVQ-EMG tokenizer produces a latent vector $\mathbf{z} \in \mathbb{R}^{d_e}$. We suppress these indices below, writing $q_k = q_{t,c,b,k}$ and $\mathbf{E}_k = \mathbf{E}_{b,k}$. Let $\mathbf{E}_k = \{\mathbf{e}_{k,j}\}_{j=0}^{M-1}$ denote the codebook at RVQ depth k with M tokens. The location of the selected token in $\mathbf{E}_k$ is denoted as $q_k \in \{0, \dots, M-1\}$. With $\mathbf{r}^{(0)} = \mathbf{z}$ and a numerical stabilizer $\varepsilon > 0$, the tokenizer applies:

$$\bar{\mathbf{r}}^{(k-1)} = \frac{\mathbf{r}^{(k-1)}}{\max(\|\mathbf{r}^{(k-1)}\|_2, \varepsilon)}, \quad (11)$$

$$q_k = \arg \min_{j \in \{0, \dots, M-1\}} \left\| \bar{\mathbf{r}}^{(k-1)} - \mathbf{e}_{k,j} \right\|_2^2, \quad (12)$$

$$\mathbf{v}^{(k)} = \mathbf{e}_{k,q_k}, \quad \mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - \mathbf{v}^{(k)}. \quad (13)$$

The reconstruction from the first K depths is defined as:

$$\hat{\mathbf{z}}^{(K)} = \sum_{k=1}^K \mathbf{v}^{(k)} \quad (14)$$

A.2 Structured EMG output

NeuroRVQ-EMG tokenizer contains $B = 4$ frequency branches. Each branch has a separate $K_{\max} = 16$ -level RVQ stack and every codebook contains $M = 8192$ token vectors of dimension $d_e = 128$. The same branch–depth codebook is shared across electrodes, but codebooks are not shared across branches or depths. NeuroRVQ-EMG is pre-trained on $C = 16$ global electrodes. At frame t , $q_{t,c,b,k} \in \{0, \dots, M-1\}$ is the code index at electrode c , branch b , and depth k . The prefix $\mathbf{Q}_t^{(K)} \in \{0, \dots, M-1\}^{C \times B \times K}$ collects the first K depths, with reconstructed vector $\hat{\mathbf{z}}_{t,c,b}^{(K)}$:

$$\mathbf{Q}_t^{(K)} = (q_{t,c,b,k})_{c=1,b=1,k=1}^{C,B,K}, \quad \hat{\mathbf{z}}_{t,c,b}^{(K)} = \sum_{k=1}^K \mathbf{E}_{b,k}[q_{t,c,b,k}]. \quad (15)$$

A.3 Identifiers and frozen geometry

Code indices are table addresses. For any permutation $\pi_{b,k}$, relabeling $q'_{t,c,b,k} = \pi_{b,k}(q_{t,c,b,k})$ and permuting the rows of $\mathbf{E}_{b,k}$ by the same map leaves every selected vector and therefore Eq. (15) unchanged. The relevant geometry is carried by the frozen vectors in the matching (b, k) codebook.

B EMG-GPT Predictive Interface

This section specifies the tensor operations summarized in Section 3. For a minibatch of N windows, the stored input has shape $[N, L, C, B, K]$, and frozen codebooks have shape $[B, K, M, d_e]$. We use d_{model} for the EMG-GPT hidden dimension.

