

Supplementary Material

A Brain Region Decoding Results

In addition to the Allen dataset discussed in the main text, we also present extended decoding results for the Neuronexus and IBL datasets (Figure 6). The confusion matrices show how each model performs across different brain regions. Lfp2vec consistently shows more focused diagonal patterns, indicating better region-specific accuracy across all datasets. While regions like the cortex and CA1 are reliably identified by all models, subfields such as CA2 and CA3 remain harder to distinguish. This is likely due to the sparsity of region labels and their lower biological separability. Notably, Lfp2vec maintains strong performance across all datasets. In the IBL dataset, all models perform well, possibly because of sparser labeling and the absence of more challenging classes. These findings support Lfp2vec’s ability to generalize across recording platforms with varied probe geometries and signal characteristics.

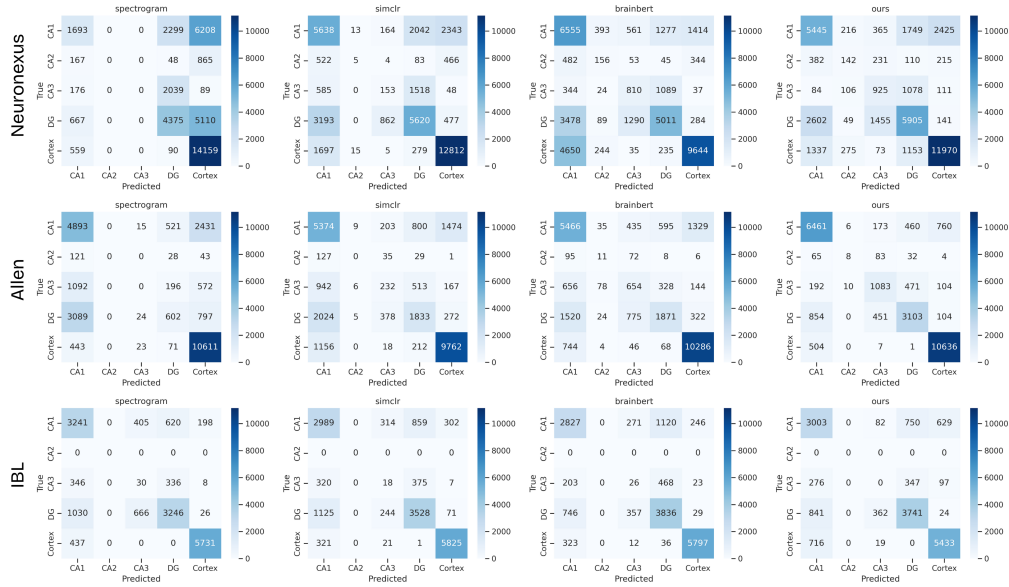


Figure 6: Decoding performance comparisons across models and datasets. Here we compare confusion matrix for anatomical region classification across datasets (Neuronexus, Allen, IBL) and models (spectrogram, SimCLR, BrainBERT, Lfp2vec). Each matrix shows predicted versus true region labels. Across all datasets, Lfp2vec shows more concentrated diagonal patterns. The cortex and CA1 regions are consistently well-identified across models, while hippocampal subfields such as CA2 and CA3 remain more challenging. Lfp2vec generalizes better than baselines, maintaining stable performance across datasets with diverse probe geometries and recording conditions.

B Baseline Settings

SimCLR We implement SimCLR [34] as a baseline for contrastive self-supervised learning. The input to SimCLR is a 2D spectrogram derived from the raw LFP signals. For data augmentation, we apply temporal masking, by randomly masking out segments along the time axis to encourage temporal invariance. The encoder architecture is a convolutional neural network (CNN) followed by a projection head composed of two linear layers with ReLU activation. During pretraining, we use the NT-Xent loss (normalized temperature-scaled cross-entropy loss) to maximize agreement between positive pairs generated from the same signal. After pretraining, we discard the projection head and fine-tune an MLP classifier on top of the encoder for brain region decoding.

