LFPformer: Geometry-Aware Self-Supervised Electrophysiology Representation Learning

Subhrajit Dey[1]* **Ping-Jung Lu**[1]* **Visweswar Sirish Parupudi**[1]*
Malhar Patel[1]* **Margaret-Estefani Conde Paredes**[2] **Mihály Vöröslakos**[3]
Anna Maslarova[3] **György Buzsáki**[3] **Saurabh Vyas**[4] **Erdem Varol**[1,3]

[1]Department of Computer Science, New York University
[2]Mortimer B. Zuckerman Mind Brain Behavior Institute, Columbia University
[3]Neuroscience Institute, Grossman School of Medicine, New York University
[4]Neuroscience Institute and Biomedical Engineering, Carnegie Mellon University

Abstract

Local field potentials (LFPs) capture neural population activity across brain regions and behavioral states. Learning transferable representations of LFPs could make it easier to compare recordings across experiments and support downstream analyses when labeled data are limited. Although many recent neural foundation models have focused on spiking activity, learning representations that generalize across LFP recordings remains an important challenge. We introduce *LFPformer*, a self-supervised framework that learns latent representations of single-channel LFPs using only probe geometry and insertion coordinates recorded at acquisition time. To train, we augment the standard wav2vec 2.0 pretraining objective with spatial relationships derived from channel spacing and estimated insertion trajectories. We evaluate representation quality through three-dimensional coordinate regression and brain-region classification. Using wav2vec 2.0 as the main backbone, we compare against supervised baselines (AnyNet, ViT, and classical spectral classifiers) across datasets, laboratories, species, and probe technologies. Geometry guided self-supervised LFP representations improve localization error relative to the standard wav2vec 2.0 objective when fine-tuned with Transformer decoders. Together, these results show that spatial relationships between recording sites can improve LFP representations and provide information about the anatomical origin of neural activity. More broadly, they support the use of spatial structure during pretraining to learn scalable representations that can be applied across whole brain recording datasets and different probe designs. Project page: `https://www.neuroinformaticslab.com/lfpformer`

1 Introduction

Extracellular electrophysiology provides a window into brain activity through voltage signals recorded by electrodes implanted in neural tissue. These recordings contain fast spikes from nearby neurons and slower local field potentials (LFPs), which reflect collective transmembrane currents, including synaptic and dendritic activity [14, 25]. LFPs provide a complementary view of population dynamics, including subthreshold activity, and can be recorded where individual spikes are difficult to detect. Their sources can extend beyond the immediate recording site, making their anatomical interpretation challenging. Learning generalizable representations of LFPs could help capture the structure of these

*These authors contributed equally.

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

signals without requiring spike sorting, supporting anatomical identification and future decoding of brain states and behavior.

Recent advances in neural representation learning demonstrate the potential of pretraining across recordings. Neural Data Transformer 2 (NDT2) [13] and POYO [11] learn from neural population activity across recording contexts, while NEMO [4] uses contrastive learning on extracellular spike waveforms and autocorrelations for cell-type and brain-region classification. These approaches focus on spiking activity. LFP analysis commonly uses spectral features such as band power and spectrograms, whose temporal resolution and frequency summaries shape the information available to a decoder. BrainBERT [10] learns representations of intracranial field potentials through masked spectrogram modeling, while Lfp2vec [5] adapts audio-pretrained wav2vec 2.0 to raw LFP for brain-region classification. Cross-modal distillation also transfers knowledge from pretrained spike models to LFP models, improving behavioural decoding [26]. These advances motivate a complementary question: can acquisition geometry guide scalable representation learning directly from raw LFP, without a spike-based teacher or histological coordinates during pretraining?

We introduce LFPformer, a framework for geometry-aware self-supervised representation learning using the Allen Brain Observatory and IBL Brain-Wide Map datasets. We compare audio-pretrained wav2vec 2.0, Whisper, and data2vec backbones [7, 8, 9]. To train LFPformer, we introduce novel self-supervised objectives that leverage probe channel geometry to guide contrastive loss functions. Namely, we use channel-to-channel distance along recording probes (`ChanDistc`), or micromanipulator insertion path Distance (`ManipDistc`) as contrastive signal to encourage embeddings to reflect spatial relationships between LFP trials. These objectives augment masked signal prediction tasks (`MaskPred`) that is traditionally used to pretrain time series foundation models like wav2vec 2.0 that we use as backbone weights. `MaskPred` task uses temporal contrastive prediction and codebook diversity rather than raw-waveform reconstruction. Neither spatial objective requires histological coordinates during pretraining. We contrast these pretraining objectives with an oracle guided pretraining task that uses the distance between ground truth recording site locations (`CCFDistc`) to measure the ceiling of representational quality. We study encoder size, decoder capacity, temporal context, and pretraining exposure, and model scaling, reaching up to 58.88 million LFP trials in training.

We evaluate representation quality through brain-region classification and continuous 3D electrode localization. Accurate localization is essential for interpreting neural activity and comparing recordings across experiments. Conventional approaches combine atlas-based surgical planning [2] with post-hoc histological reconstruction [3], requiring tissue examination after the experiment. For chronic implants, histology verifies the trajectory only at the end of the recording period, motivating complementary estimates from the signals themselves. We train linear, MLP, and Transformer readouts with heteroscedastic Gaussian regression, evaluating both frozen representations and fine-tuning. The coordinate readouts require anatomical training labels. At inference, we pool repeated channel predictions and apply probe-geometry refinement, with the dataset-specific line-fitting procedure used for Allen. We assess generalization across held-out animals and laboratories, IBL-to-Allen and NeuroNexus transfer, and mouse-to-macaque transfer after supervised adaptation. Zero-shot evaluations are distinguished from target-trained readouts and fine-tuning.

Geometry-aware pretraining yields anatomically informative representations, and Transformer readouts narrow the performance gap between the three spatial objectives while improving LFP-only localization. Combining LFP features with insertion coordinates further improves localization, although remaining errors are too large to replace histological verification. Spectral analyses link the embeddings to frequency-dependent patterns across brain regions. Scaling experiments show diminishing gains with longer pretraining, and larger encoders do not consistently improve localization. The input-duration ablation identifies a practical accuracy–cost trade-off: 3-s windows preserve comparability with Lfp2vec, while extending the window from 5 to 10 s gives a smaller further error reduction (Supplementary Figure 13b). Together, these findings support acquisition geometry as a useful training signal for LFP representation learning and motivate evaluation on broader downstream tasks, including behavioral decoding and brain-state characterization.

2 Datasets

We constructed our pretraining corpus from two complementary large-scale open Neuropixels datasets. The **Allen Brain Observatory** Visual Coding dataset [15] provides Neuropixels 1.0 recordings from 58 mice ($\sim$51 sessions, up to six probes per session, 384 channels per probe at 1,250 Hz). After removing out-of-brain electrodes, we retained 80,313 valid channels registered to the Allen Common Coordinate Framework (CCF) [16] with full 3D coordinates (AP, DV, ML in μm), spanning 116 brain regions across the Allen CCF hierarchy. We segment continuous recordings into 3-second chunks (3,750 samples) and construct pretraining sets of upto 58.5M samples. The **IBL Brain Wide Map dataset** [17] adds cross-laboratory variability from multiple institutions worldwide, with standardized insertions through frontal cortex, striatum, hippocampus, and thalamus, providing both complementary brain coverage and an independent transfer evaluation set, increasing the amount to 309 regions.

Evaluation uses subject-disjoint splits for both mouse datasets. The IBL test set contains 500,000 chunks across 309 regions from 34 held-out mice, spanning 80 sessions and 119 probes, divided into five 100,000-chunk evaluation folds. Allen evaluation uses 12 held-out mice and sessions covering 116 regions, with five test folds of approximately 94–95K chunks each. We additionally evaluate cross-probe transfer on the five-class NeuroNexus benchmark and cross-species transfer on macaque prefrontal recordings, adapting on macaque A and testing on held-out macaque B. Full dataset descriptions and splits are provided in Supplementary Section A.

3 Methods

Our approach takes a single 3-second LFP chunk from one channel, encodes it into a geometry-preserving embedding, and maps that embedding to both a continuous 3D coordinate in atlas space and a discrete brain region label. The full pipeline is illustrated in Figure 1 and involves three stages: signal preprocessing, self-supervised representation learning, and downstream localization.

3.1 Signal Preprocessing

Raw LFP recordings undergo standard destriping [19]: high-pass filtering at 2 Hz, bad-channel interpolation, and common-mode noise subtraction. The cleaned signal is segmented into 3-second windows, each treated as an independent sample.

A key design choice is resampling each 3,750-sample segment (1,250 Hz) to 48,000 samples (16 kHz) before encoding. Audio foundation models were pretrained with convolutional front-ends tuned to 16 kHz input; feeding them lower-rate signals would leave most learned filters operating outside their intended frequency range, effectively discarding pretrained knowledge. Upsampling via sinc interpolation preserves the original spectral content below the Nyquist frequency (625 Hz) while filling in intermediate samples, allowing the convolutional layers to engage their full filter bank at the temporal resolution they were designed for.

3.2 Representation Learning Objectives

We use the standard wav2vec 2.0 [7] pretraining objective `MaskPred` and investigate whether incorporating spatial information improves learned LFP respresentations (Figure 1). `MaskPred` combines masked contrastive prediction with a codebook diversity loss. Specifically, selected latent frames are masked before the Transformer, which is trained to identify their corresponding quantized representations from sampled distractors. The diversity term encourages utilization of the available codebook entries. Unlike waveform reconstruction objectives, `MaskPred` operates directly on discrete latent targets.

Each model is trained using either `MaskPred` alone or `MaskPred` augmented with one spatial objective: `ChanDistc`, `ManipDistc`, or `CCFDistc`.

Shared Spatial Loss The three spatial objectives share the same loss formulation but differ in how inter-channel distances are defined. We average the encoder's final hidden states over time and normalize each resulting chunk embeding to unit length, yielding $\mathbf{z}_i$. Physical distances define a soft target distribution over neighboring channels, assigning higher weights to closer channels.

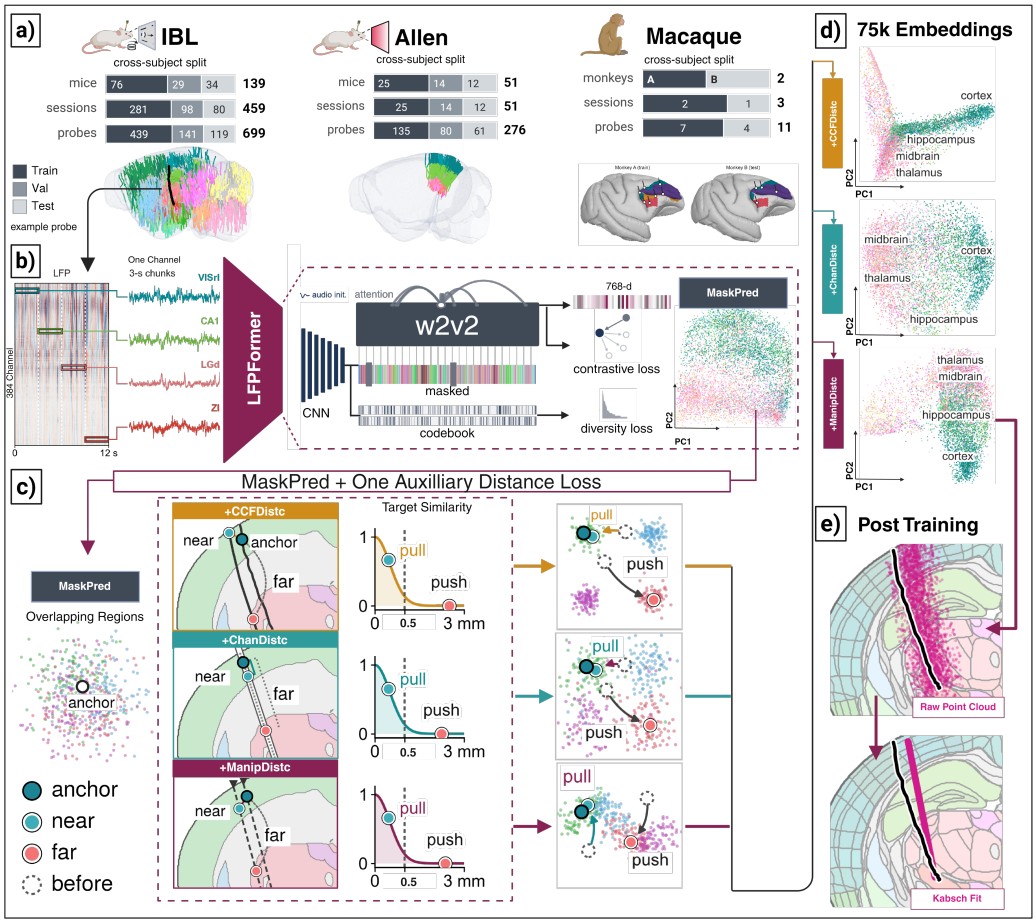


Figure 1: **LFPformer learns where a recording site is from its LFP alone.** **(a)** We pretrain on mouse Neuropixels recordings (IBL, Allen) and test transfer to new subjects, a new probe technology (NeuroNexus) and a new species (macaque). Bars show the held-out splits; probes are coloured by brain region based on Allen CCF. **(b)** Each 3s, single-channel LFP chunk is encoded by a wav2vec 2.0 model trained to predict masked frames (`MaskPred`). On its own, this objective does not separate brain regions in the embedding. **(c)** We therefore add one geometry-aware contrastive loss, which pulls together chunks recorded close in space and pushes apart distant ones (target similarity $\exp(-d^2/2\sigma^2)$, $\sigma = 0.5\,\text{mm}$). The three variants differ only in where the distance comes from: the probe's own channel layout (`ChanDistc`), the surgical micromanipulator log (`ManipDistc`), or histology (`CCFDistc`, oracle). **(d)** With these losses, held-out IBL embeddings organise by anatomy: cortex, hippocampus, thalamus and midbrain separate along the principal axes. **(e)** At test time a readout predicts a position and uncertainty for every chunk. Pooling chunks per channel (weighted by $1/\sigma^2$) and rigidly fitting the known probe geometry (Kabsch) turns this scattered cloud into a probe track that follows histology.

A corresponding distruction is computer from embedding similarities, and their cross-entropy is minimized to encourage spatially consistent representations. The loss is averaged across groups of chunks. We use $\sigma_d = 500\ \mu$ and $\tau_d = 0.2$ for all three objectives.

3.2.1 CCF Distance Contrastive (`CCFDistc`)

`CCFDistc` uses Euclidean distances between histologically registered channel coordinates, $d_{ij} = \|\mathbf{x}_i^{\text{CCF}} - \mathbf{x}_j^{\text{CCF}}\|_2$. All chunks in the contrastive batch form one group, including chunks from different probes. Because it uses anatomical coordinates during pretraining, `CCFDistc` serves as an atlas-supervised oracle reference, although it is not a guaranteed upper bound on performance.

3.2.2 Within-Probe Distance Contrastive (`ChanDistc`)

`ChanDistc` uses distances between channels in the manufactured probe layout, $d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|_2$. For Neuropixels 1.0, this includes the staggered columns and 20 μm row spacing. Each probe forms a separate comparison group; cross-probe pairs are excluded, and the losses are averaged across probes. This objective requires probe geometry but no histological coordinates.

3.2.3 Micromanipulator Distance Contrastive (`ManipDistc`)

`ManipDistc` estimates channel positions from the logged insertion point, angles, and depth. Let $\mathbf{e}$ and $\mathbf{t}$ denote the estimated entry and physical tip positions. Channel c is placed on the insertion line as

$$\tilde{\mathbf{x}}_c = \mathbf{t} + s_c \frac{\mathbf{e} - \mathbf{t}}{\|\mathbf{e} - \mathbf{t}\|_2}, \qquad s_c = (200 + 20\lfloor c/2 \rfloor)\ \mu\text{m}. \tag{1}$$

Here c is the zero-based channel index and s_c is its distance from the physical tip toward the entry point, including the 200 μm tip offset. Positions are then expressed in a common atlas frame. The loss uses distances between these approximate positions, $d_{ij} = \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|_2$, including pairs from different probes. This approximation places channels at the same depth at the same axial position. Samples without valid insertion records are excluded. Histological registration is not used, and the encoder still receives only LFP. Supplying insertion coordinates to a downstream readout is a separate condition.

Combined training objective. Each geometry-aware model retains both wav2vec 2.0 losses and adds one spatial loss:

$$\mathcal{L}_{\text{pretrain}} = \underbrace{\mathcal{L}_{\text{contrast}} + 0.1\mathcal{L}_{\text{div}}}_{\text{MaskPred}} + \lambda_t \mathcal{L}_{\text{spatial}}. \tag{2}$$

For the matched 75k recipes, λ_t is the ratio of the current `MaskPred` loss to the spatial loss, so their contributions have equal magnitude. This ratio is treated as a fixed weight during backpropagation. `MaskPred`-only training omits the spatial term.

3.3 Downstream Localization Head

After pretraining, a supervised readout predicts a coordinate mean and uncertainty for each chunk (Figure 1). We compare linear readouts, MLPs, and probe-level Transformers. The standard frozen MLP has 256 hidden units with a ReLU activation. The main direct-coordinate Transformer has four layers, width 256 and eight attention heads; it uses the embeddings, channel column and depth, and depth positional encodings to combine information across a probe. The localization and scaling comparisons keep the encoder frozen; fine-tuning is evaluated separately.

