Population Transformer: Learning Population-level Representations of Intracranial Activity

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present a self-supervised framework that learns population-level codes for intracranial neural recordings at scale, unlocking the benefits of representation learning for a key neuroscience recording modality. The Population Transformer (PopT) lowers the amount of data required for decoding experiments, while increasing accuracy, even on never-before-seen subjects and tasks. We address two key challenges in developing PopT: sparse electrode distribution and varying electrode location across patients. PopT stacks on top of pretrained representations and enhances downstream tasks by enabling learned aggregation of multiple spatially-sparse data channels. Beyond decoding, we interpret the pretrained PopT and fine-tuned models to show how it can be used to provide neuroscience insights learned from massive amounts of data. We release a pretrained PopT to enable off-the-shelf improvements in multi-channel intracranial data decoding and interpretability.

1 Introduction

Building effective representations of neural recordings is an important tool in enabling neuroscience research. We are particularly interested in modeling intracranial recordings, which rely on probes placed within the brain to provide high temporal resolution recordings of local neural activity [1, 2]. Because of its dispersed placement within the brain volume, intracranial recordings suffer from data sparsity. Moreover, there is often significant variability in probe placement across subjects [1, 2], leading to high variability in input channel meaning. Historically, constructing decoders from intracranial data has relied on supervised learning [3, 2, 4–6], but this requires experimenters to collect annotated data, which is scarce due to patient availability and labor-intensive labeling.

To improve decoding data-efficiency, self-supervised pretraining on unannotated data can be employed to first learn generic representations of the recordings. This means that the model does not have to use valuable annotated samples to learn how to do feature extraction before it can do classification, improving the reach of neuroscientific research.

In this paper, we are interested in developing generic representations of multi-channel intracranial recordings that enable efficient adaptation to a wide range of downstream decoding tasks. Prior work has shown how to pretrain subject-specific [7] or channel-specific [8] models of intracranial data, but such techniques ignore inter-channel relationships or commonalities that might exist across subjects. The most general approach would be to pretrain using data from multiple datasets, but would require tackling the aforementioned challenges of sparse electrode coverage and variable electrode placement between subjects.

We propose Population Transformer (PopT), a self-supervised pretraining approach on transformers [9] that learns subject-generic representations of arbitrary electrode ensembles (Figure 1). During

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

pretraining, we simultaneously optimize both a channel-level and ensemble-level objective, that requires the model to (1) build representations of channels in the context of other channels and (2) meaningfully distinguish temporal relationships between different ensembles of channels.

Our approach builds on top of pretrained single-channel embeddings, which has two key advantages. First, by separating the single-channel embedding and multi-channel-aggregation into different modules, we make our approach agnostic to the specific type of temporal embedding used, leaving room for future independent improvements, an approach that has been validated in video modeling [10]. Second, by taking advantage of pretrained channel embeddings, we make our population-level training lightweight, allowing for adoption in lower compute resource environments.

Empirically, we find that our pretrained PopT outperforms existing aggregation approaches, highlighting the usefulness of learning spatial relationships during pretraining. Moreover, we find that these benefits hold even for subjects not seen during pretraining, lending to its usefulenss for new subject decoding. We also find that the pretrained PopT weights themselves reveal interpretable patterns for neuroscientific study.

Our main contributions are:

1. a lightweight, generic SSL framework, Population Transformer (PopT) that learns arbitrary joint representations of channel embeddings across unannotated datasets of neural activity.
2. a demonstration that a pretrained PopT benefits downstream performance, interpretability, and training efficiency in comparison to baseline aggregation approaches.
3. a pretrained off-the-shelf model that computes population-level representations of intracranial neural recordings.

2 Population Transformer Approach

We propose a self-supervised training scheme to learn a subject-generic model that handles arbitrary electrode configurations. Our loss function has two discriminative components: (1) **channel-wise** — the model identifies outlier channels swapped with activity from a different timepoint, requiring sensitivity to surrounding channel context; (2) **ensemble-wise** — the model determines if two channel ensembles occurred consecutively, requiring ensemble-level context awareness. This objective effectively simulates many in-silico brain ablations, training the model to learn connections between regions in the presence of these ablations.

A key aspect of our method is the fact that our objective is **discriminative**, rather than reconstructive, as is often the case in self-supervision [11, 8]. We found this to be necessary, because in practice, the temporal embeddings often have low effective dimension (see [8]), and reconstruction rewards the model for overfitting to "filler" dimensions in the feature vector (see Appendix G).

To make our model subject and configuration agnostic, we provide the 3D position of each electrode, providing a common position embedding across subjects. We also vary subset sizes during sampling to ensure the model handles ensembles of different sizes, accommodating neuroscience experiments with varying electrode counts and analysis levels. Additionally, we select disjoint subsets to prevent the model from solving tasks through trivial copying.

