A scalable self-supervised method for modeling human intracranial recordings during natural behavior

Shivashriganesh P. Mahato[1]*, Jingyun Xiao[1]*, Alexandre Andre[1], Geeling Chau[2], Wenrui Ma[1], Ian J. Knight[1], Duy Nguyen[2], Lawrence Hu[3], Bingni W. Brunton[3], Michael S. Beauchamp[1], Bijan Pesaran[1], Sergey Shuvaev[1], Eva L. Dyer[1]
[1]University of Pennsylvania, [2]California Institute of Technology, [3]University of Washington
{dyer1,smahato,xiao11,aandre1,ma10,ijknight}@seas.upenn.edu
{sergey.shuvaev,bijan.pesaran,michael.beauchamp}@pennmedicine.upenn.edu
{gchau,dhnguyen}@caltech.edu
{bbrunton,jqhu}@uw.edu

Abstract

Understanding how the brain supports natural behavior is an increasingly central goal in human neuroscience. Recordings from human neurosurgical patients with intracranial EEG electrodes offer direct access to widespread electrical activity in the brain during a variety of behaviors over extended times. Despite the progress in the field, utilizing these recordings at scale to identify the neural underpinnings of natural human behavior remains difficult due to variability in electrode placement, channel geometry, and behavioral diversity across participants and sessions. To address these challenges, we introduce a self-supervised framework for multi-participant intracranial neural data. We use a Perceiver-based architecture to reconstruct masked channels of neural activity from unmasked channels using learnable embeddings of the channel identity and contextual information, capturing inter-channel dependencies without requiring labels. Finetuning of our self-supervised model has improved the decoding performance on a panel of downstream tasks, highlighting the potential of self-supervised learning to enable general-purpose neural decoding and support scalable integration of naturalistic human brain recordings.

1 Introduction

Intracranial electroencephalography (iEEG), including electrocorticography (ECoG) and stereotactic EEG (sEEG), yields high-resolution recordings of human brain activity with broad clinical and scientific utility [16]. iEEG recordings have become a technique of choice in neurosurgery due to their ability to measure human brain activity with higher resolution compared to non-invasive recordings such as EEG and MEG [3, 15, 7, 8, 13, 6]. Recent advances in data collection have made it possible to record from dozens of individuals over extended periods, capturing everyday, non-stereotyped behavior in hospital rooms, homes, and research settings [18, 4, 19]. These datasets open new opportunities for developing machine-learning models that can learn from large-scale, heterogeneous neural data without requiring rigid task structures or behavioral labeling [9, 23].

Self-supervised learning (SSL) has emerged as a tool for learning representations from neural data without labels. Prior work has shown that masking segments of neural activity and reconstructing them helps uncovering latent dynamics of neural activity [24, 26] and that predictive objectives across

*Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Foundation Models for the Brain and Body.

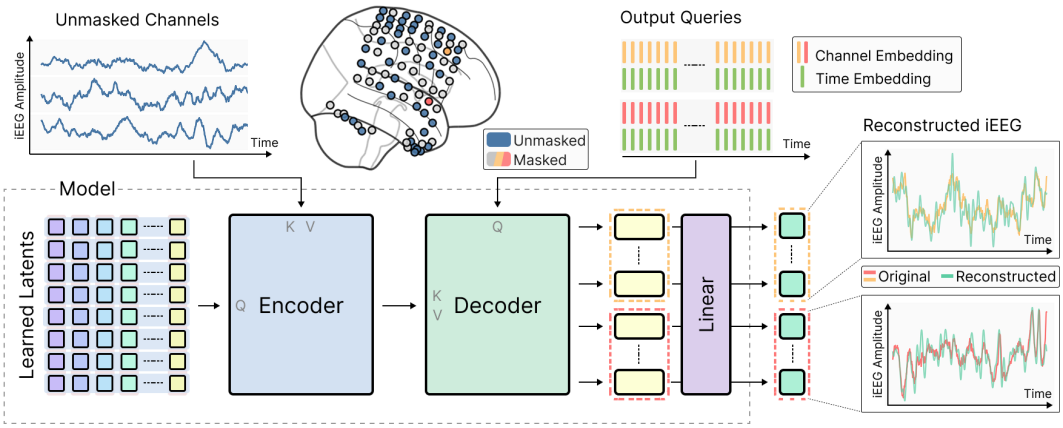

Figure 1: **Overview of our masked channel modeling framework.** During pretraining, a subset of iEEG channels is masked, and the model learns to reconstruct their activity using contextual information from the unmasked channels via learnable channel embeddings. A Perceiver-based encoder integrates spatial and temporal context across channels using cross-attention and learnable latent tokens.

time or modalities can support decoding in continuous behavior [17, 5]. These approaches, however, have largely focused on experimental settings with short recordings and stereotyped behaviors. The models developed to analyze these data may not be compatible with multi-source unlabeled data collected from patients who were not instructed to behave in accordance with any particular task.

To address this gap, we propose a SSL approach for iEEG data called `Charmander` (**CHA**nnel **M**asking **AND R**econstruction). In our approach, we mask subsets of channels and reconstruct their activity from the surrounding context. This formulation uses a Perceiver-based architecture [10] that compresses information from unmasked channels into a shared latent space and queries it using learnable channel embeddings. This combination enables the model to learn from large-scale unlabeled recordings while capturing spatial and functional relationships across brain regions and patients.

We evaluate our method on two large-scale iEEG datasets: AJILE12 [18] and the Brain Treebank [22]. Together, these datasets span 20+ participants and over 1,000+ hours of brain activity across different electrode configurations. We finetune our pretrained model on a set of downstream tasks and show that our self-supervised framework can learn transferable population-level neural representations that scale across individuals, tasks, and recording contexts.

Our main contributions are as follows:

- **A general and scalable SSL framework for iEEG.** We present a self-supervised learning framework that enables pretraining directly on raw, unlabeled intracranial recordings, accommodating variable electrode configurations and recording conditions.
- **A novel channel-level masking task for neural data.** We introduce a self-supervised objective, designed for large-scale neural data, where entire recording channels are masked and reconstructed by querying a compressed latent space using learnable channel embeddings.
- **Improved generalization through scale and cross-task transfer.** We show that our model benefits from scaling to longer recordings and more participants, increasing the performance on downstream tasks. The finetuned models outperform supervised and self-supervised baselines across modalities, sessions, and tasks, from in- and out-of-distribution settings.

2 Methods

We propose a model consisting of a neural activity tokenizer and a Perceiver-based encoder. We pretrained our model on unlabeled iEEG data using a self-supervised reconstruction objective and finetuneed it on downstream tasks. Below we detail the steps in this pipeline (Figure 1).

Data For each patient, we used iEEG recordings consisting of multivariate time-series where channels were derived from individual contacts on electrodes in the patient's brain. The placement of

the electrodes was determined by the clinical need, e.g. seizure localization for patients with epilepsy, and was fixed across long-term continuous recordings and across days; the electrode locations varied across patients. Raw voltage traces were recorded from the iEEG electrode contacts, corresponding to the bulk neural activity within the respective regions of the brain. Each channel j contained a univariate time series representing the continuous voltage signal sampled at a frequency f. For the datasets used in our work, the sampling rate was fixed across electrode types, patients, and time, but varied across datasets (Appendix A).