BrainBERT We adopt BrainBERT following the implementation in [35], which adapts a transformer architecture to 2D spectrograms from electrophysiology signal. The model is trained from scratch on our dataset using masked token prediction as the pretraining task. After pretraining, the transformer

encoder is frozen, and we train an MLP classification head for downstream decoding. The MLP consists of two fully connected layers with ReLU activation and dropout. We use Cross Entropy Loss for classification and perform Bayesian optimization over the learning rate, dropout rate, and hidden size of the MLP to select hyperparameters. This two-stage training mirrors the original BrainBERT protocol while allowing adaptation to our LFP data.

C Hyperparameters Settings

Training Schedule. We use separate training schedules for pretraining and fine-tuning in Lfp2vec. During self-supervised pretraining, the encoder is optimized using the AdamW optimizer with a learning rate of $1e-5$, a batch size of 32, and trained for 50 epochs. For fine-tuning, we use a learning rate of $3e-5$, keep the batch size at 32, and train for 10 epochs. We apply gradient accumulation over 4 steps to simulate a larger effective batch size, and use a linear warmup schedule over 10% of the total training steps. This specific set of hyperparameters are suitable for Allen dataset, and may require further hyperparameter tuning for novel datasets.

Model Architecture. The model architecture consists of four main components: a convolutional encoder, a product quantizer, a Transformer module, and an MLP classifier. The encoder is a 7-layer 1D convolutional stack with kernel sizes of (10, 3, 3, 3, 3, 2, 2) and strides of (5, 2, 2, 2, 2, 2, 2), each followed by GELU activation and LayerNorm.

Table 2: Hyperparameter Settings for Lfp2vec

Stage	Hyperparameter	Value(s)	Description
Pretraining	Learning Rate	1×10^{-5}	Step size for encoder during self-supervised learning
	Batch Size	32	Number of samples per training batch
	Epochs	50	Number of training iterations
	Optimizer	AdamW	Optimization algorithm
Fine-tuning	Learning Rate	3×10^{-5}	Step size for encoder during fine tuning
	Batch Size	32	Number of samples per training batch
	Epochs	10	Number of fine-tuning iterations
	Optimizer	AdamW	Optimization algorithm
	Gradient Accumulation	4	Batches per optimizer step
	Warmup Ratio	0.1	Fraction of steps for learning rate warmup

This produces a 512-dimensional latent representation at each time step. The quantizer discretizes these features using Gumbel-softmax sampling. A diversity loss is applied to encourage full usage of the codebooks. The Transformer module contains 12 layers with 12 attention heads, a hidden size of 768, a feedforward size of 3072, and a dropout rate of 0.1. Finally, the classifier is a two-layer MLP with 256 hidden units and ReLU activation, mapping the output to one of five brain region classes.

Table 3: Model Hyperparameters for Lfp2vec

Component	Parameter	Value
Encoder	# Layers	7
	Kernel Sizes	(10, 3, 3, 3, 3, 2, 2)
	Strides	(5, 2, 2, 2, 2, 2, 2)
	Activation	GELU
	Normalization	LayerNorm per layer
	Output Dim	512
Quantizer	Codebooks	2×320
	Total Dimension	640
	Sampling Method	Gumbel-softmax
	Auxiliary Losses	Diversity Loss
Transformer	# Layers	12
	# Attention Heads	12
	Hidden Size	768
	Feedforward Size	3072
	Dropout	0.1
Classifier	Hidden Units	256
	Activation	ReLU
	Output Classes	5 (brain regions)

D Post-processing

We post-process the model’s per-timepoint predictions $p_{t,c} = p(y = c \mid x_t)$ by averaging class probabilities across a temporal window. This acts as a lightweight denoising step that suppresses local fluctuations in predictions. For each channel c , we compute a smoothed class score:

$$\bar{p}_c = \frac{1}{|\mathcal{W}_t|} \left(\sum_{t' \in \mathcal{W}_t} p_{t',c} \pi_c \right),$$

where \mathcal{W}_t denotes a symmetric window around t . This temporal smoothing preserves softmax semantics and can be interpreted as a local belief update under the assumption of short-term temporal consistency. And we optionally incorporate a class prior π_c to bias predictions toward frequent classes. π_c may be uniform or estimated empirically from the training distribution. The final prediction is obtained by taking the most probable class:

$$\hat{y}_t = \arg \max_c \tilde{p}_{t,c}.$$

This smoothing strategy increases temporal coherence without introducing model parameters or transition assumptions. For spatial smoothing, we apply majority voting across neighboring channels within each timepoint. Specifically, each prediction $\hat{y}_{t,c}$ is replaced with the most frequent label among a fixed spatial neighborhood \mathcal{N}_c around channel c . This encourages spatial contiguity and reduces anatomically implausible local fluctuations.