Architecture A schematic of our Population Transformer (PopT) approach is shown in Figure 1. Consider a given subject with N_c channels indexed by $C = \{1, ..., N_c\}$. Activity from channel i at time t can be denoted by x_i^t. The PopT takes as input an interval of brain activity $X = \{x_i^t | i \in C\}$ from a given time t and a special `[CLS]` token. Per channel, each interval of brain activity is passed through a temporal embedding model T, in our case BrainBERT, to obtain a representation of each channel's temporal context.

Before being inputted to the PopT, each channel's 3D position is added to this embedding, so the final input is $X_B = \{T(x) + pos(i) + \mathcal{N}(0, \sigma) | x \in X\}$. Here, we add Gaussian fuzzing to prevent overfitting to a particular set of coordinates. Spatial location is given by the electrode's Left, Posterior, and Inferior coordinates [12]; see [8] for details on how these were obtained. Each coordinate is encoded using sinusoidal position encoding [9]. And the three encodings are concatenated together to form the position embedding $pos(i) = [e_{\text{left}}; e_{\text{post.}}; e_{\text{inf}}]$.

The core of PopT consists of a transformer encoder stack (see Appendix C: Architectures). The output of the PopT are spatial-contextual embeddings of the channels $Y = \{y_i\}$ as well as an embedding of

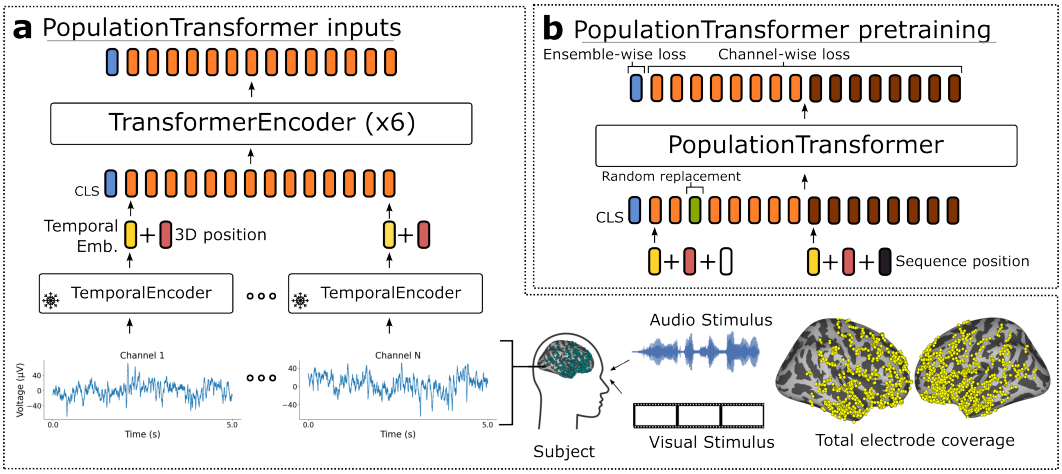

Figure 1: **Schematic of our approach**. The inputs to our model (a) are the combined neural activities from a collection of intracranial electrodes in a given time interval. These are passed to a frozen temporal embedding model, which produces a set of time-contextual embedding vectors (yellow). The 3D position of each electrode (red) is added to these vectors to produce the model inputs (orange). The PopT produces space-contextual embeddings for each electrode and a `[CLS]` token (blue), which can be fine-tuned for downstream tasks. During pretraining, (b) the PopT is trained on two objectives simultaneously: channel-wise loss and ensemble-wise loss. In channel-wise, PopT must determine whether an input channel has been replaced (green) with activity at a random other time that is inconsistent with the majority of inputs (orange or brown). In ensemble-wise, PopT determines whether two different ensembles (orange vs brown) represent consecutive or non-consecutive times.

the `CLS` token y_{cls}. During pretraining, the PopulationTransformer additionally is equipped with a linear layer head for the `[CLS]` token output and separate linear layer heads for all other individual token outputs. These produce the scalars $\tilde{y}_{cls}$ and $\tilde{y}_i$ and respectively, which are used in the objective (see Figure 1a).

Pretraining Our pretraining objective has two components: **channel-wise** and **ensemble-wise** losses (see Figure 1b). First, we describe our channel-wise discriminative learning. The model must determine whether a channels activity has been swapped with activity from a random time. Precisely, activity from each channel i is drawn from a time t_i. All channels are drawn from the same time $t_i = T$, and then 10% of the channels are randomly selected to have their activity replaced with activity from the same channel, but taken from a random point in time $t_i \neq T$. Then, given the token outputs of PopT, the channel-wise loss function $\mathcal{L}_C$ is the binary cross entropy.

