BACE: Behavior-Adaptive Connectivity Estimation from Multi-Region Neural Recordings

Mehrnaz Asadi[1], **Sina Javadzadeh**[1], **Rahil Soroushmojdehi**[1]
S. Alireza Seyyed Mousavi[1], **Terence D. Sanger**[1,2]
[1]University of California, Irvine
[2]Rady Children's Health Orange County
Mehrnaza@uci.edu

Abstract

Understanding how distributed brain regions coordinate during behavior requires models that are both predictive and interpretable. We introduce Behavior-Adaptive Connectivity Estimation (BACE), an end-to-end framework for learning behavior-conditioned directed effective connectivity from multi-region intracranial local field potentials (LFPs). BACE first encodes within-region dynamics with region-specific temporal encoders, then applies a learned adjacency matrix selected for each behavioral context, and finally forecasts future neural activity through a graph-conditioned autoregressive decoder. This framework yields explicit region-level connectivity matrices whose edges are tied to predictive dynamics rather than post-hoc explanation. On synthetic time series with known directed graphs, BACE recovers the ground-truth edge structure from forecasting alone. We then evaluate BACE on human deep-brain LFP recordings from three participants performing a cued reaching task. Across participants, BACE achieves strong neural forecasting while producing compact, behavior-specific directed graphs across task contexts. Reliability analyses further support the stability of the inferred connectivity patterns. Together, these results position BACE as a practical framework for estimating behavior-conditioned effective connectivity from high-dimensional intracranial recordings, enabling interpretable hypotheses about how deep-brain regions interact during behavior.

1 Introduction

Connectivity in neural time series is often modeled as a single graph between populations, without explicitly capturing how it varies across behavioral context. Connectivity is commonly described as structural (anatomical pathways), functional (statistical dependencies), and effective (directed influence) [17, 16]. However, these are not equally suited for this setting: structural connectivity is fixed, while functional connectivity captures undirected relationships typically estimated over aggregated time periods. As a result, methods for estimating directed connectivity that varies across behavioral conditions remain limited [36, 15].

Graph-based models are a common approach for representing connectivity in neural data, where nodes correspond to brain regions and edges encode relationships between them. This framework enables quantitative analysis of network organization, including modularity, hubs, and task-dependent variation [2, 11]. In practice, these graphs are typically constructed using pairwise correlations or related statistical measures computed over extended time periods. Consequently, they are often undirected and reflect average statistical dependencies between regions, rather than directed connectivity. This limits their ability to capture predictive dependencies and to provide interpretable representations of connectivity across contexts.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

These limitations are particularly pronounced in intracranial local field potential (LFP) recordings across multiple deep-brain regions [9, 13, 26, 27]. Multi-region LFP data introduce additional modeling challenges: (i) each anatomical region is represented by multiple micro-contacts that must be aggregated into a coherent regional representation; (ii) neural activity evolves across well-defined behavioral epochs; and (iii) spatial organization is irregular and non-Euclidean [7]. These characteristics make it necessary to estimate connectivity at the level of regions and behavioral contexts, rather than at individual channels or over aggregated time periods.

Machine learning approaches have been used to infer connectivity and interregional interactions directly from neural time series [38, 32], including through forecasting-based objectives that capture predictive dependencies in temporal data [33]. Graph-based architectures further incorporate spatial structure, enabling data-driven estimation of connectivity. Across these approaches, several challenges remain: many operate at the channel level, producing high-dimensional graphs that are difficult to interpret; others assume static or task-agnostic connectivity; and many represent dependencies implicitly through latent variables or attention mechanisms without exposing explicit, directed connectivity. As a result, methods that learn compact, directed, and context-dependent connectivity at the region level remain limited, particularly in settings where behaviorally meaningful structure must be identified.

In this work, we adopt a predictive view of connectivity: we define effective connectivity as the directed structure that improves prediction of future neural activity under a given behavioral context. This formulation links connectivity estimation directly to temporal prediction, rather than post-hoc analysis of correlations. Building on this principle, we introduce **Behavior-Adaptive Connectivity Estimation (BACE)**, an end-to-end framework that learns context-specific, directed inter-regional connectivity from multi-region intracranial LFP recordings. BACE aggregates multiple micro-contacts within each anatomical region using independent temporal encoders to produce region-level representations, and learns a separate adjacency matrix for each behavioral context. These adjacency matrices are coupled to neural dynamics through a forecasting objective, ensuring that the learned connectivity reflects predictive dependencies.

We evaluate BACE on both synthetic and real datasets. On synthetic multivariate time series with known ground-truth graphs, BACE accurately recovers directed connectivity while maintaining competitive forecasting performance. Applied to human subcortical recordings from N regions during a cued reaching task, BACE produces explicit $N \times N$ connectivity matrices for each behavioral context, enabling direct comparison of connectivity patterns across conditions. Beyond predictive accuracy, we assess reproducibility and statistical reliability, treating learned adjacency matrices as explicit model outputs rather than latent representations.

Our contributions are summarized as follows:

- **Context-dependent connectivity estimation.** We introduce a framework that explicitly models how directed connectivity varies across behavioral contexts, rather than assuming a fixed or task-agnostic graph.
- **Forecasting-based structure identification.** We show that predictive objectives can recover meaningful directed connectivity from neural data, aligning connectivity estimation with temporal prediction.
- **Region-level graph representation.** By aggregating micro-contact signals into region-level embeddings, BACE produces compact adjacency matrices that are directly interpretable and suitable for downstream analysis.
- **Application to invasive neural recordings.** We demonstrate that BACE yields reproducible and behavior-dependent connectivity estimates in multi-region human LFP data.

2 Related Work

Modeling neural population dynamics. Neural population dynamics have been modeled using a wide range of approaches, from classical linear dynamical systems and Gaussian process factor models [12, 54, 18] to modern deep learning methods. These include recurrent neural networks [19, 37], variational autoencoders [56], neural ODEs [29], and Transformers adapted for neural time series [52, 31, 20]. Predictive objectives have proven effective for learning latent representations aligned with behavior [41, 21]. Despite their success in uncovering latent dynamics, these methods

are generally not designed to yield the explicit, interpretable, and behavior-conditioned connectivity graphs that are the primary focus of this work.

Recurrent models of neural interactions. Data-constrained recurrent networks offer another route to interpreting neural interactions. CURBD fits neuron-level recurrent networks to multi-region recordings and decomposes the fitted recurrent inputs into directed currents from source regions [38]. FacRNN factorizes low-rank recurrent connectivity into components associated with latent dynamical groups, encouraging independence between groups [32]. Beiran and Litwin-Kumar [3] study how a known connectome and partial activity recordings constrain neural predictions, and show that fitting recurrent weights to recorded activity alone need not recover the generating connectivity. BACE provides a complementary approach by directly learning a compact regional adjacency for each behavioral context while sharing all other model parameters across contexts.

Dynamic graph learning for neural data. Graph neural networks (GNNs) have become a common framework for modeling brain activity across modalities. In fMRI, architectures such as STAGIN leverage spatiotemporal attention to learn dynamic representations of functional connectivity for classification tasks [28]. More recent models combine temporal sequence modeling with graph learning, such as GraphS4mer, which integrates structured state-space models with adaptive graph representations [48]. Similarly, dynamic GNN approaches for EEG, such as EvoBrain, explicitly model time-varying graph structure to capture evolving brain networks [30]. While these methods demonstrate the importance of dynamic graph representations, they are typically evaluated on classification or decoding tasks and often represent connectivity implicitly or in an undirected manner, rather than producing directed adjacency matrices explicitly conditioned on behavioral contexts.

Graph structure learning from time series. Graph structure learning (GSL) methods aim to jointly infer connectivity and temporal dynamics from multivariate time series [23]. In neuroscience, this is particularly relevant because the underlying connectivity is not directly observed. AMAG exemplifies this approach by learning a channel-level adjacency matrix jointly with forecasting neural activity, demonstrating that predictive objectives can recover connectivity from data [33]. More broadly, connections between forecasting and connectivity estimation have been studied in the context of Granger causality, where directed relationships are defined through predictive dependencies [39]. Recent neural extensions leverage deep sequence models to capture nonlinear dependencies while enforcing sparsity to extract directed connectivity [49, 39]. However, these methods typically operate at the level of individual channels and do not explicitly consider behavioral context in modeling.

Interpretability of learned neural connectivity. Interpretability in graph-based neural models has been studied primarily in the context of classification tasks. Approaches such as BrainGNN identify salient regions using pooling and sparsity regularization [34], while causal subgraph discovery methods aim to isolate task-relevant connectivity patterns [55]. Post-hoc explanation methods, including GNNExplainer [53], provide edge-level importance scores but do not directly learn interpretable connectivity as a primary model output. In contrast, our approach treats connectivity as an explicit object of interest, producing compact, directed adjacency matrices that can be compared across behavioral contexts.

3 Methods

3.1 Problem setup

Many neural recordings measure activity from multiple spatially organized units of analysis, such as anatomical regions, electrode groups, channel groups, or neural populations. We refer to these units as graph nodes. Let N denote the number of nodes, indexed by $i, j \in \{1, \ldots, N\}$. Each node i contains C_i recorded channels or contacts, and each sample consists of a recent neural activity window from all nodes.

Each activity window is associated with a behavioral context k. The goal is to forecast future neural activity while learning a directed adjacency matrix $\mathbf{A}^{(k)} \in \mathbb{R}^{N \times N}$ for each context. Formally, BACE learns $f_\theta(\mathbf{X}, k) \mapsto (\hat{\mathbf{Y}}, \mathbf{A}^{(k)})$, where $\mathbf{X}$ is the past neural activity window, $\hat{\mathbf{Y}}$ is the predicted future activity, and $\mathbf{A}^{(k)}$ is learned end-to-end as part of the forecasting model. Each adjacency is shared across examples from the same context.

An entry $A_{ij}^{(k)}$ represents the directed influence of source node j on target node i during context k. We refer to $\mathbf{A}^{(k)}$ as predictive effective connectivity: a directed connectivity graph learned by forecasting future neural activity within a given behavioral context. These edges should not be interpreted as anatomical connections or as interventional causal effects.

A method for this setting should aggregate fine-grained contacts into interpretable node-level dynamics, learn directed interactions that can vary across behavioral contexts, tie graph estimation to future neural dynamics, and produce compact adjacency matrices that can be inspected and compared across contexts. BACE addresses these goals with the encoder–graph–decoder architecture described below.

3.2 BACE overview

Figure 1 summarizes BACE. The model contains four components: regional temporal encoders, a context-specific graph learner, a graph projector, and an autoregressive forecasting head. BACE shares the encoder, graph projector, and forecasting head across behavioral contexts; only the adjacency matrix $\mathbf{A}^{(k)}$ depends on the context label. Thus, context-dependent differences are represented explicitly in the learned connectivity matrices rather than by training separate models for each context.