For target coordinate y_{ia}, predicted mean $\hat{\mu}_{ia}$ and standard deviation $\hat{\sigma}_{ia}$, we minimize the diagonal Gaussian negative log-likelihood:

$$\mathcal{L}_{\text{loc}} = \frac{1}{3N} \sum_{i=1}^{N} \sum_{a \in \{\text{AP,DV,ML}\}} \left[\frac{(y_{ia} - \hat{\mu}_{ia})^2}{2\hat{\sigma}_{ia}^2} + \log \hat{\sigma}_{ia} \right]. \tag{3}$$

Here N counts valid predictions; padded tokens and the constant Gaussian normalization are omitted. All readouts predict a coordinate mean and per-axis uncertainty and are trained with the same Gaussian negative log-likelihood. Linear and MLP heads predict log-variance, while the Transformer predicts a positive standard deviation through softplus. Predictions and uncertainties are converted to micrometers before refinement and evaluation.

3.4 Inference-Time Refinement

We first combine repeated chunk predictions for each channel c and axis a using inverse-variance weights:

$$\bar{\mu}_{ca} = \frac{\sum_t w_{cta} \hat{\mu}_{cta}}{\sum_t w_{cta}}, \qquad w_{cta} = \hat{\sigma}_{cta}^{-2}. \tag{4}$$

We then fit the known probe layout $\mathbf{p}_c$ to the pooled predictions using Kabsch alignment:

$$(R^\star, \mathbf{b}^\star) = \arg\min_{R \in SO(3),\, \mathbf{b} \in \mathbb{R}^3} \sum_c \|R\mathbf{p}_c + \mathbf{b} - \bar{\boldsymbol{\mu}}_c\|_2^2, \qquad \hat{\mathbf{x}}_c^{\mathrm{ref}} = R^\star \mathbf{p}_c + \mathbf{b}^\star. \tag{5}$$

Here R is a rotation and $\mathbf{b}$ a translation. All channels have equal weight in this fit, which preserves their physical spacing without rescaling. No test-set anatomical coordinates enter the fit. Supplementary Section D.4 gives the details, and Table 5 provides a six-channel worked example. For atlas-based classification, refined coordinates are looked up in the annotation volume. We also examine how the amount of recorded data affects localization accuracy; the supplementary ablation compares performance across different recording durations, Supplementary Figure 13b.

4 Experiments and Results

4.1 Scaling Behavior

We evaluated how localization performance changes with pretraining exposure, encoder size, decoder capacity, and compute (Figure 4). Base encoders were evaluated through 115,000 steps, corresponding to 58.88 million presentations of 3-s LFP chunks at a batch size of 512. These counts measure cumulative training exposure rather than unique examples. With frozen linear readouts, `ManipDistc` and `CCFDistc` improve during early pretraining and then largely plateau, maintaining lower localization error than `MaskPred` and `ChanDistc`. Extending Large `MaskPred` and `ChanDistc` to 100,000 steps. Larger encoders do not consistently outperform Base encoders, indicating that encoder size alone is insufficient to improve localization. At a matched pretraining budget of 35,000 steps (17.92 million chunk presentations), higher-capacity Transformer readouts generally improve performance for the geometry-aware objectives, although gains are not monotonic across architectures.

Baselines. We compare against supervised deep learning and classical methods evaluated under the same splits and post-hoc refinement. **AnyNet** [6] is a convolutional encoder trained end-to-end on LFP spectrograms; **ViT** [6] applies a Vision Transformer to the same spectrograms. On the classical side, **HistGBM/band** [1] uses histogram gradient-boosted trees on band-power features (delta through high-gamma); **kNN/band** and **kNN/PCA-PSD** are non-parametric baselines on raw and PCA-reduced spectral features; and **Ridge/PSD** is an L_2-regularized linear decoder on power spectral density.

4.2 Continuous Localization Results

Figure 2 evaluates 3D localization using frozen `MaskPred+ManipDistc` encoders with linear and Transformer readouts. Regional maps and errors by brain division show how localization accuracy varies across anatomy (panels a,b). The Transformer readout achieves lower mean per-probe error than the linear readout, while combining LFP features with micromanipulator coordinates gives the lowest mean error with a linear readout (panel c). This combination reaches approximately 710 μm, compared with 790 μm for micromanipulator coordinates alone, across the same 118 probes.

Panel (d) illustrates predictions before refinement, after inverse-variance channel pooling, and after Kabsch alignment. Kabsch fits the known probe geometry to predicted coordinates without using test-set anatomical coordinates. Panels (e–h) compare localization errors with ViT and AnyNet across regions, subjects, probes, and laboratories. Together, these results show that learned LFP features and insertion measurements provide complementary information, although substantial localization errors remain.

4.3 Cross-Probe Transfer

We evaluate IBL-pretrained encoders on the five-class NeuroNexus benchmark across seven held-out recording sessions (Figure 3a, rightmost panel). After fine-tuning, LFPformer with `ChanDistc` achieves the highest mean balanced accuracy (0.634 ± 0.089), compared with `ManipDistc` (0.606 ± 0.058), AnyNet (0.550 ± 0.110), and ViT (0.527 ± 0.069). With the encoder frozen and only the readout trained, `ChanDistc` achieves 0.570 ± 0.084, supporting the transferability of its anatomical features across recording technologies. Values are mean $\pm$ SD across held-out sessions. See the supplementary material for data splits and further dataset details.

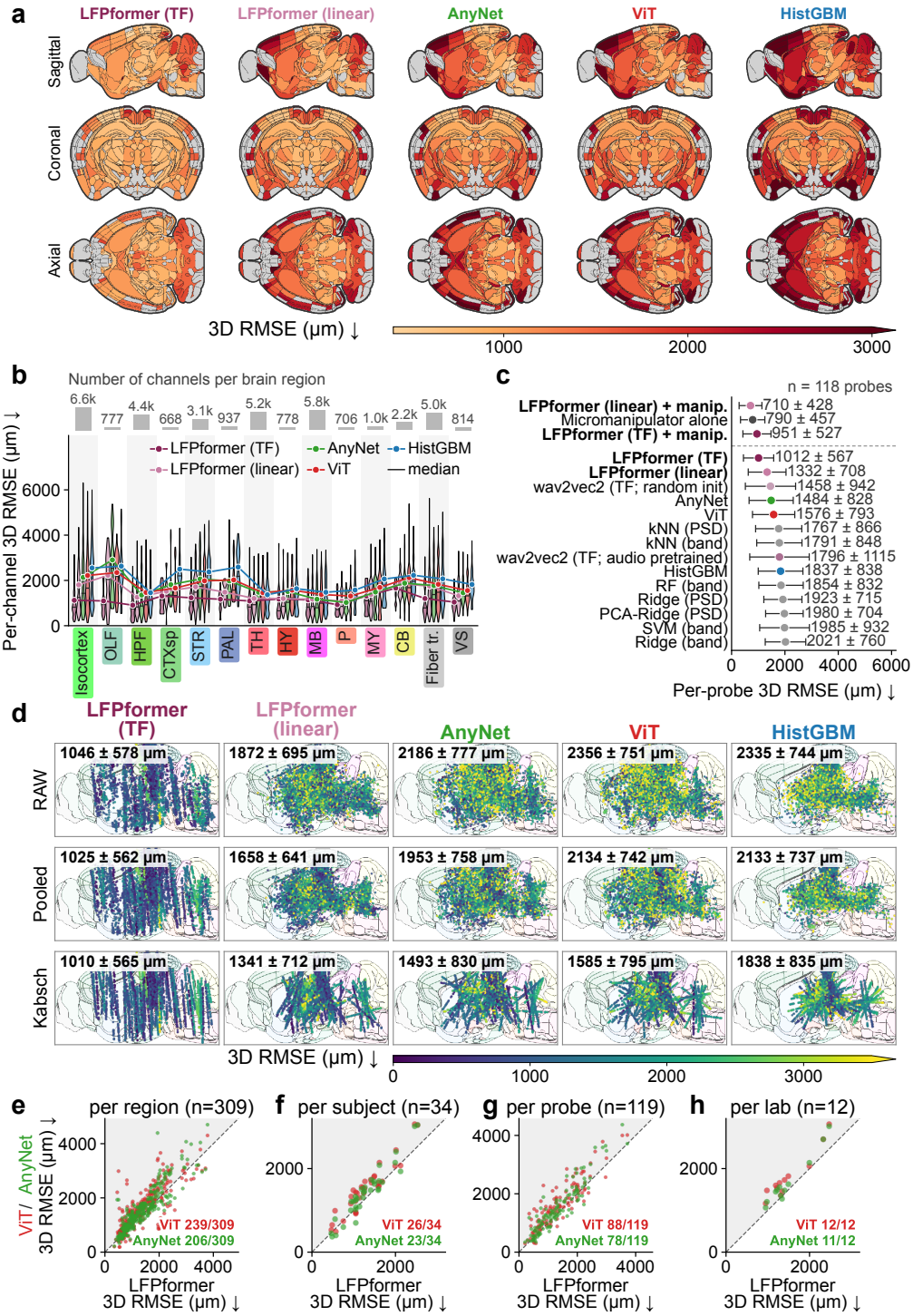


Figure 2: **IBL 3D localization.** We evaluate the frozen 75k `MaskPred` + `ManipDistc` encoder with linear and Transformer (TF) readouts; "manip." denotes micromanipulator coordinates supplied to the readout. Error is left-hemisphere-folded 3D RMSE in micrometers, averaged across decoder seeds with probes weighted equally. Kabsch refinement rigidly fits the known probe geometry to the predicted channel coordinates and does not use test-set brain coordinates.(Appendix D.4) Regional Kabsch RMSE on sagittal, coronal, and axial CCF views. **(b)** Per-channel errors grouped by Allen major division. **(c)** Per-probe RMSE. **(d)** Error before pooling, after inverse-variance channel pooling, and after Kabsch refinement. **(e–h)** Comparisons with ViT and AnyNet by region, subject, probe, and laboratory. The micromanipulator analyses include 118 probes because one test trajectory has zero recorded depth.

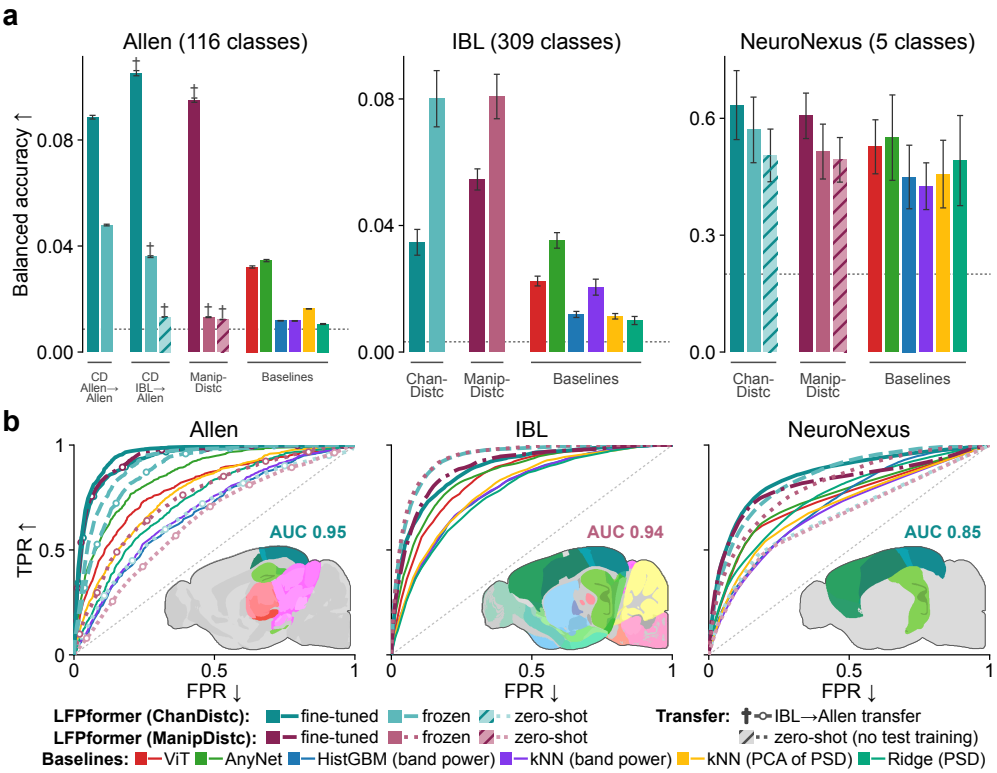


Figure 3: **Brain-region classification across datasets and probe types. (a)** Balanced accuracy on Allen (116 classes), IBL (309 classes), and NeuroNexus (5 classes). Dotted lines show chance performance; striped bars indicate zero-shot transfer, with no fitting on the target dataset. **(b)** Macro-averaged one-versus-rest ROC curves. The number in each panel is the highest macro AUC among the LFPformer variants shown. Error bars are sample SD across evaluation folds, decoder seeds, or held-out sessions, as appropriate for each dataset. Insets show the recording coverage in the Allen CCF. IBL results use per-probe Kabsch refinement; Allen results use the original per-probe PCA line fit. NeuroNexus zero-shot predictions provide hard labels only, so their ROC curves use one-hot scores.

5 Discussion

Geometry-aware pretraining improves anatomical information recovered from LFP (Figure 5). `ManipDistc` and `CCFDistc` outperform `ChanDistc` with linear readouts, but Transformer readouts narrow the gap between all three; `MaskPred` also improves yet remains substantially worse (Figure 4b). Longer pretraining gives diminishing gains, and larger encoders do not consistently help. `CCFDistc` uses atlas coordinates as an oracle reference, but `ManipDistc` and even `ChanDistc` approach its performance with a Transformer, showing that useful spatial representations can emerge without histological coordinates during pretraining. The downstream coordinate decoder still requires anatomical labels.

In Figure 3, we can see that IBL-pretrained `ChanDistc` achieves higher mean balanced accuracy than `ManipDistc` on Allen and NeuroNexus, including with frozen encoders. This supports the transferability of its embeddings and shows that source-dataset localization alone does not determine representation quality. Relative probe distances may encourage less dataset-specific features. Mouse-pretrained `ManipDistc` also achieves the highest mean macaque classification accuracy among the supplementary comparators after fine-tuning (0.37 ± 0.20; Figure 8), but only two animals limit this finding.

CCA connects embedding structure to regional spectral differences (Figure 6). C1 contrasts lower-frequency power with power above approximately 200 Hz, C2 captures broadband power, especially at high frequencies, and C3 emphasizes high gamma. The predominantly cerebellar cluster has low C1

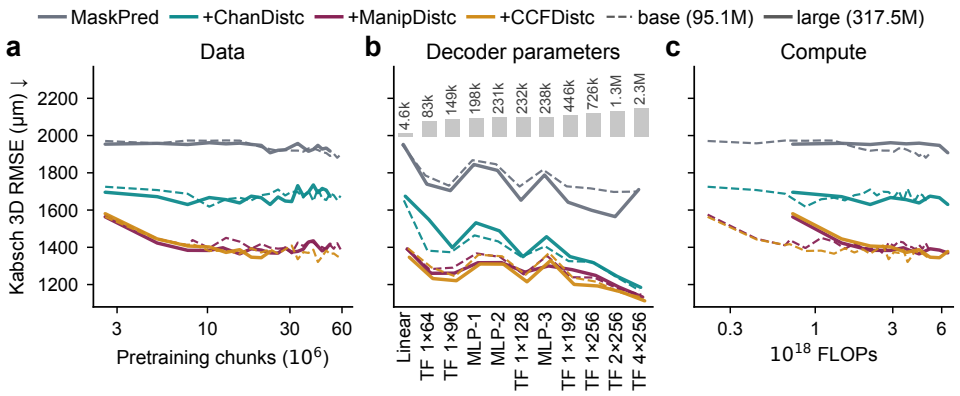


Figure 4: **Data, decoder, and compute scaling.** Colours identify `MaskPred` and the three geometry-aware variants. Dashed lines show the base encoder (95.1M parameters), and solid lines show the large encoder (317.5M parameters). Lower Kabsch-refined 3D RMSE is better. **(a)** Frozen linear-readout performance as a function of the cumulative amount of pretraining data. Each point is a checkpoint; the number of 3-s LFP chunks is the number of optimization steps multiplied by the batch size (512 chunks). **(b)** Readout capacity at a matched pretraining budget of 17.92M chunks. Linear denotes a per-channel linear Gaussian readout, MLP-k a per-channel MLP with k hidden layers, and TF $L \times d$ a probe-level Transformer with L blocks and hidden width d. Grey bars show the number of trainable decoder parameters, and readouts are ordered by this count. **(c)** Frozen linear-readout performance as a function of estimated cumulative pretraining compute. We calculate compute as $6NFBS$, where N is the total number of encoder parameters (95.1M or 317.5M), $F = 149$ is the number of latent frames in each 3-s chunk, $B = 512$ is the batch size, and S is the number of pretraining steps. The factor of six approximates the forward and backward passes. Thus, compute increases with model size, input length, and the amount of data processed. One unit on the horizontal axis is 10^{18} FLOPs. This nominal estimate excludes the additional cost of geometry losses, evaluation, and restarted training. Panels (a) and (c) use logarithmic horizontal axes; panel (c) is restricted to the compute range shared by all four objectives.

