Learning Representations of Human Neural Activity via Contrastive Neural Forecasting

Anonymous Author(s)
Affiliation
Address
email

Abstract

Electrical patterns of brain activity form the basis of everything from perception and movement to complex behaviors like decision-making and conscious thought. While terabytes of human intracranial electroencephalography (iEEG) recordings are openly available, deciphering and productively using them for downstream use cases remains a challenging problem. We present Contrastive Neural Forecasting (CNF), a simple self-supervised framework for learning representations of population-level neural activity across electrodes, time, and individuals from unlabeled data at scale. The CNF objective requires the model to predict future neural states in its latent space using cross-entropy over batched data samples. Our objective is designed to resolve two major challenges in traditional MSE-based autoencoding approaches. Forecasting in the latent space relieves the model from overfitting to noise that is inherent in the data, and the cross-entropy loss enables flexible capturing of high-dimensional, multimodal distributions underlying the evolution of neural dynamics. We validate the superior performance of the contrastive objective on BrainBERT, and then train and open-source CNF-1, a foundation model for human iEEG. We pretrain end-to-end directly from raw voltage traces, without relying on handcrafted features or frequency band filtering. While still closely followed by the linear baselines, which we found in many cases score higher than other pretrained models, CNF-1 achieves state-of-the-art performance on a suite of downstream decoding tasks. Surprisingly, and challenging assumptions made in prior work, we obtain better performance by omitting the spatial location of the electrodes from the embeddings, instead allowing the model to learn its own channel-specific parameters. We show how CNF-1 can enable novel approaches to extract neuroscientific insight from unlabeled data at scale. We envision future clinical applications such as real-time functional region mapping and model-guided electrical stimulation interventions in the operating room, as well as next-generation brain-computer interfaces. Taken together, our work paves the way for scalable brain foundation models trained entirely from observational data.

1 Introduction

The human brain continuously processes rich, overlapping streams of information: from interpreting speech and recognizing objects to reasoning about complex events [Schurz et al., 2014]. Despite considerable progress in neuroscience over the past decades, building a comprehensive computational model of the brain, where the brain state can be decoded, simulated, and interfaced with seamlessly, remains a formidable challenge [Sejnowski et al., 2014]. Through invasive and non-invasive stimulation interventions, these models could enable personalized treatments for neurological disorders such as epilepsy [Herron et al., 2024, Morrell, 2011], and through superior decoding and encoding capabilities enhance communication between brains and machines as well as between brains and other brains [Pereira et al., 2018].

Foundation models have transformed fields like natural language processing and computer vision, offering unparalleled performance across tasks and datasets [Bommasani et al., 2021, Brown et al., 2020]. Yet, their potential in neuroscience remains untapped. Even if brain foundation models (such as transformers pretrained on large volumes of data) do not provide simple and interpretable models of brain function, their capacity for capturing complex and high-dimensional relationships across brain areas, time, and individuals positions them as powerful tools for advancing medicine and neuroscience [Parvizi and Kastner, 2012].

Human intracranial encephalography (iEEG) offers brain interfacing at an unprecedented combination of spatial and temporal resolution. However, these raw voltage time series are noisy, high-dimensional, highly nonlinear, and non-trivially dependent on physiological variables [Noury et al., 2016, Buzsáki et al., 2012], which has been a major obstacle to gaining useful insight using this data and to creating new tools and treatments relying on iEEG.

Our work introduces *Contrastive Neural Forecasting (CNF)*, a simple self-supervised framework for learning population-level representations of human brain dynamics. We propose path toward general-purpose brain models that excels by learning directly from raw voltage traces, without requiring knowledge of electrode locations or prior domain-specific assumptions.

Contributions The key contributions of this work are:

- We propose a contrastive predictive learning objective tailored for neural time series data that forecasts future neural states in latent space using a cross-entropy loss over real samples, enabling flexible modeling of high-dimensional, multimodal distributions underlying neural dynamics. We validate this objective by showing its superior performance when pretraining BrainBERT [Wang et al., 2023].
- Our proposed objective enables the unification of representation learning from single channel voltage traces with population activity over the whole brain by combining information from many electrodes, without the need for pretrained channel feature extractors or spectrogram encoders, handcrafted features, or frequency-based preprocessing.
- We propose the use of learned electrode embeddings for modeling of iEEG data. Our method learns without access to spatial information about electrodes, recovering it during pretraining in a puerly data driven way, challenging the assumption that spatial embeddings are necessary for accurate neural modeling.
- We introduce **CNF-1**, a foundation model trained end-to-end on raw iEEG voltage traces across individuals and electrodes, achieving state-of-the-art performance on multiple decoding benchmarks. We release the pretrained CNF-1 model and codebase to promote further research on scalable, general-purpose brain foundation models (upon publication).

1.1 Related work

Foundation Models for Neural Data. Neuroformer introduced a multimodal, multitask generative pretrained transformer tailored for systems neuroscience, capable of associating behavioral and neural representations through joint training [Antoniades et al., 2024]. BrainBERT [Wang et al., 2023] learns representations from single channels of intracranial EEG in a self-supervised manner by predicting masked out spectrograms. Our work The Population Transformer (PopT, Chau et al. [2024]) and [Zhang et al., 2023] extended pretrained embeddings from BrainBERT to enable decoding on the population level. We provide two main improvements that raise the performance of the model: the contrastive forecasting objective as opposed to the naïve MSE approach and learnable electrode embeddings instead of positional coordinate embeddings. Learnable embeddings were introduced by [Azabou et al., 2023] for the single unit modeling. We use the same learned embeddings and adapt them for the continuous iEEG signal. NDT2 emphasized large-scale spatiotemporal pretraining for neural spiking activity, facilitating adaptation to novel contexts in decoding tasks [Ye et al., 2023]. Foundation models for neural data have been developed for single unit activity and fMRI [Liu et al., 2022, Cai et al., 2023, Dong et al., 2024], however our work focuses specifically on human intracranial EEG. For a review of brain foundation models, see Zhou et al. [2025].