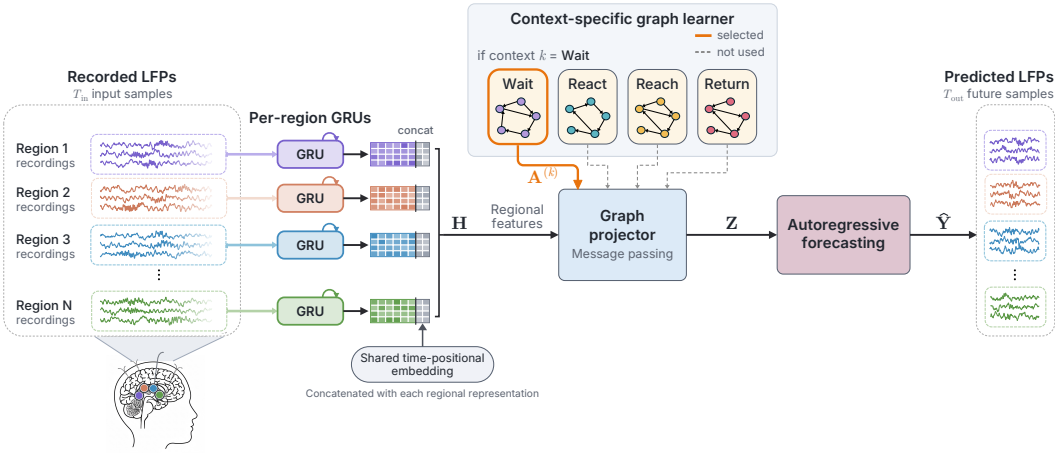


Figure 1: **BACE architecture.** Per-region temporal encoders (independent GRUs) extract local dynamics from multichannel input windows. The behavioral-context label selects a learned directed adjacency $\mathbf{A}^{(k)}$, which mixes regional features in the graph projector and regional predictions during autoregressive feedback. The forecasting head uses temporal attention and a linear readout to produce channel-space residual predictions. This end-to-end design links forecasting directly to explicit, behavior-specific connectivity estimates.

Regional temporal encoder. Neural signals recorded from multiple micro-contacts within the same anatomical region often exhibit shared local dynamics and oscillatory structure [45]. For node i, we write the input window as $\mathbf{X}_i \in \mathbb{R}^{C_i \times T_{\text{in}}}$, where C_i is the number of channels and T_{in} is the input window length in samples. BACE uses a separate GRU encoder [10] for each node, $\mathbf{H}_i = \text{GRU}_i(\mathbf{X}_i^\top)$, with time as the recurrent dimension and channels as features at each step, yielding $\mathbf{H}_i \in \mathbb{R}^{T_{\text{in}} \times d_{\text{hidden}}}$. Encoder outputs are layer-normalized and concatenated with a lightweight time-positional embedding, yielding $\mathbf{H} \in \mathbb{R}^{N \times T_{\text{in}} \times d}$. The GRU parameters are not shared across nodes, so within-node temporal dynamics are modeled independently before any cross-node mixing occurs. Inter-node information exchange is mediated only by the learned adjacency $\mathbf{A}^{(k)}$, through graph projection and autoregressive graph feedback. A held-out source-region perturbation analysis supports this architectural property (Appendix B.2).

Context-specific graph learner. BACE maintains one directed adjacency matrix $\mathbf{A}^{(k)}$ for each behavioral context k. During training and inference, the behavioral context label selects the corresponding adjacency matrix; it does not select a separate encoder, projector, or forecasting head.

Rows of each adjacency matrix are normalized to have unit ℓ_1 norm, i.e. $\sum_j |A_{ij}^{(k)}| = 1$ for every i, so each row can be interpreted as the relative pattern of incoming predictive influences onto target

node i. Diagonal entries are fixed to zero. Adjacencies are initialized from Pearson correlations between region-averaged signals computed on training trials only, followed by diagonal masking and row normalization.

Graph projector. Given $\mathbf{A}^{(k)}$, the graph projector introduces directed cross-node coupling. For node i and input time index τ,

$$\mathbf{Z}_{i,\tau} = W_{\text{self}}\mathbf{H}_{i,\tau} + \sum_{j=1}^{N} A_{ij}^{(k)} W_{\text{neigh}}\mathbf{H}_{j,\tau}.$$

Here, $\mathbf{H}_{i,\tau} \in \mathbb{R}^d$ is the encoded representation of node i, and W_{self} and W_{neigh} are learnable linear maps. The first term preserves self-dynamics, while the second aggregates transformed embeddings from source nodes j according to their directed influence on target node i. W_{neigh} transforms features identically across node pairs and contexts, whereas $A_{ij}^{(k)}$ supplies the explicit source–target-specific and context-specific routing weight. The adjacency mixes temporally encoded representations; it is not a linear state-transition matrix. We denote the activated representation by $\mathbf{Z}_{i,\tau}^{\text{act}} = \text{LeakyReLU}(\mathbf{Z}_{i,\tau})$. These activated regional representations are stacked into decoder memory $\mathbf{Z} = \text{Stack}_{i,\tau}(\mathbf{Z}_{i,\tau}^{\text{act}}) \in \mathbb{R}^{N\times T_{\text{in}}\times d_{\text{proj}}}$.

Autoregressive forecasting head. The forecasting head predicts T_{out} future samples autoregressively from the decoder memory $\mathbf{Z}$. Let $\mathbf{X}_{\text{last}} \in \mathbb{R}^{N\times C}$ be the last observed sample, where C is the common channel count per node within a participant. We initialize $\hat{\mathbf{Y}}_0 = \mathbf{X}_{\text{last}}$ and write $\hat{\mathbf{y}}_{i,t}$ and $\tilde{\mathbf{y}}_{i,t}$ for node i's rows of $\hat{\mathbf{Y}}_t$ and $\tilde{\mathbf{Y}}_t$, respectively. At each forecast step $t = 1, \ldots, T_{\text{out}}$, the previous prediction is first refined through graph feedback, $\tilde{\mathbf{Y}}_{t-1} = \hat{\mathbf{Y}}_{t-1} + \alpha_g \mathbf{A}^{(k)}\hat{\mathbf{Y}}_{t-1}$, where α_g sets the feedback strength. For each node, the previous decoder state $\mathbf{s}_{i,t-1}$ then supplies a query over that node's memory only, giving a temporal summary $\mathbf{c}_{i,t} = \sum_{\tau=1}^{T_{\text{in}}} a_{i,t,\tau} W_v \mathbf{Z}_{i,\tau}^{\text{act}}$, with weights $a_{i,t,\tau}$ normalized over input time. Finally, the decoder updates its state, and a linear readout maps it directly to a channel-space residual:

$$\mathbf{s}_{i,t} = \text{Dec}\big(\mathbf{s}_{i,t-1}, [\mathbf{c}_{i,t}; \tilde{\mathbf{y}}_{i,t-1}]\big), \qquad \Delta\hat{\mathbf{y}}_{i,t} = W_{\text{out}}\mathbf{s}_{i,t}, \qquad \hat{\mathbf{y}}_{i,t} = \hat{\mathbf{y}}_{i,t-1} + \Delta\hat{\mathbf{y}}_{i,t}.$$

Each node's output maps only to its own channels, and the loss is computed directly on these channel-space predictions.

Training and optimization. BACE is trained end-to-end with a forecasting objective over the next T_{out} samples. The objective combines weighted mean-squared prediction error with temporal regularizers:

$$\begin{aligned}\mathcal{L} = \frac{1}{T_{\text{out}}} \sum_{t=1}^{T_{\text{out}}} w_t \|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\|_2^2 + \lambda_{\text{cont}}\|\hat{\mathbf{Y}}_1 - \mathbf{X}_{\text{last}}\|_2^2 \\ + \lambda_{\text{vel}}\|\Delta\hat{\mathbf{Y}}_1 - \Delta\mathbf{Y}_1\|_2^2 + \lambda_{\text{curv}}\|\Delta^2\hat{\mathbf{Y}} - \Delta^2\mathbf{Y}\|_2^2.\end{aligned}$$

Here, $\mathbf{Y}_t$ is the ground-truth future activity, and w_t weights prediction errors across the forecast horizon. We define $\Delta\hat{\mathbf{Y}}_t = \hat{\mathbf{Y}}_t - \hat{\mathbf{Y}}_{t-1}$, with $\hat{\mathbf{Y}}_0 = \mathbf{Y}_0 = \mathbf{X}_{\text{last}}$, and define target differences analogously. The continuity term penalizes the first predicted jump from the observed window, the velocity term matches the first predicted increment to the observed increment, and the curvature term matches second differences over the forecast horizon.

All model parameters, including the context-specific adjacency matrices, are optimized jointly with Adam. We use gradient clipping and early stopping based on validation loss, retaining the checkpoint with the lowest validation loss for test evaluation. Additional model and optimization details are provided in Appendix A.1.

4 Experiments and results

4.1 Synthetic validation of graph recovery under known ground truth

We use controlled synthetic time series to test whether BACE can recover a known directed interaction graph from a forecasting objective alone. These datasets are designed to match the scale of our neural

recordings (small region-level graphs, hundreds of trials, and short segmented windows) but not the exact anatomical configuration or biological complexity of intracranial LFP. This experiment isolates the graph-learning mechanism: can the adjacency learned by BACE recover the known directed edge structure used to generate the data, even though the model is trained only to forecast future signals?

Dataset design. We generated two synthetic suites, each containing four regimes $\mathcal{D}_1$–$\mathcal{D}_4$ with distinct ground-truth adjacency matrices. In the structured suite, each regime used a sparse directed graph with the same row degree but different edge placements, producing visually distinct connectivity patterns. In the stochastic non-Gaussian suite, time series were generated from linear non-Gaussian dynamics following prior work on structure learning from neural time series [33, 6, 5, 44]. The governing dynamics take the form $\mathbf{X}_t = \mathbf{X}_{t-1} + G(\mathbf{X}_{t-1}, \mathbf{A}) + \boldsymbol{\mu}_t$, where $G(\mathbf{X}_{t-1}, \mathbf{A}) = (-\lambda \mathbf{I} + \gamma \mathbf{A})\mathbf{X}_{t-1}$ combines linear leak and adjacency-mediated coupling. The noise $\boldsymbol{\mu}_t$ is autoregressive with uniform innovations, making it colored and non-Gaussian. Since this suite uses linear dynamics, we evaluate VAR/Granger as a matched baseline.

Synthetic validation. BACE was trained using only the forecasting loss; the ground-truth adjacencies were used only for evaluation. Graph recovery was assessed with two metrics: (i) F1 @ k_{row}, which measures whether the top-k entries in each row match the ground-truth nonzero edges, and (ii) Pearson correlation between learned and ground-truth edge weights. Across the structured suite, BACE achieved perfect support recovery (mean F1 @ k_{row}=1.00) with mean correlation 0.89. Across the stochastic non-Gaussian suite, recovery remained strong, with mean F1 @ k_{row}=0.98 and mean correlation 0.90 (per-regime values in Fig. 2). As a positive-control sanity check, a matched VAR/Granger estimator nearly perfectly recovered the stochastic-suite graphs (F1 @ k_{row}=1.00, correlation 0.99). These results show that BACE can recover planted directed structure in the evaluated synthetic systems.