B.1 Frame and temporal backbone

Frozen dequantization first yields:

$$\hat{\mathbf{z}}_{n,t,c,b}^{(K)} = \sum_{k=1}^K \mathbf{E}_{b,k}[q_{n,t,c,b,k}] \in \mathbb{R}^{d_e} \quad (16)$$

The selected RVQ-sum input path applies LayerNorm before a bias-free linear projection, then adds channel and branch embeddings:

$$\mathbf{x}_{n,t,c,b} = \mathbf{W}_{\text{in}} \text{LN}_{\text{in}}(\hat{\mathbf{z}}_{n,t,c,b}^{(K)}) + \mathbf{e}_c^{\text{ch}} + \mathbf{e}_b^{\text{br}} \quad (17)$$

Reshaping contiguous $[C, B]$ axes produces the channel-major, branch-minor sequence:

$$[\mathbf{x}_{1,1}, \dots, \mathbf{x}_{1,B}, \mathbf{x}_{2,1}, \dots, \mathbf{x}_{C,B}] \quad (18)$$

A learned [FRAME] token is prepended independently at each time step. The same bidirectional transformer is applied to all frames, and its final-layer-normalized [FRAME] output is $\mathbf{f}_{n,t}$.

Absolute time embeddings are then added and dropout is applied:

$$\mathbf{U} = \text{Drop}([\mathbf{f}_{n,t} + \mathbf{e}_t^{\text{time}}]_{n,t}) \in \mathbb{R}^{N \times L \times d_{\text{model}}}. \quad (19)$$

The selected temporal backbone applies causal transformer blocks followed by a final LayerNorm:

$$\mathbf{H} = \text{LN}_{\text{out}} F_{\text{GPT}}(\mathbf{U}) = [\mathbf{h}_1, \dots, \mathbf{h}_L] \in \mathbb{R}^{N \times L \times d_{\text{model}}} \quad (20)$$

Consequently, $\mathbf{h}_t$ depends only on token frames $1:t$. The dataset uses complete contiguous windows and pairs $\mathbf{Q}_{1:L}$ with $\mathbf{Q}_{1+\Delta:L+\Delta}$. Discontinuities split sequences rather than introducing padding. If the token-frame spacing is δt , the physical forecast horizon is $\Delta \delta t$.

B.2 Depth-autoregressive codebook head

At depth k , the head creates one query for every (c, b) position. Before the decoder blocks, its implemented query is:

$$\mathbf{a}_{t,c,b,k} = P_h(\mathbf{h}_t) + \mathbf{e}_c^{\text{out, ch}} + \mathbf{e}_b^{\text{out, br}} + \mathbf{e}_k^{\text{depth}} + \mathbb{I}[k > 1] \mathbf{W}_{<k} \text{LN}_{<k}(\hat{\mathbf{z}}_{t+\Delta,c,b}^{(<k)}) \quad (21)$$

where P_h is an affine hidden-state projection and $\mathbf{W}_{<k}$ is a shared bias-free projection from d_e to d_{model} . $\text{LN}_{<k}$ is LayerNorm over the d_e -dimensional partial reconstruction $\hat{\mathbf{z}}^{(<k)}$, which is defined in Eq. (8). Its value is zero at $k = 1$.

Shared decoder blocks first apply bidirectional self-attention across the CB queries at the current depth, then cross-attend each query to the single state $\mathbf{h}_t$, and finally apply a multilayer perceptron (MLP). Same-depth queries therefore exchange hidden context, but no realized same-depth target is supplied. Training uses ground-truth coarser codes in Eq. (21), and generation replaces them with previously generated coarser codes.

Let $\mathbf{g}_{t,c,b,k}$ be the final-layer-normalized query after these blocks and let $P_q : \mathbb{R}^{d_{\text{model}}} \rightarrow \mathbb{R}^{d_e}$ be the shared affine output projection. Logits for code index j are defined as:

$$\ell_{t,c,b,k,j} = s_{b,k} \left\langle \frac{P_q(\mathbf{g}_{t,c,b,k})}{\|P_q(\mathbf{g}_{t,c,b,k})\|_2}, \frac{\mathbf{E}_{b,k}[j]}{\|\mathbf{E}_{b,k}[j]\|_2} \right\rangle, \quad s_{b,k} = \min\{\exp(\gamma_{b,k}), s_{\text{max}}\} \quad (22)$$

where $\gamma_{b,k}$ is a learned branch–depth log-scale and s_{max} is a fixed upper bound on $s_{b,k}$. Thus each branch and depth is scored only against its matching frozen codebook. Decoder blocks and P_q are shared across depth. Depth embeddings and positive logit scales are depth-specific and branch–depth-specific, respectively. Channel and branch embeddings are shared over depth. The primary loss averages branch-wise summed cross-entropies over $NLCB$ targets at each depth and then applies the normalized depth weights in Eq. (10).