Next, we describe the ensemble-wise discrimination task. Two different subsets of channels $S_A, S_B \subset C$ are chosen with the condition that they be disjoint $S_A \cap S_B = \emptyset$. During pretraining, the model receives the activities from these channels at separate times $X_A = \{x_i^t \mid i \in S_A\}$ and $X_B = \{x_i^{t'} \mid i \in S_B\}$. The objective of the task is then to determine whether these states X_A and X_B have occurred consecutively in time or are separated by some further, randomly selected interval. Given the output of the classification head, the loss function $\mathcal{L}_N$ is simply the binary cross entropy. Then, our complete objective function is $\mathcal{L} = \mathcal{L}_N + \mathcal{L}_C$.

Fine-tuning During fine-tuning, the `[CLS]` intermediate representation, $\tilde{y}_{cls}$ of the pretrained PopT is passed through a single layer linear neural network to produce a scalar $\hat{y}_{cls}$. This scalar is the input to binary cross entropy loss for our decoding tasks (see Section 3). After fine-tuning, we perform interpretability analysis on `[CLS]` attention weights with techniques outlined in Appendix F.

3 Experiment Setup

Data We use the publicly available subject data from [8]. Data was collected from 10 subjects (total 1,688 electrodes, with a mean of 167 electrodes per subject) who watched 26 movies while intracranial probes recorded their brain activity. The movie transcripts were aligned to the brain

activity so that features such as volume, pitch, etc. could be associated with the corresponding sEEG readings. 19 of the sessions are used for pretraining. 7 of the sessions are held-out for evaluation.

Decoding We evaluate the effectiveness of our pretrained PopT model by fine-tuning it on the four downstream decoding task used in the evaluation of [8]. Two of the tasks are audio focused: determining whether a word is spoken with a high or low pitch and determining whether a word is spoken loudly or softly. And two of the tasks have a more linguistic focus: determining whether the beginning of a sentence is occurring or determining whether any speech at all is occurring.

Our approach enables decoding on any arbitrary size of ensemble. We verify that our model is able to leverage additional channels for improved decoding performance that scales the number of inputs. To test this, we first order the electrodes by their individual linear decodability per task, and we increase the number of channels available to the model at fine-tuning time.

Baselines We want to determine whether the information about spatial relationships learned during pretraining was useful at fine-tuning time. For comparison, we concatenate the single-channel temporal embeddings and train a linear (Linear) or non-linear (DeepNN) aggregator on the decoding task. This sets a baseline for how much improvement is achievable from existing aggregation approaches [13]. To determine whether our performance can be attributed to using a more powerful architecture, we also fine-tune a PopT without pretraining, i.e. with randomly initialized weights.

4 Results

Decoding Performance We find that using a pretrained PopT significantly benefits downstream decoding compared to baseline channel aggregation techniques (Table 1). Additionally, while scaling performance with increasing number of channels is a challenging task for most baseline aggregation approaches, a pretrained PopT is able to scale well with increasing ensemble sizes (Figure 2a).

To gain confidence on our method's generalizability to channel encoders, we applied our framework to two different channel encoders: (1) an sEEG temporal encoder (BrainBERT [8]) and (2) a general time-series encoder (TOTEM [14]). We see significant boosts in performance with the pretrained PopT in both cases when compared with baseline aggregation approaches, across all 4 auditory-linguistic tasks (Table 1). These results suggest that our framework can generalize to benefit joint aggregation of other single-channel embeddings and neural recording modalities.

	Pitch	Volume	Sent. Onset	Speech/Non-speech
BrainBERT: single channel	0.53 ± 0.05	0.58 ± 0.08	0.68 ± 0.04	0.66 ± 0.09
Linear + BrainBERT	0.59 ± 0.08	0.66 ± 0.08	0.70 ± 0.09	0.71 ± 0.11
Deep NN + BrainBERT	0.58 ± 0.08	0.67 ± 0.08	0.71 ± 0.10	0.72 ± 0.10
Non-pretrained PopT	0.53 ± 0.06	0.61 ± 0.13	0.74 ± 0.10	0.70 ± 0.08
Pretrained PopT	$\mathbf{0.69 \pm 0.07}$	$\mathbf{0.84 \pm 0.06}$	$\mathbf{0.86 \pm 0.05}$	$\mathbf{0.89 \pm 0.07}$
TOTEM: single channel	0.53 ± 0.01	0.53 ± 0.02	0.69 ± 0.03	0.65 ± 0.04
Linear + TOTEM	0.55 ± 0.02	0.66 ± 0.03	0.79 ± 0.04	0.77 ± 0.05
Deep NN + TOTEM	0.57 ± 0.02	0.67 ± 0.03	0.78 ± 0.03	0.75 ± 0.05
Non-pretrained PopT	0.53 ± 0.02	0.64 ± 0.02	0.79 ± 0.03	0.77 ± 0.05
Pretrained PopT	$\mathbf{0.60 \pm 0.02}$	$\mathbf{0.73 \pm 0.02}$	$\mathbf{0.86 \pm 0.03}$	$\mathbf{0.84 \pm 0.06}$

