NEUROLENS: LEARNING LATENT EMBEDDINGS OF NEURAL SEMANTICS FROM CHRONIC RECORDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding how neural activity represents higher-order cognition and how these representations evolve over time has long been a central pursuit in neuroscience. However, current analytical tools cannot easily distinguish representational plasticity from recording instability in chronic neural recordings. Here, we introduce NeuroLens (**L**atent **E**mbeddings of **N**eural **S**emantics), a self-supervised model based on the Joint-Embedding Predictive Architecture (JEPA) framework that learns denoised, semantically informative latents from chronic neural recordings. An adaptive encoder maps changing neural populations into a common latent space, while a temporal predictor learns structure that supports prediction of future latent states. By predicting in latent space, NeuroLens captures temporally predictable structure and reduces sensitivity to transient, recording-specific variability. Across chronic intracortical data in mice and humans, the learned representations improve decoding of decision-making and semantic task variables. Multi-day pretraining enables generalization to future sessions, rapid few-shot adaptation to unseen neural populations, and more stable decoding over time than state-of-the-art baselines. Together, these results establish NeuroLens as a new paradigm for studying how neural representations change during learning and over long timescales.

1 INTRODUCTION

Learning continually reshapes the relationship between neural activity and behavior over extended timescales (Peters et al., 2014). Understanding this process requires tracking how task-relevant population dynamics emerge, persist, and reorganize across days (Wang et al., 2026). Advances in chronic recordings increasingly enable such longitudinal measurements (Melin et al., 2024; Bimbard et al., 2025; Skromne Carrasco et al., 2026), but electrode drift, signal degradation, and changes in the recorded neurons can obscure learning-related changes in neural activity (Karpowicz et al., 2024). This challenge motivates methods that capture genuine representational plasticity while remaining robust to recording nonstationarity.

Existing approaches address recording nonstationarity through statistical and deep-learning-based neural manifold alignment (Karpowicz et al., 2025; Gallego et al., 2020), recalibration-based test-time adaptation (Fan et al., 2023; Feghhi et al., 2026), and transformer-based in-context learning (Le et al., 2026). These methods improve decoding stability across recording days, but many are tailored to specific decoding tasks or compensate for distribution shifts in observed activity without explicitly separating task-relevant structure from recording variability. These limitations motivate a general-purpose framework that accommodates changing neural populations while preserving biological changes in the representations themselves.

Self-supervised learning (SSL) provides a foundation for learning such representations beyond a specific decoding task. However, many existing approaches reconstruct observed neural activity, which may retain recording-specific details alongside behaviorally relevant structure (Ye & Pandarinath, 2021; Zhang et al., 2024). For example, probe drift can alter which neurons are recorded even when the underlying population dynamics remain stable. Reconstruction-based models must account for these changes in the observations, although they may be unrelated to the neural computation of interest (Rule et al., 2020). Joint-embedding predictive architectures (JEPAs) instead predict masked or future representations in latent space (Assran et al., 2023). This objective em-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

phasizes predictable structure without requiring the model to reproduce every detail of the recorded activity. Applied to chronic neural recordings, it may yield latents that are less sensitive to recording instability while capturing information relevant to behavior and cognition.

Building on this principle, we introduce NeuroLens (**L**atent **E**mbeddings of **N**eural **S**emantics), a self-supervised model that learns denoised, semantically informative neural latents through causal prediction in latent space (Maes et al., 2026). The model integrates two complementary components. An adaptive neural encoder uses cross-attention to accommodate varying neuron counts across days and infers neuron identities from a few calibration trials, mapping each day's neural population into a common latent space. A causal transformer predictor then predicts future latent embeddings, with parameters shared across days to encourage a stable, predictive structure. These components enable adaptation to changing neural populations while capturing temporal regularities across days.

Across chronic intracortical data in mice (Melin et al., 2024) and humans (Willett et al., 2023; Card et al., 2024), we demonstrate that NeuroLens learns latent embeddings that improve decoding of visual decision-making and semantic task variables. After multi-day pretraining, the model generalizes to future sessions and supports rapid, gradient-free few-shot adaptation to unseen neural populations, outperforming state-of-the-art (SOTA) neural decoding baselines. Despite substantial neuron loss and day-to-day drift in neural activity, interpretability analyses indicate that NeuroLens learns temporally consistent representations with a stable effective rank over time. These results provide a foundation for tracking task-relevant population dynamics over long timescales. The contributions of this work include:

- A self-supervised framework that learns denoised, semantically informative neural representations through prediction in latent space.
- An architecture that balances flexibility and stability through an adaptive neural encoder and a shared temporal predictor, enabling rapid generalization to unseen neural populations.
- A framework for learning task-relevant representations that are robust to recording nonstationarity, enabling new analyses of how neural latent states evolve over time.

2 Related Work

Large-scale models for neural analysis. Recent advances have expanded from modeling single-session neural dynamics to large-scale pretraining across sessions, animals, and tasks (Andre et al., 2026). Supervised approaches such as POYO (Azabou et al., 2023) and POSSM (Ryoo et al., 2026) enable scalable multi-session decoding through spike tokenization combined with PerceiverIO (Jaegle et al., 2021) or recurrent state-space models. Self-supervised approaches (Ye & Pandarinath, 2021; Zhang et al., 2024; 2025) instead learn representations by masking neural activity across time, neurons, regions and modalities. Despite this progress, two challenges remain. First, supervised representations are tied to predefined decoding targets, while self-supervised objectives largely focus on reconstructing neural activity rather than capturing global structure. Second, existing multi-session models primarily address transfer across animals, tasks, or brain regions, with limited focus on adapting to temporal distribution shifts.