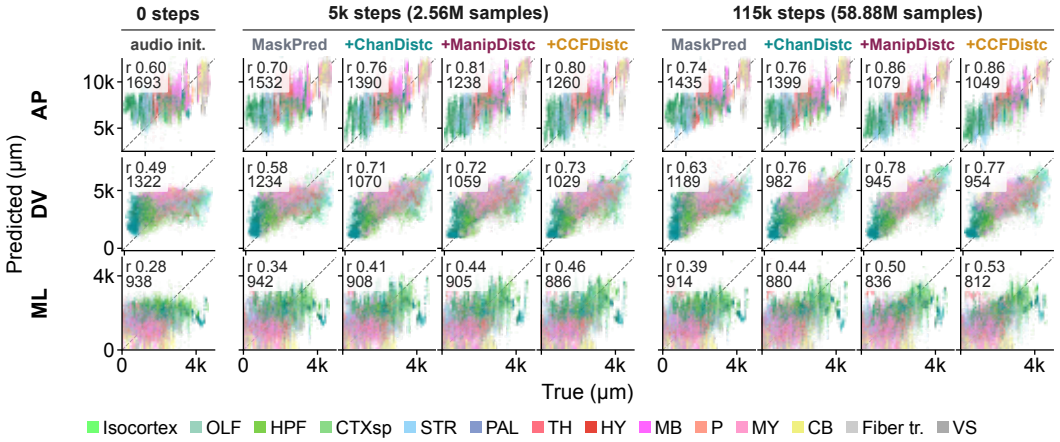


Figure 5: **Coordinate predictions during self-supervised pretraining.** Each panel compares predicted and true channel coordinates for a frozen base wav2vec 2.0 encoder with a linear Gaussian readout. Rows show anterior–posterior (AP), dorsal–ventral (DV), and medial–lateral (ML) coordinates; all panels in a row share the same axes. The left column is the audio-pretrained initialization, before LFP pretraining. The middle and right blocks show checkpoints after 5k (2.56M chunks) and 115k (58.88M chunks) training steps, respectively, for `MaskPred` and `MaskPred` with `ChanDistc`, `ManipDistc`, or `CCFDistc`. Points are coloured by Allen major division, dashed lines show equality, and each inset gives Pearson correlation and coordinate-wise RMSE in micrometers. Results use the same 500,000 held-out IBL windows and 39,615 channels in every panel, with inverse-variance pooling across repeated channel predictions before scoring.

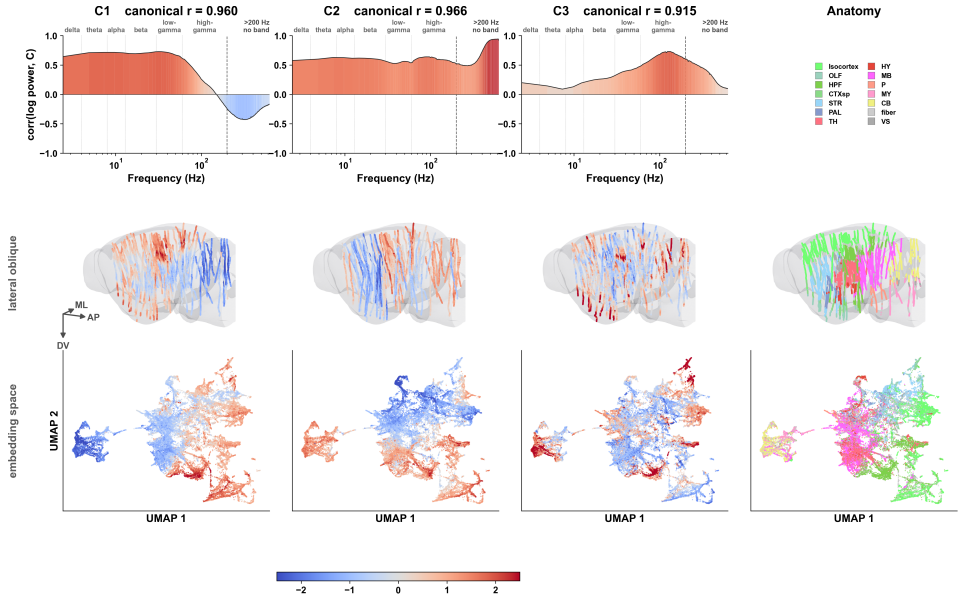


Figure 6: **Relationship between frozen LFPformer embeddings and LFP spectra.** Canonical correlation analysis relates per-channel mean log power spectra (257 Welch bins; 2.44 Hz resolution) to frozen Large `ManipDistc` embeddings (1024 dimensions) from 39,615 channels on 119 held-out IBL probes. Correlations are evaluated with five probe-grouped folds. Columns 1 to 3 are the leading canonical components. **Top:** correlation between each Welch bin and the component score. **Middle:** component scores on the probes, in a lateral-oblique view of the Allen CCF. **Bottom:** UMAP of the same channel embeddings, coloured by component score. **Column 4** repeats the brain and the UMAP with every channel coloured by its Allen CCF division, in the atlas's own colours, as an anatomical reference. The interpretation of the three components is provided in the main text.

and high C2 scores, linking its separation to a distinct spectral profile. Overlap and variation within anatomical groups suggest that embeddings capture both regional and within-region differences.

Limitations and future directions. LFPformer is a step toward replacing histology, but its errors remain large for reliable identification of small structures or anatomical boundaries. Even the best linear readout combined with micromanipulator coordinates has approximately 710 μm mean per-probe error (Figure 2c). Future work should combine LFP, spikes, probe geometry, species-specific atlases, and surgical measurements to improve accuracy, and evaluate whether the embeddings also improve behavioural decoding.

6 Broader Impact

LfpFormer can reduce reliance on histology in neuroscience, lowering the barrier to large-scale electrophysiology. Clinically, real-time electrode localization from the signal itself could improve deep brain stimulation safety by providing continuous position feedback without additional imaging. Any clinical deployment must, however, include calibrated uncertainty estimates (provided by our heteroscedastic head) and human oversight, as incorrect localization could lead to inappropriate stimulation targets. For the ML community, our results establish a new benchmark for adapting transformer-based foundation models to noisy, nonstationary biomedical time series via geometry-aware inductive biases.

Code Availability LFPformer is available as a tool for localizing recording sites and predicting brain regions from local field potentials at `https://github.com/neuroinfolab/LFPformer`. The repository provides step-by-step instructions for pretraining, post-training adaptation, and inference on new recordings.

Acknowledgments

This work was supported by the National Institutes of Health through National Institute of Mental Health award R00MH128772 to Erdem Varol and National Institute of Neurological Disorders and Stroke awards R00NS140554 to Saurabh Vyas and R01NS113782 to György Buzsáki. Cloud computing was supported by the NYU Google Cloud Platform for Research Grants Program through a GCP Research Award (RD157) to Erdem Varol for "Cloud-Accelerated NeuroFrame: Geometry-Aware Foundation Models for Extracellular Electrophysiology." We thank NYU's high-performance computing service for computational resources and support. The macaque recordings were collected by Saurabh Vyas in Mark Churchland's laboratory at Columbia University.

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[2] E. S. Lein, M. J. Hawrylycz, N. Ao, et al. Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, 445(7124):168–176, 2007.

[3] L. D. Liu, S. Chen, H. Hou, et al. Accurate localization of linear probe electrode arrays across multiple brains. *eNeuro*, 8(6), 2021.

[4] H. Yu, H. Lyu, Y. Xu, et al. In vivo cell-type and brain region classification via multimodal contrastive learning. In *ICLR*, 2025.

[5] T. He, M. Patel, C. Li, A. Maslarova, M. Vöröslakos, N. Ramanathan, W.-L. Hung, G. Buzsáki, and E. Varol. Self supervised learning for in vivo localization of microelectrode arrays using raw local field potential. In *NeurIPS*, 2025.

[6] X. Wang, J.-C. Béïque, and R. Naud. Modular organization of electrical fluctuations in the mouse brain. *bioRxiv*, 2025.10.01.679820, 2025.

[7] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *NeurIPS*, 33:12449–12460, 2020.

[8] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *ICML*, 2023.

[9] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *ICML*, 2022.

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

[11] M. Azabou, V. Arora, V. Ganesh, et al. A unified, scalable framework for neural population decoding. In *NeurIPS*, 36:44937–44956, 2023.

[12] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976.

[13] J. Ye, J. Collinger, L. Wehbe, and R. Gaunt. Neural data transformer 2: Multi-context pretraining for neural spiking activity. In *NeurIPS*, 36:80352–80374, 2023.M

[14] G. Buzsáki, C. A. Anastassiou, and C. Koch. The origin of extracellular fields and currents, EEG, ECoG, LFP and spikes. *Nature Reviews Neuroscience*, 13(6):407–420, 2012.

[15] J. H. Siegle, X. Jia, S. Durand, et al. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592(7852):86–92, 2021.

[16] Q. Wang, S.-L. Ding, Y. Li, et al. The Allen Mouse Brain Common Coordinate Framework: A 3D reference atlas. *Cell*, 181(4):936–953, 2020.

[17] The International Brain Laboratory et al. A brain-wide map of neural activity during complex behaviour. *Nature*, 2025.

[18] G. N. Angotzi, M. Vöröslakos, N. Perentos, et al. Multi-shank 1024 channels active SiNAPS probe for large multi-regional topographical electrophysiological mapping of neural dynamics. *Advanced Science*, 12(16):e2416239, 2025.

[19] K. Banga, J. Boussard, G. A. Chapuis, et al. Spike sorting pipeline for the International Brain Laboratory. *Figshare*, 2022.

[20] E. M. Trautmann, J. K. Hesse, G. M. Stine, R. Xia, S. Zhu, D. J. O'Shea, B. Karsh, J. Colonell, F. F. Lanfranchi, S. Vyas, A. Zimnik, E. Amematsro, N. A. Steinemann, D. A. Wagenaar, M. Pachitariu, A. Andrei, C. Mora Lopez, J. O'Callaghan, J. Putzeys, B. C. Raducanu, M. Welkenhuysen, M. Churchland, T. Moore, M. Shadlen, K. Shenoy, D. Tsao, B. Dutta, and T. Harris. Large-scale high-density brain-wide neural recording in nonhuman primates. *Nature Neuroscience*, 28(7):1562–1575, 2025.

[21] S. Vyas, M. D. Golub, D. Sussillo, and K. V. Shenoy. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249–275, 2020.

[22] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

[23] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, J. Ben-tzur, M. Hardt, B. Recht, and A. Talwalkar. A system for massively parallel hyperparameter tuning. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 2, 2020.

[24] A. I. Cowen-Rivers, W. Lyu, R. Tutunov, Z. Wang, A. Grosnit, R. R. Griffiths, A. M. Maraval, H. Jianye, J. Wang, J. Peters, and H. Bou-Ammar. HEBO: Pushing the limits of sample-efficient hyper-parameter optimisation. *Journal of Artificial Intelligence Research*, 74:1269–1349, 2022.

[25] O. Herreras. Local field potentials: Myths and misunderstandings. *Frontiers in Neural Circuits*, 10:101, 2016. doi: 10.3389/fncir.2016.00101.

[26] E. Erturk, S. Hashemi, and M. M. Shanechi. Cross-modal representational knowledge distillation for enhanced spike-informed LFP modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1792.

Supplementary Material

Table 1: Author contributions.

	SD	PL	VSP	MP	MCP	MV	AM	GB	SV	EV
Conceptualization	●	●	●	●						●
Data collection					●	●	●	●	●	
Data preparation	●	●		●						
Code development	●	●	●							
Model tuning	●	●	●							
Experiments	●	●	●							
Revision experiments	●		●							
Figure preparation	●	●	●							●
Writing	●	●		●						●
Editing	●	●		●						●
Funding acquisition								●	●	●

SD: Subhrajit Dey; PL: Ping-Jung Lu; VSP: Visweswar Sirish Parupudi; MP: Malhar Patel; MCP: Margaret-Estefani Conde Paredes; MV: Mihály Vöröslakos; AM: Anna Maslarova; GB: György Buzsáki; SV: Saurabh Vyas; EV: Erdem Varol.

A Dataset Details

We use four corpora. Two mouse Neuropixels datasets, the Allen Brain Observatory and the IBL Brain Wide Map, supply both pretraining and evaluation data. Two further corpora are held out entirely for transfer: NeuroNexus recordings test a different probe technology, and macaque recordings test a different species. Figure 7 gives the sample counts and splits for all four.

A.1 Allen Brain Observatory (Neuropixels Visual Coding)

The Allen Brain Observatory Visual Coding dataset [15] contains Neuropixels 1.0 recordings from 58 mice, of which we use 51 sessions, one per mouse, with up to six probes per session, 384 channels per probe, sampled at 1,250 Hz. We removed the channels that the Allen Institute's CCF registration pipeline flags as outside the brain and kept 80,313, covering 116 regions of the Allen CCF hierarchy. Each channel carries full 3D atlas coordinates (AP, DV, ML in μm).

Chunking and pretraining set. Continuous recordings are divided into non-overlapping, single-channel 3-s windows (3,750 samples at 1,250 Hz). Allen pretraining uses a fixed subset of 60,192 chunks, referred to as the Allen 75k set. The IBL pretraining subsets and scaling experiments are described below.

Evaluation splits. We use 25 sessions for training, 14 for validation and 12 for testing, with no session in more than one partition (Table 2). Each Allen session comes from a different mouse, so this is equivalently a subject-level split. The test set is divided into 5 stratified folds of roughly 94–95K chunks each, and we report the mean and standard deviation across them.

Table 2: Allen Brain Observatory, subject-level partition.

Split	**Role**	**Sessions**	**Mice**	**Probes**
SSL-train	SSL pretraining	25	25	133
Probe-val	Probe fit and early stopping	14	14	80
Test	Held-out evaluation (5×100K)	12	12	61
Total		**51**	**51**	**274**

A.2 IBL Brain Wide Map

The International Brain Laboratory (IBL) Brain Wide Map dataset [17] collects Neuropixels recordings from 12 laboratories, using standardized insertion trajectories through frontal cortex, striatum, hippocampus, thalamus, midbrain and other deep structures. Because the recordings come from many laboratories, they vary in surgical technique, probe quality and recording conditions.

Pretraining subsets and scaling. The main objective and readout ablations use a fixed IBL pretraining set of 75,000 single-channel LFP chunks. The input-duration experiment uses the 25k subset, with separate encoders trained on 1, 2, 3, 5, and 10-s windows (Supplementary Fig. 13b). For the scaling experiments, training exposure is measured by the cumulative number of chunks processed: the number of optimization steps multiplied by the batch size of 512 (Fig. 4). This quantity counts sample presentations, which may include repeated chunks, rather than the number of unique examples. Decoder-capacity comparisons use checkpoints at 35,000 steps, corresponding to 17.92 million chunk presentations.

Evaluation splits. Splits are defined at the subject level: 139 animals are partitioned into three disjoint pools of 76, 29 and 34 subjects (Table 3). No animal, session or probe appears in more than one pool, and we checked that every pairwise intersection is empty at the subject, session and probe level. This yields 5×100K-chunk test folds, 500K evaluation windows in total, across 309 canonical fine-grained region IDs in the Allen CCF. A subject-level split is stricter than a session-level one, because it tests generalization to animals with different anatomy and different surgeries.

Table 3: IBL Brain-Wide Map, subject-level partition.

Split	Role	Subjects	Sessions	Probes
SSL-train	SSL pretraining	76	281	439
Probe-val	Probe fit and early stopping	29	98	141
Test	Held-out evaluation (5×100K)	34	80	119
Total		**139**	**459**	**699**

A.3 NeuroNexus (Cross-Probe Generalization)

To evaluate transfer across recording technologies, we use the multi-shank mouse recordings associated with Angotzi et al. [18]. We refer to this region-classification benchmark as "NeuroNexus," following the evaluation files and figures; the cited study describes multi-shank SiNAPS probes. These recordings are separate from the chronic Neuropixels 2.0 dataset used for longitudinal stability analysis.

Experimental protocol. We evaluate seven recording sessions with channels labelled as cortex, CA1, CA2, CA3 or dentate gyrus. Not every region appears in every session. Inputs are single-channel 3-s LFP chunks sampled at 1,250 Hz (3,750 samples per chunk). The task is region classification; histology-aligned XYZ coordinates are not available for this benchmark.

Evaluation splits. We use seven leave-one-session-out folds. Each fold holds out one session for testing and divides the remaining six sessions into four training and two validation sessions. Within-session trial indices are partitioned into 36 training, 12 validation and 12 test trials out of 60, using seed 42. Each session subset uses only its corresponding trial indices. The splits are session-disjoint but not necessarily subject-disjoint, because some sessions come from the same animal.

The transfer models use IBL 75k `MaskPred+ChanDistc` or `MaskPred+ManipDistc` encoders. Zero-shot classification uses frozen embeddings and cosine k-NN with a labelled training-only reference bank. We select k from $\{1, 3, 5, 10, 20\}$ using validation accuracy. Adapted models train a readout or fine-tune the encoder on the training split. Figure 3 reports mean balanced accuracy and sample standard deviation across the seven held-out test sessions.