B.3 Effect of retained depth and transfer tensor

Increasing K adds vectors to Eq. (16) and ordered target levels to the head. It does not change the CB intra-frame positions, context length L or number of transferred temporal states. Transfer consumes exactly $\mathbf{H}$ from Eq. (20), after the final backbone LayerNorm and before the categorical head. It, therefore, receives neither code logits nor predicted future code indices.

C Implementation and Pretraining Details

Training slices the first K depths and constructs complete length- L windows paired with targets Δ token frames ahead. We use 25-Hz tokens with a 40-ms hop and $\Delta = 5$. This configuration predicts 200 ms ahead. Patch width is recorded separately and is not used to define this forecast interval. Missing storage blocks split a recording into separate sequences, so a window never spans a discontinuity. Codebooks are loaded as non-trainable runtime buffers with shape $[B, K, M, d_e]$. The input path and categorical head use the same frozen branch–depth tables, while all frame–encoder, temporal, and query–projection parameters are learned during pretraining.

Table 2: Configuration of the selected 25-Hz, $K = 4$ EMG-GPT model.

Field	Verified value
Token grid	25 Hz; 40-ms hop; independently encoded 200-ms patches
RVQ depth	$K = 4$ retained depths
Forecast	$L = 150$ frames (6.0 s context); $\Delta = 5$ frames (200 ms)
Offline preprocessing	Nominal 20–400 Hz third-order Butterworth with forward–reverse filtering; polyphase resampling from 2 kHz to 1 kHz
Input representation	Frozen $K = 4$ RVQ vectors summed across depth, LayerNorm, bias-free projection, learned channel/branch embeddings, CLS pooling
Temporal backbone	$d_{\text{model}} = 512$; 8 Transformer layers; 8 heads; dropout 0.1; 2-layer within-frame encoder
Output head	2-layer depth-autoregressive codebook head; uniform depth weighting
Parameters	40,494,736 trainable parameters (40.5M)
Objective	Hard categorical CE plus logit z-loss with $\lambda_z = 10^{-4}$
Optimizer	AdamW; peak LR 10^{-4} ; $\beta = (0.9, 0.95)$; weight decay 0.1
Batch and warmup	Batch 16; 2 gradient-accumulation steps; effective batch 32; 2,000-step linear warmup
Schedule	WSD through 200k, with cosine decay over steps 180k–200k to 10^{-5} ; constant 10^{-5} continuation from 200k to 400k
Precision and hardware	bfloat16 on one NVIDIA GH200 120GB; seed 123
Exact 400k exposure	12.8M windows; 1.92B predicted frames; 491.52B predicted code targets
Validation-selected GPT	Regression: 280k; Tracking: 400k

D Regression and Tracking Protocols

This section specifies the emg2pose evaluation protocols. Both tasks use the final normalized states of a pretrained $K = 4$ EMG-GPT. NeuroRVQ and its codebooks remain fixed. Inputs contain 150 token frames at 25 Hz (6 s), including 25 left-context frames (1 s). Pose targets are sampled at 50 Hz. Invalid inverse-kinematics rows are masked from optimization and scoring.