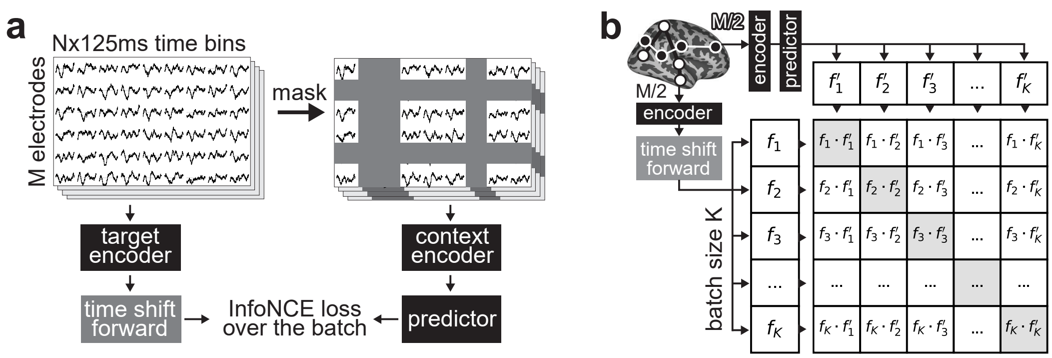

Figure 1: **Overview of the Contrastive Neural Forecasting approach.** (a) The neural activity timeseries is binned and split into two parallel streams: target and context. Each target timebin is encoded separately, and the whole timeseries is shifted forward by one timebin. The context stream is randomly masked in both the electrode and the time dimensions, and passed into the context encoder and then into the predictor. The target and context stream are compared using the InfoNCE objective, (b). The features generated by the model are compared with the corresponding electrode and timebin's features of every other item in the batch, and the InfoNCE objective requires that the corresponding pairs from the same item in the batch are close together in the feature space, and far away from the features of other items in the batch. All of the model components are trained end-to-end.

Contrastive representation learning. Contrastive Predictive Coding was first introduced by [van den Oord et al., 2019] for representation learning, and successfully applied at scale by CLIP for language-image pretraining [Radford et al., 2021]. We adapt these concepts from the ML literature to our neural data setting. STNDT [Le and Shlizerman, 2022], in a single unit spiking modeling setting, used contrastive learning as an auxillary loss to further augment the data and constrain the model. Similarly, [Vishnubhotla et al., 2023] use contrastive learning to learn representations for spike sorting of single units. We too use contrastive learning, but directly for building models of continuous iEEG signal.

2 The Contrastive Neural Forecasting approach

In this section, we overview the main components of our approach (Figure 1): predicting the future latent representation of the signal and the contrastive forecasting objective based on the cross-entropy loss. We assume that the neural data originates from a set of M channels (e.g., electrodes in the brain), sampled at a constant rate to produce T data segments of length τ each, where τ is the desired binning size for predictive modeling. Let's denote $x_t^{(m)}$ as the data sample from channel $m \leq M$ at time $t \leq T$. In the self-supervised learning setting, one is interested in modeling the distribution of masked data conditioned on unmasked data. As an illustrative example, let's say we want to autoregressively predict the joint distribution of the signal at time $t+1$ using the previous T time steps: $p(x_{t+1}^1 ... x_{t+1}^M \mid x_1^1 ... x_t^M)$. One approach, prevalent in the literature [Wang et al., 2023, Zhang et al., 2023], is to define a parametrized predictive function F (i.e. a neural network), and train the parameters to minimize the mean squared error (MSE) between the predicted and true masked datapoints:

$$\mathcal{L}_{MSE} = \sum_i \left\| \widehat{x}_{t+1}^i - x_{t+1}^i \right\|^2, \tag{1}$$

where $\widehat{x}_{t+1}^i = F(i, x_1^1 ... x_t^M)$ denotes the prediction of the model.

Despite the popularity of this approach, it has two flaws. First, in settings with an inherently high level of noise, typical for recordings from intracranial electrodes, the MSE objective punishes the model for poorly fitting the noise, encouraging overfitting to the noise pattern of the training dataset. Further, these signals tend to be temporally autocorrelated.

The second flaw is in the implicit assumption that underlies the choice of MSE as the objective: that the distribution of masked timepoints can be effectively captured with a unimodal Gaussian centered around the mean which is equal to the prediction of the model (for an overview of this equivalence, see Bishop [2006]). This assumption is not justified in our setting of interest. In practice, the dynamical system is partially observed (the dimensionality of the signal, $M < 300$, is negligible compared to the roughly 80 billion neurons in the brain), meaning that the observed input data put very mild constraints on the multimodal distribution of the future evolution of the observed data.

To overcome these challenges, we introduce Contrastive Neural Forecasting. In CNF, the input data is encoded with the context encoder $E_{context}$ and passed into a predictor P, and then compared to the encoding of target data by a target encoder E_{target}. Specifically, we use the InfoNCE objective, which pushes the predicted embedding $P(E_{context}(x_1^1...x_t^M))$ to be close to the embedding of the real target $E_{target}(x_{t+1}^1...x_{t+1}^M)$ and far from the embeddings of other random timesamples in the dataset. Formally, given a set of N random negative samples with timepoints $t'_1, ..., t'_N$, the InfoNCE loss is defined as:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp\left(P(E_{\text{context}}(x_1^1...x_t^M)) \cdot E_{\text{target}}(x_{t+1}^1...x_{t+1}^M)/\tau\right)}{\sum_{j=1}^{N} \exp\left(P(E_{\text{context}}(x_1^1...x_t^M)) \cdot E_{\text{target}}(x_{t'_j}^1...x_{t'_j}^M)/\tau\right)}, \tag{2}$$

where $\cdot$ denotes cosine similarity (dot product over the normalized features), and τ is a temperature hyperparameter. In practice, this objective is efficiently implemented as the cross-entropy loss over the batch dimension, meaning that negative samples for every item in the batch are taken from the other items encountered in the same batch.

This formulation has three advantages. First, it automatically ensures that there is no incentive for the model to encode noise in its latent space, where noise is defined as anything that is not helpful to disambiguate the true future neural state from other random examples of neural states. Second, it doesn't place assumptions (such as unimodality) on the distribution of the future timestep. Finally, it turns the hard problem of modeling the high-dimensional, continuous distribution of the neural signal into the "easy" problem of multi-class classification using the cross-entropy loss.

In the next sections, we describe the experiment setup and results that demonstrate these advantages.

3 Experimental Setup

Data To train and evaluate the performance of our objective, we use the publicly available BrainTreebank dataset [Wang et al., 2024]. The dataset consists of 43 hours of intracranial SEEG recordings from 10 human subjects (ages 4–19) implanted with a total of 1,688 electrodes while passively watching 26 full-length Hollywood films. It includes aligned audio-visual and language annotations for over 223,000 words across nearly 39,000 sentences, offering high temporal and spatial resolution data suitable for multimodal neural decoding and large-scale modeling.

Decoding evaluation tasks We evaluate models on a suite of 14 standardized neural decoding tasks spanning vision, audio, language, and multimodal domains, derived from the annotations in the BrainTreebank dataset, such as audio volume, optical flow direction, face count, word onset, LLM surprisal score, part-of-speech, speaker identity, etc. All of the tasks are formalized as binary classification by thresholding the annotations. The models are tasked with classifying the task labels from voltage traces of length 1 second aligned to each word onset. This decoding benchmark contains labeled neural data from 12 recording sessions across 6 individuals. We evaluate the models by fine-tuning on each task's training split, and testing on the non-intersecting test split that was taken from a different recording session. For more details about the decoding tasks, see Appendix A.