Table 1: **Pretraining PopT is critical to downstream decoding performance.** We test on a variety of audio-linguistic decoding tasks (see Section 3) with either a single channel (row 1) or 90 channels (rows 2-5) as input. Two different temporal encoders are used: BrainBERT [8] (top section) and TOTEM [14] (bottom section). Shown are the ROC-AUC mean and standard error across subjects. We see that all aggregation approaches (rows 2-5) outperform single-channel decoding (row 1). Pretraining PopT and then fine-tuning it for downstream decoding results in significantly better performance (bold) compared to non-pretrained aggregation approaches (rows 2-4). This gain cannot be explained by simply providing more temporal embeddings, as evidenced by the performance of Linear and Deep NN (rows 2 and 3) that take the concatenated raw temporal embeddings as input. Neither can the gain be attributed to simply using a Transformer architecture, as is shown by a comparison with a non-pretrained PopT (row 4).

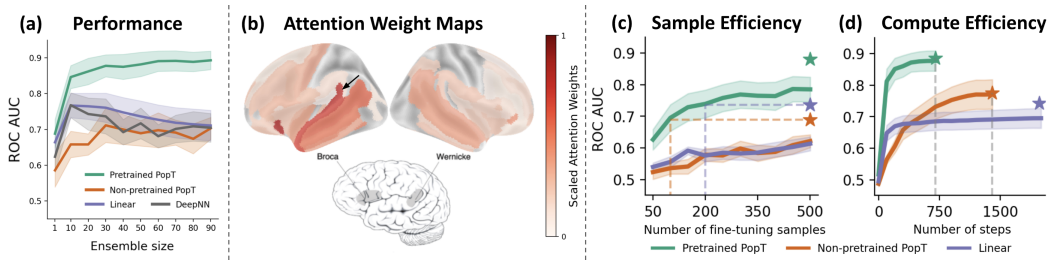

Figure 2: **(a) Pretrained PopT enables downstream performance scaling with ensemble size** Increasing channel ensemble size from 1 to 90 (x-axis), we see pretrained PopT (green) decoding performance (y-axis) not only beat non-pretrained approaches (orange, purple, grey), but also continually improve more with increasing channel count. Shaded bands show the standard error across subjects. **(b) Attention weights from a fine-tuned PopT identify candidate functional brain regions.** Candidate functional maps can be read from attention weights of a PopT fine-tuned on our decoding tasks. Note the weight placed on regions near Wernicke's area (black arrow) for this Speech vs. Non-speech tuned model. Lower brain figure highlight regions related to auditory-linguistic processing such as language production area Broca's area and language understanding Wernicke's area (adapted from [15]). **(c) Pretrained PopT is more sample efficient when fine-tuning.** Varying the number of samples available to each model at train time (x-axis), we see how the pretrained PopT is highly sample efficient, requiring only a fraction of samples to reach the full performance level of non pretrained aggregation approaches (dashed lines). Bands show standard error across test subjects. Stars indicate performance of the model trained on the full fine-tuning dataset. **(d) Pretrained PopT is consistently more compute efficient when fine-tuning.** Number of steps required for each model to reach final performance during fine-tuning (dashed lines). We find that pretrained PopT consistently requires fewer than 750 steps (each step is training on a batch size of 256) to converge, in contrast to the 2k steps required for the non pretrained PopT. Linear aggregation can be similarily compute efficient, but occasionally benefits from more training steps depending on dataset size. Bands show standard error across test subjects. Stars indicate fully trained performance.

Interpretability To analyze what our massively pretrained + fine-tuned model for sEEG data may be doing, we uncover the attention weights the model places on each input channel. We find agreement in our model's attention placement with brain regions typically involved in langauge processing (e.g. Wernicke's area), especially in the Speech vs. Non-speech downstream task (Figure 2b).

Efficiency To show that our technique is accessible to low data and compute regimes, we demonstrate that a pretrained PopT reaches the same decoding performance as other baseline approaches with an order of magnitude fewer samples and steps (Figure 2c and d). Pretraining PopT itself on more unnanotated data is also an order of magnitude more lightweight than pretraining existing end-to-end temporal-spatial models (see Appendix B). By focusing on population-level learning and leveraging the growing base of pretrained single-channel embedding techniques, our framework is efficient for learning new decoding tasks and continual pretraining.

5 Conclusion

We presented a self-supervised learning scheme for learning effective representations of intracranial activity from temporal embeddings. By decoupling temporal and spatial feature extraction, we are able to leverage existing temporal embeddings to learn spatiotemporal representations efficiently and with a smaller number of parameters. We showed that self-supervised pretraining imbues our model with knowledge of spatial relationships between these embeddings and improved downstream decoding that scales with the number of available channels. This scheme produces interpretable weights from which attention weight maps can be read to help uncover learned relationships from the massively pretrained framework. Finally, we release the pretrained weights for our PopT with BrainBERT inputs as well as our code for plug-and-play pretraining with any temporal embedding.