A.4 Chronic Neuropixels 2.0 Recordings

We additionally evaluate longitudinal prediction stability using five recording sessions from mouse M05 with a chronically implanted Neuropixels 2.0 probe. Three sessions were recorded in July and two in October, with each session lasting approximately 176–267 minutes. The encoder and readout were trained on IBL data and applied without adaptation to these recordings. Because histological ground truth is unavailable, this analysis examines the consistency of predicted anatomical labels within and across sessions rather than localization accuracy. These recordings are separate from the seven-session NeuroNexus/SiNAPS classification benchmark. The longitudinal stability analysis is presented in Section M.

A.5 Macaque (Cross-Species Generalization)

We evaluate cross-species transfer using Neuropixels-NHP recordings from two rhesus macaques, focusing on prefrontal cortex [20].

Dataset and splits. The analysis includes 11 probe recordings across three sessions. Macaque A contributes seven recordings from two sessions for training and validation, and macaque B contributes four recordings from one held-out session for testing. An additional dorsal premotor cortex (PMd) recording from macaque A is excluded.

The training data cover four cortical classes: areas 45, 46, 8, and 9/46. Area 46 is a separate training class but is not represented in macaque B. For evaluation, the two macaque B probes recorded anterior to the frontal eye field are assigned to area 8, and predictions are restricted to the three test classes: areas 45, 8, and 9/46.

Each run uses 94,313 training windows and 9,432 validation windows from macaque A, sampled separately from the same seven probes. Training samples are approximately balanced across the four classes. The test set contains 24,000 windows from macaque B, with 6,000 windows per probe. Each window contains 3 s of single-channel LFP. The million-scale counts in the dataset overview describe the available recording corpus rather than the sampled windows used in each run.

Evaluation rationale. This setting tests whether representations pretrained on mouse LFP support anatomical classification in macaques after supervised adaptation on macaque A and evaluation on the held-out macaque B. It therefore combines cross-species representation transfer with cross-subject evaluation. Because the dataset contains only two animals, the results provide a limited assessment of generalization across macaques.

A.6 Terminology: Allen CCF Hierarchy and Depth Levels

The Allen Common Coordinate Framework organizes brain regions into a tree with 10 depth levels, from d0 at the root to d9 at the finest leaf. We report classification at several depths. Depth-0 assigns every channel to "brain" and is trivial; mapping the 309 IBL fine labels up the tree leaves 31 classes at depth 3, 107 at depth 5 and 309 at depth 9, where individual cortical layers and fine nuclei are resolved. Balanced accuracy at each depth shows how performance changes with anatomical resolution.

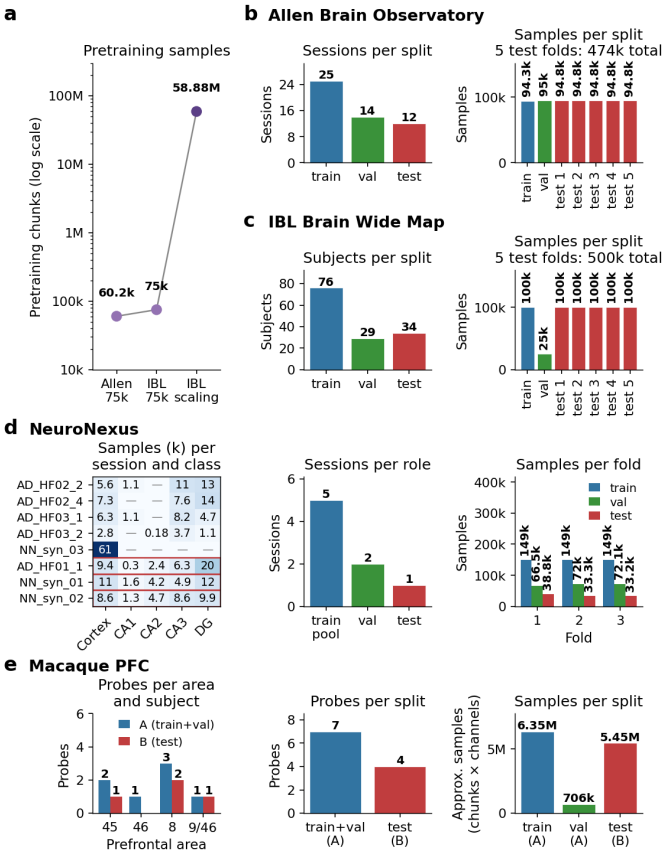


Figure 7: **Datasets and evaluation splits. (a)** Number of 3-s LFP chunks used for the Allen and IBL 75k pretraining sets and for the largest scaling run. **(b–c)** Session and subject splits, respectively, for the Allen Brain Observatory and IBL Brain-Wide Map datasets, together with the number of windows in each split. The five held-out folds contain 474k Allen windows and 500k IBL windows in total. **(d)** NeuroNexus sessions, class coverage, and the three rotating held-out-session folds used for cross-probe transfer. **(e)** Macaque prefrontal-cortex probes and approximate numbers of channel windows. Macaque A supplies the training and validation data, and macaque B is held out for testing. Splits are disjoint by session, subject, or probe as appropriate for each dataset.

B Cross-Species Transfer to Macaque Recordings

We next ask whether representations learned from mouse LFP are useful for identifying brain regions in macaques. We fine-tune the mouse-pretrained LFPformer on recordings from macaque A and evaluate it on a second, held-out macaque B. The task distinguishes three cortical regions: areas 45, 8, and 9/46. This evaluates transfer with labeled macaque training data, rather than zero-shot prediction.

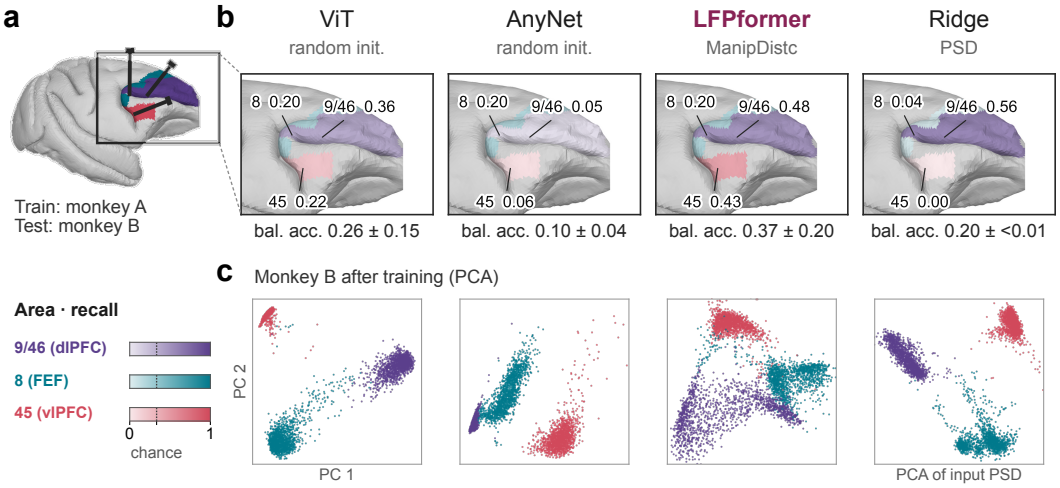


Figure 8: **Macaque cross-subject transfer after fine-tuning. (a)** Models are trained on recordings from macaque A and evaluated on the held-out macaque B in prefrontal areas 45, 8, and 9/46. **(b)** Per-area recall and balanced accuracy for ViT, AnyNet, mouse-pretrained LFPformer with `ManipDistc`, and a ridge classifier using power-spectral-density features. Values are mean ± SD across ten runs. ViT and AnyNet are trained from random initialization; LFPformer is initialized with the mouse-pretrained encoder and fully fine-tuned on macaque A. **(c)** PCA projections of the held-out macaque-B representations, coloured by cortical area; the ridge panel shows the PSD input features.

Figure 8 shows that LFPformer achieves the highest mean balanced accuracy among the evaluated models: 0.37 ± 0.20, compared with 0.26 ± 0.15 for ViT, 0.10 ± 0.04 for AnyNet, and 0.20 ± 0.003 for Ridge. LFPformer has higher mean recall in areas 45 and 8, while Ridge has higher recall in area 9/46. These results suggest that mouse-pretrained representations can support anatomical classification in macaques after fine-tuning. Anatomical classification may help researchers interpret neural population dynamics within the CTD framework [21] by identifying the brain areas contributing to a recording. This could be useful when a probe spans multiple cortical areas. Our macaque experiments evaluate anatomical labels at the probe level; identifying boundaries within individual probes and using these predictions in CTD analyses remain future directions. The macaque dataset contains only two subjects, which limits the diversity of training data and may make higher-capacity models more prone to overfitting. Additional recordings from more animals could improve generalization, reduce run-to-run variability, and enable a more reliable assessment of cross-species transfer.

C Group-Wise Localization and Depth-5 Confusion Analysis

C.1 Localization Across Regions, Subjects, Probes, and Laboratories

Aggregate localization error can obscure differences across anatomical structures and recording groups. Figure 9 compares localization across Allen fine regions ($n = 309$), subjects ($n = 34$), probes ($n = 119$), and laboratories ($n = 12$). Each point represents one group, with the error of `MaskPred+ManipDistc` on the horizontal axis and a comparator on the vertical axis. Points above the diagonal indicate lower error for `MaskPred+ManipDistc`.

All five encoder systems use Gaussian linear readouts with 4,614 parameters. Each readout predicts coordinate means and variances from a 768-dimensional embedding independently for each window. Repeated predictions for each physical channel are combined using inverse-variance weights, followed

by an unweighted Kabsch fit of the known probe geometry to the pooled predictions. Within each group, RMSE is computed over each probe's channels and then averaged across probes with equal weight. For the combined-objective encoders, these group-level results are averaged across eight decoder seeds.

`MaskPred+ManipDistc` has lower error than `MaskPred` alone in 266 of 309 regions, `ChanDistc` alone in 266 regions, and `CCFDistc` alone in 270 regions. Compared with `MaskPred+ChanDistc`, it has lower error in 252 of 309 regions, 29 of 34 subjects, 95 of 119 probes, and all 12 laboratories. Its localization advantage therefore persists when the readout architecture and geometric refinement are matched.

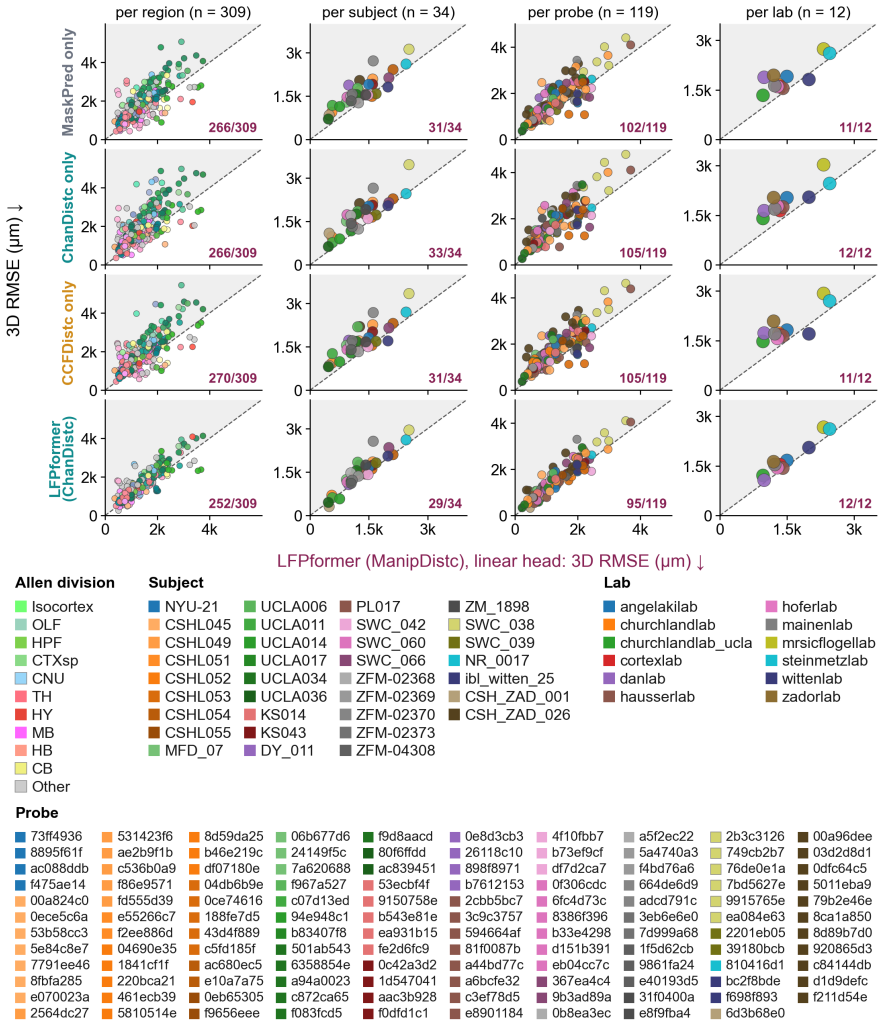


Figure 9: **Group-wise localization with Gaussian linear readouts.** Each point compares the localization error of `MaskPred+ManipDistc` (horizontal axis) with a comparator (vertical axis). Rows show `MaskPred` only, `ChanDistc` only, `CCFDistc` only, and `MaskPred+ChanDistc`; columns group results by Allen fine region, subject, probe, and laboratory. Points above the diagonal indicate lower error for `MaskPred+ManipDistc`, and annotations count these groups. All five systems use Gaussian linear readouts. Window predictions are combined within each channel using inverse-variance weights, followed by Kabsch fitting of known probe geometry and left-hemisphere-folded scoring. Within each group, channel RMSE is computed separately for each probe and averaged across probes with equal weight. Combined-objective results are averaged across eight decoder seeds; single-objective results use one saved decoder fit each. Colours identify the groups shown in the legends. Ground-truth test coordinates are used only for evaluation.

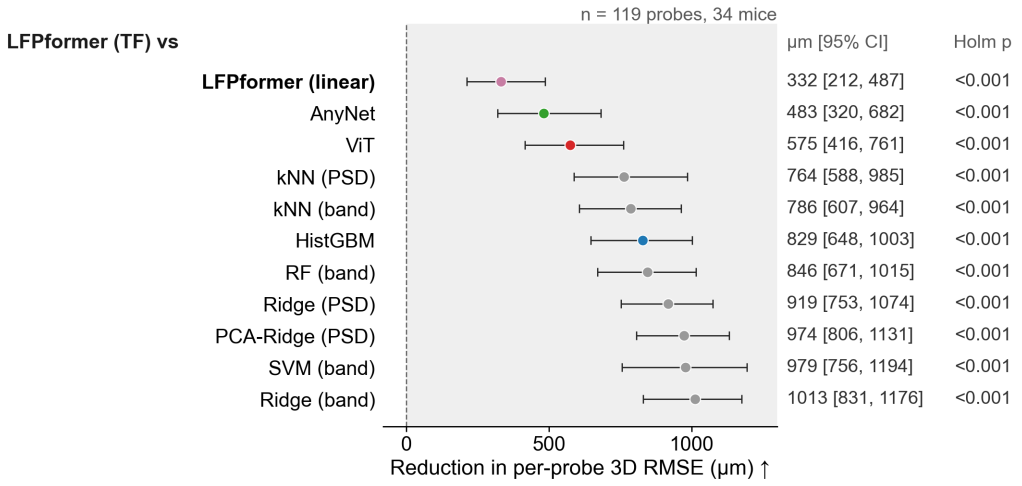


Figure 10: **Localization error reductions relative to LFP-only comparators.** Dots show the mean reduction in probe RMSE (comparator minus LFPformer); positive values favour LFPformer. Bars show pointwise 95% confidence intervals from 10,000 bootstrap resamples of mice. Two-sided tests use 100,000 sign flips at the mouse level, with Holm correction across the 11 comparisons. Uncertainty reflects variation across test mice, conditional on the fitted models, rather than variation across encoder pretraining runs.

C.2 Statistical comparison of localization methods

LFPformer with a transformer decoder achieved lower mean localization error than all 11 LFP-only comparators shown here (Holm-adjusted $p < 0.001$). We compared the full set of 119 test probes from 34 mice, averaging each probe's error over the available decoder seeds. We resampled mice rather than individual probes to account for probes from the same animal being related.

C.3 Depth-5 Classification from Predicted Coordinates

A predicted coordinate can be close to its target yet fall in a different anatomical structure. Figure 11 examines this distinction by mapping channel-pooled, Kabsch-refined coordinates to the Allen atlas at hierarchy depth 5. Every system uses the same weighted channel pooling and Kabsch procedure. Refined channel predictions are assigned back to their corresponding test windows, and all systems are scored on the same 362,314 windows from the 309 canonical fine regions.

Six regions are displayed individually, and all remaining regions are pooled into Other. These seven displayed categories therefore do not represent the complete depth-5 label set. Rows are normalized within each true category: diagonal entries give the percentage of test windows assigned correctly, while off-diagonal entries show their alternative assignments. For the combined-objective systems, confusion counts are averaged across eight decoder seeds before row normalization.