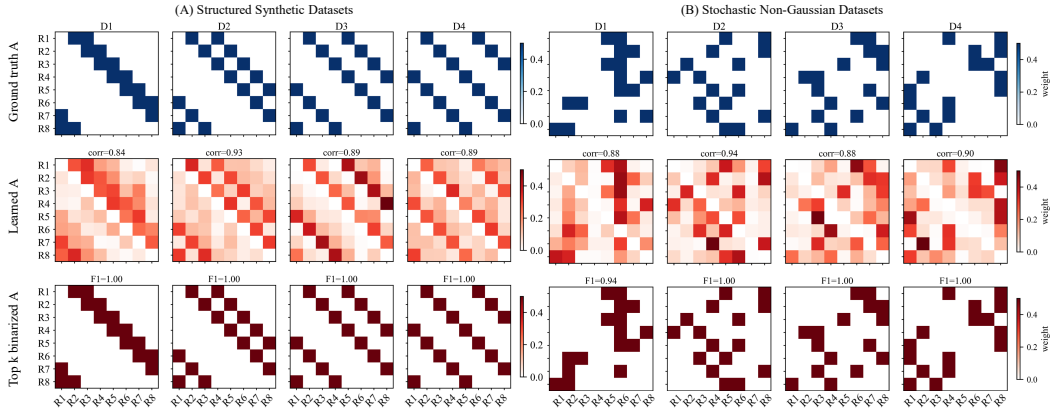


Figure 2: **Adjacency recovery on synthetic datasets.** (**A**) Structured suite; (**B**) stochastic non-Gaussian suite. Columns correspond to the four dataset regimes $\mathcal{D}_1$–$\mathcal{D}_4$. Rows show, from top to bottom: the ground-truth adjacency $\mathbf{A}$, BACE's learned adjacency $\hat{\mathbf{A}}$, and the top-k binarization of $\hat{\mathbf{A}}$, with k set to the ground-truth row degree (k=2). Values above the learned-adjacency panels report Pearson correlation with the ground-truth weights; values above the binarized panels report F1 @ k_{row}.

Graph recovery under nonlinear dynamics. To test whether recovery extends to nonlinear interactions, we additionally evaluated an eight-node system with four context-specific graphs, two planted parents per target, and centered $\tanh^2$ inter-regional coupling (Appendix A.2). Across three runs, BACE achieved F1 @ $k_{\text{row}} = 1.000 \pm 0.000$ ($k = 2$), compared with 0.474 ± 0.018 for context-specific VAR(10), whose lag range included the true structural lag. Pearson correlation between absolute inferred and ground-truth edge weights was 0.989 ± 0.002 for BACE versus 0.381 ± 0.095 for VAR(10). These results extend graph recovery to a nonlinear setting in which BACE outperforms the tested linear comparator.

4.2 Evaluation on human LFP data

Dataset and preprocessing. We evaluated BACE on 3 human intracranial LFP datasets collected during DBS treatment as part of clinical care. Recordings were acquired during a cued reaching task and segmented into 4 behavioral contexts: Wait, the 400 ms pre-cue baseline; React, the 400 ms after the go cue before movement; Reach, the forward reach to the target; and Return, the movement back to the home button. Contacts were grouped into anatomically defined nodes, yielding 8–10 nodes per participant with 8–10 retained channels per node after cleaning. BACE was trained separately for each participant to accommodate differences in electrode coverage [see 22, for cross-subject aggregation]. Node indices are therefore not homologous across participants.

Signals were filtered from 1–50 Hz, common-average referenced, downsampled to 1 kHz, and cleaned by removing noisy channels and outlier trials. Each 400 ms context window was divided into sliding forecasting examples with $T_{\text{in}} = 100$, $T_{\text{out}} = 20$, and stride 20. Within each behavioral context, trials were divided into 70% training, 15% validation, and 15% test sets before sliding windows were generated, preventing leakage through overlapping windows; the same split was used for BACE, ablations, and baselines. For each participant, all analyzed trials were recorded within a single session, so reported results reflect generalization to held-out trials within that session. Participant-level recording and preprocessing details are provided in Appendix B.1.

Benchmark methods. To contextualize BACE's forecasting performance, we compare against a targeted set of baselines spanning the main modeling families relevant to multivariate neural time-series prediction: classical linear dynamics, recurrent neural networks, Transformer-based sequence models, and graph-based neural forecasting. As a classical effective-connectivity reference, we include a VAR/Granger model. From the recurrent family, we include a standard LSTM [19], together with a context-aware LSTM variant that receives the same behavioral context labels as BACE. From the Transformer family, we evaluate the Neural Data Transformer (NDT) [52], designed for neural sequence modeling with masked reconstruction objectives. For graph-based methods, we include DCRNN [35], which integrates graph diffusion over a fixed adjacency into recurrent updates; GWNet [51], which combines dilated causal convolutions with adaptive graph learning; and AMAG [33], which adaptively learns channel-level graphs for neural forecasting. We also include a context-aware AMAG variant that receives the same behavioral context labels as BACE. These baselines provide reference points for predictive accuracy, while BACE is designed to additionally return compact, behavior-indexed directed region-level graphs for downstream connectivity analysis.

Experimental setup. All neural models were trained separately for each participant and evaluated using matched participant-specific train/validation/test splits. The VAR/Granger baseline was evaluated across three different trial splits. Values report mean $\pm$ standard deviation across runs. Training used early stopping based on validation loss, and the checkpoint with the best validation performance was used for test evaluation. Hyperparameters and implementation details for BACE and all baselines are provided in Appendices A.1 and A.3. We report mean squared error (MSE), coefficient of determination (R^2), and Pearson correlation (Corr): MSE measures absolute prediction error, R^2 quantifies explained variance, and Corr measures temporal alignment between predicted and ground-truth activity.

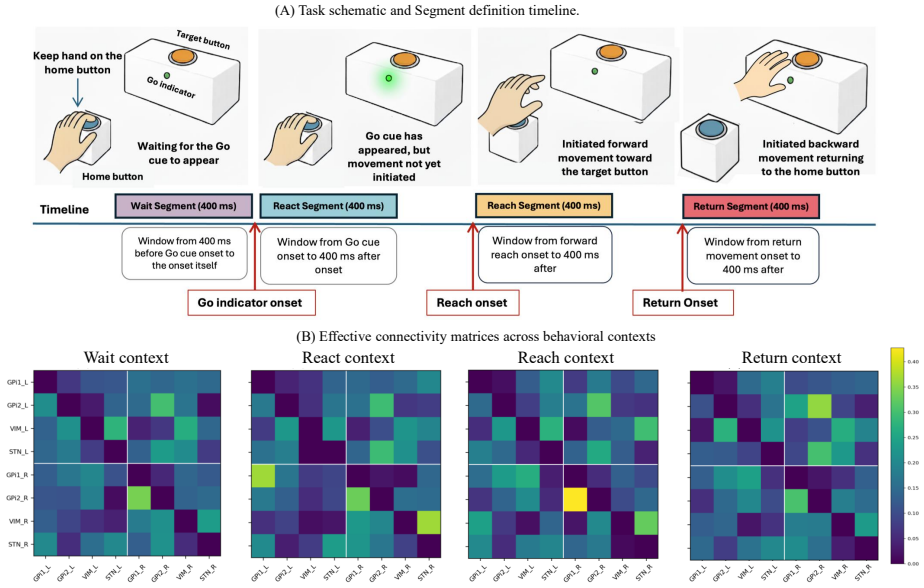


Figure 3: (**A**) Task schematic and behavioral segmentation timeline. Each trial comprised 4 stages with a 400 ms analysis window: Wait (-400 ms to go cue onset), React (0–400 ms after go cue onset), Reach (0–400 ms after forward reach onset), and Return (0–400 ms after return onset). (**B**) Representative context-specific $N \times N$ predictive effective connectivity matrices estimated by BACE for one participant. Matrix $\mathbf{A}_{ij}^{(k)}$ denotes directed influence from source node j to target node i. Node labels GPi1 and GPi2 denote anterior and posterior GPi, respectively; _L/_R indicate hemisphere.

Model performance. Table 1 summarizes 20-step forecasting performance across the 3 human intracranial LFP datasets. The neural models were evaluated on the same participant-specific train/validation/test splits, and values report mean $\pm$ standard deviation across runs. BACE achieved the lowest MSE and highest R^2 and Corr for all 3 participants, showing that its forecasting performance was consistent across different datasets.

Among non-graph models, LSTM and NDT provided reasonable forecasts but were consistently outperformed by graph-based methods. VAR/Granger was a strong classical baseline, showing that LFP forecasting contains substantial linear predictive structure. However, learned graph-based models improved further, suggesting that modeling interactions across recorded channels or regions is beneficial. Among the graph-based baselines, GWNet and AMAG were the strongest competitors, with AMAG providing the closest comparison because it also learns adaptive graphs for neural forecasting. Because BACE uses the behavioral context only to select the graph used in the graph-mixing layer, we added context-aware baselines that received the behavioral context labels for a more balanced comparison. The context-aware variants improved over their original versions, especially for AMAG, indicating that behavioral context helps prediction. Nevertheless, BACE achieved the best performance across all 3 datasets. Overall, BACE provides strong forecasting accuracy while also returning compact context-specific graphs that can be used to study directed connectivity between recorded regions.

Table 1: 20-step forecasting results across three human intracranial LFP datasets. Values report mean $\pm$ standard deviation across three runs. Ctx denotes context-aware variants. Best results are shown in bold.