Pose decoders and alignment. A learned projection maps each 512-dimensional backbone state to 64 features. Both tasks then use a two-layer LSTM with hidden width 512 to predict 20 joint angles (see Figure 1). *Regression* linearly interpolates token features to 50 Hz with endpoint alignment. It receives no ground-truth initial pose: the decoder starts from zeros, predicts absolute pose for 12 steps, and subsequently integrates angular increments scaled by 0.01. The scored tail contains 250 pose steps (5 s), and windows advance by 150 token frames (6 s). *Tracking* uses a causal zero-order hold, repeating every 25-Hz feature for two pose steps. It receives the ground-truth pose at the scoring boundary, 1.04 s after the token-window start, and then autoregressively integrates increments scaled by 0.01. Its one-token-frame offset gives 40 ms token-to-pose latency. The scored tail is again 5 s, while windows advance by 125 token frames (5 s).

Objectives. The angular term is masked joint-angle L1 in radians. The fingertip term is the mean distance in millimetres over the five fingertips. Regression uses angular L1 plus 0.01 fingertip distance. Tracking adds velocity loss with weight 4. Full adaptation trains the RVQ–input mapping and intra-frame encoder, all temporal GPT blocks, temporal and structural embeddings, and final backbone LayerNorm. It uses Adam with an effective batch of 32, head learning rate 5×10^{-5} , backbone learning rate 5×10^{-6} , and a 5k-step cosine schedule.

E Transfer Evaluation

E.1 Tracking Results

Table 3: Tracking test results, where the boundary pose is given. vEMG2Pose values are from [Salter et al., 2024].

Test set	Model	Angular MAE ($^{\circ}$) $\downarrow$	Landmark (mm) $\downarrow$
User	vEMG2Pose	7.7 ± 1.0	10.3 ± 1.5
	EMG-GPT	8.9 ± 0.9	11.6 ± 1.2
Stage	vEMG2Pose	11.2 ± 1.4	15.2 ± 1.9
	EMG-GPT	12.4 ± 1.4	16.9 ± 1.8
User, Stage	vEMG2Pose	11.0 ± 1.0	15.4 ± 1.4
	EMG-GPT	12.2 ± 1.3	16.9 ± 1.4

E.2 Matching Evaluation Protocol with vEMG2Pose

Published vEMG2Pose values in the main tables (Tables 1 and 3) are external protocol references. To control for differences in evaluation sampling, we also evaluate the released vEMG2Pose Regression and Tracking checkpoints, without retraining, at the canonical test target indices used for EMG-GPT. Within each task, the ordered target-index hashes, valid-value counts, metric implementation, and equal-user aggregation are identical between models. Table 4 controls the effective test samples and aggregation locally. This local comparison is not a reproduction of the published protocol: vEMG2Pose retains its released raw-EMG frontend and weights, while EMG-GPT uses fixed, independently tokenized 200-ms patches, 25-Hz GPT features, and source-aligned 50-Hz scoring. Zero-phase filtering also makes the complete token pipeline offline. We therefore treat both comparisons as descriptive and do not make significance or benchmark-superiority claims.

Table 4: Locally matched test evaluation of the released vEMG2Pose checkpoints and EMG-GPT. Values are mean $\pm$ sample SD across users within each generalization condition. Bold indicates the lower mean; rounded ties are bolded together. Each method retains its native input frontend.

Test set	Model	Angular MAE ($^{\circ}$) $\downarrow$	Landmark (mm) $\downarrow$
Regression			
User	vEMG2Pose	12.80 ± 1.31	16.29 ± 1.85
	EMG-GPT	12.80 ± 1.08	16.30 ± 1.29
Stage	vEMG2Pose	15.40 ± 1.59	20.43 ± 2.09
	EMG-GPT	16.30 ± 1.57	22.02 ± 1.92
User, Stage	vEMG2Pose	15.73 ± 1.38	21.25 ± 1.79
	EMG-GPT	16.64 ± 1.30	22.70 ± 1.39
Tracking			
User	vEMG2Pose	8.57 ± 0.95	11.11 ± 1.35
	EMG-GPT	8.89 ± 0.93	11.59 ± 1.24
Stage	vEMG2Pose	11.50 ± 1.41	15.45 ± 1.81
	EMG-GPT	12.42 ± 1.43	16.88 ± 1.80
User, Stage	vEMG2Pose	11.33 ± 0.97	15.64 ± 1.33
	EMG-GPT	12.18 ± 1.25	16.93 ± 1.36

On held-out users, the matched Regression angular errors are equal after rounding to two decimals. vEMG2Pose has lower mean error in the remaining Regression comparisons and in all Tracking comparisons.