With Gaussian linear readouts, balanced accuracy is 0.335 for `MaskPred` only, 0.321 for `ChanDistc` only, and 0.292 for `CCFDistc` only. `MaskPred+ChanDistc` achieves 0.330 ± 0.012, and `MaskPred+ManipDistc` achieves 0.401 ± 0.001, reported as mean $\pm$ sample standard deviation across eight decoder seeds. `MaskPred+ManipDistc` retains the highest mean balanced accuracy in this comparison, while `MaskPred+ChanDistc` is close to `MaskPred` alone. The confusion matrices show that performance varies across anatomical categories, complementing the distance-based localization analysis.

Ground-truth test coordinates are used to determine evaluation labels and errors, never to fit the probe trajectory. Kabsch refinement uses only known probe geometry and predicted coordinates.

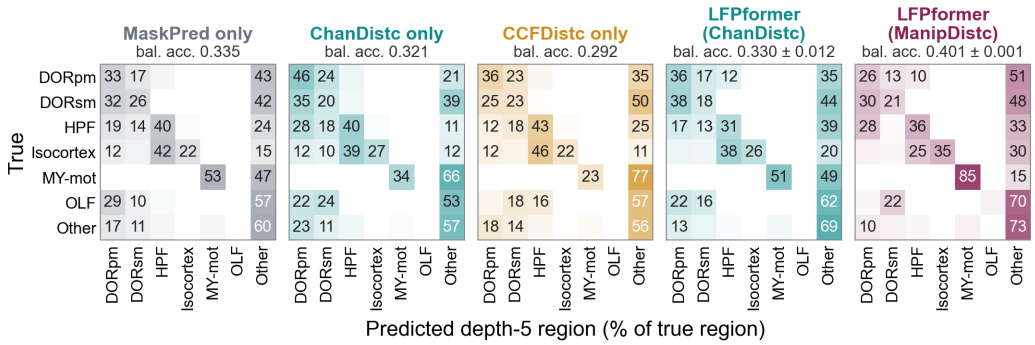


Figure 11: **Depth-5 anatomical classification with Gaussian linear readouts.** Rows indicate true anatomical categories and columns indicate categories obtained by mapping predicted coordinates to the Allen atlas. Six selected regions are displayed individually, and all remaining regions are combined as Other. Cells show row-normalized percentages; numerical annotations below 10% are omitted. All five systems use Gaussian linear readouts, inverse-variance channel pooling, and Kabsch fitting of known probe geometry. Refined coordinates are folded into the left hemisphere before atlas lookup. All systems are evaluated on the same 362,314 test windows from the 309 canonical fine regions. For the combined-objective systems, confusion counts are averaged across eight decoder seeds before normalization, and titles report mean ± sample standard deviation of balanced accuracy. Single-objective systems use one saved decoder fit each.

D Model Architecture and Training Pipeline

D.1 Backbone Encoders

All three backbones are used as single-channel encoders: each LFP channel is embedded on its own, and nothing in the encoder sees more than one channel at a time. We compare three audio-pretrained backbones at the base scale and run the wav2vec 2.0 line at the large scale as well. Table 4 gives the three base configurations, which share a signal-processing front end and differ in their Transformer core.

Table 4: **Backbone architectures.** All backbones use pretrained audio foundation model weights and share the same convolutional feature extractor design (7 layers, GELU activations, group normalization). Parameter counts include the feature extractor, Transformer encoder, and quantizer/projector modules.

Backbone	Pretrained ID	Layers	Heads	d_{model}	d_{ff}	Params
Wav2Vec 2.0	`facebook/wav2vec2-base`	12	12	768	3072	94.4 M
Data2Vec	`facebook/data2vec-audio-base`	12	12	768	3072	94.4 M
Whisper	`openai/whisper-base` (encoder)	6	8	512	2048	72.6 M

Convolutional feature extractor. The raw waveform passes through a 7-layer 1D convolutional stack with 512 output channels in every layer. Kernel sizes are $(10, 3, 3, 3, 3, 2, 2)$ and strides $(5, 2, 2, 2, 2, 2, 2)$, a total downsampling factor of $\prod_i s_i = 320$. Each convolution is followed by group normalization and a GELU activation, and the convolutional layers carry no bias. A convolutional relative positional encoding layer (kernel 128, 16 groups) sits between the stack and the Transformer blocks.

Signal preprocessing. Each 3-second LFP chunk (3,750 samples at 1,250 Hz) is resampled on GPU to 16,000 Hz with `torchaudio.functional.resample`, giving the 48,000-sample waveform that the pretrained audio models expect. The convolutional stack turns this into 149 latent frames, the value F used in the compute estimate of Figure 4c. We tried per-sample z-score normalization and left it off, because the pretrained audio backbones expect unnormalized waveforms.

Large encoder. The scaling results also use a large wav2vec 2.0 encoder, which embeds each chunk in 1,024 dimensions rather than 768 and holds 317.5M parameters against 95.1M for the base model. Pretraining batches hold 512 chunks. The large encoder supplies the solid lines in Figure 4 and the embeddings analyzed in Figure 6, and the best frozen localization we report comes from its `ManipDistc` checkpoint.

D.2 Embedding Structure During Pretraining

Supplementary Figure 12 follows the representations of the same 100,000 held-out IBL windows from step 5,000 to step 45,000. For each encoder size, we fit one PCA basis to a balanced reference containing 1,000 matched windows per checkpoint across four objectives and nine checkpoints (36,000 embeddings). We then project all windows onto the first two axes of this shared basis. Keeping the axes fixed makes it possible to follow changes over training without fitting a new projection at every checkpoint.

How the explained variance changes. For `ManipDistc`, the fraction of variance captured by these two axes falls from 37% to 31% for Base and from 43% to 33% for Large between steps 5k and 45k. For `CCFDistc`, it falls from 38% to 32% and from 45% to 35%, respectively. These are overall trends rather than steady decreases at every checkpoint. The other objectives behave differently: Base `MaskPred` stays near 10%, Base `ChanDistc` rises from 16% to 18%, and Large `ChanDistc` remains near 32%; Large `MaskPred` fluctuates and ends below its initial value.

The percentages describe the share of variation visible in this particular two-dimensional projection. A lower percentage means that a larger share lies outside the two displayed axes. It does not, by itself, show that total embedding variance has decreased, that the representation has collapsed, or that its intrinsic dimensionality has increased. Because the axes are fitted jointly across objectives and checkpoints, they are not necessarily the two directions that explain the most variance at any individual checkpoint. We use these plots to describe representation changes alongside the downstream scaling results, rather than as a measure of localization quality.

D.3 Downstream Regression Head

Within a 3-second chunk the backbone averages its latent frames, so every channel-chunk leaves the encoder as one embedding: 768-dimensional for the base wav2vec 2.0 and data2vec models, 512 for Whisper, 1,024 for the large encoder. This temporal average is a separate step from the later pooling of coordinate predictions across repeated chunks of the same channel.

Three readouts consume that embedding. A linear Gaussian readout maps it straight to a coordinate and a variance. A per-channel MLP reads each embedding on its own. A probe-level Transformer instead takes the channels of one probe as a sequence ordered by on-probe depth and attends across them, so it mixes channels inside a probe and never between probes.

The per-channel MLP on a base embedding has a shared trunk:

$$\mathbf{h}_0 = \mathrm{Pool}(\mathbf{z}) \in \mathbb{R}^{768} \;\rightarrow\; \mathbf{h}_1 \in \mathbb{R}^{512} \;\rightarrow\; \mathbf{h}_2 \in \mathbb{R}^{256} \;\rightarrow\; \mathbf{h}_3 \in \mathbb{R}^{128},$$

with ReLU activations between each linear layer. Two heads branch off $\mathbf{h}_3$, each mapping it to $\mathbb{R}^3$: a mean head $\boldsymbol{\mu}$ predicting the AP, DV and ML coordinates in μm, and a log-variance head $\log \boldsymbol{\sigma}^2$ predicting a per-axis variance. They are trained together with the heteroscedastic Gaussian negative log-likelihood

$$\mathcal{L}_{\mathrm{reg}} = \sum_{k \in \{AP,DV,ML\}} \frac{(y_k - \mu_k)^2}{2\sigma_k^2} + \frac{1}{2} \log \sigma_k^2,$$

so the model produces a coordinate and an uncertainty for each axis.

Readout capacity spans more than an order of magnitude. On the large encoder, the three-layer per-channel MLP has 303,942 trainable parameters and the four-block Transformer ($d = 256$, 8 heads, feed-forward width 512) has 2,372,870. Averaged over eight decoder seeds, the MLP reaches 1,359 μm Kabsch error against the Transformer's 1,071 μm, and the gap is far wider before refinement: 1,988 against 1,104 μm raw. Attending over the channel sequence helps most where the geometric prior has not yet been applied. Figure 4b compares the full family at a matched pretraining budget, and Appendix G sets out what each comparison does and does not isolate.

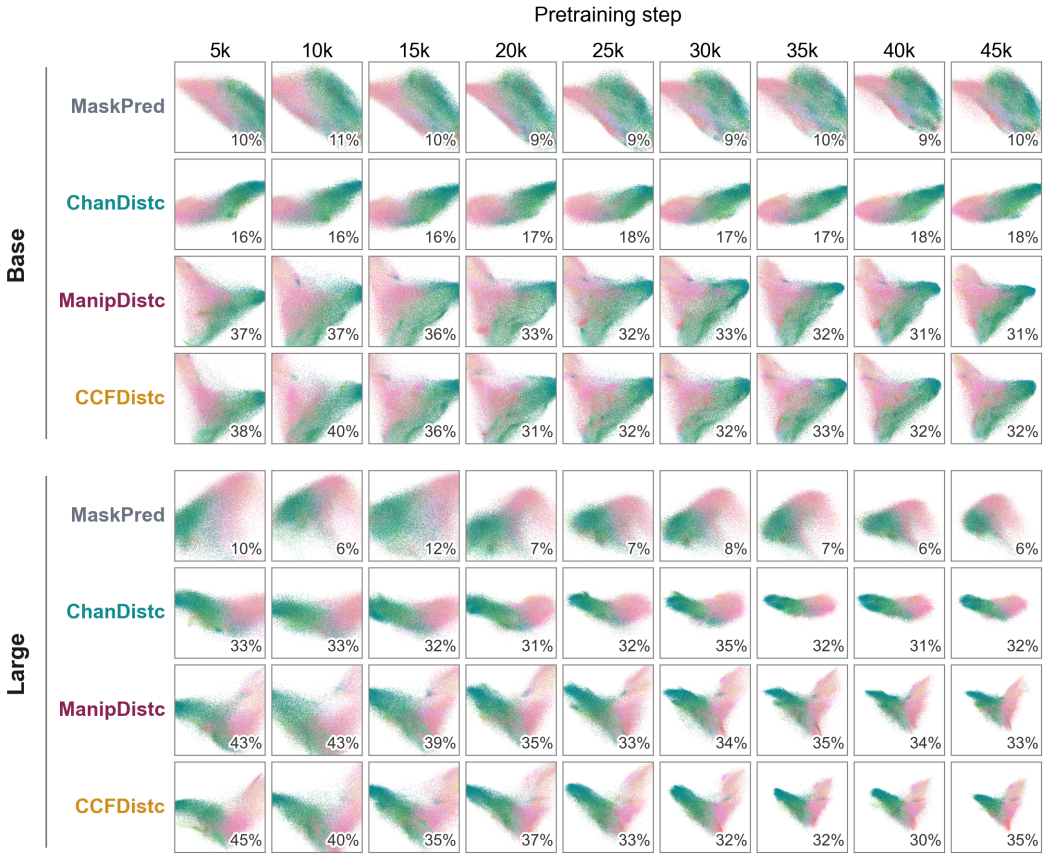


Figure 12: **Evolution of LFP embeddings during pretraining.** Each panel shows the same 100,000 held-out IBL windows, coloured by brain region. Rows compare `MaskPred` and its combinations with `ChanDistc`, `ManipDistc` and `CCFDistc` for Base (top) and Large (bottom) encoders; columns show pretraining steps from 5k to 45k. All checkpoints of a given encoder size use the same two PCA axes. Percentages give the fraction of each checkpoint's embedding variance captured by these axes. This fraction generally decreases for `ManipDistc` and `CCFDistc` as pretraining progresses, although the decrease is not monotonic. Plotting limits are fixed across steps within each row and zoomed to the pooled 2nd–98th percentiles, cropping some outer points. Plotting ranges differ between rows, and Base and Large use separate PCA fits.

D.4 Kabsch Refinement

Kabsch refinement [12] takes exactly two inputs: the known Neuropixels channel geometry and the model's own predictions. No ground-truth coordinate enters the fit. The step snaps the template probe geometry onto the predicted cloud, which amounts to a strong geometric prior: whatever the model predicts, the answer is required to be a physically realizable probe.

Setup. Channel i has a fixed offset $\mathbf{p}_i \in \mathbb{R}^3$ on the shank, known from the device specification before any recording is made. For Neuropixels 1.0 this is the staggered four-column layout, $\mathbf{p}_i = (x_i, 0, z_i)$ with $x_i \in \{11, 27, 43, 59\}$ μm and z_i advancing in 20 μm steps along the shank. Let $\hat{\mathbf{x}}_i \in \mathbb{R}^3$ be the predicted CCF coordinate for channel i, after inverse-variance pooling over that channel's time chunks. Per probe we find the rigid pose of the physical shank that best explains the predicted cloud,

$$\mathbf{R}^\star, \mathbf{t}^\star \;=\; \arg\min_{\mathbf{R}\in SO(3),\, \mathbf{t}\in\mathbb{R}^3} \sum_{i=1}^{N} \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \hat{\mathbf{x}}_i\|_2^2 , \tag{6}$$

All channels receive equal weight in the Kabsch fit. Inverse-variance weighting is applied only when pooling repeated window predictions within each channel.

Solution. The minimiser is closed-form. Center both sets at their unweighted centroids $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_{\hat{x}}$, form the cross-covariance $\mathbf{H} = \sum_i (\mathbf{p}_i - \boldsymbol{\mu}_p)(\hat{\mathbf{x}}_i - \boldsymbol{\mu}_{\hat{x}})^\top$, take its SVD $\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, and set

$$\mathbf{R}^\star = \mathbf{V}\,\mathrm{diag}\big(1, 1, \det(\mathbf{V}\mathbf{U}^\top)\big)\,\mathbf{U}^\top, \qquad \mathbf{t}^\star = \boldsymbol{\mu}_{\hat{x}} - \mathbf{R}^\star \boldsymbol{\mu}_p. \tag{7}$$

Discarding $\boldsymbol{\Sigma}$ is what makes the transform rigid: the probe may rotate and translate, but it cannot stretch to fit the predictions. The $\det(\cdot)$ term rules out reflections. We additionally enforce the known insertion direction, channel 0 deeper in DV than channel 383, by applying a $180°$ flip whenever the unconstrained fit returns an inverted pose. The refined coordinate of channel i is $\hat{\mathbf{x}}_i^{\mathrm{ref}} = \mathbf{R}^\star \mathbf{p}_i + \mathbf{t}^\star$.

Inputs and outputs. Stated as a map, the procedure is $(\{\mathbf{p}_i\}, \{\hat{\mathbf{x}}_i\}) \mapsto \{\hat{\mathbf{x}}_i^{\mathrm{ref}}\}$: the template and the predictions go in, refined predictions come out. Ground-truth coordinates are used only afterwards, to score the result.

Worked example. Table 5 follows six channels through the procedure. A refined position is computed from its *template* point, not from its own prediction; individual predictions enter only through $\mathbf{R}^\star$ and $\mathbf{t}^\star$. The effect is to impose the hardware constraint. Channels 0 and 1 sit 32 μm apart on the physical probe, but the raw predictions place them 647 μm apart, which no probe can realize; after fitting they are 32 μm apart again.

Table 5: Kabsch refinement, worked example (six channels, μm). The template is the manufactured channel layout. Refined positions are $\mathbf{R}^\star \mathbf{p}_i + \mathbf{t}^\star$: each is computed from its own template point, and the predictions enter only through the single pose $\mathbf{R}^\star, \mathbf{t}^\star$ that the whole cloud determines.

Channel	Template $\mathbf{p}_i$	Predicted $\hat{\mathbf{x}}_i$	Refined $\hat{\mathbf{x}}_i^{\mathrm{ref}}$
ch0	(11, 0, 0)	(6332, 2179, 514)	(6048, 2090, 504)
ch1	(43, 0, 0)	(5705, 2050, 421)	(6025, 2068, 497)
ch2	(27, 0, 20)	(6017, 1987, 502)	(6036, 2085, 481)
ch3	(59, 0, 20)	(5976, 1864, 658)	(6014, 2063, 475)
ch4	(11, 0, 40)	(6183, 2408, 543)	(6047, 2102, 465)
ch5	(43, 0, 40)	(5981, 2002, 245)	(6024, 2080, 459)

Oracle variant. Fitting the template to the true channel coordinates, instead of to the predictions, gives 107.2 µm. That is the best this refinement can produce: the residual measures how far the atlas -registered channel positions depart from any rigid Neuropixels 1.0 shank, so it is identical for every decoder we evaluate and no model can go below it under this protocol. It needs test-set ground truth, so it is not a method a user could run, and it never contributes to a headline number.