	P1			P2			P3		
Model	$R^2 \uparrow$	Corr $\uparrow$	MSE $\downarrow$	$R^2 \uparrow$	Corr $\uparrow$	MSE $\downarrow$	$R^2 \uparrow$	Corr $\uparrow$	MSE $\downarrow$
LSTM	$0.779 \pm 4e^{-4}$	$0.886 \pm 5e^{-4}$	$0.1645 \pm 3e^{-4}$	$0.748 \pm 1e^{-3}$	$0.871 \pm 1e^{-3}$	$0.1739 \pm 1e^{-3}$	$0.636 \pm 3e^{-3}$	$0.799 \pm 3e^{-3}$	$0.3014 \pm 3e^{-3}$
Ctx-LSTM	$0.781 \pm 3e^{-3}$	$0.887 \pm 1e^{-3}$	$0.1631 \pm 2e^{-3}$	$0.751 \pm 3e^{-3}$	$0.873 \pm 3e^{-3}$	$0.1717 \pm 2e^{-3}$	$0.638 \pm 2e^{-3}$	$0.801 \pm 1e^{-3}$	$0.2998 \pm 2e^{-3}$
NDT	$0.786 \pm 1e^{-3}$	$0.889 \pm 6e^{-4}$	$0.1594 \pm 1e^{-3}$	$0.760 \pm 2e^{-3}$	$0.877 \pm 5e^{-4}$	$0.1656 \pm 2e^{-3}$	$0.649 \pm 3e^{-4}$	$0.807 \pm 1e^{-4}$	$0.2908 \pm 2e^{-4}$
VAR/Granger	$0.837 \pm 4e^{-3}$	$0.916 \pm 2e^{-3}$	$0.1209 \pm 3e^{-3}$	$0.825 \pm 4e^{-3}$	$0.909 \pm 2e^{-3}$	$0.1207 \pm 3e^{-3}$	$0.773 \pm 1e^{-2}$	$0.879 \pm 8e^{-3}$	$0.1812 \pm 1e^{-2}$
DCRNN	$0.842 \pm 4e^{-2}$	$0.918 \pm 2e^{-2}$	$0.1179 \pm 3e^{-2}$	$0.837 \pm 8e^{-3}$	$0.918 \pm 6e^{-3}$	$0.1123 \pm 6e^{-3}$	$0.685 \pm 2e^{-2}$	$0.835 \pm 2e^{-2}$	$0.2612 \pm 2e^{-2}$
GWNet	$0.859 \pm 8e^{-4}$	$0.940 \pm 4e^{-3}$	$0.1051 \pm 6e^{-4}$	$0.829 \pm 1e^{-3}$	$0.930 \pm 4e^{-3}$	$0.1184 \pm 7e^{-4}$	$0.793 \pm 1e^{-3}$	$0.914 \pm 4e^{-3}$	$0.1715 \pm 1e^{-3}$
AMAG	$0.861 \pm 7e^{-3}$	$0.928 \pm 4e^{-3}$	$0.1030 \pm 5e^{-3}$	$0.832 \pm 8e^{-3}$	$0.929 \pm 5e^{-3}$	$0.1160 \pm 6e^{-3}$	$0.797 \pm 4e^{-3}$	$0.899 \pm 4e^{-3}$	$0.1680 \pm 4e^{-3}$
Ctx-AMAG	$0.880 \pm 5e^{-3}$	$0.939 \pm 3e^{-3}$	$0.0890 \pm 4e^{-3}$	$0.841 \pm 1e^{-2}$	$0.929 \pm 6e^{-3}$	$0.1100 \pm 8e^{-3}$	$0.814 \pm 2e^{-2}$	$0.904 \pm 1e^{-2}$	$0.1539 \pm 2e^{-2}$
BACE	$\mathbf{0.894 \pm 3e^{-4}}$	$\mathbf{0.946 \pm 1e^{-4}}$	$\mathbf{0.0787 \pm 2e^{-4}}$	$\mathbf{0.874 \pm 3e^{-4}}$	$\mathbf{0.935 \pm 1e^{-4}}$	$\mathbf{0.0867 \pm 2e^{-4}}$	$\mathbf{0.845 \pm 6e^{-4}}$	$\mathbf{0.919 \pm 3e^{-4}}$	$\mathbf{0.1285 \pm 5e^{-4}}$

Ablation: effect of behavior-adaptive graph structure. To assess the contributions of graph structure and behavioral-context conditioning, we evaluated four BACE variants across all three participants: NoGraph removes graph mixing and graph feedback; Ctx-NoGraph adds the behavioral context label to the NoGraph forecasting decoder; SingleGraph learns one adjacency matrix shared across behavioral contexts; and Full BACE learns one $\mathbf{A}^{(k)}$ per context while sharing the encoder, graph projector, and forecasting head across contexts. All variants retain the same regional temporal backbone and forecasting objective and use matched training settings and identical data splits.

As shown in Table 2, Full BACE achieved the lowest mean MSE for all three participants, improving over both NoGraph and SingleGraph. Full BACE also consistently outperformed the context-aware graph-free control, Ctx-NoGraph, supporting the contribution of graph-mediated inter-regional information beyond behavioral-context conditioning alone. Additional ablation details and a complementary source-region perturbation analysis are provided in Appendix B.2.

Table 2: Ablation of graph structure across 3 participants. Ctx-NoGraph receives the behavioral context label but has no graph mixing or graph feedback. Values report mean $\pm$ standard deviation over three seeds.

	P1			P2			P3		
Model	$R^2 \uparrow$	Corr $\uparrow$	MSE $\downarrow$	$R^2 \uparrow$	Corr $\uparrow$	MSE $\downarrow$	$R^2 \uparrow$	Corr $\uparrow$	MSE $\downarrow$
NoGraph	$0.862 \pm 3e^{-4}$	$0.928 \pm 2e^{-4}$	$0.103 \pm 2e^{-4}$	$0.855 \pm 3e^{-4}$	$0.924 \pm 1e^{-4}$	$0.100 \pm 2e^{-4}$	$0.765 \pm 6e^{-4}$	$0.874 \pm 3e^{-4}$	$0.195 \pm 5e^{-4}$
Ctx-NoGraph	$0.882 \pm 4e^{-4}$	$0.939 \pm 2e^{-4}$	$0.088 \pm 3e^{-4}$	$0.865 \pm 8e^{-4}$	$0.931 \pm 4e^{-4}$	$0.093 \pm 6e^{-4}$	$0.827 \pm 2e^{-3}$	$0.910 \pm 1e^{-3}$	$0.143 \pm 2e^{-3}$
SingleGraph	$0.886 \pm 3e^{-4}$	$0.941 \pm 2e^{-4}$	$0.085 \pm 3e^{-4}$	$0.862 \pm 4e^{-4}$	$0.928 \pm 1e^{-4}$	$0.095 \pm 3e^{-4}$	$0.831 \pm 5e^{-4}$	$0.911 \pm 3e^{-4}$	$0.140 \pm 4e^{-4}$
Full BACE	$\mathbf{0.894 \pm 3e^{-4}}$	$\mathbf{0.946 \pm 1e^{-4}}$	$\mathbf{0.079 \pm 2e^{-4}}$	$\mathbf{0.874 \pm 3e^{-4}}$	$\mathbf{0.935 \pm 1e^{-4}}$	$\mathbf{0.087 \pm 2e^{-4}}$	$\mathbf{0.845 \pm 6e^{-4}}$	$\mathbf{0.919 \pm 3e^{-4}}$	$\mathbf{0.129 \pm 5e^{-4}}$

Computational complexity. BACE is lightweight relative to the neural forecasting baselines, with 0.13M trainable parameters and 0.20 GB peak GPU memory in our profiling setup, while also achieving the best forecasting results in Table 1. Full parameter counts, memory measurements, runtime details, and hardware specifications are provided in Appendix A.4.

4.3 Analysis of learned connectivity

Reproducibility. We first asked whether BACE's learned connectivity graphs were stable across resampled subsets of trials. For each participant and behavioral context, we repeated split-half resampling and compared the resulting absolute adjacency matrices $|\mathbf{A}^{(k)}|$. Across the 3 participants and 4 behavioral contexts, split-half correlations ranged from $r = 0.83$ to 0.91, corresponding to Spearman–Brown corrected repeatability values of 0.91 to 0.95. These results show that BACE learns context-specific graphs that are reproducible across independent subsets of trials. Full participant-level results are provided in Appendix B.3.

Edge reliability and context-dependent changes. We next tested whether the learned directed edges differed across behavioral contexts. For each participant and each adjacency matrix $\mathbf{A}^{(k)}$, we used nonparametric bootstrap resampling over trials to estimate confidence intervals for edge weights. We then tested pairwise differences between behavioral contexts with FDR correction [4]. Across all 3 participants, several directed edges showed reliable context-dependent changes. Fig. 4A shows one representative participant. These results indicate that BACE does not return a single fixed connectivity pattern, but instead learns graphs that vary with task segment.

Spatial organization. We next examined whether the learned graphs showed interpretable spatial structure across participants. We first compared within-hemisphere and cross-hemisphere edges. Within-hemisphere edges connect nodes in the same hemisphere, whereas cross-hemisphere edges connect nodes across hemispheres. BACE estimated stronger within-hemisphere than cross-hemisphere connectivity across behavioral contexts and participants (Fig. 4C), consistent with the general observation that brain networks often show stronger local or intra-hemispheric coupling than inter-hemispheric coupling [8, 47]. We also compared outgoing source strength from hemispheres ipsilateral and contralateral to the moving hand. Across participants, contralateral sources showed stronger outgoing influence than ipsilateral sources during React, Reach, and Return, while Wait was more balanced (Fig. 4B). This pattern is consistent with contralateral motor organization [43, 50]. Together, these analyses suggest that BACE produces directed predictive graphs that are not only accurate for forecasting, but also structured in ways that align with known spatial organization of motor networks.

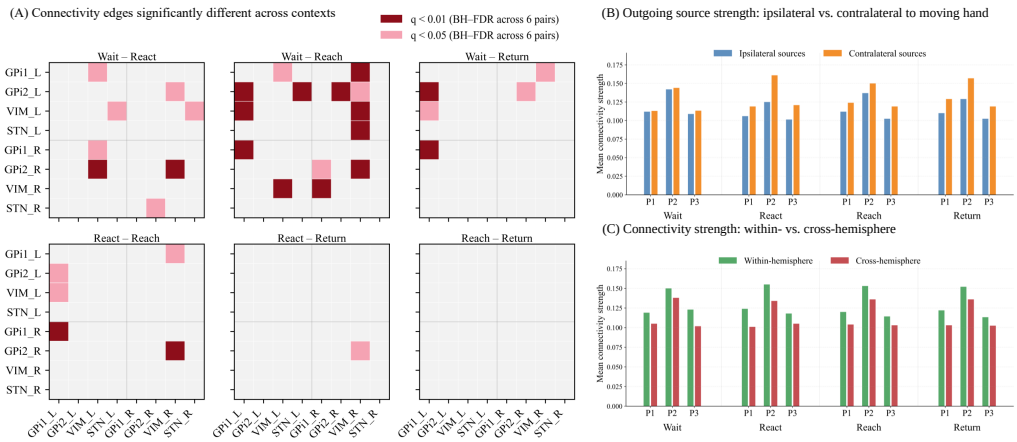


Figure 4: **Behavior-specific connectivity analyses.** (**A**) Representative participant showing edges with significant differences across behavioral contexts, based on bootstrap confidence intervals and Benjamini–Hochberg FDR correction. (**B**) Across-participant average outgoing connectivity strength from ipsilateral vs. contralateral hemisphere sources, defined relative to the moving hand. (**C**) Across-participant average within-hemisphere vs. cross-hemisphere connectivity strength.

5 Discussion

This work introduced BACE, a framework for estimating behavior-adapted connectivity from multi-region neural recordings. BACE couples forecasting with graph estimation, learning compact, directed inter-regional graphs whose parameters encode predictive interactions. By operating at the region level, the model consolidates many micro-contacts into interpretable embeddings and then learns connectivity among regions. The resulting adjacency matrices are behavior-specific and interpretable, supporting network analyses that can be inspected by domain experts.

In synthetic experiments, BACE recovered ground-truth connectivity while trained with a forecasting objective, suggesting that prediction-based objectives can recover connectivity structure from multivariate time series. Applied to 3 human subcortical LFP datasets during a cued reaching task, the model produced one connectivity matrix per behavioral context for each participant. Comparisons revealed behavior-dependent modulation of inter-regional influence, suggesting that deep-brain networks change their directed connectivity across task segments. The learned graphs also reflected interpretable spatial structure: within-hemisphere edges were stronger than cross-hemisphere edges, consistent with wiring-cost and modularity constraints [8, 47]. Methodologically, several design choices contribute to interpretability: independent regional encoders prevent cross-region leakage prior to graph mixing, context-specific adjacency matrices isolate behavior-dependent graph structure, and lightweight temporal regularizers stabilize multi-step predictions.