Tokenizer To process large data arrays of high-resolution iEEG signals, we split them into fixed-sized segments (temporal patches) $\mathbf{p}_{ij} = \left[\mathbf{u}_{iP,j}, \mathbf{u}_{iP+1,j}, \ldots, \mathbf{u}_{(i+1)P-1,j}\right]^\top \in \mathbb{R}^P$. Here $\mathbf{u}_{t,j}$ is the raw voltage amplitude of channel j at timestep t and P is the fixed size of a patch. Over T time-steps we obtained $N = \left\lfloor \frac{T}{P} \right\rfloor$ patches which were then projected onto a shared embedding space $\mathbb{R}^d$ using a learnable projection matrix $\mathbf{W}_p \in \mathbb{R}^{d \times P}$.

Taking inspiration from `Poyo+` [2], we combined the patch embeddings of neural activity with information about the corresponding channel identity and recording time as follows. For each patch $\mathbf{p}_{ij}$, where $i \in [0, \ldots, N-1]$ and $j \in [0, \ldots, K-1]$, we concatenated the projection of the neural activity $\mathbf{W}_p\mathbf{p}_{ij}$ with a learned embedding of the channel identity $\mathbf{c}_j \in \mathbb{R}^d$, unique for each of the K channels but constant in time, to form a joint embedding $\mathbf{x}_{ij} = \mathrm{Concat}\big(\mathbf{W}_p\mathbf{p}_{ij},\ \mathbf{c}_j\big)$. Relative timing information was incorporated by linking the resulting embedding $\mathbf{x}_{ij}$ with a $(2d)$-dimensional sinusoidal rotary embedding t_{ij} [20], derived from the center time-step of the patch relative to the beginning of the context window, to form a token $(\mathbf{x}_{ij}, t_{ij})$. We flattened all the tokens into a single sequence to be provided as input to the `Charmander` encoder, described below.

Encoder To transform the tokens of neural activity to compressed low-dimensional representations that would enable the decoding of behavioral variables in downstream tasks, we used the `Poyo+` encoder, featuring a Perceiver-based architecture with linear-time compute complexity w.r.t. the context window size. It computes cross-attention between the neural data and a low-dimensional set of learnable latents which, similar to the input iEEG tokens, were formed as all possible concatenations of H d-dimensional feature embeddings ("virtual channels") and M d-dimensional sinusoidal temporal embeddings ("virtual timestamps"). The result was a $H \times M$ grid of d-dimensional embeddings where $H \ll K$ and $M \ll N$. Each of these $H \times M$ latent embeddings served as a query to attend to iEEG activity tokens (serving as keys and values) using rotary [20] cross-attention. After this step, the architecture consisted of standard Transformer blocks performing rotary [20] self-attention, keeping the data within the dimensionality of the latent space.

A novel masking-based pretraining objective To capture population-level structure in neural neural activity dynamics across the spatially distributed electrodes, we randomly masked subsets of channels and trained the model to reconstruct their voltage signals, based on their learned channel embeddings, from the remaining unmasked channels. We randomly masked a subset $\mathcal{M}$ of input channels and passed the unmasked tokens $(\mathbf{x}_{ij}, t_{ij})$ for $j \notin \mathcal{M}$ as inputs to the `Charmander` encoder. To reconstruct each masked token, we formed a query using the corresponding channel embedding $\mathbf{c}_j$ and its associated timestep t_{ij} omitting the raw signal $\mathbf{p}_{ij}$. We passed these tokens, along with the outputs of the `Charmander` encoder, through a single cross-attention decoder block, and finally through a linear projection to predict the raw voltage values for each masked channel. The model was trained using the mean squared error (MSE) loss on the differences between the predicted $(\hat{\mathbf{u}}_{t,j})$ and the actual $(\mathbf{u}_{t,j})$ voltages.

3 Results

We evaluated the pretrained models on several downstream tasks, spanning classification of activities of daily living and auditory or lingual feature decoding tasks during movie-watching.

Self-supervised pretraining improves decoding of human behavior from neural activity To quantify the effect of self-supervised pretraining on the model's performance in downstream tasks, we implemented a behavioral classification task using labels from the AJILE12 dataset [18] describing activities of daily living (Appendix A.1) with simultaneous iEEG recordings (primarily ECoG). We

Table 1: **Human activity decoding from neural data.** F1-scores from a 5-way multilabel classification task after either end-to-end supervised training, or self-supervised pretraining and finetuning. Results are reported for three patients (P2, P3, P4). We test in three settings: **same session** (test data comes from same sessions used during pretraining of SSL models); **novel finetuned** (more clinically relevant setting where test data comes from sessions held-out of pretraining); and **novel zero-shot** (same as finetuning across sessions, except models trained/finetuned on held-in sessions are directly inferred on new sessions). Reported errors are SEM.

	Model	Same session			Novel finetuned			Novel zero-shot		
		P2	P3	P4	P2	P3	P4	P2	P3	P4
Supervised	MLP	0.189 ± 0.008	0.301 ± 0.022	0.260 ± 0.009	0.251 ± 0.007	0.256 ± 0.047	0.300 ± 0.017	0.252 ± 0.006	0.268 ± 0.045	0.300 ± 0.015
	TCN	0.196 ± 0.016	0.457 ± 0.044	0.277 ± 0.025	0.286 ± 0.078	0.545 ± 0.036	0.445 ± 0.035	0.065 ± 0.009	0.240 ± 0.014	0.353 ± 0.015
	`Seegnificant` [12]	0.437 ± 0.004	0.581 ± 0.004	0.500 ± 0.015	0.539 ± 0.007	0.778 ± 0.007	0.654 ± 0.008	0.281 ± 0.009	0.251 ± 0.003	0.404 ± 0.015
	`Seegnificant` [12] (MP3)	0.431 ± 0.009	0.596 ± 0.009	0.594 ± 0.008	0.569 ± 0.010	0.795 ± 0.001	0.639 ± 0.018	0.274 ± 0.010	0.277 ± 0.007	0.448 ± 0.025
	`Poyo+` [1]	0.447 ± 0.011	0.597 ± 0.062	0.682 ± 0.022	0.611 ± 0.012	0.838 ± 0.010	0.707 ± 0.024	0.196 ± 0.013	0.188 ± 0.020	0.441 ± 0.017
	`Poyo+` [1] (MP3)	0.425 ± 0.011	0.614 ± 0.063	0.671 ± 0.033	0.592 ± 0.020	0.793 ± 0.034	0.668 ± 0.010	0.262 ± 0.011	0.259 ± 0.027	0.489 ± 0.000
SSL	`PopT` [5]	0.384 ± 0.007	0.478 ± 0.008	0.444 ± 0.012	0.510 ± 0.013	0.628 ± 0.012	0.572 ± 0.012	0.240 ± 0.010	0.414 ± 0.002	0.467 ± 0.004
	`PopT` [5] (MP3)	0.385 ± 0.005	0.519 ± 0.003	0.478 ± 0.005	0.518 ± 0.015	0.690 ± 0.007	0.608 ± 0.007	0.246 ± 0.012	**0.476** ± 0.008	0.487 ± 0.003
	`Charmander`	0.498 ± 0.001	0.629 ± 0.126	0.748 ± 0.024	0.620 ± 0.002	0.830 ± 0.009	0.695 ± 0.009	0.245 ± 0.017	0.266 ± 0.018	0.522 ± 0.009
	`Charmander` (MP3)	0.491 ± 0.006	0.644 ± 0.117	0.717 ± 0.052	0.597 ± 0.004	0.836 ± 0.009	**0.793** ± 0.006	0.245 ± 0.006	0.248 ± 0.002	0.550 ± 0.001
	`Charmander` (MP8)	**0.525** ± 0.002	**0.672** ± 0.128	**0.768** ± 0.028	**0.625** ± 0.023	**0.869** ± 0.001	0.715 ± 0.025	**0.317** ± 0.002	0.316 ± 0.005	**0.558** ± 0.010

framed the task as a multilabel classification problem with 5 activity labels: sleeping/resting, eating, talking, using a computer/phone, and watching TV.