Refinement removes each channel's scatter about the fitted probe, so the error that remains reflects how well the model places and orients the insertion as a whole. It does not remove a bias shared by every channel on that probe: the pose is estimated from the predictions themselves, so a shared offset carries the fit along with it.

E Hyperparameter Tuning Pipeline

We tune hyperparameters separately for each backbone with Ray Tune [22], using HEBO [24] as the search algorithm and ASHA, Asynchronous Successive Halving [23], as the trial scheduler. ASHA gives every trial a grace period of 3 epochs, then at each rung keeps the best third and stops the rest; the maximum trial length is the run's epoch budget. Earlier exploratory runs used Optuna, but every configuration reported in this paper was selected with HEBO.

The search minimizes `loss_per_mask`, the validation loss divided by `mask_time_prob` $\times$ `mask_time_length`. Raw validation loss falls when fewer positions are masked, so a search on validation loss alone drifts toward trivial masking rates; dividing by the masked fraction removes that incentive.

Scale-dependent warmup. Warmup steps and the warmup start factor are the two settings we did not hold at their searched values. Warmup is lengthened for larger-scale training, from 400 steps to 1,200, while the start factor drops from 0.01 to 0.001, so that warmup stays a roughly constant fraction of training as the number of steps per epoch rises with scale.

Geometry kernel width. The `ChanDistc` kernel width σ sets how fast the target similarity falls off with on-probe distance. We chose it on the downstream task rather than on the pretext metric: sweeping $\sigma \in \{100, 200, 300, 500, 750, 1000\}$ μm at the 75K scale and scoring frozen Kabsch-refined 3D RMSE on the validation set, and we get 500 μm. Away from that minimum the curve is shallow, with the other five widths costing between 57 and 169 μm, so the width is worth choosing but the result does not hinge on hitting it exactly.

Table 6: Hyperparameter sweep (Ray Tune + HEBO). Selected values are those used in the reported IBL `MaskPred` + `ChanDistc` runs, listed in full in Table 7; † Warmup was lengthened for larger-scale training.

Hyperparameter	**Search range**	**Selected value**
learning rate	5×10^{-5} – 1×10^{-3}	1.02×10^{-4}
weight decay	1×10^{-4} – 0.1	0.080
hidden dropout	0.0 – 0.3	0.097
attention dropout	0.0 – 0.3	0.084
final dropout	0.0 – 0.3	0.0
mask probability	0.05 – 0.50	0.079
mask span length	5 – 60	5
unfrozen top-k layers	$\{0, 2, 4, 6, 8, 12\}$	12
warmup steps	$\{50, 100, 200, 400\}$	$400 \rightarrow 1200$ †
warmup start factor	0.001 – 0.1	$0.01 \rightarrow 0.001$ †
`ChanDistc` kernel width σ	$\{100, 200, 300, 500, 750, 1000\}$ μm	500 μm

Table 7: IBL SSL hyperparameters: 75K cross-negative SSL + `ChanDistc` joint, σ=500 μm. Wav2Vec2-base initialized from `facebook/wav2vec2-base` and fully unfrozen (12/12 layers); jointly optimized with masked InfoNCE prediction (λ_{ssl}=1.0) and the channel-distance contrastive loss (λ_{dist}=1.0, σ=500 μm, τ=0.2) over Neuropixels 1.0 4-column shank coordinates. Quantizer uses 2 groups $\times$ 320 codevectors with diversity weight 0.1 (HuggingFace `wav2vec2-base` default). Training: 200 epochs target, batch 64 (4 probes $\times$ 16 samples), AdamW (lr= 1.02×10^{-4}, wd= 0.08), 400-step warmup, plateau LR schedule with floor $\eta_{\min}$=10^{-5}, gradient clipping 1.0, AMP off.

Backbone	
backbone	wav2vec2
pretrained_model	`wav2vec2-base`
rand_init	false
hidden_size	768
num_hidden_layers	12
num_attention_heads	12
intermediate_size	3072
unfreeze_top_k_layers	12 (full)
Quantizer	
codevectors / group	320
codevector groups	2
codebook size	640
diversity_loss_weight	0.1
gumbel_temp_init	2.0
gumbel_temp_min	0.5
gumbel_temp_decay	0.999995
Loss weights	
ssl_lambda (`MaskPred`)	1.0
dist_lambda (`ChanDistc`)	1.0
Masking	
mask_time_prob	0.079
mask_time_length	5
num_negatives	100
cross_sample_negatives	true
min_masks	2

Optimizer	
lr_ssl	1.02×10^{-4}
weight_decay	0.080
hidden_dropout	0.097
attention_dropout	0.084
final_dropout	0.0
warmup_steps	400
warmup_start_factor	0.01
grad_clip_norm	1.0
lr_schedule	plateau
plateau_factor	0.5
plateau_patience	5
eta_min	1.0×10^{-5}
ChanDistc	
dist_temperature	0.2
dist_sigma (μm)	500.0
dist_target_fraction	1.0
samples_per_probe	16
num_probes	4
coord_dim	2
queue_size	0
Training	
batch_size_ssl	64
ssl_epochs	200
amp	false
atlas_depth	9
num_workers	8
prefetch_factor	16
embed_interval	5
probe_interval	5
probe_epochs	20
early_stop_patience	30

F Inference cost and input duration

Inference timing. We benchmarked inference components on 500,000 single-channel, 3-s LFP chunks from 119 held-out IBL probes, using a fine-tuned wav2vec 2.0 checkpoint trained on the IBL 75k set with joint masked-prediction, channel-distance and coordinate-regression objectives, with linear MSE and Gaussian heads. The channel-distance kernel width is 500 micrometers.

Table 8 separates GPU resampling from 1250 Hz to 16 kHz, encoder inference, coordinate prediction, and CPU Kabsch fitting. The head timer includes both linear heads. The CPU benchmark fits probe geometry directly to repeated Gaussian-mean chunk predictions, without the uncertainty-weighted channel-pooling step used in the final localization pipeline. These timings therefore characterize this benchmark configuration, not the complete `ManipDistc`–Transformer inference pipeline.

The encoder accounts for approximately 99.4% of the A100 runtime and 90.5% of the H200 runtime. The estimated encoder computation is 41.56 GFLOPs per 3-s chunk, or 20.78 PFLOPs across 500,000 chunks. Batch size and worker count differ between the two runs, so the two columns are not a like-for-like hardware comparison: the A100 column is batched throughput and the H200 column is single-request latency. We counted supported forward-pass operations with PyTorch

Table 8: Component timings on the held-out IBL test set. The A100-80GB run used batch size 64 and four data-loading workers; the H200 run used batch size 1 and one worker. Per-chunk values are total stage time divided by 500,000, rounded to three decimals. They are amortized processing costs, not individual request latencies. GPU stages use CUDA-event timers; the CPU fit uses a wall-clock timer. The total is their sum and excludes data-loading, host-to-device transfer, prediction decoding and aggregation outside those timers.

	A100-80GB, batch 64		H200, batch 1	
Stage	Total (s)	ms/chunk	Total (s)	ms/chunk
Preprocessing/resampling	4.4	0.009	202.5	0.405
Audio encoder	933.9	1.868	2311.8	4.624
Linear heads (MSE + Gaussian)	1.3	0.003	41.1	0.082
Direct chunk-level Kabsch fit	0.3	0.001	0.4	0.001
Sum of timed stages	939.9	1.880	2555.8	5.112

`FlopCounterMode` on one 3,750-sample input, resampled to 48,000 samples. The encoder count is separate from preprocessing and both linear heads; CPU Kabsch operations are excluded.

Processing cost, recording duration and whole-probe latency are three different quantities. Each input needs 3 s of recorded signal before it can be processed at all, and a probe produces many channel chunks at once. The summed duration of the evaluated chunks is 416.7 channel-hours, which is not 416.7 hours of distinct recording time. These offline timings motivate streaming evaluation; they do not measure acquisition, probe-to-GPU transfer, or a complete online localization system.

Input duration. Longer windows give the encoder more context per prediction and cost more to train on. Table 9 reports both sides of that trade at the 25k scale.

Table 9: Timing details accompanying the window-length experiment in Supplementary Fig. 13b. Separate `MaskPred+ChanDistc` encoders use the 25k pretraining set and frozen linear readouts. Inference values are median wall times on an H100 for data loading, host-to-device transfer and encoder embedding extraction, with batch size 1 and one worker (100 repetitions after 20 warm-up iterations). The latency benchmark does not run a coordinate head, channel pooling or Kabsch refinement. SSL training time is reported per epoch. Kabsch RMSE is the Gaussian-head value on the IBL test set, for reference; the MSE-head values and the regional breakdown are in the figure.

Input duration (s)	SSL training (min/epoch)	Inference (ms/chunk)	Kabsch RMSE (μm)
1	1.9	4.3	2067.5
2	5.1	4.6	1939.6
3	6.5	4.7	1909.4
5	9.4	5.1	1776.3
10	18.9	6.9	1734.6

Going from 1 s to 10 s lowers Kabsch error by 333 μm, but training time per epoch rises tenfold while inference rises only about 1.6-fold, from 4.3 to 6.9 ms per chunk. Training cost, not inference cost, is what makes long windows expensive at scale. We used 3-s windows to match the lfp2vec baseline, and longer windows also need a longer acquisition interval before any prediction can be made. Supplementary Fig. 13b shows the same trade-off for the Gaussian readout with the regional results.

G Decoder architecture and loss ablations

Two choices sit between the frozen encoder and a coordinate: how much capacity the readout has, and which loss trains it. This appendix separates them, because the figures that address them use different encoders and are easy to read as one experiment.

Supplementary Fig. 13c compares MSE and Gaussian heads for linear, MLP and Transformer readouts on four frozen 75k encoders. It reports MSE RMSE minus Gaussian RMSE at three inference stages,

so positive values favour the Gaussian head. The error bars combine the seed standard deviations of the two methods and are not paired confidence intervals. Panel a of the same figure evaluates region classification at several depths of the Allen hierarchy.

Raw RMSE evaluates the prediction for each channel chunk. Channel-pooled RMSE evaluates one prediction per channel after combining repeated chunks over time: per-axis inverse-variance pooling for Gaussian heads and mean pooling for MSE heads. Kabsch RMSE evaluates the probe trajectory after rigidly fitting known probe geometry to the pooled predictions, without using ground-truth test coordinates in the fit. Because the loss comparison also changes the pooling rule, it does not isolate the training loss from the effect of pooling.

Within each chunk, the audio encoder averages its temporal features to produce one embedding. This is separate from the later pooling of coordinate predictions across repeated chunks. A per-channel MLP reads each embedding independently; the Transformer uses self-attention within a depth-ordered probe sequence.

Figure 4b compares linear, deeper MLP and Transformer readouts, with trainable parameter counts. That experiment uses base and large encoders at step 35,000 (17.92 million pretraining chunks), whereas Supplementary Fig. 13c uses the 75k encoders. The two figures therefore address different questions: readout capacity at a matched scaling budget, and loss choice at the 75k pretraining scale. The non-linear readouts predict absolute XYZ directly.

On the large encoder the ordering is clear. A three-layer per-channel MLP with 303,942 parameters reaches 1,359 μm Kabsch error, and the four-block Transformer with 2,372,870 reaches 1,071 μm, each averaged over eight decoder seeds. Before refinement the two are 1,988 and 1,104 μm, so most of the Transformer's advantage is already present in the raw predictions rather than created by the rigid fit. Earlier readout-capacity numbers from the rebuttal are not comparable with these: they were measured with pose-decomposed Transformer readouts on a 768-dimensional base encoder, and should not be read as describing the direct-XYZ readouts used here.

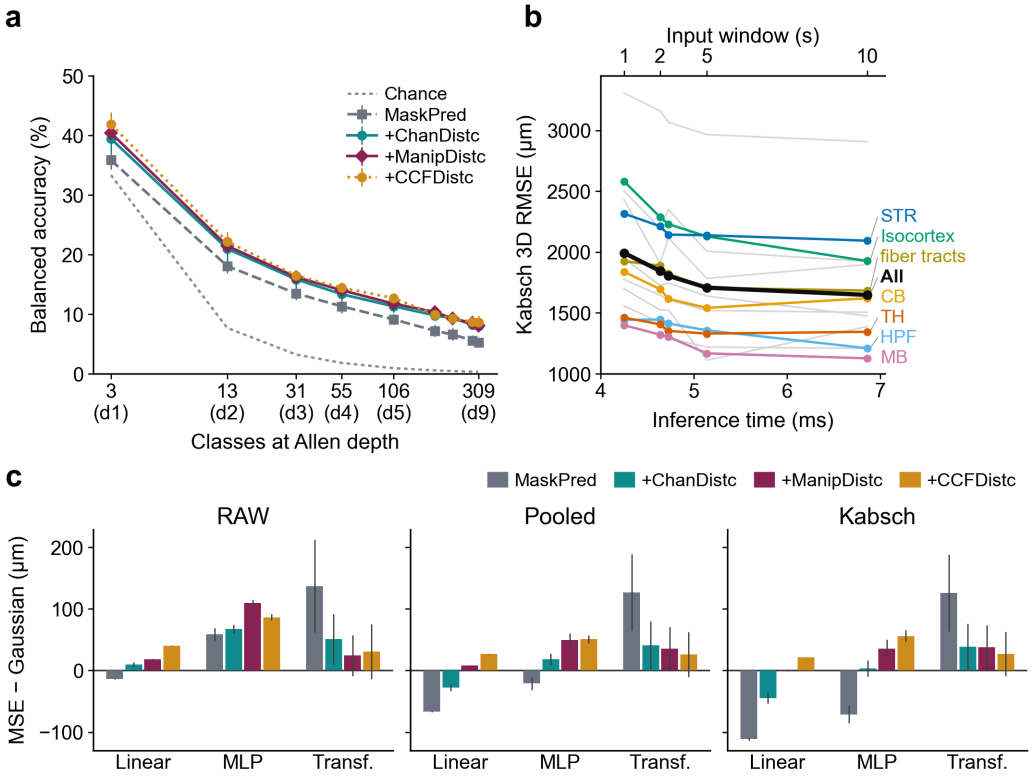


Figure 13: **Pretraining objectives, input duration, and decoder loss. (a)** Brain-region balanced accuracy at different depths of the Allen anatomical hierarchy for four frozen encoders pretrained on the 75k set, each evaluated with a Transformer Gaussian readout. Coordinates are combined by inverse-variance channel pooling, refined with Kabsch, and mapped to the Allen atlas. The horizontal axis gives the number of evaluated classes at each hierarchy depth (d1–d9). Lines and error bars show the mean and sample standard deviation across eight decoder seeds; the dotted line indicates chance performance. **(b)** Input-duration experiment using separate `MaskPred+ChanDistc` encoders pretrained on the 25k set with 1, 2, 3, 5, and 10 s windows, each evaluated with a frozen linear Gaussian readout. Lines show Kabsch-refined 3D RMSE against median embedding-extraction time on an H100 with batch size 1 and one data-loading worker. Timing includes data loading, host-to-device transfer, and encoder inference; it excludes the coordinate readout, channel pooling, and Kabsch refinement. The top axis indicates input duration. Coloured lines show the seven Allen major divisions with at least 1,700 test channels, grey lines show smaller divisions, and black shows all channels. **(c)** MSE-head RMSE minus Gaussian-head RMSE for linear, MLP, and Transformer readouts on the four frozen 75k encoders, evaluated before pooling, after channel pooling, and after Kabsch refinement. Positive values favour the Gaussian head. Error bars combine the two methods' seed standard deviations in quadrature. Gaussian heads use per-axis inverse-variance pooling; MSE heads use mean pooling. All Transformer readouts predict absolute XYZ coordinates directly. Kabsch fits known probe geometry to predicted coordinates without using ground-truth test locations. RMSE is evaluated after left-hemisphere folding. `CCFDistc` uses histology-registered atlas coordinates during pretraining and serves as an oracle pretraining control.

H Nearest-neighbour transfer on NeuroNexus

We evaluate the frozen IBL 75k `MaskPred+ChanDistc` and `MaskPred+ManipDistc` encoders used for the NeuroNexus zero-shot bars in Figure 3. No coordinate head is used. Each of seven folds holds out one recording session for testing, with four other sessions used for training and two for validation. Trial indices are also partitioned with seed 42: 36 training, 12 validation and 12 test indices out of 60; each session subset uses only its corresponding trial indices. The labelled reference bank contains only training embeddings. Validation and test embeddings are queries.

We select k from $\{1, 3, 5, 10, 20\}$ by validation accuracy, choosing the smaller k in an exact tie. Each prediction is the majority label of the k nearest training embeddings by cosine similarity; tied class votes are resolved by the smallest class index. Thus, zero-shot refers to a frozen encoder without target-specific parameter training, while classification still requires labelled reference examples.

Table 10: Validation accuracy and selected k for the NeuroNexus zero-shot models in Figure 3. Each row identifies the held-out test session; validation uses two other sessions. Values are rounded to three decimals after selecting k using full-precision accuracy.