Models We bin the neural data sampled at 2048 Hz into bins of 256 samples each (125 ms). The target encoder E_{target} is a simple linear layer from the raw 256-dimensional feature vector into the $d_{model} = 192$ dimensional latent space. For the initial validation experiments, we reimplement a context encoder scaled-down version of the BrainBERT architecture [Wang et al., 2023] for computational efficiency, which we call BrainBERT-mini. BrainBERT-mini is a transformer encoder stack [Vaswani et al., 2023] with $N = 4$ layers and the hidden dimension size 192 with 12 attention heads per layer. For CNF-1, the context encoder (Figure 2a) is a transformer with 4 layers, hidden dimension d_{model}, that takes as input tokens which are 16 consecutive samples of the input data and produces the latent representations of dimensionality d_{model}, and 4 attention heads per layer. The outputs of the context encoder are concatenated to produce chunks of 256 samples for the next model

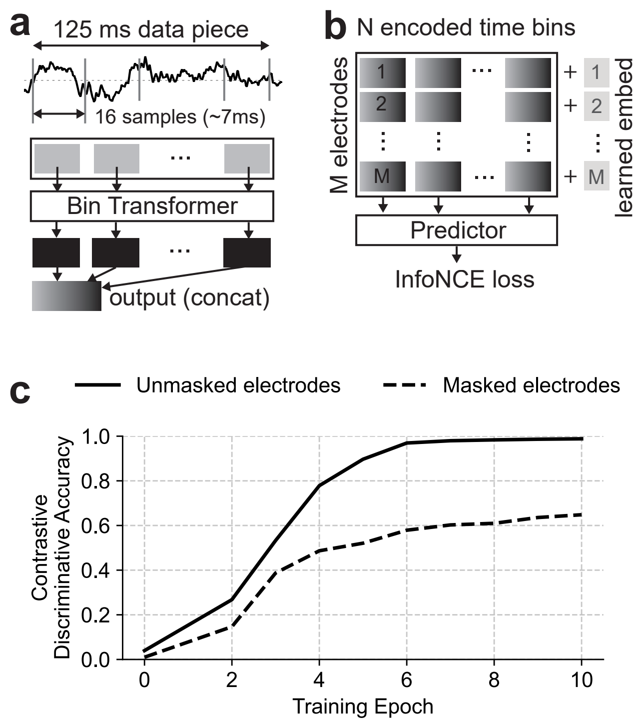

Figure 2: **Architecture and training dynamics of CNF-1.** Neural data sampled at 2048 Hz is binned into 125 ms segments (256 samples). The target encoder is a linear projection from 256 to a 192-dimensional latent space. The context encoder (a) is a 5-layer Transformer with 12 attention heads and hidden size 192, operating on tokens of 16 consecutive samples. The predictor (b) takes per-channel, per-timebin representations, adds learned electrode embeddings, and processes them through another 5-layer Transformer. Its outputs are compared with future target embeddings using the InfoNCE loss.

stage. The predictor (Figure 2b) takes in these representations for each channel $i \leq M$ and timebin $t \leq T = 1$ second, which are added to the learned per-channel electrode embeddings K_i (see next subsection), and again outputs the features of dimensionality d_{model} for every token by passing them through 5 layers of 12 attention heads each.

Pretraining For BrainBERT-mini experiments, we follow the approach of [Wang et al., 2023]. For the masking scheme, we set all data in $p = 10\%$ of timebins to 0, only passing the masked timebins into the objective (either MSE or our InfoNCE contrastive objective for this experiment). For CNF-1, the representations from its predictor output layer are then compared with the corresponding target embeddings *for the following future timebin*, using the InfoNCE objective as described in the previous section. For all models, we use a context of neural data of length 2 seconds. BrainBERT-mini is trained for 100 epochs on a small dataset containing just one subject's session data. CNF-1 is trained for 10 epochs (CNF-1) on data from 20 sessions from all 10 subjects in the dataset. We train all networks with learning rate 0.003, the Muon optimizer [Jordan et al., 2024], and learning rate scheduling of 100 steps of warmup followed by linear decay to 0. The networks are trained on a single A100 GPU for 10 hours (CNF-1) or 2 hours (BrainBERT-mini).

Electrode embeddings To provide the Predictor transformer with the information about which channel each signal comes from in the brain, we allow the model to learn additional separate vectors of dimensionality d_{model} for each channel $1 \leq i \leq M$ in the dataset, implementing the technique used by [Azabou et al., 2023] for the single unit modeling. We contrast this approach with the prior iEEG-based approaches [Zhang et al., 2023, Chau et al., 2024] that provide coordinates of each channel via cosine positional embeddings. Empirically, we find higher performance with fully learned embeddings, when not providing any spatial information of the electrodes into the model (see Appendix B).

Baselines and previous methods We compare the performance of our model to six different baselines and previous methods for feature learning on human intracranial EEG data:

- Linear regression from the raw voltage segments, aligned to the word onset.
- Linear regression from the spectrogram of the signal, normalized per frequency bin.
- Linear regression from the Fourier transform features (which include both magnitude and phase information of the frequency bands).
- Population Transformer Chau et al. [2024], a previous state of the art in representation learning from human intracranial EEG on the BrainTreebank dataset. We compare against frozen PopT (only fine tuning the output linear layer and keeping the model weights frozen), and a end-to-end finetuned PopT for each task.
- BrainBERT [Wang et al., 2023], a single-electrode representation extractor from iEEG on the same BrainTreebank dataset. For this evaluation, the features from every electrode are concatenated together before passing them into the linear regression layer to obtain the final prediction of the task label.

For more detail on baselines and previous methods, see Appendix C.

4 Results

Training objective	Mean decoding AUROC (14 tasks)
MSE loss (voltage)	0.638 ± 0.008
MSE loss (spectrogram)	0.598 ± 0.009
Contrastive (voltage, latent space)	$\mathbf{0.653 \pm 0.010}$
Contrastive (spectrogram, latent space)	0.631 ± 0.011
Contrastive (voltage, data space)	$\mathbf{0.664 \pm 0.011}$
Contrastive (spectrogram, data space)	0.632 ± 0.010

Table 1: **In pretraining BrainBERT-mini, the contrastive objective performs better than the traditional MSE loss across 14 decoding tasks.** Trained for 100 epochs using the Muon optimizer. The weights of the models are frozen after pretraining with no labels, and a linear regression is applied on the features of the frozen models to obtain the AUROC (mean $\pm$ SEM). The bolded entries indicate best performance (within one SEM of each other).