To ensure that potential SSL-related gains in the decoding performance of `Charmander` (our model) are not trivial, we first ensured that the `Poyo+` model itself, using supervised learning alone, delivers high performance on our task. We hypothesized that its performance may be high because `Poyo+` has previously shown state-of-the-art (SOTA) results on various decoding tasks from electrophysiology and optophysiological recordings [1][2]. We thus used our task to evaluate `Poyo+`'s performance and to compare it to several baselines, including simple supervised models (MLP, TCN) and a prior SOTA supervised model (`Seegnificant` [12]) specifically for iEEG recordings. As an additional baseline, we have also considered a recent SOTA self-supervised model for iEEG (`PopT` [5]). We found that among these baselines, the best performance was delivered by supervised models, specifically by `Poyo+` and closely followed by `Seegnificant` (Table 1).

We then evaluated the impact of SSL on the downstream performance in the classification of behavioral activities. To this end, we compared our method, `Charmander`, which includes SSL pretraining, to `Poyo+`, its direct counterpart that only involves supervised training. To evaluate the impact of SSL across different training regimes, we considered the models trained on single-patient and multi-patient (MP3, 3 patients) data. We found that in both cases SSL has led to an improvement of the decoding accuracy (Table 1).

Finally, we extended our evaluation to a setting reflecting a distribution shift in the data: novel recording sessions held-out of pretraining. Starting from the models trained on held-in sessions (either from scratch or finetuned), we further finetuned on the new sessions and found consistent improvement in decoding accuracy due to our SSL framework. Then, we evaluated whether the learned representations were robust to the distribution shifts without any additional training (zero-shot) and found a similar pattern for two out of the three patients whose data we evaluated on. Overall, our results suggest that the SSL pretraining is beneficial even on out-of-distribution data.

Scaling numbers of patients and sessions improves generalization To understand how varying the data volume influences the ability of the model to generalize to new sessions, we considered different configurations of multi-participant pretraining where we scaled the number of participants and the amount of per-participant data. We sought to assess properties of the learned representations in all models using the same novel zero-shot evaluation on held-out sessions as was done when comparing to baselines. To this end, we normalized F1 scores by the minimum score per patient and took the average across patients. We found that improvements came with the increased data scale when sufficient data was used per-patient (Figure 2). Refer to Appendix A.1.3 for details on the pretraining scales. Note that the model size was fixed to 8M parameters (see Appendix B.2) for all data scales.

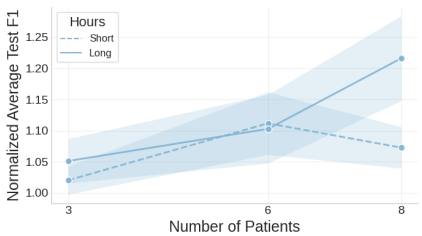

Figure 2: **Impact of the number of patients and sessions on decoding.** Here we used frozen weights from pretrained models at varying data scale, and each model was evaluated on shared test splits.

Our model sets a new SOTA performance on low-level acoustic feature classification tasks To contextualize the performance of `Charmander`, we tested it in the settings used in other works, namely on the Brain Treebank dataset [22]. We evaluated `Charmander` on the same task suite and splits as in [5] which included four auditory/linguistic feature decoding tasks: (1) pitch classification, (2) volume classification, (3) sentence onset detection, and (4) word onset detection (Appendix A.2). All tasks involved binary classification and performance was measured in terms of AUROC. The data was recorded from patients watching movies with simultaneously recorded iEEG (exclusively sEEG) signals. Our model outperforms all considered baselines on low-level acoustic feature classification tasks, namely pitch and volume (Table 2). For higher-level linguistic segmentation, `Charmander` achieves performance comparable to `PopT` on word onset detection and matches `Brant` on sentence onset detection. Overall, these results show that our framework generalizes across tasks, brain regions, and recording modalities.

Table 2: **Feature decoding from the Brain Treebank.** Results are shown for four different downstream tasks, spanning audiory and lingual feature decoding during movie-watching, and compared across SOTA methods. Baseline results (all BrainBERT-based models and `Brant`) are taken from [5]. `Poyo+` is trained single-session, and `Charmander` is pretrained on the same pretraining set as [5]. All reported numbers reflect AUROC on respective tasks.

Model	Pitch	Volume	Sentence On	Word On
BrainBERT:				
Linear [5]	0.59 $\pm$0.03	0.66 $\pm$0.03	0.70 $\pm$0.04	0.71 $\pm$0.04
MLP [5]	0.56 $\pm$0.03	0.64 $\pm$0.04	0.71 $\pm$0.03	0.70 $\pm$ 0.04
`PopT` [5]	<u>0.74</u> $\pm$0.03	<u>0.87</u> $\pm$0.03	**0.90** $\pm$0.01	**0.93** $\pm$0.02
End-to-end:				
`Brant` [26]	0.61 $\pm$0.03	0.74 $\pm$0.03	0.80 $\pm$0.04	0.80 $\pm$0.03
`Poyo+` [1]	0.71 $\pm$0.04	0.82 $\pm$0.04	0.67 $\pm$ 0.05	0.87 $\pm$0.02
`Charmander`	**0.88** $\pm$0.03	**0.93** $\pm$0.03	<u>0.81</u> $\pm$0.03	<u>0.91</u> $\pm$0.06

4 Discussion

In this work, we presented a novel SSL framework, called `Charmander`, for modeling multi-patient intracranial recordings of neural activity. By training our models on large-scale datasets of iEEG (ECoG and sEEG) recordings during naturalistic behavior, we have shown the effectiveness, scalability, and transferability of our approach across different conditions and downstream tasks. Our results highlight the potential of SSL to overcome the inherent challenges of working with iEEG data, including the variability in electrode placement and limited coverage within individuals.

We tested our models on the AJILE12 and Brain Treebank datasets featuring iEEG recordings of neural activity in the brain matched with various behavioral labels. Among publicly available iEEG datasets, these datasets offered the largest collection of multi-participant intracranial recordings of neural activity with long-term behavioral annotations, offering a clinically-relevant benchmark for large-scale models of neural activity. We leveraged a Perceiver-based encoder along with learnable channel embeddings that are paramount to the operation of our SSL masking objective. While previous works typically used strict priors on spatial embeddings, namely normalized electrode contact coordinates in a fixed space (e.g., MNI [11]), our approach encouraged the model to flexibly learn channel embeddings with the information it deems relevant.

Looking ahead, our work lays the foundation for large-scale generalist models for integrating and interpreting diverse neural data. Applying this framework to broader datasets, including recordings across different spatiotemporal scales may lead to insights into how neural circuits encode information and coordinate activity across individuals. By using SSL approaches, we can continue to bridge the gaps between data heterogeneity and model scalability, advancing our understanding of the brain.