References

[1] Josef Parvizi and Sabine Kastner. Promises and limitations of human intracranial electroencephalography. *Nature neuroscience*, 21(4):474–483, 2018.

[2] Christian Herff, Dean J Krusienski, and Pieter Kubben. The potential of stereotactic-eeg for brain-computer interfaces: current progress and future directions. *Frontiers in neuroscience*, 14: 483258, 2020.

[3] Sina Faezi, Rozhin Yasaei, and Mohammad Abdullah Al Faruque. Htnet: Transfer learning for golden chip-free hardware trojan detection. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1484–1489. IEEE, 2021.

[4] Stephanie Martin, Iñaki Iturrate, José del R Millán, Robert T Knight, and Brian N Pasley. Decoding inner speech using electrocorticography: Progress and challenges toward a speech prosthesis. *Frontiers in neuroscience*, 12:367292, 2018.

[5] Sean L Metzger, Kaylo T Littlejohn, Alexander B Silva, David A Moses, Margaret P Seaton, Ran Wang, Maximilian E Dougherty, Jessie R Liu, Peter Wu, Michael A Berger, et al. A high-performance neuroprosthesis for speech decoding and avatar control. *Nature*, 620(7976): 1037–1046, 2023.

[6] Francis R Willett, Erin M Kunz, Chaofei Fan, Donald T Avansino, Guy H Wilson, Eun Young Choi, Foram Kamdar, Matthew F Glasser, Leigh R Hochberg, Shaul Druckmann, et al. A high-performance speech neuroprosthesis. *Nature*, 620(7976):1031–1036, 2023.

[7] Trung Le and Eli Shlizerman. Stndt: Modeling neural population activity with spatiotemporal transformers. *Advances in Neural Information Processing Systems*, 35:17926–17939, 2022.

[8] Christopher Wang, Vighnesh Subramaniam, Adam Uri Yaari, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. Brainbert: Self-supervised representation learning for intracranial recordings. In *The Eleventh International Conference on Learning Representations*, 2022.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[10] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[11] Andy T. Liu, Shang-Wen Li, and Hung-yi Lee. TERA: self-supervised learning of transformer encoder representation for speech. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:2351–2366, 2021. doi: 10.1109/TASLP.2021.3095662. URL https://doi.org/10.1109/TASLP.2021.3095662.

[12] Graham Wideman. Orientation and voxel-order terminology: Ras, las, lpi, rpi, xyz and all that, 2024. URL http://www.grahamwideman.com/gw/brain/orientation/orientterms.htm.

[13] Gaurav R Ghosal and Reza Abbasi-Asl. Multi-modal prototype learning for interpretable multivariable time series classification. *arXiv preprint arXiv:2106.09636*, 2021.

[14] Sabera Talukder, Yisong Yue, and Georgia Gkioxari. Totem: Tokenized time series embeddings for general time series analysis. *arXiv preprint arXiv:2402.16412*, 2024.

[15] What is aphasia? — types, causes and treatment, Mar 2017. URL https://www.nidcd.nih.gov/health/aphasia.

[16] Ran Liu, Mehdi Azabou, Max Dabagia, Jingyun Xiao, and Eva Dyer. Seeing the forest and the tree: Building representations of both individual and collective dynamics with transformers. *Advances in neural information processing systems*, 35:2377–2391, 2022.

[17] Sabera J Talukder and Georgia Gkioxari. Time series modeling at scale: A universal representation across tasks and domains. 2023.

[18] Hsiang-Yun Sherry Chien, Hanlin Goh, Christopher M Sandino, and Joseph Y Cheng. Maeeg: Masked auto-encoder for eeg representation learning. *arXiv preprint arXiv:2211.02625*, 2022.

[19] Demetres Kostas, Stephane Aroca-Ouellette, and Frank Rudzicz. Bendr: using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data. *Frontiers in Human Neuroscience*, 15:653659, 2021.

[20] Ke Yi, Yansen Wang, Kan Ren, and Dongsheng Li. Learning topology-agnostic eeg representations with geometry-aware modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[21] Armin Thomas, Christopher Ré, and Russell Poldrack. Self-supervised learning of brain dynamics from broad neuroimaging data. *Advances in Neural Information Processing Systems*, 35:21255–21269, 2022.

[22] Xuan Kan, Wei Dai, Hejie Cui, Zilong Zhang, Ying Guo, and Carl Yang. Brain network transformer. *Advances in Neural Information Processing Systems*, 35:25586–25599, 2022.