E.3 Qualitative Results

We select validation clips using pre-defined user and stage error percentiles, then choose windows nearest to the corresponding aggregate error. Figures 3–7 show the motion-capture target, the locked 25-Hz EMG-GPT Tracking model, and a locally evaluated vEMG2Pose reference on the same clips. Figures 8–10 apply the same selection rule to Regression, with target equality verified for each aligned comparison segment.

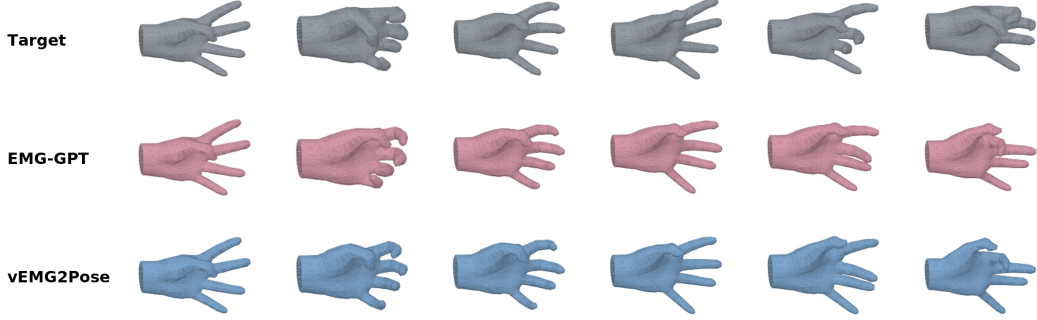


Figure 3: Median held-out user and median held-out stage (*FingerFreeform*). Top: motion capture; middle: EMG-GPT Tracking; bottom: vEMG2Pose. Six poses unroll from left to right over a five-second validation clip.

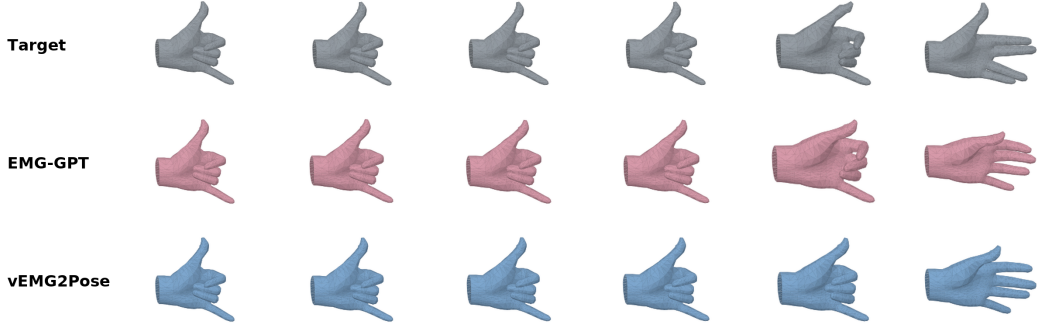


Figure 4: Median held-out user and lower-error held-out stage (*ShakaVulcanPeace*). Rows and temporal layout follow Figure 3.