The current implementation of BACE uses experimentally defined behavioral contexts to select the corresponding adjacency matrix, while keeping the encoder and decoder shared across contexts; future extensions could relax this requirement by learning contexts in a weakly supervised or continuous manner. The estimated graphs capture predictive effective connectivity rather than anatomical or interventional causal effects. Future work should evaluate BACE across additional recording sessions, larger cohorts, additional tasks, physiology-informed screening [14], and other recording modalities, and could incorporate anatomical priors or multimodal behavioral signals such as EMG and video. Clinically, behavior-adapted connectivity estimates may provide a foundation for hypothesis generation and exploratory studies of brain-network analysis, but any translational use will require substantial validation.

Reproducibility statement

Code for model implementation, training, and evaluation is available at `https://github.com/mehrnazasadi95/BACE`.

Acknowledgments and Disclosure of Funding

This work was supported by the Once Upon a Time Foundation through the Raynor Cerebellum Project.

References

[1] Mehrnaz Asadi. Graph-based modeling of deep brain dynamics during motor behavior. Master's thesis, University of California, Irvine, 2025. URL https://www.proquest.com/docview/3281304876.

[2] Danielle S. Bassett and Olaf Sporns. Network neuroscience. *Nature Neuroscience*, 20(3): 353–364, 2017. doi: 10.1038/nn.4502.

[3] Manuel Beiran and Ashok Litwin-Kumar. Prediction of neural activity in connectome-constrained recurrent networks. *Nature Neuroscience*, 28(12):2561–2574, December 2025. doi: 10.1038/s41593-025-02080-4. URL https://doi.org/10.1038/s41593-025-02080-4.

[4] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

[5] Rahul Biswas and Eli Shlizerman. Statistical perspective on functional and causal neural connectomics: A comparative study. *Frontiers in Systems Neuroscience*, 16:817962, 2022.

[6] Rahul Biswas and Eli Shlizerman. Statistical perspective on functional and causal neural connectomics: The time-aware pc algorithm. *PLOS Computational Biology*, 18(11):e1010653, 2022.

[7] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[8] Ed Bullmore and Olaf Sporns. The economy of brain network organization. *Nature reviews neuroscience*, 13(5):336–349, 2012.

[9] 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.

[10] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[11] Michael W. Cole, Jeremy R. Reynolds, Jonathan D. Power, Grega Repovs, Alan Anticevic, and Todd S. Braver. Multi-task connectivity reveals flexible hubs for adaptive task control. *Nature Neuroscience*, 16(9):1348–1355, 2013. doi: 10.1038/nn.3470.

[12] Rainer Dahlhaus. Graphical interaction models for multivariate time series. *Metrika*, 51(2): 157–172, 2000. ISSN 1435-926X. doi: 10.1007/s001840000055. URL https://doi.org/10.1007/s001840000055.

[13] Gaute T Einevoll, Christoph Kayser, Nikos K Logothetis, and Stefano Panzeri. Modelling and analysis of local field potentials for studying the function of cortical circuits. *Nature Reviews Neuroscience*, 14(11):770–785, 2013.

[14] Saba A. Farahani, Elahe Khatibi, Manoj Vishwanath, Amir M. Rahmani, and Hung Cao. Beyond AHI: An interpretable causal-discovery-guided framework for sleep recovery in connected health, 2026. URL https://arxiv.org/abs/2606.18506.

[15] Karolina Finc, Kamil Bonna, Xiaosong He, David M Lydon-Staley, Simone Kühn, Włodzisław Duch, and Danielle S Bassett. Dynamic reconfiguration of functional brain networks during working memory training. *Nature communications*, 11(1):2435, 2020.

[16] Alex Fornito, Andrew Zalesky, and Edward T. Bullmore. *Fundamentals of Brain Network Analysis*. Academic Press, San Diego, CA, 2016. ISBN 978-0-12-407908-3.

[17] Karl J. Friston. Functional and effective connectivity: A review. *Brain Connectivity*, 1(1):13–36, 2011. doi: 10.1089/brain.2011.0008.

[18] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. *Advances in neural information processing systems*, 29, 2016.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[20] Mohammad Hosseini, Eray Erturk, Saba Hashemi, and Maryam M. Shanechi. Cross-subject modeling for widefield calcium imaging via atlas-aligned spatiotemporal tokenization. In *Forty-third International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=pZq2RMptsQ`.

[21] Sayed Mohammad Hosseini and Maryam Shanechi. Dynamical modeling of behaviorally relevant spatiotemporal patterns in neural imaging data. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=k4KVhQd19x`.

[22] Sina Javadzadeh, Rahil Soroushmojdehi, S. Alireza Seyyed Mousavi, Mehrnaz Asadi, Sumiko Abe, and Terence D Sanger. Functional embeddings enable aggregation of multi-area SEEG recordings over subjects and sessions. *arXiv preprint arXiv:2510.27090*, 2025. doi: 10.48550/arXiv.2510.27090. URL `https://arxiv.org/abs/2510.27090`.

[23] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 66–74, 2020.

[24] Maral Kasiri, Sina Javadzadeh, Jaya Nataraj, Seyyed Alireza Seyyed Mousavi, and Terence Sanger. Correlated activity in globus pallidus and thalamus during voluntary reaching movement in three children with primary dystonia. *Dystonia*, 2:11117, 2023.

[25] Maral Kasiri, Sumiko Abe, Rahil Soroushmojdehi, Estefania Hernandez-Martin, Seyyed Alireza Seyyed Mousavi, and Terence D Sanger. Local field potential signal transmission is correlated with the fractional anisotropy measured by diffusion tractography. *Brain Communications*, 7(5): fcaf336, 2025.

[26] Maral Kasiri, Mehrnaz Asadi, Jaya Nataraj, Sumiko Abe, and Terence D. Sanger. Optimized DBS in pediatric dystonia restores balance in transmission of signals within pallidothalamic network by modulating neural oscillations in deep brain regions. *medRxiv*, 2025. doi: 10.1101/2025.10.07.25337522. URL `https://www.medrxiv.org/content/early/2025/10/09/2025.10.07.25337522`. Preprint.

[27] Maral Kasiri, Jessica Vidmark, Estefania Hernandez-Martin, S. Alireza Seyyed Mousavi, and Terence D Sanger. Deep brain stimulation in globus pallidus internus travels to thalamus and subthalamic nuclei along physiological pathways. *Frontiers in Neuroscience*, 19:1592689, 2025.

[28] Byung-Hoon Kim, Jong Chul Ye, and Jae-Jin Kim. Learning dynamic graph representation of brain connectome with spatio-temporal attention. In *Advances in Neural Information Processing Systems*, volume 34, pages 4314–4327, 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/hash/22785dd2577be2ce28ef79febe80db10-Abstract.html`.

[29] Timothy D Kim, Thomas Z Luo, Jonathan W Pillow, and Carlos D Brody. Inferring latent dynamics underlying neural population activity via neural differential equations. In *International Conference on Machine Learning*, pages 5551–5561. PMLR, 2021.

[30] Rikuto Kotoge, Zheng Chen, Tasuku Kimura, Yasuko Matsubara, Takufumi Yanagisawa, Haruhiko Kishima, and Yasushi Sakurai. Evobrain: Dynamic multi-channel eeg graph modeling for time-evolving brain networks. *arXiv preprint arXiv:2509.15857*, 2025.

[31] Trung Le and Eli Shlizerman. STNDT: Modeling neural population activity with spatiotemporal transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 17926–17939, 2022. doi: 10.52202/068431-1303. URL https://papers.neurips.cc/paper_files/paper/2022/hash/72163d1c3c1726f1c29157d06e9e93c1-Abstract.html.

[32] Chengrui Li, Yunmiao Wang, Yule Wang, Weihan Li, Dieter Jaeger, and Anqi Wu. A factorized low-rank RNN framework for uncovering independent neural latent dynamics and connectivity. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 70983–70998. PMLR, 2026. URL https://proceedings.mlr.press/v306/li26ha.html.

[33] Jingyuan Li, Leo Scholl, Trung Le, Pavithra Rajeswaran, Amy Orsborn, and Eli Shlizerman. Amag: Additive, multiplicative and adaptive graph neural network for forecasting neuron activity. In *Advances in Neural Information Processing Systems*, volume 36, pages 8988–9014, 2023.

[34] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H Staib, Pamela Ventola, and James S Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 74:102233, 2021. doi: 10.1016/j.media.2021.102233.

[35] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[36] Hamed Mohammadi and Waldemar Karwowski. Graph Neural Networks in brain connectivity studies: Methods, challenges, and future directions. *Brain Sciences*, 15(1):17, 2025. ISSN 2076-3425. doi: 10.3390/brainsci15010017. URL https://www.mdpi.com/2076-3425/15/1/17.

[37] 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.

[38] Matthew G. Perich, Charlotte Arlt, Sofia Soares, Siyan Zhou, Manuel Beiran, Aaron S. Andalman, Tyler Benster, Megan E. Young, Clayton P. Mosher, Juri Minxha, Eugene Carter, Ueli Rutishauser, Peter H. Rudebeck, Christopher D. Harvey, Karl Deisseroth, and Kanaka Rajan. Inferring brain-wide interactions using data-constrained recurrent neural network models. *Neuron*, 2026. doi: 10.1016/j.neuron.2026.07.016. Published online 7 August 2026; ahead of print.

[39] Harsh Poonia, Felix Divo, Kristian Kersting, and Devendra Singh Dhami. Exploring neural granger causality with xlstms: Unveiling temporal dependencies in complex data. *arXiv preprint arXiv:2502.09981*, 2025.

[40] Terence D Sanger, Aaron Robison, Enrique Arguelles, Diana Ferman, and Mark Liker. Case report: targeting for deep brain stimulation surgery using chronic recording and stimulation in an inpatient neuromodulation monitoring unit, with implantation of electrodes in GPi and Vim in a 7-year-old child with progressive generalized dystonia. *Journal of child neurology*, 33(12):776–783, 2018.

[41] Omid G Sani, Hamidreza Abbaspourazad, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature neuroscience*, 24(1):140–149, 2021.

[42] S. Alireza Seyyed Mousavi, Sina Javadzadeh, Mehrnaz Asadi, Richard B. Ivry, and Terence D. Sanger. Behaviorally coupled oscillations reveal distinct timing roles of basal ganglia and cerebellothalamic circuits in dystonia. *Dystonia*, Volume 4, 2025. ISSN 2813-2106. doi: 10.3389/dyst.2025.15390. URL https://www.frontierspartnerships.org/journals/dystonia/articles/10.3389/dyst.2025.15390.

[43] Hiroshi Shibasaki and Mark Hallett. What is the bereitschaftspotential? *Clinical neurophysiology*, 117(11):2341–2356, 2006.

[44] Le Song, Mladen Kolar, and Eric P. Xing. Time-varying dynamic Bayesian networks. In *Advances in Neural Information Processing Systems*, volume 22, 2009.