Superior performance of the contrastive pretraining objective when training BrainBERT-mini First, we seek to validate the performance of the contrastive objective on an established architecture from past literature - a scaled-down version of BrainBERT [Wang et al., 2023]. After the pretraining phase with no labels concludes, we freeze the model and fine-tune only a single linear layer on top of the model features on our downstream tasks of interest, in order to assess the quality of the generated representations, with results shown in Table 1. Our findings demonstrate that the contrastive loss performed better than its MSE counterpart in all experimental conditions (training on raw voltage vs spectrogram of the signal). Furthermore, we find that pretraining on raw voltage instead of the spectrogram of the signal, as often done in prior work, is beneficial for downstream performance across many tasks. Taking insight from this smaller scale experiment, we next scale up our pretraining to a larger chunk of the dataset, larger models, and expand to the population level information as opposed to only single electrode with CNF-1.

CNF-1 achieves state-of-the-art performance across the decoding tasks We train CNF-1 on the BrainTreebank dataset for 10 epochs and note that the model gets better at discriminating the true next timestep from random samples over the course of training (Figure B). To assess the quality of the representations learned by our model, and compare it to previously published models, we finetune

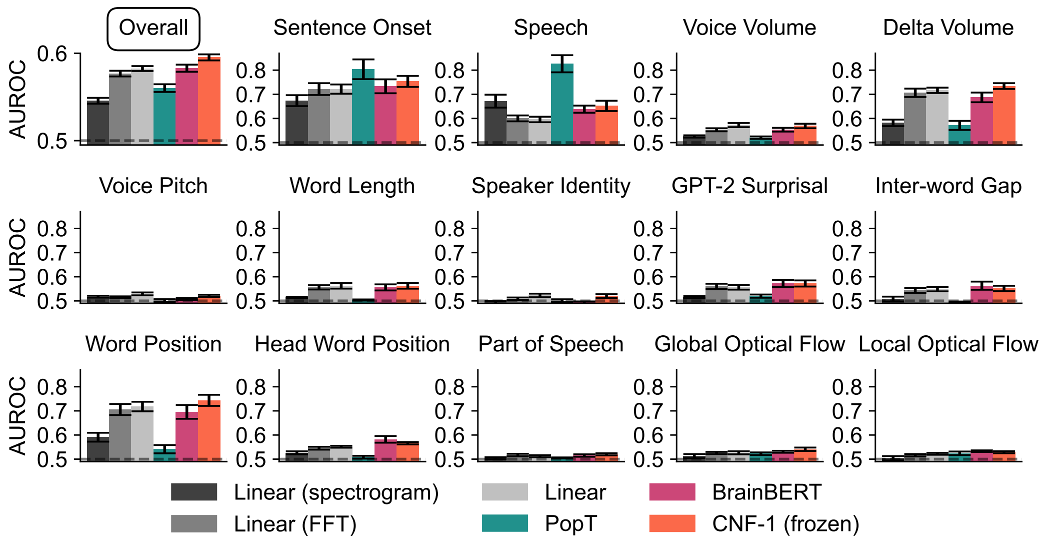

Figure 3: **CNF-1 achieves state-of-the-art performance on a suite of benchmark decoding tasks, across subjects and sessions.** We evaluate CNF-1, as well as baseline models and models from previous work, on 14 tasks that span language and visual domains, on 12 total recording sessions from 6 human subjects. We find that generally the linear baseline performs surprisingly well, at times outperforming pretrained models. CNF-1 (our model) outperforms all the considered pretrained models as well as all of the considered baselines overall.

the final linear layer that projects the features into a single output dimension on a suite of downstream decoding tasks (Figure 3). We find that generally the linear baseline performs surprisingly well, and surprisingly find that previous state of the art models, while outperforming all of the considered baseline methods on some tasks, fall behind them on others. we associate the lower performance of the BrainBERT and PopT models on some tasks with the low performance of the spectrogram regression baseline: spectrogram of the input signal is the base representation that both of these models use as input, and the baseline task decoding performance for some tasks (e.g. GPT-2 Surprisal, Word Length, Head Word Position etc) requires phase information as well, which is lost when taking the power spectrogram. In our experiments, we also found that pretraining often boosts decoding performance for some tasks (especially Onset and Speech) while decreasing the downstream performance on most other tasks.

While still close to the baselines, CNF-1 outperforms all the considered pretrained models and baselines (Figure 3, top left corner). While for PopT the performance peaks at some tasks and drops for others, CNF-1 shows a more uniform pattern of performance, suggesting that it contains representations that capture more aspects of the neural processing. The state-of-the-art performance of CNF-1 shows the potential and behind the Contrastive Neural Forecasting approach.

Investigating the learned electrode embeddings We now turn to what can be discovered in the data-driven way using our foundation model. An innovation from prior work is our entirely learned electrode embeddings, which replace the traditional coordinate positional embeddings.

We conjecture that over the course of training, the model may discover relationships between the input channels, and use the learned embedding parameters to store them across batches and employ them to improve the performance on the predictive objective. To test this hypothesis, we freeze the pretrained model and examine its learned electrode embeddings (example subject is shown in Figure 4). Across all pairs of electrodes, we find that the distance in embedding space of the model is strongly correlated with the physical distance between the electrodes in the brain (Figure 4a, $r = 0.400, p < 0.001$), despite the fact that spatial information was never available in pretraining. Furthermore, a t-SNE dimensionality analysis reveals spatially clustered groups of electrodes (Figure 4b) that are roughly corresponding to the gross anatomical and functional subdivision of the brain (Figure 4c). We note the consistent difference in the embeddings for the frontal, temporal and occipital lobe electrodes,

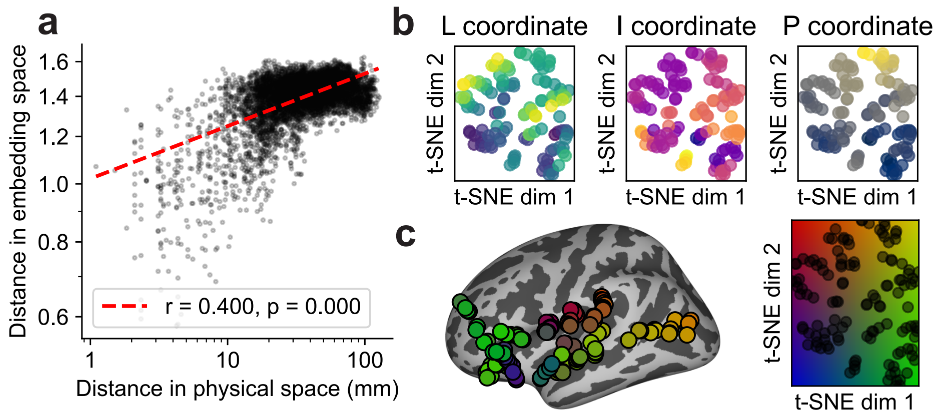

Figure 4: **Learned electrode embeddings correlate with rough anatomical and functional brain regions, recovering them in a pure data-driven way.** (a) Distance in the embedding space of the learned electrode embeddings is correlated with the distance in physical space in the brain, even though the spatial information was never made available during training. (b) Dimensionality reduction (t-SNE) reveals clustering of the electrode embeddings in the latent space, with the clusters generally grouping together according to the coordinate in the physical space. (c) Visualization of the t-SNE reduction result on an inflated map of the brain, which shows the anatomical locations of the embeddings. The results are shown for a representative Subject 3.