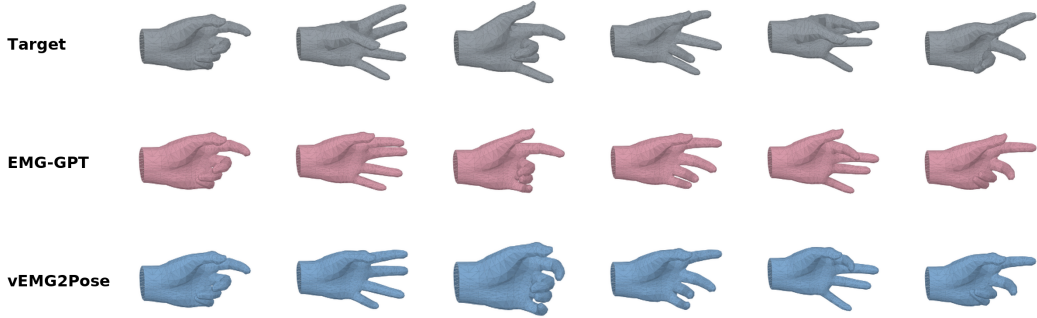


Figure 5: Median held-out user and higher-error held-out stage (*CountingUpDownFingerWigglingSpreading*). Rows and temporal layout follow Figure 3.

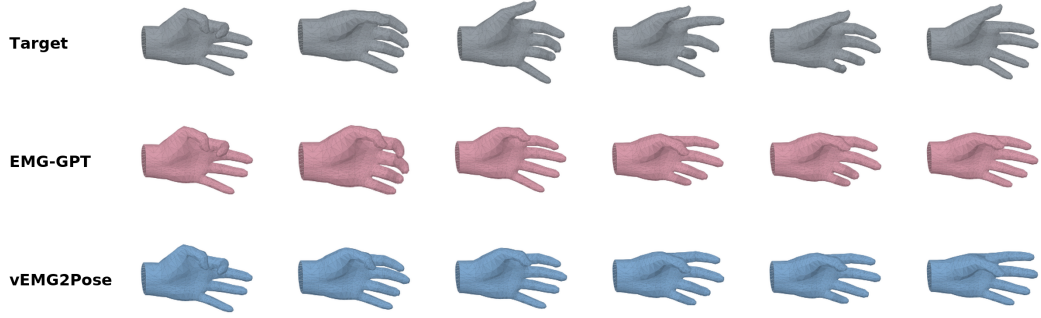


Figure 6: Lower-error held-out user and median held-out stage (*HandOverHandCountingUpDown-FingerWigglingSpreading*). Rows and temporal layout follow Figure 3.

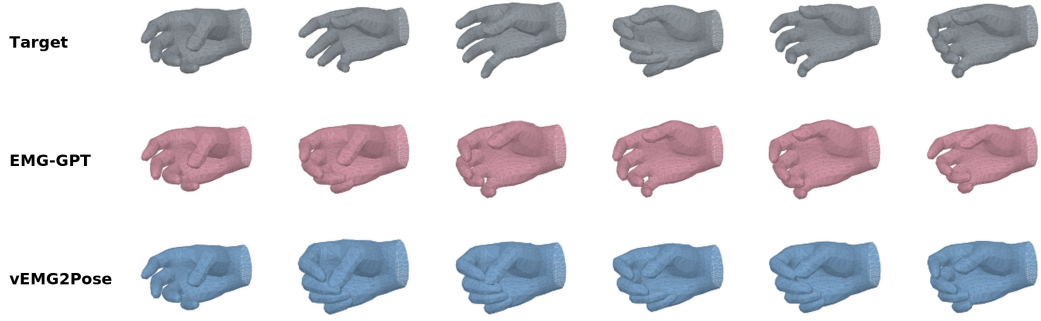


Figure 7: Higher-error held-out user and median held-out stage (*FingerFreeform*). Rows and temporal layout follow Figure 3.