[23] Josue Ortega Caro, Antonio Henrique Oliveira Fonseca, Christopher Averill, Syed A Rizvi, Matteo Rosati, James L Cross, Prateek Mittal, Emanuele Zappala, Daniel Levine, Rahul M Dhodapkar, et al. Brainlm: A foundation model for brain activity recordings. *bioRxiv*, pages 2023–09, 2023.

[24] Antonis Antoniades, Yiyi Yu, Joseph Canzano, William Wang, and Spencer LaVere Smith. Neuroformer: Multimodal and multitask generative pretraining for brain data. *arXiv preprint arXiv:2311.00136*, 2023.

[25] Daoze Zhang, Zhizhang Yuan, Yang Yang, Junru Chen, Jingjing Wang, and Yafeng Li. Brant: Foundation model for intracranial neural signal. *Advances in Neural Information Processing Systems*, 36, 2024.

[26] Chaoqi Yang, M Westover, and Jimeng Sun. Biot: Biosignal transformer for cross-data learning in the wild. *Advances in Neural Information Processing Systems*, 36, 2024.

[27] Weibang Jiang, Liming Zhao, and Bao liang Lu. Large brain model for learning generic representations with tremendous EEG data in BCI. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=QzTpTRVtrP.

[28] Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. Neural data transformer 2: multi-context pretraining for neural spiking activity. *Advances in Neural Information Processing Systems*, 36, 2024.

[29] Donghong Cai, Junru Chen, Yang Yang, Teng Liu, and Yafeng Li. Mbrain: A multi-channel self-supervised learning framework for brain signals. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 130–141, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599426. URL https://doi.org/10.1145/3580305.3599426.

[30] Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, 36, 2024.

[31] Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

[32] Brianna M Karpowicz, Yahia H Ali, Lahiru N Wimalasena, Andrew R Sedler, Mohammad Reza Keshtkaran, Kevin Bodkin, Xuan Ma, Lee E Miller, and Chethan Pandarinath. Stabilizing brain-computer interfaces through alignment of latent dynamics. *BioRxiv*, pages 2022–04, 2022.

[33] Alan D Degenhart, William E Bishop, Emily R Oby, Elizabeth C Tyler-Kabara, Steven M Chase, Aaron P Batista, and Byron M Yu. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nature biomedical engineering*, 4(7):672–685, 2020.

[34] Justin Jude, Matthew G Perich, Lee E Miller, and Matthias H Hennig. Robust alignment of cross-session recordings of neural population activity by behaviour via unsupervised domain adaptation. feb 2022. doi: 10.48550. *arXiv preprint arXiv.2202.06159*.

[35] Xuan Ma, Fabio Rizzoglio, Kevin L Bodkin, Eric Perreault, Lee E Miller, and Ann Kennedy. Using adversarial networks to extend brain computer interface decoding accuracy over time. *elife*, 12:e84296, 2023.

[36] Sabera Talukder, Jennifer J Sun, Matthew Leonard, Bingni W Brunton, and Yisong Yue. Deep neural imputation: A framework for recovering incomplete brain recordings. *arXiv preprint arXiv:2206.08094*, 2022.

[37] Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. *arXiv preprint arXiv:2110.05357*, 2021.

[38] Geeling Chau, Yujin An, Ahamed Raffey Iqbal, Soon-Jo Chung, Yisong Yue, and Sabera Talukder. Generalizability under sensor failure: Tokenization+ transformers enable more robust latent spaces. *arXiv preprint arXiv:2402.18546*, 2024.

[39] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

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

[41] ildoonet. ildoonet/pytorch-gradual-warmup-lr: Gradually-warmup learning rate scheduler for pytorch, 2024. URL https://github.com/ildoonet/pytorch-gradual-warmup-lr.

[42] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[43] Christophe Destrieux, Bruce Fischl, Anders Dale, and Eric Halgren. Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature. *Neuroimage*, 53(1): 1–15, 2010.

[44] Nilearn, 2015. URL https://nilearn.github.io/stable/index.html.

A Related Work

Self-supervised learning on neural data Channel independent pretrained models are a popular approach for neural spiking data [16], intracranial brain data [8, 17], and general time-series [14]. Additionally, in fixed-channel neural datasets, approaches exist for EEG [18–20], fMRI [21–23], and calcium imaging [24] datasets. However, all of this work do not learn population-level interactions across datasets with different recording layouts due to the single-channel focus or the ability to assume fixed-channel setups. Several works pretrain spatial and temporal dimensions across datasets with variable inputs [25–29], but most simultaneously learn the temporal embeddings with the spatial modeling, which make them challenging to interpret and computationally expensive to train. As far as we know, we are the first to study the problem of building pretrained channel aggregation models on top of pre-existing temporal embeddings trained across datasets with variable sampling of input channels, allowing for modeling of high quality (>2kHz sampling rate) intracranial data.