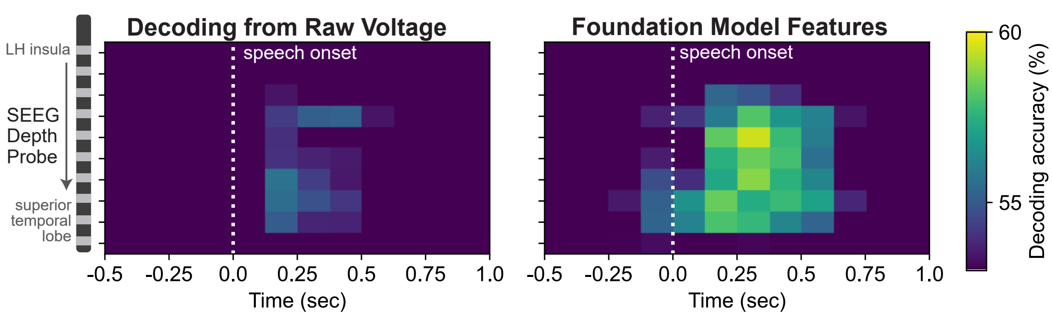

Figure 5: **Foundation models enable effective functional mapping of brain regions.** Validation of the model on an open human intracranial electroencephalography data (StereoEEG). For a given StereoEEG depth probe, we simulate a language mapping setting where the patient engages in an experimental task with two conditions: speech processing and non-speech. Then, we use either raw voltage (left) or features from our frozen foundation model (right) to decode GPT2 surprisal, as an indication of language processing, from every 125 ms timebin and every contact on the probe. The resulting model enabled stronger decoding of speech onset events than the raw voltage inputs. The probe spans multiple locations in the patient's brain, enabling localization of the functional language processing region. Model: causal transformer, 5 layers, model hidden dimension 128, 4 attention heads per block, sampling rate 2048 Hz. Trained for 1000 steps with batch size 128 with Contrastive Neural Forecacsting, using the Muon optimizer, learning rate 0.003, no weight decay.

as well as the language-selective parts of the superior temporal lobe, which suggests that the model rediscovered the gross anatomical layout of the brain.

Mapping function of brain regions with the foundation model features Next, we validate a practical application of our foundation model for functional brain mapping in a clinical setting. A standard protocol in neurosurgery involves identifying brain regions involved in critical functions, such as language processing, to guide tissue resection [Aron et al., 2021]. Traditionally, this relies on visually inspecting raw intracranial recordings for stimulus-locked activity differences, which may be subtle or ambiguous.

We propose using features extracted from our pretrained model to improve the clarity of this mapping. In Figure 5, we compare the decodability of one of our features connected with language processing (GPT-2 surprisal) across time and electrode contacts using raw voltage versus foundation model representations. The model-derived features yield a much sharper spatial and temporal decoding profile, revealing a more localized and time-locked peak in language-related activity along the probe. This result demonstrates that we can enhance functional mapping by amplifying task-relevant signals that may be difficult to detect in the raw data.

5 Discussion

Our results show that Contrastive Neural Forecasting (CNF) is a viable and scalable framework for learning population-level representations of human intracranial neural activity directly from raw voltage data. By forgoing handcrafted features and instead predicting future neural states in latent space using a contrastive objective, CNF avoids several limitations of traditional approaches, chiefly overfitting to noise and inflexibility in modeling multimodal dynamics. CNF-1 achieves state-of-the-art decoding performance across a diverse suite of language and vision tasks.

One striking finding of our work is the effectiveness of learned electrode embeddings in the absence of spatial coordinate information, reaffirming findings by Azabou et al. [2023]. Not only does this challenge prevailing assumptions in neural modeling, but it also suggests that useful structural priors can emerge from data alone when trained at scale, opening new opportunities for interpretability in foundation models of brain activity. Future work will examine the possibility of delineating functional and/or anatomical brain regions [Glasser et al., 2016] based solely on the activity statistics using foundation models such as CNF-1.

Limitations Our work has several limitations and directions for future research. First, our model outperforms linear baselines by only a small amount, and there is clearly room to grow. We anticipate that training on datasets beyond the BrainTreebank, as well as incremental architecture and training process improvements will greatly enhance the performance of our models. In addition, future work may explore multimodal extensions that incorporate neural data with information about the sensory inputs such as the viewed video. This can be achieved by incorporating CLIP representations of the visual inputs and/or wav2vec or other audio representations (this data is available in datasets such as BrainTreebank, but not used in this work).

More broadly, we view CNF as part of an emerging class of tools that treat the brain as a sequence-generating system that is amenable to the same powerful modeling techniques that have revolutionized NLP and vision. In this framing, iEEG signals become the neural analogue of text or pixels: high-dimensional, temporally structured data with rich latent dynamics.

Broader impacts Importantly, our results support the broader vision of brain foundation models (BFMs): pretraining once on large-scale observational recordings and reusing these representations for a wide range of downstream clinical and scientific applications [Zhou et al., 2025]. For example, we show that CNF-1 can enable real-time functional brain mapping, which could be used in clinical settings such as operating rooms during brain resection surgeries [Richardson, 2022] to define surgical and non-surgical targets. Thus, we anticipate this approach could accelerate workflows in neurosurgery, diagnosis, and closed-loop brain-computer interfaces. The usecases of foundation models should be strictly vetted to adhere to the ethical regulations, especially in the medical usecases and when involved in decision making impacting human lives [Gordon and Seth, 2024].

Beyond the clinic, foundation models like CNF-1 offer exciting opportunities in basic neuroscience. By unifying single-channel and population-level representations in a single model, CNF-1 can help researchers probe the functional roles of specific brain regions, and simulate the evolution of neural dynamics under different conditions, and generate new hypotheses to be tested in vivo based on the findings in the foundation models (inception loops; Wang et al. [2025], Walker et al. [2019]). Moreover, as brain foundation models grow larger and more expressive, they may serve as computational proxies for in silico experimentation.

Taken together, CNF represents a step toward a general-purpose framework for modeling brain dynamics, supporting the development of robust, scalable, and clinically useful brain-computer interfaces and tools in neuroscience and medicine.

Acknowledgements

This work has been supported by ONR award N00014-19-1-2584, by NSF-CISE award IIS-2151077 under the Robust Intelligence program, by the ARO-MURI award W911NF-23-1-0277, by the Simons Foundation SCGB program 1181110, and the K. Lisa Yang ICoN Center.