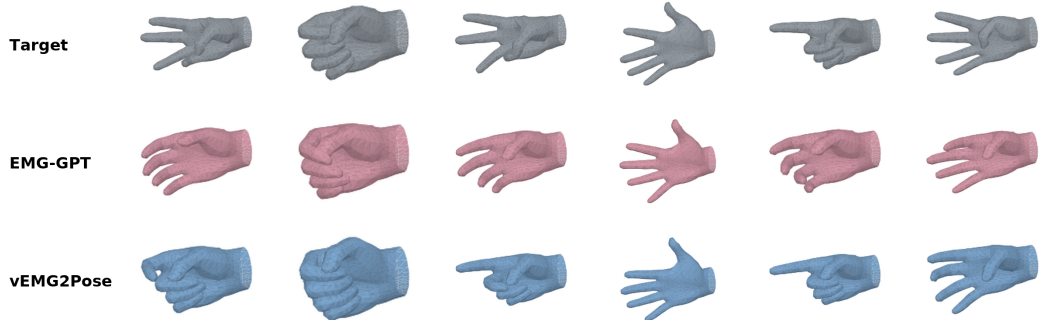


Figure 8: Regression example for a median held-out user and median held-out stage. Top: motion-capture target; middle: EMG-GPT Regression; bottom: vEMG2Pose Regression. Six poses unroll from left to right over the same target-aligned five-second validation clip.

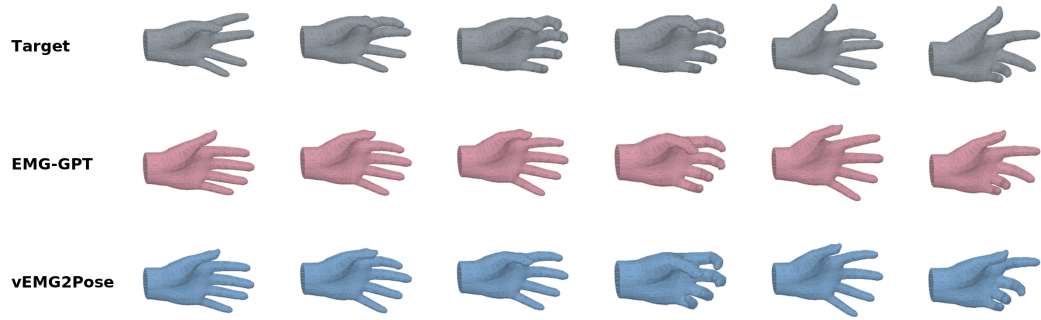


Figure 9: Regression example for a median held-out user and lower-error held-out stage. Rows and target-aligned temporal layout follow Figure 8.

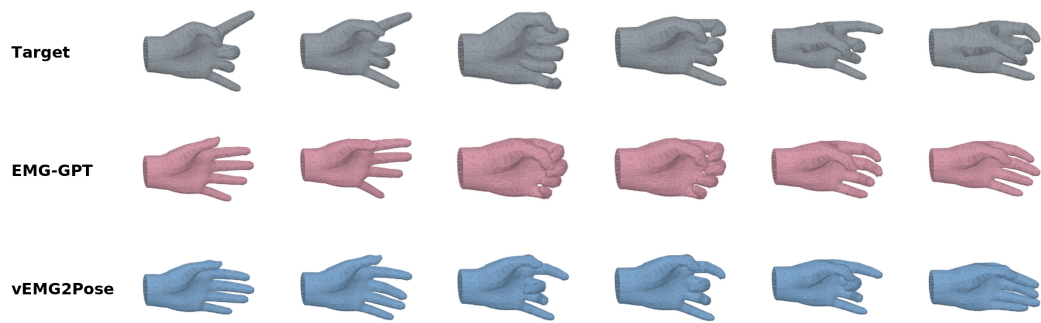


Figure 10: Regression example for a median held-out user and higher-error held-out stage. Rows and target-aligned temporal layout follow Figure 8.