Acknowledgements

We are grateful to Divyansha Lachi, Vinam Arora, and Zihao Chen for insightful discussions and feedback.

References

[1] M. Azabou, V. Arora, V. Ganesh, X. Mao, S. Nachimuthu, M. Mendelson, B. Richards, M. Perich, G. Lajoie, and E. Dyer. A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, 36, 2023.

[2] M. Azabou, K. X. Pan, V. Arora, I. J. Knight, E. L. Dyer, and B. A. Richards. Multi-session, multi-task neural decoding from distinct cell-types and brain regions. In *The Thirteenth International Conference on Learning Representations*.

[3] S. Baillet, J. C. Mosher, and R. M. Leahy. Combined MEG/EEG source imaging of visual evoked responses. *Human brain mapping*, 14(suppl 1):42–47, 2001.

[4] J. M. Bernabei, A. Li, A. Y. Revell, R. J. Smith, K. M. Gunnarsdottir, I. Z. Ong, K. A. Davis, N. Sinha, S. Sarma, and B. Litt. "hup ieeg epilepsy dataset", 2022.

[5] G. Chau, C. Wang, S. Talukder, V. Subramaniam, S. Soedarmadji, Y. Yue, B. Katz, and A. Barbu. Population transformer: Learning population-level representations of neural activity. *ArXiv*, pages arXiv–2406, 2024.

[6] K. Glomb, J. Cabral, A. Cattani, A. Mazzoni, A. Raj, and B. Franceschiello. Computational models in electroencephalography. *Brain Topography*, 35(1):142–161, 2022.

[7] R. Grech, T. Cassar, J. Muscat, K. P. Camilleri, S. Fabri, M. E. Zervakis, P. Xanthopoulos, V. Sakkalis, and B. Vanrumste. Review of advanced techniques in EEG source localization. *IEEE Engineering in Medicine and Biology Magazine*, 27(3):38–51, 2008.

[8] J. F. Hipp, D. J. Hawellek, M. Corbetta, M. Siegel, and A. K. Engel. Large-scale cortical correlation structure of spontaneous oscillatory activity. *Nature neuroscience*, 15(7):1049–1063, 2012.

[9] C. Holdgraf, S. Appelhoff, S. Bickel, K. Bouchard, S. D'ambrosio, O. David, O. Devinsky, B. Dichter, A. Flinker, B. L. Foster, et al. ieeg-bids, extending the brain imaging data structure specification to human intracranial electrophysiology. *Scientific data*, 6(1):102, 2019.

[10] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[11] J. Mazziotta, A. Toga, A. Evans, P. Fox, J. Lancaster, K. Zilles, R. Woods, T. Paus, G. Simpson, B. Pike, et al. A probabilistic atlas and reference system for the human brain: International Consortium for Brain Mapping (ICBM). *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 356(1412):1293–1322, 2001.

[12] G. Mentzelopoulos, E. Chatzipantazis, A. G. Ramayya, M. J. Hedlund, V. P. Buch, K. Daniilidis, K. P. Kording, and F. Vitale. Neural decoding from stereotactic eeg: accounting for electrode variability across subjects. *arXiv preprint arXiv:2411.10458*, 2024.

[13] A. Mirzababaie, I. Solomon, P. Dugan, V. Kremen, M. Stead, f. adolescence, f. Epilepsy, i. adults, d. team, d. The, and d. network. Dynamical network models from eeg and meg for epilepsy surgery—a quantitative approach. *Frontiers in Neurology*, 13, 2022.

[14] S. Mitchell, M. OSullivan, and I. Dunning. Pulp: a linear programming toolkit for python. *The University of Auckland, Auckland, New Zealand*, 65:25, 2011.

[15] P. L. Nunez and R. Srinivasan. *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford University Press, USA, 2006.

[16] J. Parvizi and S. Kastner. Human intracranial eeg: promises and limitations. *Nature neuroscience*, 21(4):474, 2018.

[17] S. M. Peterson, R. P. Rao, and B. W. Brunton. Learning neural decoders without labels using multiple data streams. *Journal of Neural Engineering*, 19(4):046032, 2022.

[18] S. M. Peterson, S. H. Singh, B. Dichter, M. Scheid, R. P. Rao, and B. W. Brunton. Ajile12: Long-term naturalistic human intracranial neural recordings and pose. *Scientific data*, 9(1):184, 2022.

[19] V. R. Rao, M. K. Leonard, J. K. Kleen, B. A. Lucas, E. A. Mirro, and E. F. Chang. Chronic ambulatory electrocorticography from human speech cortex. *NeuroImage*, 153:273–282, 2017.

[20] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.

[21] C. Wang, V. Subramaniam, A. U. Yaari, G. Kreiman, B. Katz, I. Cases, and A. Barbu. Brainbert: Self-supervised representation learning for intracranial recordings. *arXiv preprint arXiv:2302.14367*, 2023.

[22] C. Wang, A. Yaari, A. Singh, V. Subramaniam, D. Rosenfarb, J. DeWitt, P. Misra, J. Madsen, S. Stone, G. Kreiman, et al. Brain treebank: Large-scale intracranial recordings from naturalistic language stimuli. *Advances in Neural Information Processing Systems*, 37:96505–96540, 2024.

[23] N. X. Wang, J. D. Olson, J. G. Ojemann, R. P. Rao, and B. W. Brunton. Unsupervised decoding of long-term, naturalistic human neural recordings with automated video and audio annotations. *Frontiers in human neuroscience*, 10:165, 2016.

[24] J. Ye and C. Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.

[25] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

[26] D. Zhang, Z. Yuan, Y. Yang, J. Chen, J. Wang, and Y. Li. Brant: Foundation model for intracranial neural signal. *Advances in Neural Information Processing Systems*, 36:26304–26321, 2023.

Appendix

A Dataset details

A.1 AJILE12 dataset

The *Annotated Joints in Long-term Electrocorticography for 12 human participants (AJILE12)* [18] dataset is a large-scale collection of naturalistic iEEG recordings from 12 human patients. The patients were recorded passively during clinical epilepsy monitoring at Harborview Medical Center in Seattle, WA, USA. Each patient was implanted with a series of electrode groups, with modality and distribution determined by their surgeons based on clinical need. Electrode types and placement varied significantly between patients, with broad coverage of cortical areas in several gyri (Precentral, Postcentral, Middle Temporal, and Inferior Temporal Gyrus) via ECoG grids / strips, and subcortical areas via depth electrodes (see Table A1 for statistics from each patient, and Figure A1 for a plot of electrode placement for the AJILE12 patients considered in this work).

Table A1: **AJILE12 participant statistics.** Basic information for each of the 12 patients from the AJILE12 dataset, including: # of surface and depth electrodes; % of timepoints that are missing across all sessions, average duration of activity epochs ($\pm$ SEM across epochs, see Appendix A.1.2), % of epochs that are labeled with multiple activities (out of all non-Blocklist epochs, as well as non-Sleep/rest epochs); # of sessions per participant, and whether the participant's data was used in pretraining.