[45] Rahil Soroushmojdehi, Sina Javadzadeh, Mehrnaz Asadi, and Terence D Sanger. Disentangling shared and private neural dynamics with SPIRE: a latent modeling framework for deep brain stimulation. *arXiv preprint arXiv:2510.25023*, 2025. doi: 10.48550/arXiv.2510.25023. URL `https://arxiv.org/abs/2510.25023`.

[46] Rahil Soroushmojdehi, Jessica SL Vidmark, S Alireza Seyyed Mousavi, Sumiko Abe, and Terence D Sanger. Frequency-dependent modulation of evoked potentials during GPi-DBS in pediatric dystonia. *Scientific Reports*, 15(1):33551, 2025.

[47] Olaf Sporns and Richard F Betzel. Modular brain networks. *Annual review of psychology*, 67 (1):613–640, 2016.

[48] Siyi Tang, Jared A Dunnmon, Qu Liangqiong, Khaled K Saab, Tina Baykaner, Christopher Lee-Messer, and Daniel L Rubin. Modeling multivariate biosignals with graph neural networks and structured state space models. In *Conference on health, inference, and learning*, pages 50–71. PMLR, 2023.

[49] Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B Fox. Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2021.

[50] Guy Vingerhoets. Contribution of the posterior parietal cortex in reaching, grasping, and using objects and tools. *Frontiers in psychology*, 5:151, 2014.

[51] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

[52] Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.

[53] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

[54] Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in neural information processing systems*, 21, 2008.

[55] Kaizhong Zheng, Shujian Yu, and Badong Chen. Ci-gnn: A granger causality-inspired graph neural network for interpretable brain network-based psychiatric diagnosis. *Neural Networks*, 172:106147, 2024.

[56] Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-vae. *Advances in neural information processing systems*, 33:7234–7247, 2020.

A Method Details

A.1 Model and Training Details

Adjacency parameterization and initialization. For each behavioral context k, BACE learns a directed adjacency matrix $\mathbf{A}^{(k)} \in \mathbb{R}^{N \times N}$. During training, the diagonal entries are fixed to zero, and each row is normalized to unit ℓ_1 norm,

$$\sum_j |A_{ij}^{(k)}| = 1,$$

so that each row represents the relative pattern of incoming predictive influence onto target node i. The same masking and normalization are used for all behavioral contexts. Adjacency matrices are initialized from region-wise Pearson correlations computed using training trials only. Correlations are computed between region-averaged signals, the diagonal is set to zero, and the resulting matrices are row-normalized before training. Validation and test trials are not used for graph initialization.

Time-positional embedding. After each region-specific GRU encoder, hidden states are layer-normalized and concatenated with a lightweight time-positional embedding. Normalized time indices $\tau/T_{\text{in}} \in [0, 1]$ are passed through a linear projection followed by a $\tanh$ activation. This gives the forecasting head access to the relative position of each sample within the input window while preserving the separation between regions before graph mixing.

Temporal attention in the forecasting head. Attention is computed over input time separately within each node. The previous decoder state supplies the query $\mathbf{q}_{i,t} = W_q \mathbf{s}_{i,t-1}$, while the activated graph-projected representations supply keys $\mathbf{k}_{i,\tau} = W_k \mathbf{Z}^{\text{act}}_{i,\tau}$ and values $\mathbf{v}_{i,\tau} = W_v \mathbf{Z}^{\text{act}}_{i,\tau}$:

$$a_{i,t,\tau} = \frac{\exp(\beta\, \mathbf{q}_{i,t}^\top \mathbf{k}_{i,\tau})}{\sum_{\tau'=1}^{T_{\text{in}}} \exp(\beta\, \mathbf{q}_{i,t}^\top \mathbf{k}_{i,\tau'})}, \qquad \mathbf{c}_{i,t} = \sum_{\tau=1}^{T_{\text{in}}} a_{i,t,\tau} \mathbf{v}_{i,\tau}.$$

Here, β is a learnable temperature. The temporal summary is concatenated with the adjacency-refined previous prediction to update the regional decoder state and produce the next channel-space residual forecast.

Temporal differences in the loss. The continuity, velocity, and curvature terms in the training objective are computed along the forecast horizon. We define

$$\Delta \hat{\mathbf{Y}}_t = \hat{\mathbf{Y}}_t - \hat{\mathbf{Y}}_{t-1}, \qquad \Delta \mathbf{Y}_t = \mathbf{Y}_t - \mathbf{Y}_{t-1},$$

with $\hat{\mathbf{Y}}_0 = \mathbf{X}_{\text{last}}$ and $\mathbf{Y}_0$ defined analogously. Second-order differences are computed as

$$\Delta^2 \hat{\mathbf{Y}}_t = \Delta \hat{\mathbf{Y}}_t - \Delta \hat{\mathbf{Y}}_{t-1}, \qquad \Delta^2 \mathbf{Y}_t = \Delta \mathbf{Y}_t - \Delta \mathbf{Y}_{t-1}.$$

The horizon weights w_t are log-spaced, with larger weights assigned to later forecast steps.

Optimization. All BACE experiments used Adam with learning rate 10^{-3}, weight decay 10^{-4}, and gradient clipping at 1.0. Training used early stopping based on validation loss, and the checkpoint with the lowest validation loss was used for test evaluation. Normalization statistics and graph initialization were computed from training trials only. Held-out test trials were used only for final evaluation.

Interpretation of the learned adjacency. Because $\mathbf{A}^{(k)}$ is optimized jointly with the regional encoders, the graph projector, and the forecasting head, it is not uniquely identifiable: different parameterizations of the surrounding model could, in principle, yield similar forecasts with different graphs. We therefore interpret $\mathbf{A}^{(k)}$ as a reproducible, context-dependent directed predictive connectivity map within BACE's parameterization, rather than as a uniquely determined causal or anatomical graph. Within this parameterization, the shared projection W_{neigh} transforms source-node features identically for all node pairs and contexts, so source–target-specific and context-specific routing is carried only by $\mathbf{A}^{(k)}$. Edge magnitudes reflect relative incoming predictive influence after row-wise ℓ_1 normalization and should not be interpreted as absolute coupling strength or as excitatory or inhibitory synaptic sign.

A.2 Nonlinear synthetic graph-recovery experiment

Data generation. We simulated an eight-node system with four context-specific directed graphs $\mathbf{A}^{(k)}$, in which each target node had two planted incoming parents and no self-edges. Each node was represented by a single signal, with dynamics

$$x_i(t+1) = 0.35\, x_i(t) + 0.50 \sum_j A^{(k)}_{ij} \left[\tanh^2\big(3x_j(t)\big) - \mu_\phi\right] + \eta_i(t),$$

where μ_ϕ is a centering constant and $\eta_i(t)$ is an innovation term. Each context contained 326 trials. Ground-truth graphs were used only for evaluation.

Training and comparison. BACE was trained using only the forecasting objective, with $T_{\text{in}} = 100$ and $T_{\text{out}} = 20$. The comparator was a context-specific VAR(10), whose lag range included the true structural lag; its directed edge scores aggregated coefficient magnitudes across lags.

Graph-recovery evaluation. Support recovery was assessed using $\mathrm{F1}@k_{\mathrm{row}}$ with $k = 2$, matching the number of planted parents per target. Edge-magnitude agreement was assessed using Pearson correlation between absolute inferred and ground-truth edge weights. Metrics were averaged across the four contexts; the main text reports mean $\pm$ standard deviation across three independently generated datasets, and Fig. 5 shows a representative run.

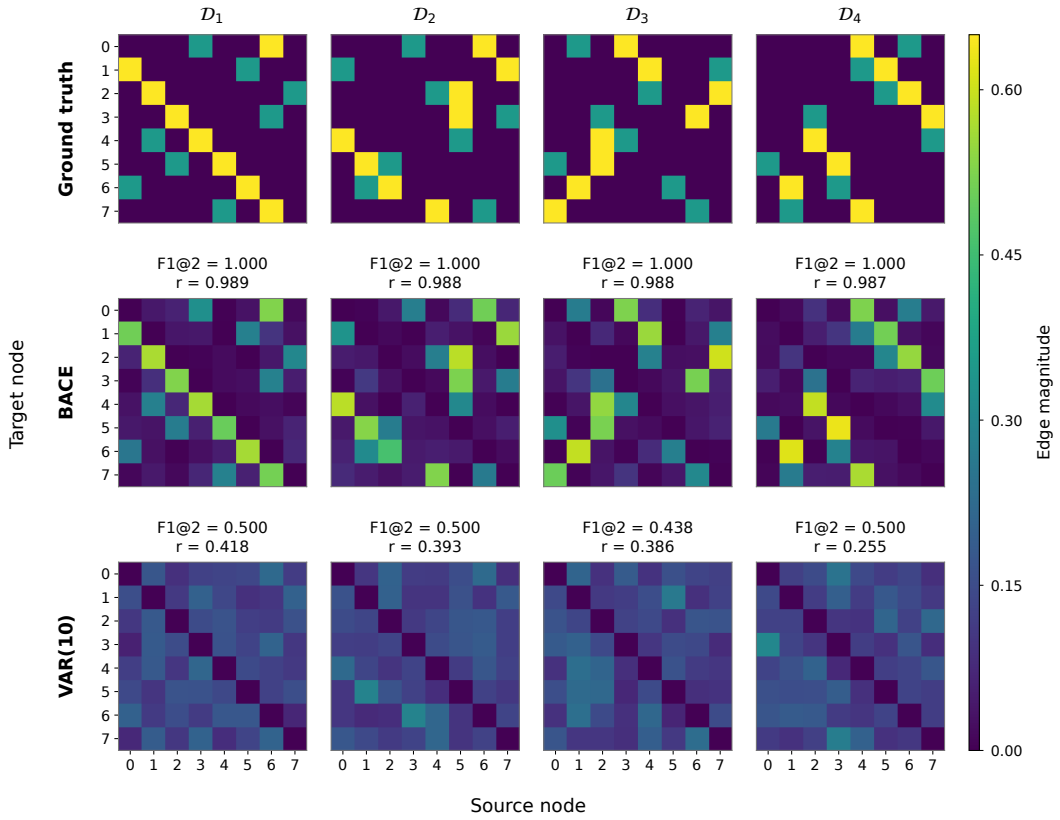


Figure 5: **Graph recovery in an example nonlinear synthetic run.** Columns correspond to four synthetic contexts. Rows show absolute ground-truth weights, absolute BACE adjacency weights, and VAR(10) edge scores. Within each matrix, rows denote targets and columns denote sources; all panels share a common color scale. Annotations report $\mathrm{F1}@k_{\mathrm{row}}$ ($k = 2$) and Pearson correlation between edge magnitudes (r).

A.3 Baseline Implementation Details

Details of Comparison Models. Except for VAR/Granger, comparison models were trained and evaluated using the same participant-specific train/validation/test splits as BACE. Unless otherwise stated, models used the same input window length ($T_{\mathrm{in}} = 100$), forecasting horizon ($T_{\mathrm{out}} = 20$), and 80 retained channels. Hyperparameters were selected using validation performance, and the checkpoint with the best validation loss was used for test evaluation.