References

- Konstantinos Barmpas, Na Lee, Dimitrios Chalatsis, William Raftery, Yannis Panagakis, Dimitrios A. Adamos, Nikolaos Laskaris, Alexandros Koliouisis, Dario Farina, and Stefanos Zafeiriou. Neurorvq: Multi-scale biosignal tokenization for generative foundation models, 2026. URL <https://arxiv.org/abs/2510.13068>.
- Wenhui Cui, Woojae Jeong, Philipp Thölke, Takfarinas Medani, Karim Jerbi, Anand A. Joshi, and Richard M. Leahy. Neuro-GPT: Towards a foundation model for EEG. In *IEEE International Symposium on Biomedical Imaging*, pages 1–5, 2024. doi: 10.1109/ISBI56570.2024.10635453.
- Néstor J. Jarque-Bou, Joaquín L. Sancho-Bru, and Margarita Vergara. A systematic review of emg applications for the characterization of forearm and hand muscle activity during activities of daily living: Results, challenges, and open issues. *Sensors*, 21(9), 2021. ISSN 1424-8220. doi: 10.3390/s21093035. URL <https://www.mdpi.com/1424-8220/21/9/3035>.
- Wei-Bang Jiang, Li-Ming Zhao, and Bao-Liang Lu. Large brain model for learning generic representations with tremendous EEG data in BCI. In *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=QzTpTRVtrP>.
- Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022. doi: 10.1109/CVPR52688.2022.01123.
- Yilin Liu, Shijia Zhang, and Mahanth Gowda. NeuroPose: 3d hand pose tracking using EMG wearables. In *Proceedings of the Web Conference 2021*, pages 1471–1482, 2021. doi: 10.1145/3442381.3449890.
- Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers, 2023. URL <https://arxiv.org/abs/2312.05251>.
- Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild, 2025. URL <https://arxiv.org/abs/2409.12259>.
- Fernando Quivira, Toshiaki Koike-Akino, Ye Wang, and Deniz Erdogmus. Translating sEMG signals to continuous hand poses using recurrent neural networks. In *IEEE EMBS International Conference on Biomedical and Health Informatics*, pages 166–169, 2018. doi: 10.1109/BHI.2018.8333395.
- 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, 2024. doi: 10.52202/079017-1770.
- Chen Si, Yulin Liu, Bo Ai, Jianwen Xie, Rolandos Alexandros Potamias, Chuanxia Zheng, and Hao Su. Anyhand: A large-scale synthetic dataset for rgb (-d) hand pose estimation. *arXiv preprint arXiv:2603.25726*, 2026.
- Raul C. Simpetru, Marius Osswald, Dominik I. Braun, Daniela S. Oliveira, Andre L. Cakici, and Alessandro Del Vecchio. Accurate continuous prediction of 14 degrees of freedom of the hand from myoelectric signals through convolutional neural networks. In *44th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 702–706, 2022. doi: 10.1109/EMBC48229.2022.9870937.
- Rayane Tchanchane, Hao Zhou, Shen Zhang, and Gursel Alici. A review of hand gesture recognition systems based on noninvasive wearable sensors. *Advanced Intelligent Systems*, 5(10):2300207, 2023. doi: <https://doi.org/10.1002/aisy.202300207>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL <https://proceedings.neurips.cc/paper/7181-attention-is-all-you-need>.

- Chaoqi Yang, M. Brandon Westover, and Jimeng Sun. BIOT: Biosignal transformer for cross-data learning in the wild. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3420.
- Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. SUPERB: Speech processing universal PERFORMANCE benchmark. In *Interspeech 2021*, pages 1194–1198, 2021. doi: 10.21437/Interspeech.2021-1775.
- Tongtian Yue, Xuange Gao, Shuning Xue, Yepeng Tang, Longteng Guo, Jie Jiang, and Jing Liu. BrainGPT: Unleashing the potential of EEG generalist foundation model by autoregressive pre-training. *arXiv preprint arXiv:2410.19779*, 2024. doi: 10.48550/arXiv.2410.19779. Version 2, preprint.
- Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2022. doi: 10.1109/TASLP.2021.3129994.