References

Antonis Antoniades, Yiyi Yu, Joseph Canzano, William Wang, and Spencer LaVere Smith. Neuroformer: Multimodal and Multitask Generative Pretraining for Brain Data, March 2024.

Olivier Aron, Jacques Jonas, Sophie Colnat-Coulbois, and Louis Maillard. Language mapping using stereo electroencephalography: A review and expert opinion. *Frontiers in Human Neuroscience*, 15:619521, 2021. doi: 10.3389/fnhum.2021.619521. URL `https://doi.org/10.3389/fnhum.2021.619521`.

Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael J. Mendelson, Blake Richards, Matthew G. Perich, Guillaume Lajoie, and Eva L. Dyer. A Unified, Scalable Framework for Neural Population Decoding, October 2023.

Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

György Buzsáki, Costas A Anastassiou, and Christof Koch. The origin of extracellular fields and currents-eeg, ecog, lfp and spikes. *Nature Reviews Neuroscience*, 13(6):407–420, 2012.

Donghong Cai, Junru Chen, Yang Yang, Teng Liu, and Yafeng Li. MBrain: A Multi-channel Self-Supervised Learning Framework for Brain Signals, June 2023.

Geeling Chau, Christopher Wang, Sabera Talukder, Vighnesh Subramaniam, Saraswati Soedarmadji, Yisong Yue, Boris Katz, and Andrei Barbu. Population Transformer: Learning Population-level Representations of Neural Activity, October 2024.

Zijian Dong, Ruilin Li, Yilei Wu, Thuan Tinh Nguyen, Joanna Su Xian Chong, Fang Ji, Nathanael Ren Jie Tong, Christopher Li Hsian Chen, and Juan Helen Zhou. Brain-jepa: Brain dynamics foundation model with gradient positioning and spatiotemporal masking, 2024. URL `https://arxiv.org/abs/2409.19407`.

Matthew F. Glasser, Timothy S. Coalson, Emma C. Robinson, et al. A multi-modal parcellation of human cerebral cortex. *Nature*, 536:171–178, 2016. doi: 10.1038/nature18933. URL `https://doi.org/10.1038/nature18933`.

Emma C. Gordon and Anil K. Seth. Ethical considerations for the use of brain–computer interfaces for cognitive enhancement. *PLOS Biology*, 22(10):e3002899, 2024. doi: 10.1371/journal.pbio.3002899. URL `https://doi.org/10.1371/journal.pbio.3002899`.

Joel Herron, Vahe Kremen, John D. Simeral, et al. The convergence of neuromodulation and brain–computer interfaces. *Nature Reviews Bioengineering*, 2:628–630, 2024. doi: 10.1038/s44222-024-00187-0. URL `https://doi.org/10.1038/s44222-024-00187-0`.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL `https://kellerjordan.github.io/posts/muon/`.

Tim Large, Yang Liu, Minyoung Huh, Hyojin Bahng, Phillip Isola, and Jeremy Bernstein. Scalable optimization in the modular norm. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 73501–73548. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/8629b0fff229b8a27efb1422e990605f-Paper-Conference.pdf`.

Trung Le and Eli Shlizerman. STNDT: Modeling Neural Population Activity with a Spatiotemporal Transformer, June 2022.

Ran Liu, Mehdi Azabou, Max Dabagia, Jingyun Xiao, and Eva L. Dyer. Seeing the forest and the tree: Building representations of both individual and collective dynamics with transformers, October 2022.

Martha J Morrell. Responsive cortical stimulation for the treatment of medically intractable partial epilepsy. *Neurology*, 77(13):1295–1304, 2011.

Norbert Noury, Joerg F Hipp, and Markus Siegel. Physiological processes non-linearly affect electrophysiological recordings. *eNeuro*, 3(4):ENEURO.0191–16.2016, 2016.

Josef Parvizi and Sabine Kastner. Promises and limitations of human intracranial electroencephalography. *Nature Neuroscience*, 15(2):264–272, 2012.

Francisco Pereira, Brian Lou, Brian Pritchett, et al. Toward a universal decoder of linguistic meaning from brain activation. *Nature Communications*, 9:963, 2018. doi: 10.1038/s41467-018-03068-4. URL `https://doi.org/10.1038/s41467-018-03068-4`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL `https://arxiv.org/abs/2103.00020`.

R. Mark Richardson. Closed-loop brain stimulation and paradigm shifts in epilepsy surgery. *Neurologic Clinics*, 40(2):355–373, 2022. doi: 10.1016/j.ncl.2021.12.002. URL `https://doi.org/10.1016/j.ncl.2021.12.002`.

Matthias Schurz, Joaquim Radua, Markus Aichhorn, Fabio Richlan, and Josef Perner. Fractionating theory of mind: A meta-analysis of functional brain imaging studies. *Neuroscience & Biobehavioral Reviews*, 42:9–34, 2014.

Terrence J Sejnowski, Patricia S Churchland, and J Anthony Movshon. Putting big data to good use in neuroscience. *Nature Neuroscience*, 17(11):1440–1441, 2014.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL `https://arxiv.org/abs/1807.03748`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL `https://arxiv.org/abs/1706.03762`.

Ankit Vishnubhotla, Charlotte Loh, Liam Paninski, Akash Srivastava, and Cole Hurwitz. Towards robust and generalizable representations of extracellular data using contrastive learning. *bioRxiv*, 2023. doi: 10.1101/2023.10.30.564831. URL `https://doi.org/10.1101/2023.10.30.564831`. Preprint, not peer reviewed.

E. Y. Walker, F. H. Sinz, E. Cobos, et al. Inception loops discover what excites neurons most using deep predictive models. *Nature Neuroscience*, 22:2060–2065, 2019. doi: 10.1038/s41593-019-0517-x. URL `https://doi.org/10.1038/s41593-019-0517-x`.

Christopher Wang, Vighnesh Subramaniam, Adam Uri Yaari, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. BrainBERT: Self-supervised representation learning for intracranial recordings, February 2023.

Christopher Wang, Adam Uri Yaari, Aaditya K Singh, Vighnesh Subramaniam, Dana Rosenfarb, Jan DeWitt, Pranav Misra, Joseph R. Madsen, Scellig Stone, Gabriel Kreiman, Boris Katz, Ignacio Cases, and Andrei Barbu. Brain treebank: Large-scale intracranial recordings from naturalistic language stimuli, 2024. URL `https://arxiv.org/abs/2411.08343`.

Evelyn Y. Wang, Patrick G. Fahey, Zheng Ding, et al. Foundation model of neural activity predicts response to new stimulus types. *Nature*, 640:470–477, 2025. doi: 10.1038/s41586-025-08829-y. URL `https://doi.org/10.1038/s41586-025-08829-y`.

Joel Ye, Jennifer L. Collinger, Leila Wehbe, and Robert Gaunt. Neural Data Transformer 2: Multi-context Pretraining for Neural Spiking Activity, September 2023.