Test session	$k = 1$	$k = 3$	$k = 5$	$k = 10$	$k = 20$	Selected k
`MaskPred+ChanDistc`						
AD_HF01_1	0.672	0.704	0.718	0.724	0.734	20
AD_HF02_2	0.635	0.650	0.655	0.653	0.650	5
AD_HF02_4	0.567	0.573	0.570	0.563	0.558	3
AD_HF03_1	0.688	0.707	0.713	0.705	0.704	5
AD_HF03_2	0.649	0.659	0.665	0.657	0.657	5
NN_syn_01	0.645	0.659	0.661	0.659	0.658	5
NN_syn_02	0.647	0.661	0.663	0.661	0.660	5
`MaskPred+ManipDistc`						
AD_HF01_1	0.683	0.731	0.740	0.753	0.760	20
AD_HF02_2	0.619	0.624	0.628	0.625	0.626	5
AD_HF02_4	0.556	0.558	0.568	0.564	0.569	20
AD_HF03_1	0.657	0.666	0.664	0.656	0.655	3
AD_HF03_2	0.622	0.631	0.631	0.622	0.626	5
NN_syn_01	0.625	0.633	0.637	0.630	0.630	5
NN_syn_02	0.624	0.633	0.636	0.630	0.630	5

Test balanced accuracy (mean $\pm$ sample SD across seven held-out sessions) is 0.505 ± 0.068 for `ChanDistc` and 0.493 ± 0.057 for `ManipDistc`, matching the corresponding Figure 3 bars. This classification experiment does not predict atlas coordinates.

I Compute Resources

Compute resources: We used NYU's high-performance computing cluster for hyperparameter tuning, ablation studies, experimental design, and running and testing experiments. Google Cloud Platform (GCP) supported data processing, pretraining, and evaluation of the four methods used in the scaling experiments, including the computations underlying the scaling plots. We estimate approximately 43,000 GPU-hours across the broader research effort: roughly 35,000 on NYU HPC and 7,600 on GCP. These estimates measure allocated GPU time rather than active utilization. Cloud computing was supported by the NYU Google Cloud Platform for Research Grants Program through a GCP Research Award (RD157) to Erdem Varol, providing $20,000 in cloud credits.

J AP, DV, ML Regression Backbone Ablation

Table 11: **Electrode localization on IBL with geometry-aware pretraining.** We compare data2vec, Whisper-small, and the wav2vec 2.0 backbone of LFPformer after pretraining on 75k LFP samples, together with audio-only and random-initialized wav2vec 2.0 controls without LFP pretraining. Mean and median Euclidean error, R^2, and Spearman correlation (ρ) describe predictions before geometric refinement; R^2 and ρ are averaged over the three coordinate axes. Kabsch RMSE measures error after uncertainty-weighted averaging of repeated channel predictions and rigid fitting of known probe geometry, without using ground-truth test coordinates; ML coordinates are then folded into a common hemisphere. Raw metrics use 500,000 test samples from 119 probes; Kabsch RMSE uses the 430,759 samples in the 309-region subset. Values are mean ± sample SD across five test folds. LFPformer results are averaged over eight decoder seeds within each fold before computing these summaries; the no-LFP-pretraining controls each use one decoder fit. PT and RI denote audio-pretrained and random initialization. LFPformer uses linear, MLP, and Transformer (TF) readouts; the wav2vec 2.0 controls use linear readouts. **Bold** marks the best mean in each column (lower error; higher R^2 and ρ). Training budgets and readout configurations differ across model families.

Backbone	Pretraining	Eval/readout	Init	Mean (μm)	Med. (μm)	R^2	Spearman (ρ)	Kabsch RMSE (μm)
data2vec	`MaskPred`	frozen / MLP	PT	2416 ± 3	2222 ± 5	0.186 ± 0.003	0.468 ± 0.003	1958 ± 1
data2vec	`MaskPred`	full FT / MLP	PT	2221 ± 6	1927 ± 7	0.252 ± 0.003	0.560 ± 0.002	1718 ± 10
data2vec	`MaskPred` + `ChanDistc`	frozen / MLP	PT	2294 ± 6	2062 ± 4	0.239 ± 0.004	0.516 ± 0.003	1780 ± 2
data2vec	`MaskPred` + `ChanDistc`	full FT / MLP	PT	2267 ± 6	1979 ± 6	0.223 ± 0.003	0.541 ± 0.002	1690 ± 3
data2vec	`MaskPred` + `ManipDistc`	frozen / MLP	PT	2184 ± 5	1910 ± 4	0.287 ± 0.003	0.552 ± 0.003	1712 ± 7
data2vec	`MaskPred` + `ManipDistc`	full FT / MLP	PT	2214 ± 7	1894 ± 9	0.250 ± 0.004	0.556 ± 0.003	1701 ± 8
data2vec	`MaskPred`	frozen / MLP	RI	2296 ± 5	2067 ± 9	0.240 ± 0.003	0.506 ± 0.003	1797 ± 2
data2vec	`MaskPred` + `ChanDistc`	frozen / MLP	RI	2221 ± 5	1962 ± 7	0.271 ± 0.003	0.540 ± 0.002	1705 ± 3
data2vec	`MaskPred` + `ChanDistc`	full FT / MLP	RI	2192 ± 5	1889 ± 5	0.270 ± 0.003	0.569 ± 0.002	1687 ± 7
data2vec	`MaskPred` + `ManipDistc`	frozen / MLP	RI	2184 ± 4	1898 ± 7	0.284 ± 0.002	0.550 ± 0.002	1783 ± 3
data2vec	`MaskPred` + `ManipDistc`	full FT / MLP	RI	2191 ± 6	1880 ± 7	0.257 ± 0.004	0.567 ± 0.002	1758 ± 3
whisper-small	`MaskPred`	frozen / linear	PT	2425 ± 3	2245 ± 4	0.218 ± 0.002	0.476 ± 0.003	2210 ± 3
whisper-small	`MaskPred`	full FT / MLP	PT	2371 ± 5	2162 ± 6	0.211 ± 0.003	0.486 ± 0.003	2023 ± 3
whisper-small	`MaskPred` + `ChanDistc`	frozen / linear	PT	2350 ± 6	2133 ± 6	0.241 ± 0.002	0.505 ± 0.002	2068 ± 2
whisper-small	`MaskPred` + `ChanDistc`	full FT / MLP	PT	2328 ± 6	2083 ± 6	0.211 ± 0.004	0.485 ± 0.002	1874 ± 2
whisper-small	`MaskPred` + `ManipDistc`	frozen / linear	PT	2360 ± 5	2114 ± 4	0.216 ± 0.002	0.482 ± 0.001	2015 ± 2
whisper-small	`MaskPred` + `ManipDistc`	full FT / MLP	PT	2322 ± 4	2062 ± 3	0.218 ± 0.002	0.492 ± 0.002	1973 ± 2
wav2vec 2.0	None	frozen / linear	PT	2604 ± 6	2460 ± 6	0.106 ± 0.003	0.391 ± 0.005	2366 ± 4
wav2vec 2.0	None	frozen / linear	RI	2500 ± 5	2368 ± 7	0.186 ± 0.002	0.465 ± 0.002	2196 ± 2
LFPformer	`MaskPred`	frozen / linear	PT	2198 ± 2	2029 ± 3	0.345 ± 0.002	0.572 ± 0.002	1971 ± 2
LFPformer	`MaskPred`	frozen / MLP	PT	2188 ± 4	1958 ± 2	0.320 ± 0.004	0.557 ± 0.004	1913 ± 1
LFPformer	`MaskPred`	frozen / TF	PT	1605 ± 15	1271 ± 21	0.570 ± 0.005	0.759 ± 0.005	1676 ± 6
LFPformer	`MaskPred` + `ChanDistc`	frozen / linear	PT	2082 ± 3	1869 ± 3	0.395 ± 0.002	0.613 ± 0.002	1799 ± 5
LFPformer	`MaskPred` + `ChanDistc`	frozen / MLP	PT	1962 ± 4	1708 ± 4	0.426 ± 0.002	0.636 ± 0.002	1639 ± 6
LFPformer	`MaskPred` + `ChanDistc`	frozen / TF	PT	1299 ± 19	1080 ± 23	0.696 ± 0.008	$\mathbf{0.850 \pm 0.006}$	$\mathbf{1187 \pm 5}$
LFPformer	`MaskPred` + `ManipDistc`	frozen / linear	PT	1948 ± 6	1713 ± 5	0.440 ± 0.004	0.644 ± 0.003	1587 ± 6
LFPformer	`MaskPred` + `ManipDistc`	frozen / MLP	PT	1927 ± 4	1657 ± 4	0.435 ± 0.003	0.643 ± 0.002	1547 ± 8
LFPformer	`MaskPred` + `ManipDistc`	frozen / TF	PT	$\mathbf{1261 \pm 15}$	$\mathbf{985 \pm 16}$	$\mathbf{0.703 \pm 0.007}$	0.849 ± 0.006	1193 ± 9

Table 11 compares downstream XYZ regression across backbones and readouts. For practical use in neuroscience, localization should recover a physically plausible probe trajectory within the brain, rather than an unconstrained cloud of channel predictions. Kabsch refinement rigidly fits the known probe geometry to the predicted channel coordinates, preserving the probe's shape and channel spacing while estimating its position and orientation. The fit uses no ground-truth test coordinates; these are used only to evaluate localization error. We report raw per-channel metrics alongside Kabsch-refined 3D RMSE to distinguish prediction accuracy from the benefit of geometric refinement, using the evaluation subsets specified in the caption. Adding `ChanDistc` or `ManipDistc` to `MaskPred` improves downstream XYZ regression, measured by Kabsch RMSE, across data2vec, Whisper-small, and the wav2vec 2.0 backbone of LFPformer. For frozen, audio-initialized data2vec, `ChanDistc` and `ManipDistc` reduce Kabsch RMSE from 1958 μm to 1780 μm and 1712 μm (9.1% and 12.6% reductions). For frozen Whisper-small, the corresponding reductions are 6.4% and 8.8%. The same trend holds for all three LFPformer readouts: linear (1971 to 1799 and 1587 μm), MLP (1913 to 1639 and 1547 μm), and Transformer (1676 to 1187 and 1193 μm). For data2vec, the improvement is also observed with random initialization. LFPformer with a geometry-trained encoder and Transformer readout achieves lower Kabsch RMSE than every data2vec and Whisper-small

variant listed in the table, including their full-fine-tuning variants. Its `ChanDistc` and `ManipDistc` configurations achieve 1187 µm and 1193 µm, respectively, compared to 1687 µm for the best listed data2vec variant and 1874 µm for the best listed Whisper-small variant. Figure 14 complements

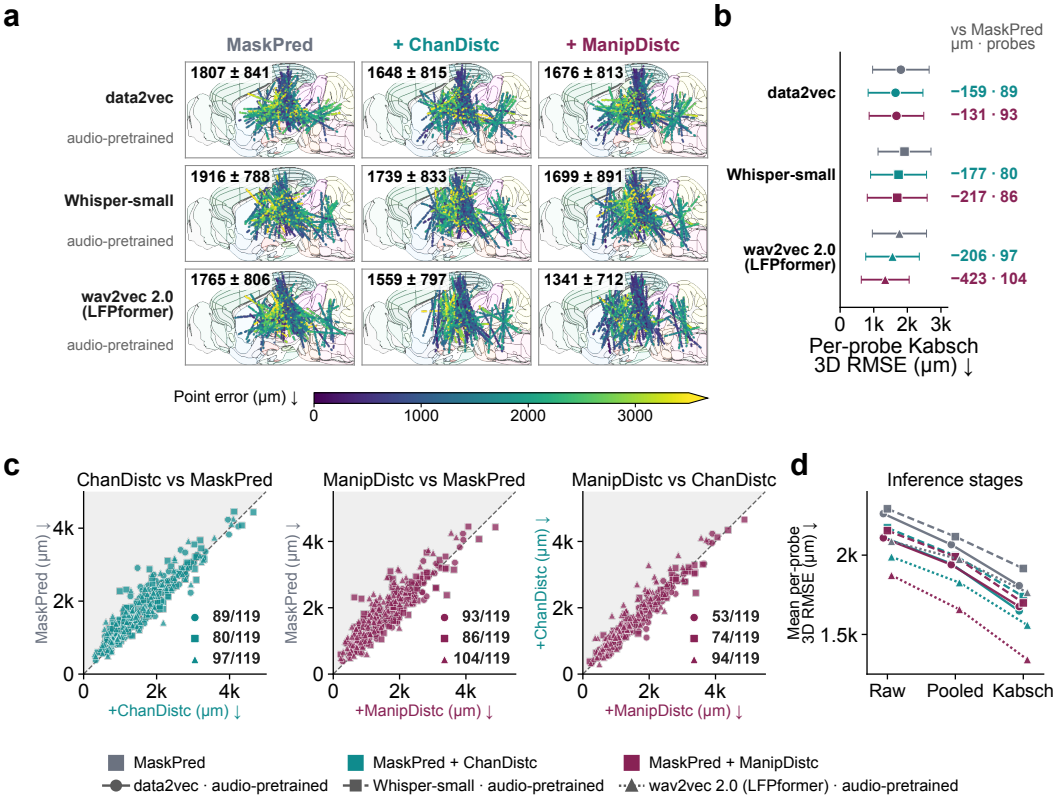


Figure 14: **Geometry-aware objectives across audio-initialized backbones.** Frozen linear Gaussian readouts are evaluated on 119 held-out IBL probes. **(a)** Kabsch-refined probe trajectories for `MaskPred`, `MaskPred + ChanDistc`, and `MaskPred + ManipDistc`; colours show per-channel localization error and labels give mean ± SD per-probe RMSE. **(b)** Per-probe Kabsch RMSE for each objective and its change relative to `MaskPred`. **(c)** Paired per-probe comparisons between objectives; annotations give the number of probes improved by the method on the horizontal axis. **(d)** Mean error before pooling repeated predictions, after inverse-variance channel pooling, and after Kabsch refinement. Symbols distinguish data2vec, Whisper-small, and the wav2vec 2.0 backbone used in LFPformer. Kabsch fits the known probe geometry to predicted coordinates without using ground-truth test locations. The data2vec `ManipDistc` result uses the historical checkpoint shown in the figure and should not be treated as a controlled evaluation of the corrected checkpoint in Table 11.

Table 11 with frozen linear readouts and equal-probe summaries for audio-initialized encoders. For data2vec, mean per-probe Kabsch RMSE is 1807 µm for `MaskPred`, 1648 µm for `MaskPred + ChanDistc`, and 1676 µm for the historical `MaskPred + ManipDistc` checkpoint. Relative to `MaskPred`, the corresponding reductions are 159 µm (89/119 probes improved) and 131 µm (93/119 probes improved). Whisper-small and wav2vec 2.0 retain mean reductions of 177 and 206 µm with `ChanDistc` and 217 and 423 µm with `ManipDistc`. Both the figure and table cover 119 test probes, but this figure weights probes equally and uses the full evaluation channel cohort, whereas the table reports fold-averaged Kabsch RMSE on the 309-region subset. Their means and standard deviations are therefore not directly comparable.

K IBL Probe Case Study: LFP and Micromanipulator Fusion

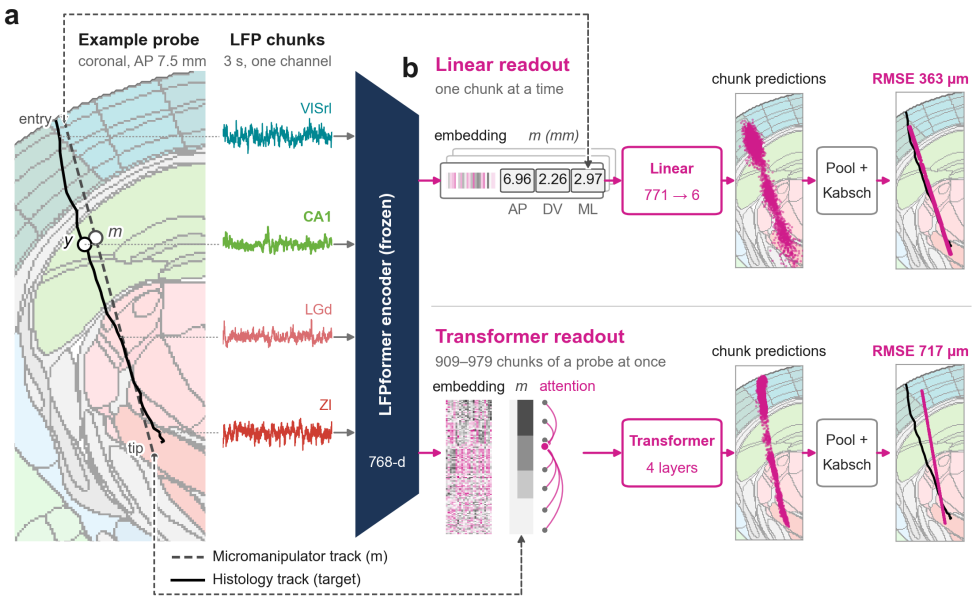


Figure 15: **Combining LFP representations with the micromanipulator record. (a)** An example IBL probe in a coronal atlas section. The dashed line is its logged micromanipulator trajectory (m), and the solid line is the histology trajectory (y). Four 3-s LFP chunks from increasing depths are passed through the frozen LFPformer encoder. **(b)** A linear readout predicts each chunk location independently from its embedding and micromanipulator coordinates. A Transformer readout receives the ordered chunks from an entire probe and can use information from other depths through self-attention. For both readouts, repeated chunk predictions are combined within channel by inverse-variance pooling, then the known probe geometry is fitted with Kabsch. The resulting probe path is compared with histology; the displayed RMSE values are for this example probe.