Modeling across variable input channels Modeling spatial representations on top of temporal embeddings have been found to be beneficial for decoding [3, 7, 30], but prior works use supervised labels, so do not leverage large amounts of unannotated data. The brain-computer-interface field has been studying how to align latent spaces [31–35] which either still requires creating an alignment matrix to learn across datasets or only provides post-training alignment mechanisms rather than learning across datasets. Other approaches impute missing channels or learn latent spaces robust to missing channels [36–38], but these are more suited for the occasional missing channel rather than largely varying sensor layouts. We directly learn spatial-level representations using self-supervised learning across datasets to leverage massive amounts of unannotated intracranial data.

B Model and Compute Requirements

	e5	e50	e90
PopT		20M	
Deep NN	3M	20M	36M
Linear	3.8k	38k	69k
Brant [25]		500M	
LaBraM [27]		350M	

Table 2: **Parameter counts**. Since PopT takes existing temporal embeddings as input, the number of parameters that must be trained is an order of magnitude less than recent end-to-end approaches.

	GPU count	GPU type	Time to train	TFLOPS
PopT	1	NVIDIA TITAN RTX (24GB)	2 days	2.1M
Brant [25]	4	NVIDIA Tesla A100 (80G)	2.8 days	18.8M
LaBraM [27]	8	NVIDIA Tesla A800 (40G)	–	–

Table 3: **Pretraining compute requirements** Based on published train times (none were given for LaBraM) it is evident that PopT has smaller hardware and shorter training time requirements.

C Architectures and training

Pretrained PopT The core Population Transformer consists of a transformer encoder stack with 6 layers, 8 heads. All layers ($N = 6$) in the encoder stack are set with the following parameters: $d_h = 512$, $H = 8$, and $p_{\text{dropout}} = 0.1$. We pretrain the PopT model with the LAMB optimizer [39] ($lr = 1e - 4$), with a batch size of $n_{\text{batch}} = 256$, and train/val/test split of 0.98, 0.01, 0.01 of the data. We pretrain for 500,000 steps, and record the validation performance every 1,000 steps. Downstream evaluation takes place on the weights with the best validation performance. We use the intermediate representation at the `[CLS]` token $d_h = 512$ and put a linear layer that outputs to $d_{out} = 1$ for fine-tuning on downstream tasks. These parameters for pretraining were the same for any PopT that needed to be pretrained (hold-one-out subject, subject subsets, ablation studies).

Non-pretrained PopT The architecture for the non-pretrained PopT is the same as the pretrained PopT (above). However, no pretraining is done, and the weights are randomly initialized with the default initializations.

Linear The linear baseline consists of a single linear layer that outputs to $d_{out} = 1$. The inputs are flattened and concatenated BrainBERT embeddings $d_{emb} = 756$ or TOTEM embeddings $d_{emb} = 64$ from a subset of channels $S \subset C$. Thus, the full input dimension is $d_{input} = d_{emb} * |S|$.

Deep NN The inputs are the same as above, but the decoding network now consists of 5 stacked linear layers, each with $d_h = 512$ and a GeLU activation.

Downstream Training For both PopT models, we train with these parameters: AdamW optimizer [40], $lr = 5e^{-4}$ where transformer weights are scaled down by a factor of 10 ($lr_t = 5e^{-5}$), $n_{batch} = 256$, a Ramp Up scheduler [41] with warmup 0.025 and Step LR gamma 0.99, reducing 100 times within the 2000 total steps that we train for. For Linear and DeepNN models, we train with these parameters: AdamW optimizer [40], $lr = 5e^{-4}$, $n_{batch} = 256$, a Ramp Up scheduler [41] with warmup 0.025 and Step LR gamma 0.95, reducing 25 times within the 17,000 total steps we train for. For all downstream decoding, we use a fixed train/val/test split of 0.8, 0.1, 0.1 of the data.

Compute Resources To run all our experiments (data processing, pretraining, evaluations, interpretability), one only needs 1 NVIDIA Titan RTXs (24GB GPU RAM). Pretraining PopT takes 2 days on 1 GPU. Our downstream evaluations take a few minutes to run each. For the purposes of gathering all the results in the paper, we parallelized the experiments on roughly 8 GPUs.

D Decoding tasks

We follow the same task specification as in Wang et al. [8], with the modification that the pitch and volume examples are determined by percentile (see below) rather than standard deviation in order to obtain balanced classes.

Pitch The PopT receives an interval of activity and must determine if it corresponds with a high or low pitch word being spoken. For the duration of a given word, pitch was extracted using Librosa's `piptrack` function over a Mel-spectrogram (sampling rate 48,000 Hz, FFT window length of 2048, hop length of 512, and 128 mel filters). For this task, for a given session, positive examples consist of words in the top-quartile of mean pitch and negative examples are the words in the bottom quartiles.

Volume The volume of a given word was computed as the average intensity of root-mean-square (RMS) (`rms` function, frame and hop lengths 2048 and 512 respectively). As before, positive examples are the words in the top-quartile of volume and negative examples are those in the bottom quartiles.