Daoze Zhang, Zhizhang Yuan, Yang Yang, Junru Chen, Jingjing Wang, and Yafeng Li. Brant: Foundation Model for Intracranial Neural Signal. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

Xinliang Zhou, Chenyu Liu, Zhisheng Chen, Kun Wang, Yi Ding, Ziyu Jia, and Qingsong Wen. Brain foundation models: A survey on advancements in neural signal processing and brain discovery, 2025. URL `https://arxiv.org/abs/2503.00580`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We describe our pretraining approach, and then validate and show its performance and outline the applications of foundation models in the Results section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We included a limitations sections in our Discussion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not introduce any theory results or theorems, only showing empirical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our approach and architectures used for pretraining are described in the Approach and Experimental Setup sections in detail, as well as in the included Appendix; in addition, the code will be released upon publication.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We release the Github repository with all of the code required to train the models and reproduce the experiments, as well as the model weights, upon publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide information about our pretraining, architecture, and hyperparameters, and optimizer in the Experimental Setup section in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For our empirical results, we report standard error across cross-val folds, as well as across subjects in case of evaluating on data from multiple subjects.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: we specify the computational requirements when describing pretraining in our Experimental Setup section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We adhere to the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have included a broader impacts subsection in our Discussion section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Our paper describes ethical consideration with use of brain foundation models in the Discussion section.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The authors own all the assets in this paper, and credit with references whenever the openly available resources are used for the experiments or datasets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We document all of the code created for this paper in the comments and README files in the Github repository to be shared upon publication.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper doesn't involve any crowdsourcing for our experiments and does not perform new experiments with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We use a public dataset that is openly published and available on the internet to construct our benchmark and pretrain the models (BrainTreebank, https://braintreebank.dev). As such, we did not require any IRB approvals or equivalent to conduct our research.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We do not use LLMs as core components of our methods. One of our tasks is "GPT2 Surprisal", tasking the model with decoding the LLM negative log likelihood of the words in the dataset, however this feature was extracted from the sentences following standard protocol.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Decoding Tasks and Split Construction

Decoding Tasks The decoding benchmark used in this paper includes 14 decoding tasks derived from multimodal annotations in the BrainTreebank dataset [Wang et al., 2024]. These tasks span audio, vision, and language modalities. All tasks are cast as binary classification problems to ensure uniformity in evaluation across task types and models.

- **Scalar features** (e.g., GPT2 surprisal, pitch, volume): For each session, values are thresholded such that the top 25% of the distribution are labeled as the positive class, and the bottom 25% as the negative class. The middle 50% of values are excluded from training and evaluation to reduce ambiguity around class boundaries.
- **Categorical features** (e.g., part-of-speech, speaker identity): For each feature, a single target class is selected (typically the most frequent), and the task is defined as a one-vs-rest binary classification problem.

All tasks are aligned to word onsets. Neural data is segmented into 1-second windows starting at the onset of each word. Unless otherwise stated, all decoding experiments use these 1-second segments of neural activity as model inputs, and the corresponding BrainTreebank annotations as binary labels. See more information about the decoding tasks in the tables below.

Subj.	Age (yrs.)	# Electrodes	Movie	Recording time (hrs)	Used in benchmark
1	19	154	Thor: Ragnarok	1.83	x
			Fantastic Mr. Fox	1.75	
			The Martian	0.5	x
2	12	162	Venom	2.42	x
			Spider-Man: Homecoming	2.42	
			Guardians of the Galaxy	2.5	
			Guardians of the Galaxy 2	3	x
			Avengers: Infinity War	4.33	
			Black Panther	1.75	
			Aquaman	3.42	
3	18	134	Cars 2	1.92	x
			Lord of the Rings 1	2.67	x
			Lord of the Rings 2 (extended edition)	3.92	
4	12	188	Incredibles	1.15	
			Shrek 3	1.68	x
			Megamind	2.43	x
5	6	156	Fantastic Mr. Fox	1.5	
6	9	164	Megamind	2.58	
			Toy Story	1.33	
			Coraline	1.83	
7	11	246	Cars 2	1.75	x
			Megamind	1.77	x
8	4.5	162	Sesame Street Episode	1.28	
9	16	106	Ant Man	2.28	
10	12	216	Cars 2	1.58	x
			Spider-Man: Far from Home	2.17	x

Table S1: **Subject statistics** Subjects in the BrainTreebank dataset, and the trials used in the benchmark tasks. Table adapted from Wang et al. [2023]. The second column shows the total number of electrodes. The average amount of recording data per subject is 4.3 (hrs).

#	Feature	Description	Benchmark Task
1	global_flow *(visual)*	A camera motion proxy. The maximal average dense optical flow vector magnitude	Same as above
2	local_flow *(visual)*	A large displacement proxy. The maximal optical flow vector magnitude	Same as above
3	volume *(auditory)*	Average root mean squared watts of the audio	Binary classification: low (0%-25%) vs high (75%-100%)
4	pitch *(auditory)*	Average pitch of the audio	Same as above
5	delta_volume *(auditory)*	The difference in average RMS of the 500ms windows pre- and post-word onset	Same as above
6	speech *(language)*	Whether any speech is present in the given time interval	Binary classification
7	onset *(language)*	Whether a new sentence starts in the interval, or there is no speech at all	Binary classification
8	gpt2_surprisal *(language)*	Negative-log transformed GPT-2 word probability (given preceding 20s of language context)	Binary classification: low (0%-25%) vs high (75%-100%)
9	word_length *(language)*	Word length (ms)	Same as above
10	word_gap *(language)*	Difference between previous word offset and current word onset (ms)	Same as above
11	word_index *(language)*	The word index in its context sentence	2-way classification: 0 (the first word in the sentence), or other (1)
12	word_head_pos *(language)*	The relative position (left/right) of the word's dependency tree head	Binary classification
13	word_part_speech *(language)*	The word Universal Part-of-Speech (UPOS) tag	2-way classification: verb (0), or other (1)
14	speaker *(multimodal)*	The movie character that speaks the given word.	2-way classification: most frequent speaker (0), or other (1)

Table S2: **Extracted visual, auditory, and language features used to create the evaluations.** For all classification tasks, the classes were rebalanced. The difference between local and global flow is that global is the averaged optical flow, with the average being taken over all optical flow vectors on the screen, whereas local is the largest individual optical flow vector on the screen. The table is adapted from Chau et al. [2024].

Train/Test Split Construction To probe model generalization under increasingly challenging conditions, we define the following split strategies:

- **Same Subject / Same Movie (SS/SM):** Training and testing data are drawn from the same subject and same movie (trial of recording). A contiguous 80/20 train-test split is applied, ensuring the training block precedes the test block to reduce temporal autocorrelation. Performance is computed via 5-fold cross-validation.
- **Same Subject / Different Movie (SS/DM):** Data is drawn from the same subject across two different movies. For the two movies selected for every subject for evaluation, both ways to split the pair into the train and test movie are used, and the resulting AUROC is averaged between the two splits.