Participant	# Surface	# Depth	% NA	Avg Epoch (m)	% Multilabel	% Multilabel (/ non-sleep)	# Sessions	Pretraining
1	86	8	0.19	2.01 ± 0.17	5.43	8.87	4	
2	70	16	0.26	1.99 ± 0.08	2.87	8.60	4	✓
3	80	16	1.62	2.02 ± 0.23	15.59	20.34	4	✓
4	84	0	0.62	1.87 ± 0.65	10.74	19.25	5	✓
5	106	0	1.31	2.00 ± 0.17	6.79	15.68	3	
6	80	0	1.28	2.01 ± 0.19	7.41	44.25	5	✓
7	64	0	0.67	1.92 ± 0.52	2.51	18.26	5	✓
8	92	0	0.64	2.10 ± 0.47	4.58	19.38	5	✓
9	98	28	1.31	2.01 ± 0.24	1.00	7.52	5	
10	86	40	1.93	2.00 ± 0.17	1.80	10.50	5	✓
11	106	0	0.78	1.99 ± 0.12	0.15	0.73	5	✓
12	92	32	2.13	2.00 ± 0.13	0.38	4.55	5	

A.1.1 Preprocessing

We utilized the publicly available release of the AJILE12 dataset with minimal preprocessing beyond what was done already in the public release. In particular, the raw voltage traces were sampled at 1kHz, and preprocessed before public release in a pipeline consisting of: downsampling to 500Hz, band pass filtering (0-200Hz, encompassing theta to high-gamma bands), line noise (60Hz) removal via notch filtering, and re-referencing to Common Median Reference (CMR). More details on the data collection and preprocessing pipeline from the public release can be found in [18]. To further prepare the data for a deep learning workflow, we performed the following steps in our preprocessing pipeline for numerical stability:

1. z-score normalization over whole sessions.
2. Clipped amplitude values to within 10σ to remove obscene outliers.
3. A small proportion of the timepoints were missing iEEG data across most-to-all channels. These sometimes intersected with "Blocklist" segments, where either recordings were paused for private time, restroom breaks, or unrelated research experiments; or when activity detection (see Appendix A.1.2) was likely inaccurate. Such segments were filtered out during preprocessing, but some missing timepoints still remained scattered throughout the 24hrs. Due to the sparsity of these segments (see Table A1), we opted to interpolate voltage traces between the time points right before and right after as a form of imputation.

A.1.2 Activity classification task

Along with the iEEG recordings, the AJILE12 dataset contains annotations for activities of daily living (ADL) during epochs of time. The epochs are roughly 2m in length (breakdown by patient in

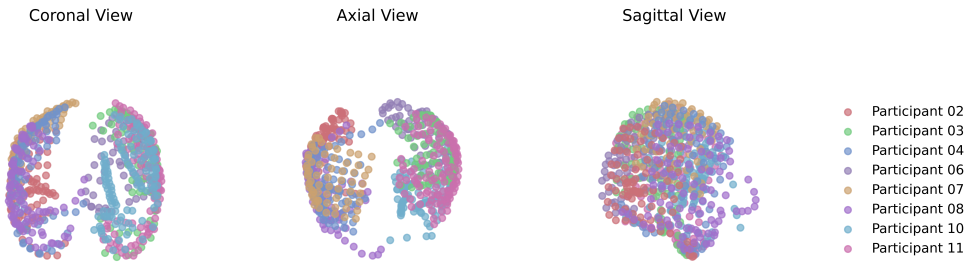

Figure A1: **AJILE12 electrode configurations are diverse and span a wide area of the brain.** For each of the 8 participants used for pretraining (e.g. MP8-short and MP8-long), we plot the location of each electrode in MNI space, along three distinct anatomical planes (Coronal, Axial, Sagittal). We can see the electrode configurations vary significantly across participants, and in aggregation spans wide coverage of the cortical surface, as well as some subcortical areas.

Table x) and consist of the following activities: Sleep/rest, Inactive, Eat, Talk, TV, Computer/phone, Other activity (Blocklist and unlabeled epochs were omitted when generating evaluation splits). Epochs may be labeled with one or more activities yielding a multilabel classification problem. Importantly, the "Inactive" and "Other activity" labels serve as broad negative categories, in the sense that they indicate absence of some type of meaningful behavior, and hence are mutually exclusive from all other labels. Including them in the multilabel classification task would result in unbalanced and semantically inconsistent label distributions. Further, our interest was more in identifying specific and meaningful activities in the positive, as is beneficial for behaviorally grounded decoding and downstream applications such as assistive technologies or patient monitoring. Hence, we finalized on the 5-activity (Sleep/rest, Eat, Talk, TV, Computer/phone) multilabel classification problem. In particular, the task is to determine the activities performed by the patient given a sampled 1s chunk of iEEG recording during those activities. The models output a multi-hot encoding representing the activity predictions, and (treating each activity as a binary classification problem) they are compared with the ground truth using Binary Cross Entropy (BCE) during training, and we report macro-averaged F1 score during validation and testing.

A.1.3 Pretraining splits

For pretraining, we randomly sampled 100s non-overlapping segments from the full 24 hours to generate the train / validation / test splits. In pretraining, as in the activity classification task, we sample 1s chunks (from the 100s segments) when forming training batches. See Table A2 for a patient-wise breakdown of the number of pretraining hours of iEEG recordings utilized for the various models.

Table A2: **Pretraining splits from AJILE12.** Durations of pretraining splits for short and long pretrained models. For a breakdown of sessions used for pretraining (held-in), see Figure A2.

		short **pretraining split durations (m)**				long **pretraining split durations (m)**			
Participant	**# Sessions**	**Avg Train**	**Total Train**	**Total Val**	**Total Test**	**Avg Train**	**Total Train**	**Total Val**	**Total Test**
2	3	115.6	346.7	90.0	110.0	360.0	1080.0	155.0	305.0
3	3	115.6	346.7	88.3	116.7	345.0	1035.0	51.7	131.7
4	3	151.7	455.0	115.0	141.7	360.0	1080.0	155.0	305.0
6	4	28.8	115.0	28.3	30.0	360.0	1440.0	206.7	406.7
7	5	61.7	308.3	100.0	88.3	360.0	1800.0	258.3	508.3
8	5	108.3	541.7	145.0	185.0	360.0	1800.0	258.3	508.3
10	5	106.3	531.7	141.7	178.3	360.0	1800.0	258.3	508.3
11	5	65.3	326.7	86.7	101.7	360.0	1800.0	258.3	508.3

A.1.4 Finetuning splits

The label distributions vary widely across activity labels, as well as across patients (see Figure A3). To train decoders that are not underrepresented for any activity, we sampled balanced splits for finetuning evaluation from the full 24 hours. First, we chose patients 2, 3, and 4 for evaluation, since

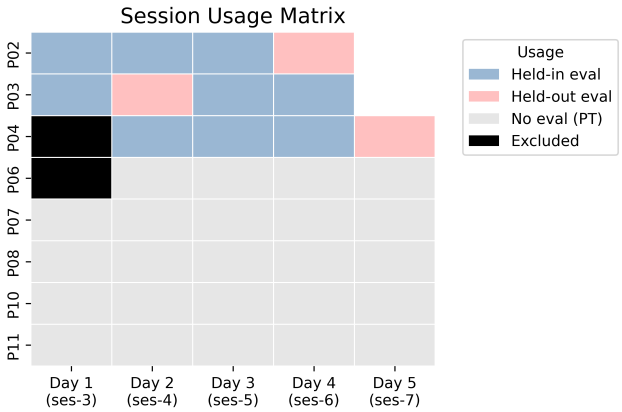

Figure A2: **Evaluation settings for each session used from AJILE12.** Among the 8 participants whose data were used from the AJILE12 dataset, data from the 3 with most balanced label distributions (with respect to entropy over the distributions) were used for evaluation. Each were designated a single session held-out from pretraining. Models were evaluated within-distribution (on Held-in eval sessions, i.e. "same session" in Table 1) and out-of-distribution (on Held-out eval sessions, i.e. "novel" in Table 1). The remaining sessions were only used for pretraining.

they exhibited the most balanced label distributions (with respect to entropy). We then sampled epochs per a balanced multilabel undersampling procedure (detailed below) to generate roughly 1hr of training data per session. A session matrix of the patient data used for different modes of evaluation can be found in Figure A2.