LSTM. We used a two-layer LSTM forecaster with hidden dimension 128. The model takes past windows across all 80 channels and predicts the next 20 samples jointly for every channel. Training used mean squared error loss with Adam, learning rate 10^{-3}, weight decay 10^{-4}, and gradient clipping at 1.0. Models were trained with early stopping based on validation loss.

Context-aware LSTM. To control for access to behavioral context labels, we also trained a context-aware LSTM. This model uses the same LSTM architecture as the standard LSTM, but receives an additional learned embedding of the behavioral context label. This embedding is concatenated to the input representation before forecasting. The model therefore tests whether providing behavioral context alone improves prediction without learning explicit context-specific graphs.

NDT. We implemented a Neural Data Transformer (NDT) baseline under a masked-forecasting setup. Although NDT is typically used to reconstruct randomly masked neural activity, we adapted it for forecasting by treating the final 20 samples of each window as the masked segment. The model uses a three-layer Transformer with attention dimension 128 over windows of length $100 + 20$. During training, the past 100 samples are provided and the final 20 samples are zero-masked; the model is optimized to reconstruct only the masked future segment. Training used Adam with learning rate 10^{-4}, weight decay 10^{-5}, and early stopping based on validation loss.

Graph WaveNet. We implemented Graph WaveNet (GWNet) as a graph-based forecasting baseline for the cued reaching LFP datasets from the 3 participants. The architecture consists of 3 layers, each composed of 2 spatiotemporal blocks, for 6 blocks in total. Each block includes a gated dilated temporal convolution, a graph convolutional module for spatial mixing, and residual and skip connections. Temporal convolutions use kernel size 2, with 64 channels in the residual and dilated paths and 128 channels in the skip path. The skip outputs are aggregated and passed through a two-stage output head consisting of a 1×1 convolution with 64 channels followed by a readout convolution mapping to the 20-step forecasting horizon. Training was performed with Adam using an initial learning rate of 5×10^{-4}, decayed by 0.95 every 50 epochs, and no weight decay. Static graph supports were constructed from training data using correlation-based k-nearest neighbors, and an adaptive adjacency matrix was learned jointly during training.

DCRNN. We implemented the Diffusion Convolutional Recurrent Neural Network (DCRNN) using a sequence-to-sequence architecture with diffusion-convolutional GRU (DCGRU) cells. Both the encoder and decoder consisted of two stacked DCGRU layers with hidden dimension 64. Each DCGRU layer used diffusion convolution with $K = 2$ steps. The model was trained with Adam using learning rate 5×10^{-4}, learning-rate decay of 0.95 every 50 epochs, no weight decay, and inverse-sigmoid scheduled sampling with $\tau = 3000$.

AMAG. We implemented AMAG as a channel-level adaptive graph forecasting baseline. The 80 retained LFP channels were treated as 80 graph nodes, with one signal feature per node. Following the AMAG design, the model used three modules: temporal encoding, spatial interaction, and temporal readout. The spatial interaction module learned additive and multiplicative adjacency matrices and used a sample-dependent adaptor to modulate interactions between channel pairs. The adaptor was computed from a compact 5-step pooled summary of the past encoder states, while the temporal encoder and temporal readout operated on the full $T_{\text{in}} = 100$ input window and $T_{\text{out}} = 20$ forecasting horizon. Additive and multiplicative adjacency matrices were initialized from channel-wise correlations computed on the training data only. The temporal encoder and temporal readout used single-layer Transformer modules with hidden size 64 and 2 attention heads. Training used Adam with initial learning rate 5×10^{-4}, learning-rate decay of 0.95 every 50 epochs, weight decay 10^{-5}, gradient clipping at 1.0, and early stopping based on validation loss.

Context-aware AMAG. We also implemented a context-aware AMAG variant to control for access to behavioral context labels. This model used the same channel-level AMAG architecture described above, with 80 retained LFP channels treated as graph nodes. The only modification was the addition of a learned embedding for the behavioral context label, which was added to the input representation before forecasting. All other components, including the temporal encoding, spatial interaction, temporal readout, additive and multiplicative adjacency matrices, and training procedure, were kept the same as in AMAG.

A.4 Compute Resources and Computational Complexity

All neural forecasting models were profiled on a single workstation equipped with an AMD Ryzen Threadripper PRO 7955WX CPU with 16 cores, 128 GiB RAM, and one NVIDIA RTX 4500 Ada GPU with 24 GiB VRAM. Training and profiling were performed using PyTorch 2.2 with CUDA 12.2.

We compare the number of trainable parameters, #P(M), and peak GPU memory, M (GB), for neural forecasting models in Table 3. Parameter counts were computed from trainable model weights. Peak GPU memory was measured during training-style forward and backward passes using the same batch size and input/output window lengths used in the forecasting experiments.

Table 3: Computation complexity for neural forecasting models. Parameter count and peak GPU memory are reported under the same training setup.

	LSTM	NDT	GWNet	DCRNN	AMAG	Ctx-AMAG	BACE
#P (M)	0.45	0.63	0.48	0.30	0.253	0.291	**0.13**
Peak memory (GB)	**0.15**	0.84	5.54	5.22	17.26	17.26	0.20

BACE has a small computational footprint, with 0.13M trainable parameters and 0.20 GB peak GPU memory, while also achieving the best forecasting results in the main benchmark table. This efficiency follows from its region-level design: BACE learns compact $N \times N$ context-specific graphs and maps the resulting regional representation back to the channel level for forecasting.

Among graph-based baselines, AMAG and context-aware AMAG have moderate parameter counts but require substantially more GPU memory in our implementation. GWNet and DCRNN also use larger graph-based operations than BACE. In contrast, BACE keeps the temporal encoder and forecasting head shared across behavioral contexts, and only the small adjacency matrices vary with context. These results indicate that BACE's gains arise from its architectural design rather than increased model capacity, combining efficient forecasting with interpretable graph outputs.

For BACE, the reported runs used 200 training epochs. A training-style epoch, measured using forward and backward passes for profiling, required approximately 27.15 seconds on the NVIDIA RTX 4500 Ada GPU. Validation and test forward-only evaluation required approximately 2.06 and 2.12 seconds, respectively.

B Additional Dataset and Analysis Details

B.1 Human LFP Dataset and Preprocessing

Participants and experimental protocol. We analyzed three human intracranial local field potential (LFP) datasets collected from pediatric male participants undergoing deep brain stimulation (DBS) treatment as part of clinical care [40, 24, 42, 26]. All recordings were collected under Institutional Review Board-approved protocols, with informed consent obtained from the participants' legal guardians.

All participants performed a cued reaching task. At the beginning of each trial, the participant held a home button. After a variable delay, a visual go cue prompted the participant to release the home button and reach forward to press a target button. After pressing the target, the participant returned the hand to the home button to prepare for the next trial. Trials with incomplete task execution or excessive artifacts were excluded before model fitting.

Recording electrodes and acquisition. Recordings were obtained from up to 10 temporary stereo-electroencephalography (sEEG) depth leads (AdTech MM16C), implanted into candidate DBS targets using standard stereotactic procedures [40]. Each lead included ten high-impedance micro-contacts (50 μm, 70–90 kΩ). Signals were digitized at 24,414 Hz with a PZ5M 256-channel digitizer and RZ2 processor, and stored on an RS4 system (Tucker-Davis Technologies) [25, 46]. Electrode coverage differed across participants because implantation was clinically determined. Therefore, contacts were grouped into participant-specific anatomical nodes before model training, and BACE was trained separately for each participant.

Table 4 summarizes the node definitions and retained channels for each dataset.

Table 4: Participant-level recording and preprocessing details. Internal clinical identifiers are not shown.

Dataset	Original channels	Retained channels	Nodes	Trials/context after cleaning	Anatomical nodes
P1	80	80	8	326	GPi anterior-L/R, GPi posterior-L/R, VIM-L/R, STN-L/R
P2	100	80	10	329	GPi anterior-L/R, GPi posterior-L/R, VOa–VOp-L/R, VIM-L/R, CMCL-L/R
P3	100	80	10	398	CMP-L/R, GPi anterior-L/R, GPi posterior-L/R, VIM-L/R, VOa–VOp-L/R

Abbreviations: GPi, globus pallidus internus; VIM, ventral intermediate nucleus; STN, subthalamic nucleus; VOa–VOp, ventral oral anterior/posterior thalamic nuclei; CMCL, centromedian–central lateral thalamic region; CMP, centromedian–parafascicular thalamic region.

Preprocessing and cleaning. Signals were band-pass filtered from 1–50 Hz, common-average referenced, and downsampled to 1 kHz. Each dataset was cleaned before model fitting. For datasets with 100 original channels, two noisy channels were removed from each anatomical group, yielding eight retained contacts per node and 80 retained channels in total. Outlier trials were removed using robust amplitude-based criteria, and the final retained trials were balanced across behavioral contexts within each participant.

Trial segmentation. Each trial was segmented into four behavioral contexts of equal duration [1]:

- **Wait:** 400 ms preceding go cue onset.
- **React:** 0–400 ms following go cue onset, before movement initiation.
- **Reach:** 0–400 ms after forward reach onset, defined as the start of movement toward the target button.
- **Return:** 0–400 ms after return onset, defined as the start of movement from the target button back to the home button.

This segmentation produced event-aligned windows for comparing predictive connectivity across behavioral contexts.

Sliding windows, splitting, and normalization. Within each 400 ms behavioral context, data were divided into sliding forecasting examples with $T_{\mathrm{in}} = 100$ input samples, $T_{\mathrm{out}} = 20$ prediction samples, and stride 20. Train/validation/test splits were performed at the trial level rather than the window level to prevent leakage between overlapping windows from the same trial. Within each behavioral context, trials were shuffled using a fixed split seed and divided into 70% training, 15% validation, and 15% test sets, and the same split was used for BACE, ablations, and baseline models. All trials for a given participant were recorded during a single session on one day. Normalization parameters were computed from the training trials only and then applied to validation and test trials.

B.2 Graph and Context Ablation Analyses

Implementation of the graph-free controls. In NoGraph and Ctx-NoGraph, the regional encoders process each region's input independently, decoder attention operates over time separately within each region, and each regional output is mapped to that region's own channels. Both variants omit adjacency-based message passing and autoregressive graph feedback; Ctx-NoGraph additionally receives the behavioral-context label through the decoder. This preserves context-conditioned regional forecasting while removing the explicit graph-mediated routes for exchanging information between regions.

Source-region perturbation analysis. We interpret $\mathbf{A}^{(k)}$ as describing how regions exchange information (regional effective connectivity) within the BACE model. We therefore checked whether this exchange occurs only through the pathways that use $\mathbf{A}^{(k)}$. If information could also pass between regions through other parts of the model, such as the encoders, temporal attention, or decoder, then $\mathbf{A}^{(k)}$ would capture only part of the model's inter-regional structure.