The effectiveness of these post-processing strategies is summarized in Figure 7. The temporal smoothing ensures that each channel receives a single, consistent anatomical label across trials. And the spatial smoothing ensures that there are no discontinuities in channel prediction. As shown in panels b, d, and f, this significantly reduces noise-driven fluctuations in the raw predictions and produces cleaner anatomical maps, particularly in deeper structures where variability across trials is most pronounced. The accuracy plots (a, c, e) confirm that this approach yields measurable improvements, especially for Lfp2vec, which produces rich but sometimes inconsistent predictions. These findings highlight that post-processing does not merely boost metrics, but selectively compensates for dataset-specific and region-specific noise patterns inherent in neural data.

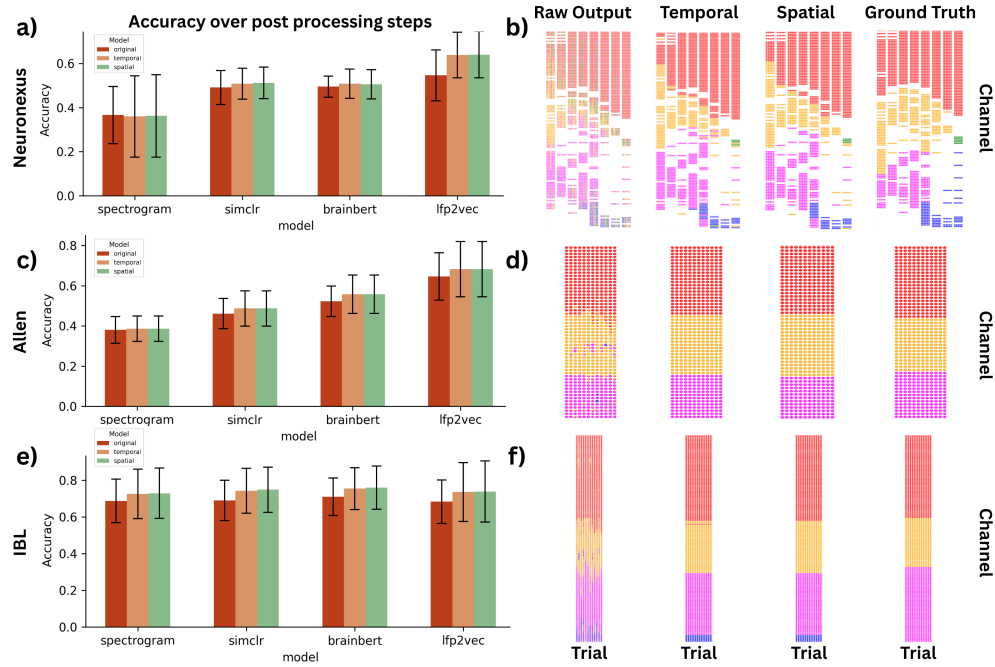


Figure 7: Effect of post-processing on anatomical decoding across datasets and models.. a, c, e) Classification accuracy before and after post-processing (temporal smoothing, spatial smoothing) across datasets (Neuronexus, Allen, IBL) and models. Accuracy consistently improves or remains stable after post-processing, with the largest gains seen for Lfp2vec. b, d, f) Visualization of predicted anatomical labels across channels and trials for an example session, before and after post-processing, compared to ground truth. Raw predictions exhibit spatial and temporal discontinuity, particularly in deeper regions. Temporal and spatial smoothing align predictions more closely with anatomical boundaries, reducing local inconsistencies and increasing interpretability.