L Allen Brain Observatory: Per-Group Comparison and Probe Case Study

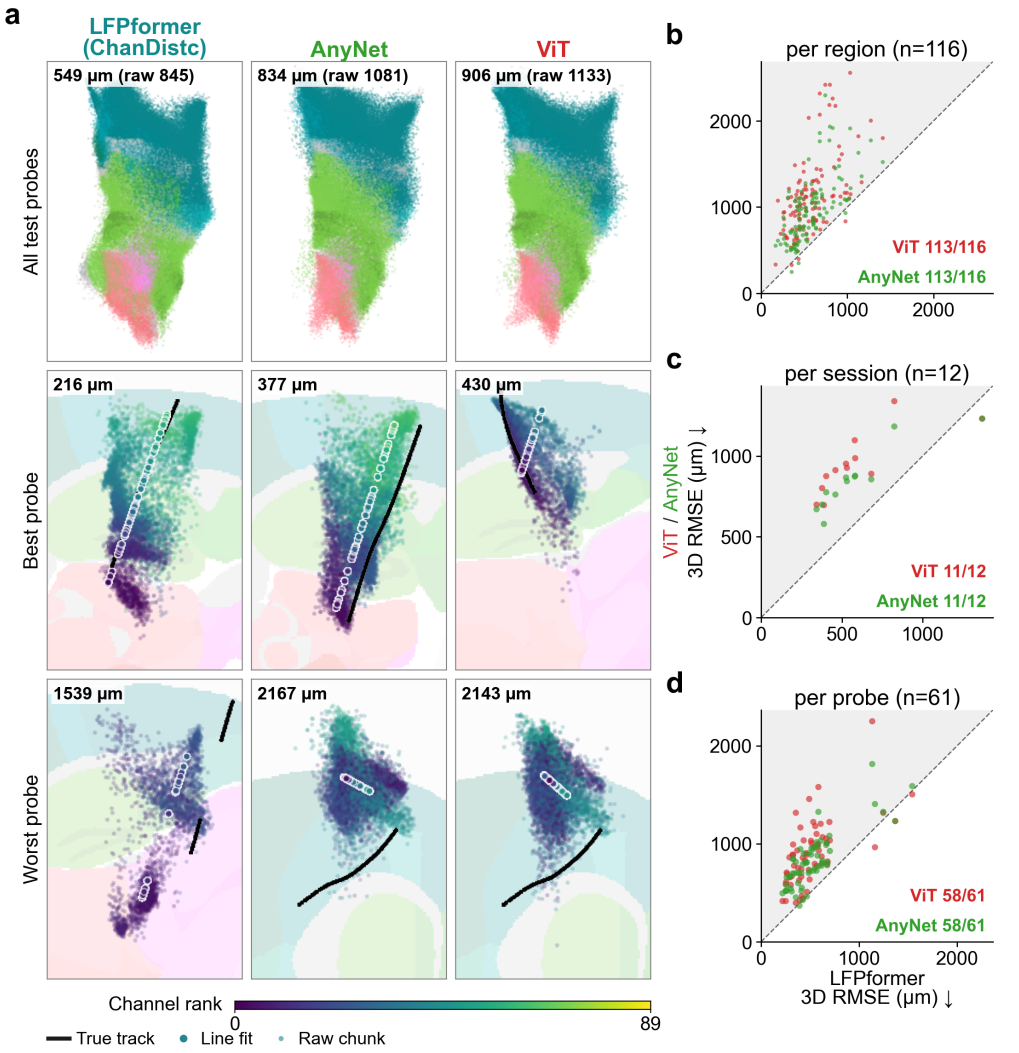


Figure 16: **Allen localization at the 75k pretraining scale.** LFPformer is the `MaskPred + ChanDistc` encoder pretrained on 75k Allen windows and fully fine-tuned with a Gaussian read-out; ViT and AnyNet are trained on the same Allen data. Predictions are refined by a per-channel inverse-variance ($1/\sigma^2$) weighted mean followed by projection of each probe's channels onto their first principal axis (line fit). This is the original Allen post-processing, rather than the pooling and Kabsch protocol used for the main-text RMSE. **(a)** Probe-localization examples in sagittal projection (AP–DV). Top row: raw per-chunk predictions of all test probes, coloured by the Allen region at the predicted location; labels give the line-fit RMSE and, in brackets, the raw RMSE. Middle and bottom rows: each model's test probe with the lowest and highest line-fit RMSE among probes with at least 50 test chunks. Raw chunk predictions and line-fit channel positions are coloured by channel rank along the probe. The histology track is shown in black, with the Allen sagittal section at the probe's median mediolateral position in the background. Labels give the line-fit RMSE for each probe. RMSE in panel a is left-hemisphere-folded. **(b–d)** Group-wise 3D RMSE of ViT (red) and AnyNet (green) against LFPformer on the held-out Allen test set, grouped by brain region (b; 116 regions with at least 20 test chunks), recording session (c), and probe (d). Points in the shaded area above the dashed identity line indicate groups where LFPformer has lower error; annotations give the number of such groups. Overall RMSE across all test chunks is 549 μm for LFPformer, 952 μm for ViT, and 840 μm for AnyNet. The ViT and AnyNet examples in panel a use earlier training runs than panels b–d, with overall line-fit RMSE of 906 and 834 μm, respectively.

M Zero-Shot Chronic Stability on Neuropixels 2.0

A chronic implant is the setting where post-hoc histology helps least. The probe stays in the animal for months, and the registration, when it happens at all, happens at the end. This appendix asks what the model reports there, on recordings that share no animal, no laboratory and no probe type with anything it was trained on.

Recording. A collaborating laboratory recorded mouse M05 on five days through one chronically implanted Neuropixels 2.0 probe (NP2010, 384 channels), three days in July 2024 and two in October, 19.4 h in total. The hardware is unlike every probe in our training data. Neuropixels 2.0 has two site columns instead of four, a 15 μm site pitch instead of 20 μm, and spans 2,865 μm instead of 3,840 μm. We decimate the LF band from 2,500 to 1,250 Hz and cut it into 3 s chunks, which gives 23,288 chunks per channel and 8.94 M channel $\times$ chunk predictions. No histology exists for these recordings, so nothing in this appendix is scored against a true coordinate.

Readout. The encoder is the large `MaskPred + ManipDistc` model from the main results, frozen. The readout is the three-layer per-channel MLP with a Gaussian head described in Appendix D, averaged over eight decoder seeds. We deliberately do not use the probe-sequence Transformer here. That decoder takes channel column and depth in μm as explicit inputs and was fitted on the Neuropixels 1.0 layout, so running it on Neuropixels 2.0 would make it extrapolate across a hardware change inside a figure whose subject is change over time, and the two effects could not be separated afterwards. The MLP sees the embedding and nothing else. What that costs is known: on the IBL test folds the same MLP scores 1,359 μm Kabsch error against the Transformer's 1,071 μm. Figure 17 is a statement about stability, and its coordinates should not be read against the main-text error bars.

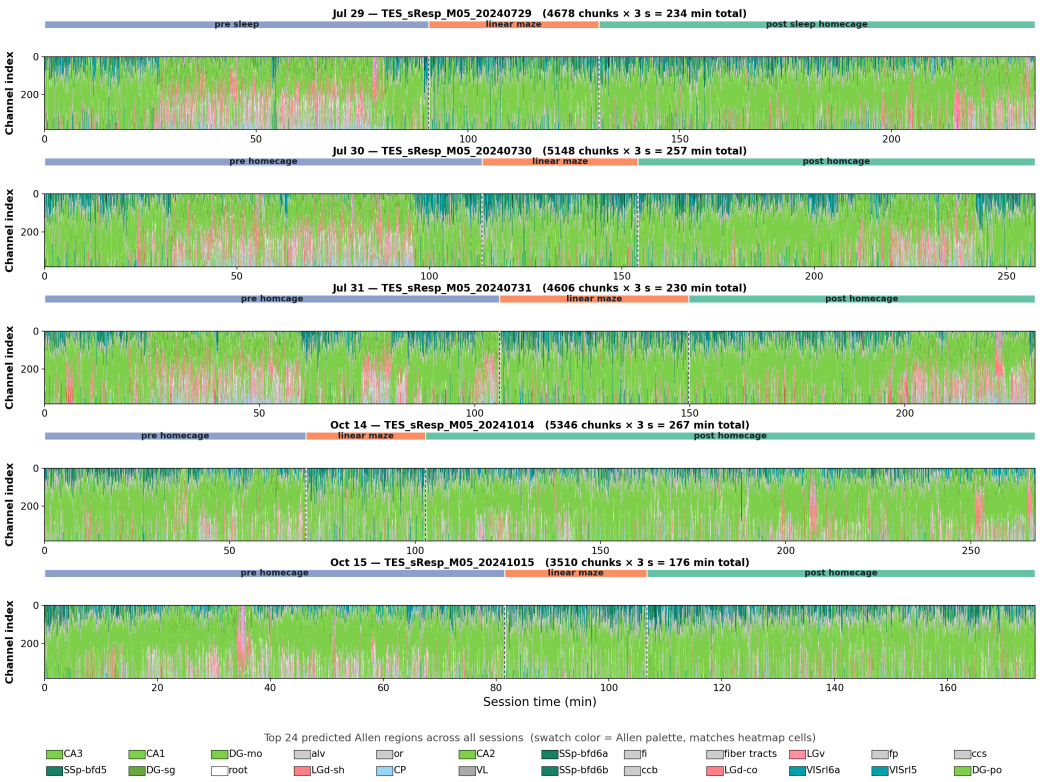


Figure 17: **Zero-shot region readout on a chronically implanted Neuropixels 2.0 probe.** Five full sessions from mouse M05, three in July 2024 and two in October, recorded by a collaborator on hardware LFPformer never saw during training. Each row is one session, 176 to 267 min long: the LFP is cut into 3 s chunks and every channel of every chunk is coloured by the Allen CCF region at its predicted coordinate, with the base of the shank at the top and the tip at the bottom. The strip above each row marks the three recording blocks (pre-task, linear maze, post-task); these are block labels, not sleep scoring. Encoder and readout were trained on IBL alone, and nothing is fit to this animal. There is no histology for these recordings, so the figure reports predictions without an error measure. The predicted layout places deep cortex on the shallow channels and dorsal hippocampus over most of the shank, and it holds across hours within a session and days within each week, while the July and October groups differ.

M.1 Behavioural Context in the Chronic Representation

Each session is three sub-recordings that the rig wrote as separate files: a pre-task block in the home cage, a linear maze run, and a post-task block back in the home cage. These are the animal's behavioural context, not scored sleep stages, and they are what the strip above each row of Figure 17 marks. The encoder was trained to localize and has never seen a behavioural label, so we can ask whether its representation carries the context anyway.

We pool each 3 s chunk over the 384 channels and fit a logistic regression to tell maze chunks from home cage chunks, scoring leave-one-session-out so that no fold shares a recording day with its training set. Pooling by a plain channel mean gives one 1,024-dimensional vector per chunk and a balanced accuracy of 0.764 ± 0.051 against a chance level of 0.5. Averaging within four depth bands instead, which keeps where along the shank a change happened, raises this to 0.863 ± 0.036 (Table 12).

Those numbers alone would also be consistent with slow drift rather than behaviour, since each block is a contiguous stretch of time and a separate file on disk. The second row of the table is the control. Pre-task and post-task home cage are the two blocks furthest apart in time within a session and are separate files in the same way, so anything driven by settling, temperature or impedance should separate them at least as easily as it separates the maze block from the two neighbours it sits between.

It separates them less well, 0.665 ± 0.051 and 0.692 ± 0.070, and depth-band pooling barely helps it while it adds ten points to the behavioural contrast. The context signal is structured along the probe and is not a restatement of elapsed time.

We are describing what the representation distinguishes and not the mechanism behind it. A maze run differs from the home cage in locomotion, arousal and hippocampal theta at once, and movement also reaches the LFP as artefact. Separating those contributions needs the sleep and behaviour scoring held by the recording laboratory, which was not available to us for this revision.

Table 12: Behavioural context decoded from the frozen chronic representation. Logistic regression on pooled 3 s embeddings, scored leave-one-session-out across the five sessions; chance is 0.5. Depth-band pooling averages channels within four bands along the shank instead of over all 384. Row 2 is the control: the two home cage blocks are the furthest apart in time within a session, so slow drift would separate them at least as easily as it separates the maze block from its two neighbours.

			Balanced accuracy	
Contrast	Role	Chunks	Channel mean	Depth bands
Maze vs. home cage	behavioural context	3,642 / 19,646	0.764 ± 0.051	0.863 ± 0.036
Pre vs. post home cage	elapsed-time control	9,246 / 10,400	0.665 ± 0.051	0.692 ± 0.070

The coordinates move with it, a little. The predicted anatomy should be indifferent to what the animal is doing, because the brain does not move between blocks. Averaging each channel's Kabsch-refined coordinate within a block, the per-channel coordinate shifts by a median of 147, 330 and 219 μm in AP, DV and ML across the three blocks, against a chunk-to-chunk scatter inside a single block of 341, 379 and 370 μm. A single chunk is therefore noisier than the whole behavioural effect, which is why the layout in Figure 17 looks stable across a session. With hundreds to thousands of chunks per block the shift is still well above the standard error on a block mean, so the readout is not fully invariant to the animal's state, and the largest shift is in DV, along the shank.

Limits. There is no histology for this animal, so the layout in Figure 17 is checked for consistency across hours and across days rather than against a true coordinate. The geometry-free readout costs roughly 290 μm of Kabsch error relative to our best decoder, so the figure should not be read as a localization-accuracy result. Chunk-to-chunk predictions also still jitter enough that change between days does not clearly dominate the variation within a session; pooling over longer windows would reduce that, at the price of changing what the panel claims.

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 clearly highlight and convey the contributions and scope of this paper: LFP data contains enough anatomical information to localize and decode to 3D anatomical CCF coordinates, which translates across probe technologies, labs, species through geometry aware self supervised learning objectives, all of which are substantiated by the results of our extensive experiments with scaling, backbones, training methodologies and evaluations.

   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: We include and discuss limitations and future scope in Section 5, emphasizing data imbalance for rare classes, quality of histological registration to the CCF atlas and further scope of extension to larger cross species datasets with extended training and larger models.

   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 new theoretical results; it rather focuses on leveraging probe geometry for self supervised learning objectives, thus providing novel representation learning objectives.

   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: We provide all necessary steps and methodologies for training publicly available model architectures, pre-processing techniques, and post training procedures, including our novel representation learning contributions through detailed figures and explanations as detailed in Section 3 and Section 4 as well as the Supplementary material (A-M), ensuring full reproducibility of our results.

   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: The repository linked in the Code Availability paragraph provides code and documentation for pretraining, downstream adaptation, and inference, including the geometry-aware objectives and post-hoc refinement procedures. We will also release pretrained model weights and instructions for accessing the datasets and backbones used in our experiments.

   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: Section 2 and Appendix A describe dataset-specific training, validation, and test splits, including subject-disjoint evaluation for Allen and IBL. Section 3 specifies the pretraining objectives, spatial kernel width and temperature, Gaussian localization loss, readout architectures, and inference-time refinement. Appendices D–G provide further architecture details, hyperparameter selection, training settings, and decoder ablations. Frozen-encoder evaluation, full fine-tuning, and transfer protocols are distinguished in the corresponding experiments.

   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: Our figures, plots and results include errors bars with standard deviation for all quantitative and qualitative comparisons of scaling, backbones, representation learning objectives and evaluation methods, computed with consistent train/test/validation splits with multiple folds, along with explanation and interpretation of the results.

   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: We have added information about our compute, GPU types, GPU hours in the supplementary in Section I.

   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: All animal experiments were conducted under institutional IACUC or equivalent approval at collaborating labs, and we adhere to community guidelines on data handling and privacy for neural recordings.

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: Section 6 discusses positive impacts (reducing histology burden, enabling real-time DBS feedback) and negative impacts (risk of incorrect localization in clinical settings), along with mitigation via calibrated uncertainty estimates.

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: [N/A]

Justification: The model localizes electrodes in mouse/primate brains and poses minimal misuse risk. No high-risk generative capabilities are involved.

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: We use the Allen Brain Observatory and IBL Brain Wide Map datasets, along with the publicly available pretrained audio based transformer backbones including Wav2Vec2, Data2Vec, Whisper, and baselines ViT, AnyNet; all cited with their original publications.

    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: We provide code and documentation for pretraining, downstream adaptation, and post-hoc refinement in the repository linked in the Code Availability paragraph. Pretrained model checkpoints will also be released there.

    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: [N/A]

Justification: No human or crowdsourced experiments are involved; our study majorly involves animal electrophysiology recordings from mice and monkeys.

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: [N/A]

Justification: No human trials and subjects were involved; all animal electrophysiology recordings and experiments were conducted under appropriate institutional animal care and approvals.

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 are not part of our study and methodology; we only use self-supervised audio based transformer backbones for our experimentation and results.

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.