Sampling a balanced distribution from a multilabel epoch set is challenging, since choosing any particular epoch may contribute to the representation of multiple activities from the set. To address this challenge, we established a set of constraints that would guarantee balance, up to availability of labels. For a given session, there is a set $\mathcal{E}$ of *target labels*, which may include multiple *activities* (from activity set $\mathcal{A}$) at a time. Let e_i be the number of epochs that are labeled $\mathcal{E}_i$, and let a_j be the number of epochs for which the label contains the activity $\mathcal{A}_j$. The goal is to find a solution vector $c \in \mathbb{N}^{|\mathcal{E}|}$ which contains the number of epochs to sample per target label, such that:

(I) It respects the upper bound constraint that each target label $\mathcal{E}_i$ appears at most e_i times, i.e.

$$c_i \leq e_i \qquad (\forall i \in \mathcal{E})$$

(II) For a given $\mathcal{C} \in \mathbb{N}$, each activity $\mathcal{A}_j$ appears exactly $\mathcal{C}$ times, unless $a_j < \mathcal{C}$, i.e.

$$\sum_{i:\mathcal{A}_j \subset \mathcal{E}_i} c_i = \min(\mathcal{C}, a_j) \qquad (\forall j \in \mathcal{A})$$

This constraint attempts to guarantee balance across the activities, though it must respect the availability of activities in the session.[1]

In addition, it can be tempting for a solver to resolve these two constraints alone by favoring single-activity labels over mixed-activity, e.g. to resolve "Eat", first sample all "Eat" labels without touching any "Eat, TV" or "Eat, Talk". To encourage the solver to sample evenly from the target labels (in addition to the hard constraint to balance activities established by (II)), we further impose an objective on the solution set c:

(III) Try to minimize the absolute deviation from the mean $\bar{c}$ (recall that this is over the target label set $\mathcal{E}$, so an optimal solution will have all target label counts in c as close to equal as possible), i.e.

$$\min_{c \in \mathbb{N}^{|\mathcal{E}|}} \sum_{i \in [|\mathcal{E}|]} |c_i - \bar{c}|.$$

[1]Note: $\mathcal{A}_j \subset \mathcal{E}_i$ means that the activity $\mathcal{A}_j$ is one of the activities found in the target label $\mathcal{E}_i$.

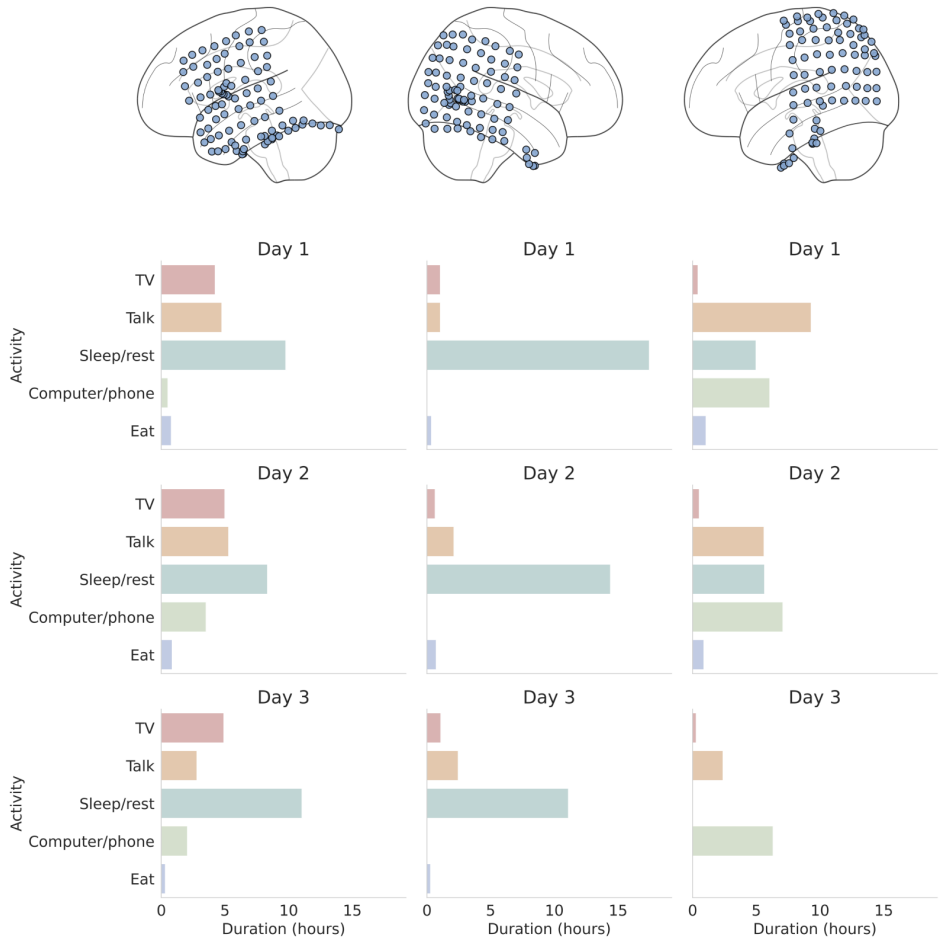

Figure A3: **Behavior variability across patients in AJILE12.** Each plot depicts the activity distribution of the 5 activities used for multilabel classification, across 3 separate days each from 3 different participants (P1, P2, and P3 left-to-right). Each bar represents the sum of durations of any epoch *including* that label.

Combining (I), (II), and (III), we have a linear program that will provide an optimal solution c to the balanced undersampling problem. We iterate $\mathcal{C}$ until we get a solution whose total duration over sampled epochs is above the 1hr threshold. The linear program was solved using the default CBC solver in the `pulp` library [14].

After resolving a number of samples per target label, we randomly select epochs under each label into train / val / test sets. Note that for some sessions, some labels are so scarce that they cannot be represented in all 3 sets. In this case, there is an order of precedence as to which split will receive the label: train $\rightarrow$ test $\rightarrow$ validation. These splits are fixed and shared across all models, either for finetuning from pretrained models (in the case of `Ours` and `PopT`) or training from scratch. Note also that the same procedure is used regardless of evaluation setting, though the choice of sessions varies (see Figure A2).

A.2 Brain Treebank Dataset

Brain Treebank [22] is a dataset consisting of iEEG recordings from patients watching movies. The movies are aligned and annotated with visual and language features, which can be decoded from the iEEG signals. The iEEG recordings are sampled at 2048Hz and Laplacian referenced. We use the same pretraining / finetuning splits and downstream decoding tasks as in [5] and [21]. These tasks are auditory and linguistic binary decoding tasks. More details on the dataset can be found in

[22], and more details on the tasks and splits can be found in [5]. In addition to the preprocessing steps replicated from the previous works, since our architecture models raw voltage signals instead of spectral features transformed a priori, we further normalized the iEEG signals over the entire session and clipped values to within 10σ (as we did with AJILE12 as well) for numerical stability.