Sentence onset Negative examples are intervals of activity from 1s periods during which no speech is occurring in the movie. Positive examples are intervals of brain activity that correspond with hearing the first word of a sentence.

Speech vs. Non-speech Negative examples are as before. Positive examples are intervals of brain activity that correspond with dialogue being spoken in the stimuli movie.

E Data

Subj.	Age (yrs.)	# Electrodes	Movie	Recording time (hrs)	Held-out
1	19	91	Thor: Ragnarok Fantastic Mr. Fox The Martian	1.83 1.75 0.5	 x
2	12	100	Venom Spider-Man: Homecoming Guardians of the Galaxy Guardians of the Galaxy 2 Avengers: Infinity War Black Panther Aquaman	2.42 2.42 2.5 3 4.33 1.75 3.42	 x
3	18	91	Cars 2 Lord of the Rings 1 Lord of the Rings 2 (extended edition)	1.92 2.67 3.92	x
4	9	135	Megamind Toy Story Coraline	2.58 1.33 1.83	 x
5	11	205	Cars 2 Megamind	1.75 1.77	x
6	12	152	Incredibles Shrek 3 Megamind	1.15 1.68 2.43	 x
7	6	109	Fantastic Mr. Fox	1.5	
8	4.5	72	Sesame Street Episode	1.28	
9	16	102	Ant Man	2.28	
10	12	173	Cars 2 Spider-Man: Far from Home	1.58 2.17	x

Table 4: **Subject statistics** Subjects used in PopT training, and held-out downstream evaluation. Table taken from [8]. The number of uncorrupted, electrodes that can be Laplacian re-referenced are shown in the second column The average amount of recording data per subject is 4.3 (hrs).

F Interpretation Methods

Candidate functional brain regions from attention weights After fine-tuning our weights on a decoding task, we can examine the attention weights of the `[CLS]` output for candidate functional brain regions. We obtain a normalized Scaled Attention Weight metric (see next section) across all subjects to be able to analyze candidate functional brain regions across sparsely sampled subject datasets. The Scaled Attention Weight is computed from raw attention weights at the `[CLS]` token passed through the attention rollout algorithm [42]. The resulting weights from each channel are then grouped by brain region according to the Destrieux layout [43].

Scaled Attention Weight First, we obtain an attention weight matrix across all trials which includes weights between all tokens. Then, we perform attention rollout [42] across layers to obtain the contributions of each input channel by the last layer. We take the resulting last layer of rollout weights for all channels, where the target is the `[CLS]` token, normalize within subject, and scale by ROC AUC to obtain the Scaled Attention Weight per channel. Finally, we plot the 0.75 percentile weight per region, as mapped by the Destrieux atlas [43] using Nilearn [44].

G Ablation study

Ablation of loss components and position information An ablation study confirms that both the network-wise and channel-wise component of the pretraining objective contribute to the downstream performance (Table 5). We also find that including the 3D position information for each channel is critical for decoding. Additionally, we find that the discriminative nature of our loss is necessary for decoding. Attempting to add an L1 reconstruction term to our pretraining objective results in poorer performance, perhaps because the model learns to overfit on low-entropy features in the embedding. Our discriminative loss requires the model to understand the embeddings in terms of how they can be distinguished from one another, which leads the model to extract more informative representations.

	Pitch	Volume	Sent. Onset	Speech/Non-speech
PopT	$\mathbf{0.69 \pm 0.07}$	$\mathbf{0.84 \pm 0.06}$	$\mathbf{0.86 \pm 0.05}$	$\mathbf{0.89 \pm 0.07}$
PopT w/o group-wise loss	0.66 ± 0.07	0.83 ± 0.06	0.84 ± 0.04	0.88 ± 0.08
PopT w/o channel-wise loss	0.67 ± 0.06	0.81 ± 0.08	0.84 ± 0.06	0.87 ± 0.09
PopT w/o position encoding	0.59 ± 0.07	0.67 ± 0.10	0.75 ± 0.08	0.79 ± 0.08
PopT with reconstruction loss	0.60 ± 0.11	0.73 ± 0.11	0.81 ± 0.05	0.83 ± 0.09
PopT with L1 reconstruction only	0.56 ± 0.04	0.65 ± 0.08	0.73 ± 0.10	0.74 ± 0.10

Table 5: **PopT ablation study.** We individually ablate our losses and positional encodings during pretraining then decode on the resulting models. Shown are ROC-AUC mean and standard error across subjects. The best performing model across all decoding tasks uses all three of our proposed components, showing that they are all necessary. Removing our positional encoding during pretraining and fine-tuning drops the performance the most, indicating that position encoding is highly important for achieving good decoding. Additionally, we attempt adding a reconstruction component to the loss or purely using the L1 mask loss, but find that this leads to poorer performance (last two rows).