BrainBERT-mini decoding experiments were run on the SS/SM split. CNF-1 (Contrastive Neural Forecasting) and Functional mapping analyses (i.e., the spatiotemporal decoding maps shown in Figure 5) evaluations were run on the SS/DM split.

We discard the data from electrodes which were labeled as corrupted by the BrainTreebank authors [Wang et al., 2024].

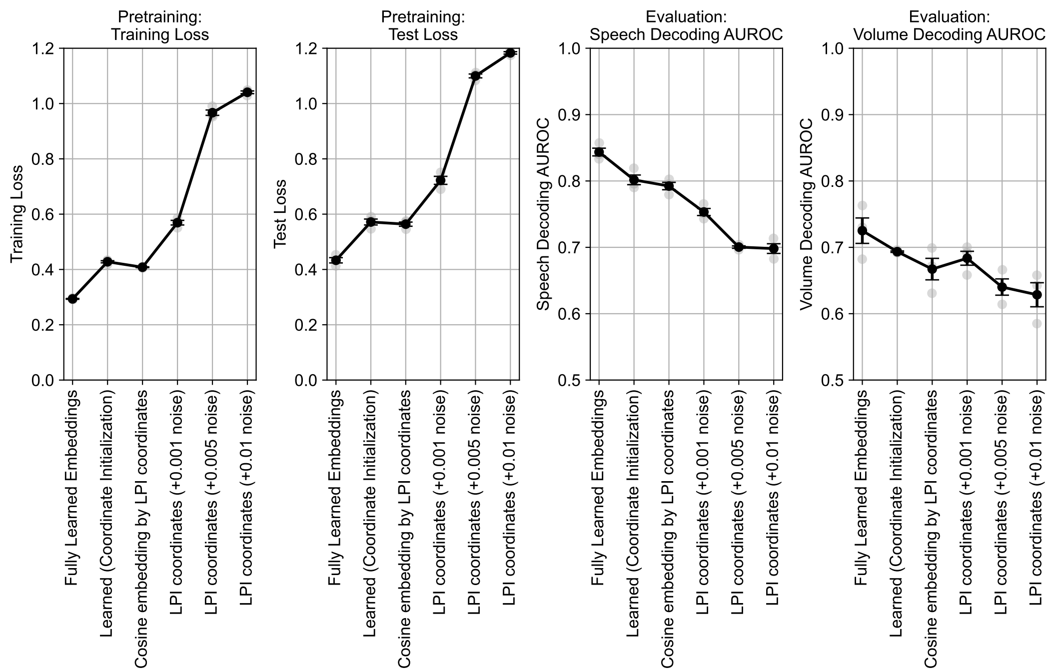

Figure S1: **Fully learned electrode embeddings perform the best in Contrastive Neural Forecasting across both pretraining and evaluation.** The two left graphs show the training and test loss during autoregressive pretraining, respectively. Fully learned embeddings outperform both traditional positional cosine embeddings with electrode LPI coordinates and the approach where the embeddings are initialized with the cosine embeddings but then are allowed to be updated during pretraining. Adding noise to the positional electrode embedding only increases the train and test pretraining error. The two right plots show the evaluation decoding AUROC (with frozen model weights), and demonstrate that the evaluation performance also decreases with increased pretraining loss. The error bars show the mean and s.e.m. across 3 random seeds. This experiment was performed in a model where the CNF objective was applied at the CLS token.

Functional mapping experiment To estimate the time course of information processing in the brain across space in a brain area (Figure 5), we used a sliding window of 125 ms across neural activity, in steps of 125 ms from -500 ms to $+1000$ ms relative to word onset. For each time bin and electrode, a separate linear decoder is trained for each task, either with raw voltage traces acting as features, or from the model features. The resulting decoding scores are averaged across cross-validation folds. The analysis was run specifically with subject 3, trial 0.

B Details on the Model and Pretraining

Pretraining hyperparameters All models were pretrained with learning rate 0.003 which we found works best across the range of different model sizes and architectures (which might be a feature of the Muon optimizer, for a discussion see Jordan et al. [2024], Large et al. [2024]). We use batch size 100, and for every electrode we use batch norm to normalize the input voltage traces across the batch and timesamples dimensions. We discard the data from electrodes which were labeled as corrupted by the BrainTreebank authors [Wang et al., 2024].

Learned electrode embeddings We found in our experiments that fully learned electrode embeddings resulted in lower pretraining loss and higher decoding performance compared to the traditional approach from prior work [Chau et al., 2024, Zhang et al., 2023] which provides cosine positional embeddings from the electrode physical coordinates in 3D space (Supplementary Figure S1).

C Comparison to baselines and previous methods

Linear For this evaluation, raw voltage traces sampled at 2048 Hz were taken from the BrainTreebank data, then line noise was removed at 60 ± 5 Hz and the 4 harmonics, and the resulting vectors of sampled features were fed as input to the linear regression. We found almost identical results when removing line noise or passing the data raw to the linear regression.

Linear (STFT) For this baseline evaluation, the features are the STFT of the raw signal with the following parameters (given that the sampling rate is 2048Hz):

- nperseg=256
- noverlap=0
- window=boxcar

After this step, the data turns into an array of arrays where first dimension is the time bin and the second dimension is the STFT result (a complex number); for the downstream regression, all of these features are concatenated together, with the real and imaginary parts of the complex features being split into two features each.

Linear (spectrogram) For this baseline evaluation, first the STFT of the raw voltage signal was taken as in the Linear (STFT) description, and then the absolute value of each complex number was taken to obtain the final real number features for each example.

BrainBERT For this evaluation, the BrainTreebank data was Laplacian rereferenced (as described in the original BrainBERT paper by Wang et al. [2023]), with line noise removed, and then passed into the BrainBERT model as provided by Wang et al. [2023]. The output features were concatenated and used as input to the linear regression. For the electrodes which could not be Laplacian rereferenced, non-rereferenced data was inputted into BrainBERT. The BrainBERT model was frozen and only the final linear regression layer was fine tuned, in order to compare the quality of features generated by the foundation model.

For all linear regression, we used the sklearn package, class LinearRegression, with the tolerance parameter set as 0.001. In all cases, the features were first normalized using the sklearn StandardScaler. We found that it helps with convergence and often produces higher regression values for the baselines.

Population Transformer For Population Transformer, we followed the implementation and used the weights from [Chau et al., 2024]. The fine-tuning protocol is taken to be directly the same as in the authors' original paper (including linear rate, number of epochs, a factor of 10 between learning rates of the linear output layer vs the transformer blocks, etc). We found that frozen Population Transformer's performance was almost always at chance and that pretraining through the whole model was necessary to achieve comparable performance to other methods.