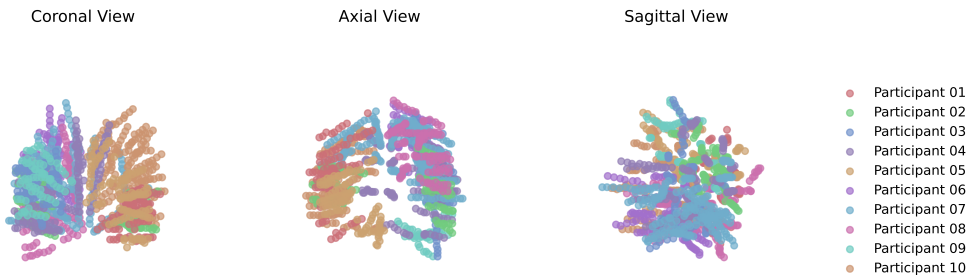

Figure A4: **Brain Treebank electrode configurations span subcortical brain areas.** For every participant (10) in the Brain Treebank dataset, we plot the location of each electrode from its position in LPI space ($L \to x$, $P \to y$, $I \to z$), along three distinct anatomical planes (Coronal, Axial, Sagittal). We can see the electrode configurations vary significantly across participants, and in aggregation spans especially wide coverage of subcortical areas.

B Model Details and Experimental Setup

B.1 Model hyperparameters

Throughout all of our experiments, we use a context window of 1s and do not segment data into trials during training. We train the model with H latent tokens and a dimension d.

Our models are trained with the hyperparameters from Table A3, following hyperparameter tuning via a random search:

Table A3: **Hyperparameters used for `Charmander` and `Poyo+` models.**

Hyperparameter	Value
Patch size (P)	5
Embedding dimension (d)	128
Number of latents (H)	32
Number of virtual timesteps (M)	8
Transformer depth (L)	16
Attention heads (h)	8
FFN dropout (p_{FFN})	0.20
Linear dropout (p_{Lin})	0.40
Attention dropout (p_{Attn})	0.20

B.2 Comparison of model sizes

In Table A4 we compare the size of the Transformer-based models (in terms of number of parameters) used throughout the text. Note that in scaling experiments, we attempted to scale up the model size of `Charmander` with the following configurations:

1. **base**: base hyperparameters as defined in A3 (8M params).
2. **large**: same hyperparameters as in **base** but with $d = 256$, $L = 24$ (33M params).
3. **venti**: same hyperparameters as in **base** but with $d = 512$, $L = 32$ (142M params).

However, we found no benefit to increasing model scale in terms of downstream performance. We did observe improved reconstruction quality during SSL with larger models, especially with also scaling the nunber of latents H.

Table A4: **Number of parameters in Transformer-based models.** We note that `Charmander` was able to achieve top performance with the least number of trainable parameters.

Model	# of parameters
`Seegnificant`	115M
`PopT`	
`Poyo+`	8M
`Charmander`	8M

B.3 Details of masking strategy

During self-supervised pretraining, we employ a random channel masking strategy in which exactly 50% of the input channels are randomly selected and masked for each training example. These masked channels are entirely removed from the input sequence but retained as reconstruction targets. The masking is applied without regard to spatial adjacency or temporal dynamics, forcing the model to learn robust statistical and contextual dependencies across all channels.

By reconstructing the masked signals solely from the unmasked ones, the model is encouraged to leverage inter-channel correlations and build a richer understanding of population dynamics. This strategy mimics partial observation scenarios and improves the model's generalization to real-world neural decoding tasks, where some channels may be missing or noisy.

B.4 Rotary position embeddings (RoPE)

Position of tokens in our model are determined by their timestamps. Thus, we allow the positions to be continuous values, which is different from the conventional application of RoPE in domains such as language and vision. For this, we define a custom rotation matrix $\mathbf{R}(t)$ as follows:

$$\mathbf{R}(t) = \begin{bmatrix} \mathbf{R}_{2\times2}(t, T_1) & 0 & \cdots & 0 \\ 0 & \mathbf{R}_{2\times2}(t, T_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{R}_{2\times2}(t, T_{d/2}) \end{bmatrix}$$

With each 2×2 sub-matrix being defined as:

$$\mathbf{R}_{2\times2}(t, T) = \begin{bmatrix} \cos(2\pi t/T) & -\sin(2\pi t/T) \\ \sin(2\pi t/T) & \cos(2\pi t/T) \end{bmatrix}$$

Here, $T_1, T_2, \ldots, T_{d/2}$ denote the "time-period" of different sinusoids. These decide what resolution and range the model is capable of resolving. We set these time-periods to be logarithmically spaced from $0.1ms$ to $20s$.

B.5 Training details

POYO supervised training We have a supervised setup of training POYO. We train our model using a multi-label loss, which is appropriate for our iEEG dataset where we aim to decode multiple behavioral activities simultaneously. For each training step, the model computes a prediction, which is then compared against binary ground truth labels using a Binary Cross-Entropy with Logits Loss for optimization. The training utilizes the Lamb optimizer [25] with weight decay of 1×10^{-4}. The initial learning rate is set to 3.125×10^{-3}, held constant during the first 150 epochs and decayed over the last 150 epochs using a cosine decay schedule. To select the best model, we apply an early drop strategy: during training, we monitor validation performance at each checkpoint and retain the model with the highest validation F1 score. This selected checkpoint is then used for final evaluation on the test set, and the corresponding test F1 score is reported.

POYO self-supervised pretraining We pretrain POYO using the masking strategy described above (Appendix B.3), where the reconstruction objective minimizes the Mean Squared Error (MSE)

between the predicted and ground-truth iEEG signals of the masked channels. Unless otherwise noted, all pretraining experiments are conducted for 300 epochs. Due to compute and time constraints, the MP8-long variant was only trained for 45 epochs. Training uses the Lamb optimizer with weight decay of 1×10^{-4} and an initial learning rate of 3.125×10^{-4}, held constant for the first 150 epochs and then decayed using a cosine schedule over the remaining 150 epochs.

POYO finetuning Our fine-tuning strategy is designed to efficiently adapt pretrained POYO models to new sessions or tasks. Initially, the latent space, cross-attention, and self-attention layers of the pretrained encoder are frozen. We replace the pretraining decoder with a randomly initialized, task-specific decoder (e.g., a multi-label classifier). To accommodate novel subjects and sensor layouts, we initialize new subject and channel embeddings. For the first 50 epochs, only the decoder and the new embeddings are updated. This controlled warm-up avoids disrupting the pretrained representations. Subsequently, we unfreeze the last four self-attention layers and continue training up to 300 epochs. Optimization uses the LAMB optimizer with weight decay of 1×10^{-4}, with an initial learning rate of 3.125×10^{-3}. The learning rate is constant in the first 150 epochs and then is decayed via cosine schedule in another 150 epochs. To select the best model, we adopt an early drop strategy: validation performance is monitored at each checkpoint, and the model achieving the highest validation F1 score is retained. This checkpoint is then used for final evaluation on the test set, and the corresponding test F1 score is reported.