This concern leads to a direct, testable prediction. In Full BACE, perturbing the input of one region can change the forecasts of other regions because that perturbation can travel through the graph pathways. In contrast, if these are the only routes for regions to interact, then in a matched model with these pathways removed, the same perturbation should change only the perturbed region's own forecast and leave every other region unaffected. Ctx-NoGraph provides this control: it is a matched ablation that retains the regional encoders, context conditioning, attention mechanism, decoder, and forecasting objective, but removes the $\mathbf{A}^{(k)}$-mediated graph pathways—both graph mixing and graph feedback.

To test this prediction, we perturbed one source region at a time on held-out examples by replacing all channels and time samples of its normalized input history with zero, corresponding to the training mean, while leaving the behavioral-context label and all other regions' inputs unchanged. For each source–target pair, we measured the mean absolute change in the target region's forecast, averaged over held-out examples, forecast steps, and channels within the target region. Changes in the perturbed region's own forecast measure the local effect; changes in other regions' forecasts measure cross-region propagation.

The results matched the prediction (Table 5). Local effects were similar in the two models, confirming that both responded to the perturbation to a similar degree. In Full BACE, the perturbation also changed the forecasts of other regions for all three participants. In Ctx-NoGraph, the effect stayed entirely within the perturbed region, with zero cross-region change to numerical precision. Removing the graph pathways therefore eliminated the observed cross-region effects while preserving a similar local response, supporting the conclusion that in the fitted models, regions exchange information only through the adjacency-mediated pathways.

Finally, we tested whether the size of each cross-region effect followed the strength of the corresponding learned edge. Within each participant and behavioral context, we correlated the off-diagonal forecast changes with the edge magnitudes $|A_{ij}^{(k)}|$, where j is the perturbed source and i the target. Correlations were consistently high (Pearson r = 0.914–0.953 for P1, 0.903–0.957 for P2, and 0.928–0.947 for P3): perturbing a source region changed a target's forecast more when the learned edge between them was stronger. Together, these results support the conclusion that inter-regional interaction in BACE is both confined to the adjacency-mediated pathways and organized by the learned adjacency, consistent with interpreting $\mathbf{A}^{(k)}$ as the model's directed predictive connectivity between regions.

Table 5: Source-region perturbation on held-out examples. Values are mean absolute changes in normalized forecasts, pooled across the four behavioral contexts. Local effects refer to the perturbed region's own forecast; cross-region effects refer to forecasts of other regions. Ctx-NoGraph cross-region effects were zero to numerical precision.

	Full BACE		Ctx-NoGraph	
Participant	Local	Cross-region	Local	Cross-region
P1	0.580	0.017	0.584	0.000
P2	0.559	0.011	0.556	0.000
P3	0.561	0.020	0.579	0.000

B.3 Statistical analysis of learned connectivity

Split–half reproducibility. For each participant and behavioral context k, we performed $K = 50$ random half-splits of trials. For each split, BACE graphs were estimated separately on each half, and reproducibility was measured as the Pearson correlation between the vectorized absolute adjacency matrices $|\mathbf{A}^{(k)}|$ from the two halves. We then applied the Spearman–Brown correction,

$$r_{\mathrm{SB}} = \frac{2r}{1+r},$$

to estimate full-data repeatability.

Mean split-half correlations were: P1: Wait $r = 0.91$, React $r = 0.90$, Reach $r = 0.88$, Return $r = 0.85$; P2: Wait $r = 0.90$, React $r = 0.88$, Reach $r = 0.91$, Return $r = 0.83$; P3: Wait $r = 0.90$, React $r = 0.91$, Reach $r = 0.89$, Return $r = 0.84$.

The corresponding Spearman–Brown corrected repeatability values were: P1: 0.95, 0.95, 0.94, and 0.92; P2: 0.95, 0.94, 0.95, and 0.91; P3: 0.95, 0.95, 0.94, and 0.91. These values indicate that BACE's context-specific graphs are reproducible across different subsets of trials.

Bootstrap resampling. To quantify uncertainty in individual edges, we performed $B = 1000$ nonparametric bootstrap resamples within each behavioral context. For each bootstrap sample, we re-estimated the graph parameters while holding the trained encoder, graph projector, and forecasting head fixed. This procedure measures how stable the learned adjacency estimates are under trial resampling. For each directed edge $(j \to i)$, we computed percentile 95% confidence intervals for the absolute edge strength $|A_{ij}^{(k)}|$.

Edgewise precision. We summarized bootstrap uncertainty using the width of the 95% confidence interval for each edge. Because very small median edge values can make relative uncertainty unstable, we computed relative half-width only for edges whose median magnitude exceeded a threshold τ, set to the 66th percentile of edge magnitudes:

$$\mathrm{rHW} = \frac{\mathrm{hi} - \mathrm{lo}}{2\,|\mathrm{median}|}.$$

This analysis provides a scale-normalized measure of uncertainty for non-negligible edges and complements the across-context tests reported below.

Across-context edge differences. To test whether learned edge strengths differed across behavioral contexts, we compared each pair of context-specific adjacency matrices within each participant. For each ordered edge $(j \to i)$ and each context pair (k_1, k_2), we computed the difference in absolute predictive edge strength,

$$\Delta_{ij}^{(k_1,k_2)} = \left|A_{ij}^{(k_1)}\right| - \left|A_{ij}^{(k_2)}\right|.$$

Uncertainty was estimated with nonparametric bootstrap resampling over trials within each context. For each bootstrap sample, BACE adjacency matrices were re-estimated and the edgewise difference $\Delta_{ij}^{(k_1,k_2)}$ was recomputed. Edges were considered context-dependent if the bootstrap confidence interval excluded zero and the corresponding two-sided test survived Benjamini–Hochberg FDR correction at $q = 0.05$. Correction was applied within each participant and context-pair comparison across all directed edges.

Fig. 4A shows an example participant, displaying only edges that survived this procedure.

Ethics statement

Human neural data were collected under IRB-approved protocols at participating medical centers. Informed consent was obtained from participants or legal guardians for minors. All data were de-identified before analysis, stored on secure access-controlled servers, and used solely for research in compliance with HIPAA and institutional privacy policies. The study did not involve clinical decision-making, non-standard interventions, or recruitment beyond patients already evaluated for DBS, and no re-identification was attempted. Raw recordings cannot be publicly shared due to ethical and regulatory restrictions, but de-identified datasets may be provided upon reasonable request under IRB-approved data use agreements. The authors confirm adherence to the NeurIPS Code of Ethics and declare no conflicts of interest or external sponsorship that could have influenced the research.

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: The abstract and Introduction state that BACE learns behavior-conditioned effective connectivity among anatomically defined brain nodes and is evaluated on three synthetic datasets and three human intracranial datasets. The claims are supported by the experimental results and are stated within the scope of the evaluated datasets and tasks.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The Discussion states that BACE has been evaluated on three human intracranial datasets, but that the cohort remains small and clinically heterogeneous. It also notes that the current model requires experimentally defined behavioral context labels, which limits its use to labeled datasets. Future work should test BACE on larger cohorts and other behavioral contexts.

   Guidelines:

   - The answer [N/A] 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: [N/A]

   Justification: The paper does not present formal theoretical results or proofs. The mathematical expressions define the model architecture, forecasting objective, and evaluation setup.

   Guidelines:

   - The answer [N/A] 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: The paper describes the datasets, preprocessing pipeline, model architecture, training procedures, hyperparameters, evaluation metrics, baselines, and compute resources in the main text and appendix. The BACE project repository is available at `https://github.com/mehrnazasadi95/BACE`.

   Guidelines:

   - The answer [N/A] 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: Because this study uses human intracranial recordings, the raw data cannot be publicly released due to ethical restrictions and institutional approvals. De-identified data may be shared upon reasonable request under IRB-approved data use agreements. Code for preprocessing, training, evaluation, and analysis, along with documentation, is available at `https://github.com/mehrnazasadi95/BACE`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the trial-level train/validation/test splits, sliding-window construction, input and prediction horizons, evaluation metrics, optimizer, and checkpoint selection in the main text. The appendix provides additional details on preprocessing, normalization, BACE hyperparameters, graph initialization, and baseline implementations.

   Guidelines:

   - The answer [N/A] 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: The paper reports variability measures and statistical analyses for the main experimental results, including mean and standard deviation across runs for forecasting and ablation experiments, split-half reproducibility for learned connectivity, bootstrap confidence intervals for edge reliability, and corrected p-values using Benjamini–Hochberg false discovery rate (FDR) correction. Additional participant-level reliability and statistical results are reported in the appendix.

Guidelines:

- The answer [N/A] 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: The paper reports compute information for the main neural forecasting experiments, including CPU/GPU specifications, system and GPU memory, software environment, trainable parameter counts, peak GPU memory, and approximate execution time for BACE training, validation, and test evaluation. These details are summarized in the computation complexity table and compute-resources appendix.

   Guidelines:

   - The answer [N/A] 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: The study was conducted under IRB-approved protocols, with informed consent obtained from participants or their legal guardians. Human neural data were de-identified before analysis, stored on secure access-controlled servers, and used only for research.

Guidelines:

- The answer [N/A] 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: The paper discusses the broader implications of BACE as a general framework for learning interpretable, behavior-conditioned connectivity from neural time series. Potential positive impacts include supporting neuroscience research, enabling hypothesis generation about brain-region interactions during behavior, and providing a model that can be adapted to other multi-region neural recordings beyond the datasets studied here. The paper also discusses ethical considerations around human intracranial data, including privacy, controlled data sharing and cohort heterogeneity.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: The paper does not release a high-risk pretrained model, generative model, or scraped dataset. The human intracranial recordings are not publicly released; access to de-identified data is controlled through reasonable request and execution of a data use agreement consistent with IRB approvals.

Guidelines:

- The answer [N/A] 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: Existing assets used in the paper, including open-source machine learning frameworks and scientific computing libraries such as PyTorch and standard Python analysis packages, are used in accordance with their respective licenses. The paper also cites the existing methods and models used as baselines. The human intracranial recordings were collected under IRB-approved clinical research protocols and are not publicly redistributed due to ethical and privacy restrictions.

    Guidelines:

    - The answer [N/A] 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: The paper introduces and releases a code repository, including preprocessing, training, and evaluation code, along with documentation, environment specifications, and synthetic data examples. The human intracranial recordings are not released as a public asset because of ethical and privacy restrictions, but de-identified data may be shared upon reasonable request under an IRB-consistent data use agreement.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: The paper analyzes previously collected human intracranial recordings obtained under IRB-approved protocols with informed consent. The authors did not conduct new participant recruitment, crowdsourcing, surveys, and compensation. The task protocol and trial structure are described in the dataset and preprocessing appendix.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: As described in the Ethics statement, the human intracranial recordings analyzed in this study were collected under IRB-approved protocols with informed consent from participants or their legal guardians. Potential risks associated with the original clinical recordings were governed by the approved clinical research protocols. All data used in this work were de-identified before analysis, stored securely, and used only for research.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

Justification: LLMs were not used in developing the BACE model, running experiments, analyzing data, implementing baselines, or producing the reported results. Any AI-assisted tools were limited to language polishing, editing, and formatting, and did not affect the methodology, experimental design, or scientific conclusions.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.