PopT We pretrained `PopT` on 1s windows from the same pretraining splits as used in `Ours`. We first train a temporal encoder `BrainBERT` [21] using all default parameters besides $nperseg = 100$, $noverlap = 87$, and $d_h = 252$. Taking the best `BrainBERT` model (lowest validation loss) for each subject or MP group, we freeze the model and pass individual channel time-series through it to encode the time-series into a $d = 252$ vector. Then, we pre-train `PopT` on top of these encodings using the same pretraining parameters as specified in the original work [5]. When finetuning on a decoding task, we append a linear layer to the `[CLS]` token to project the representation to the classification output dimension. We use the same scheduler as in the original papers for fine-tuning, but train with $batchsize = 128$ for $epochs = 25$, where the number of steps per epoch varies by the number of samples in the full dataset.

Seegnificant The Seegnificant baseline model first converts each electrode's 50 Hz time series into a 128-dimensional vector using a 1×50 convolution with batch normalization, average pooling and 0.3 dropout. It then applies two transformer encoder layers over time and two over space; each layer comprises a 4-head self-attention block (with layer normalization and 0.2 dropout) followed by a feed-forward network (hidden dimension 512, GELU activation, 0.2 dropout). Between the time and space blocks, an MNI positional encoding adds each electrode's (x, y, z) coordinates via Gaussian kernels and a linear projection. Finally, the features are flattened and passed through a two-layer MLP (128 units, ReLU activation, 0.2 dropout) and routed to ten subject-specific linear heads to produce the class logits.

Seegnificant is trained for 600 epochs using random seeds 41, 42 and 43. We use the Lamb optimizer with learning rate 3.125×10^{-4} and weight decay 1×10^{-4}. A OneCycleLR scheduler with a 50% warm-up period and cosine annealing adjusts the learning rate. For each seed, we select the checkpoint with the highest validation F1 and report the average test F1 across all seeds.

C Expanded Results

C.1 AJILE12 Activity Classification

Table A5: **Per-session AJILE12 activity decoding.** F1-scores from 5-way multilabel classification task on same session (held-in sessions) setting. This table shows a session-by-session breakdown of Table 1 in the main text. Reported errors are SEM.

	Model	Same session								
		P2 S3	P2 S4	P2 S5	P3 S3	P3 S5	P3 S6	P4 S4	P4 S5	P4 S6
Supervised	MLP	0.184 ± 0.007	0.218 ± 0.006	0.164 ± 0.005	0.280 ± 0.035	0.271 ± 0.040	0.352 ± 0.035	0.257 ± 0.007	0.235 ± 0.009	0.288 ± 0.010
	TCN	0.213 ± 0.022	0.166 ± 0.032	0.209 ± 0.029	0.522 ± 0.042	0.290 ± 0.011	0.560 ± 0.014	0.212 ± 0.010	0.252 ± 0.022	0.366 ± 0.018
	`Seegnificant`	0.463 ± 0.005	0.395 ± 0.006	0.453 ± 0.015	0.673 ± 0.006	0.360 ± 0.002	0.710 ± 0.006	0.431 ± 0.030	0.516 ± 0.014	0.553 ± 0.006
	`Seegnificant` (MP3)	0.461 ± 0.010	0.379 ± 0.010	0.453 ± 0.012	0.665 ± 0.019	0.373 ± 0.003	0.749 ± 0.007	0.507 ± 0.018	0.588 ± 0.001	0.688 ± 0.005
	`Poyo+`	0.449 ± 0.016	0.458 ± 0.014	0.434 ± 0.028	0.735 ± 0.013	0.357 ± 0.007	0.699 ± 0.044	0.615 ± 0.042	0.702 ± 0.008	0.729 ± 0.018
	`Poyo+` (MP3)	0.439 ± 0.020	0.412 ± 0.029	0.423 ± 0.008	0.737 ± 0.029	0.366 ± 0.020	0.740 ± 0.002	0.587 ± 0.067	0.679 ± 0.039	0.747 ± 0.028
SSL	`PopT`	0.385 ± 0.005	0.339 ± 0.009	0.429 ± 0.011	0.573 ± 0.009	0.285 ± 0.006	0.577 ± 0.009	0.293 ± 0.004	0.495 ± 0.013	0.543 ± 0.019
	`PopT` (MP3)	0.403 ± 0.001	0.329 ± 0.008	0.424 ± 0.012	0.618 ± 0.007	0.301 ± 0.001	0.639 ± 0.007	0.305 ± 0.007	0.529 ± 0.005	0.602 ± 0.010
	`Charmander`	0.499 ± 0.006	0.499 ± 0.023	0.496 ± 0.005	0.743 ± 0.009	0.377 ± 0.003	0.766 ± 0.002	0.734 ± 0.008	0.717 ± 0.011	0.795 ± 0.014
	`Charmander` (MP3)	0.503 ± 0.016	0.481 ± 0.027	0.489 ± 0.007	0.767 ± 0.010	0.410 ± 0.004	0.757 ± 0.023	0.620 ± 0.010	0.733 ± 0.020	0.797 ± 0.009
	`Charmander` (MP8)	0.524 ± 0.004	0.529 ± 0.015	0.522 ± 0.005	0.794 ± 0.021	0.416 ± 0.010	0.807 ± 0.004	0.730 ± 0.024	0.753 ± 0.008	0.822 ± 0.008

Table A6: **Per-session AJILE12 activity decoding.** F1-scores from 5-way multilabel classification task on novel finetuned (held-out sessions) setting. Reported errors are SEM.

	Model	Novel finetuned		
		P2 S6	P3 S4	P4 S7
Supervised	MLP	0.251 ± 0.007	0.256 ± 0.047	0.300 ± 0.017
	TCN	0.286 ± 0.078	0.545 ± 0.036	0.445 ± 0.035
	`Seegnificant`	0.281 ± 0.009	0.251 ± 0.003	0.404 ± 0.012
	`Seegnificant` (MP3)	0.274 ± 0.010	0.277 ± 0.007	0.448 ± 0.025
	`Poyo+`	0.612 ± 0.012	0.838 ± 0.010	0.707 ± 0.024
	`Poyo+` (MP3)	0.592 ± 0.021	0.793 ± 0.034	0.668 ± 0.011
SSL	`PopT`	0.510 ± 0.013	0.628 ± 0.012	0.572 ± 0.012
	`PopT` (MP3)	0.518 ± 0.015	0.690 ± 0.007	0.608 ± 0.007
	`Charmander`	0.620 ± 0.002	0.830 ± 0.009	0.695 ± 0.009
	`Charmander` (MP3)	0.597 ± 0.004	0.836 ± 0.009	0.793 ± 0.006
	`Charmander` (MP8)	0.625 ± 0.023	0.869 ± 0.001	0.715 ± 0.020

C.2 Scaling

Table A7: **Per-session AJILE12 activity decoding.** F1-scores from 5-way multilabel classification task on novel zero-shot (held-in sessions) setting across varying data scale. This table reflects the numbers used to generate Figure 2 in the main text. Reported errors are SEM.

	Model	Novel session zero-shot		
		P2	P3	P4
SSL	MP3-short	0.245 ± .011	0.248 ± .003	0.550 ± .002
	MP6-short	0.278 ± .011	0.293 ± .016	0.527 ± .016
	MP8-short	0.263 ± .018	0.277 ± .019	0.532 ± .015
	MP3-long	0.241 ± .021	0.278 ± .013	0.544 ± .014
	MP6-long	0.296 ± .012	0.257 ± .011	0.550 ± .012
	MP8-long	0.317 ± .003	0.316 ± .009	0.558 ± .008