Self-supervised representation learning for neural activity. Self-supervised learning enables transferable representations from unlabeled data. Observation-space methods such as GPT (Radford et al., 2018) predict future inputs, while masked autoencoders (MAEs) (Radford et al., 2018) reconstruct missing inputs. Although these methods scale effectively, they require modeling low-level details that may be irrelevant to downstream tasks. Joint-Embedding Predictive Architectures (JEPAs) instead predict masked targets in latent space, encouraging higher-level structure (Bardes et al., 2024). I-JEPA (Assran et al., 2023) demonstrated semantic visual representations without hand-crafted augmentations, while LeJEPA (Maes et al., 2026) introduced a theoretically grounded regularizer to prevent representation collapse. In neuroscience, self-supervised models (Ye et al., 2023; Zhang et al., 2025) have largely adopted reconstruction-based objectives to model neural population activity. Given the noise and variability in neural recordings, latent-space prediction may emphasize dynamics that capture higher-level representations and support generalization.

Nonstationarity in chronic neural recordings. Methods for addressing nonstationarity in chronic neural recordings have largely been developed for task-specific decoding and fall into three

categories. First, manifold-alignment methods map activity across days into a shared latent space using statistical or deep learning techniques; NoMAD maps activity to a stable reference manifold (Karpowicz et al., 2025), while CEBRA learns consistent embeddings from temporal context (Schneider et al., 2023). Second, test-time recalibration methods continually update decoders, with CORP and DietCORP using language-model-corrected pseudo-labels for online adaptation (Fan et al., 2023). Third, transformer-based methods, including SPINT and POSSM, use causal prediction objectives (Le et al., 2026; Ryoo et al., 2026). However, these approaches have limited generality: manifold alignment imposes restrictive assumptions about shared cross-day dynamics, recalibration requires task-specific pseudo-labels, and SPINT and POSSM rely on supervision for predefined tasks. Learning general-purpose representations that remain stable under long-term changes in neural population composition while supporting accurate decoding therefore remains an open challenge.

3 DATASETS

IBL chronic Neuropixels data. This dataset tracks a mouse learning a visual decision-making task in which it turns a wheel to report whether a stimulus appears on the left or right side of a screen, receiving a reward for correct choices (Melin et al., 2024). Chronic Neuropixels recordings capture spiking activity from hundreds of neurons over 3 months as the mouse progresses from novice to proficient performance. We analyze 53 daily sessions ($\sim$47 hours) from this mouse, including both trial and non-trial (spontaneous) periods. Each session contains 366 trials on average, each spanning 2 seconds around stimulus onset; non-trial periods are divided into contiguous 2-second windows. Spikes are binned at approximately 16.7 ms (60 bins per second), and units with mean firing rates below 1 Hz are excluded. Each recording contains 159$\sim$498 units across approximately 6$\sim$20 anatomical regions. Neural activity is paired with synchronized left- and right-camera videos of the face and upper body. Behavioral variables include trial-level binary choice and time-bin-level wheel speed, whisker motion energy from each camera, and paw and nose speeds from the left camera.

Human chronic Utah array data. These datasets contain neural recordings from two participants (T12 and T15) with ALS-induced anarthria as they attempted to speak displayed sentences. For T12, chronic recordings were obtained from four intracortical Utah arrays implanted in the ventral motor cortex (Willett et al., 2023); we retain the 128 channels in area 6v. We analyze 9,680 trials across 23 daily sessions over four months, totaling 16.2 hours. For T15, we analyze 9,498 trials across 45 sessions over 20 months, using all 256 electrodes in the precentral gyrus (Card et al., 2024). For both datasets, neural features include threshold-crossing spike counts and spiking-band power (SBP), binned at 20 ms and z-scored to mitigate recording nonstationarity (Appendix C). Each sentence is labeled with an unaligned phoneme sequence using a 41-class vocabulary that includes the Connectionist Temporal Classification (CTC) blank token. For semantic decoding, we obtain a 384-dimensional sentence embedding from all-MiniLM-L6-v2 (Reimers & Gurevych, 2019), such that sentences with similar meanings have similar embeddings.

4 METHODS

NeuroLens is a self-supervised model that learns task-relevant neural representations by predicting future latent embeddings. Building on LeJEPA (Maes et al., 2026), it addresses nonstationarity in chronic recordings via a day-adaptive encoder, which maps daily activity into a common latent space by conditioning on neural population size and identity, and a shared autoregressive (AR) predictor, which extracts temporally predictable structure to form a stable subspace across days. Training combines prediction mean squared error (MSE) with sketched isotropic Gaussian regularization (SIGReg) to prevent representation collapse (Balestriero & LeCun, 2025). At inference, the encoder supports few-shot, gradient-free adaptation to unseen populations. We detail the architecture and training objectives below.

4.1 MODEL ARCHITECTURE

Day-adaptive cross-attention encoder. To accommodate changes in recorded neurons and their responses across days, we map variable-size neural populations into a common latent space. Let $\mathbf{x}_t^{(d)} \in \mathbb{R}^{N_d}$ denote spike counts for N_d neurons on day d at time bin t. A two-layer MLP f_{proj} with GELU activation maps each spike count to a neuron token $\mathbf{u}_{t,n}^{(d)} = f_{\text{proj}}\left(x_{t,n}^{(d)}\right)$, yielding

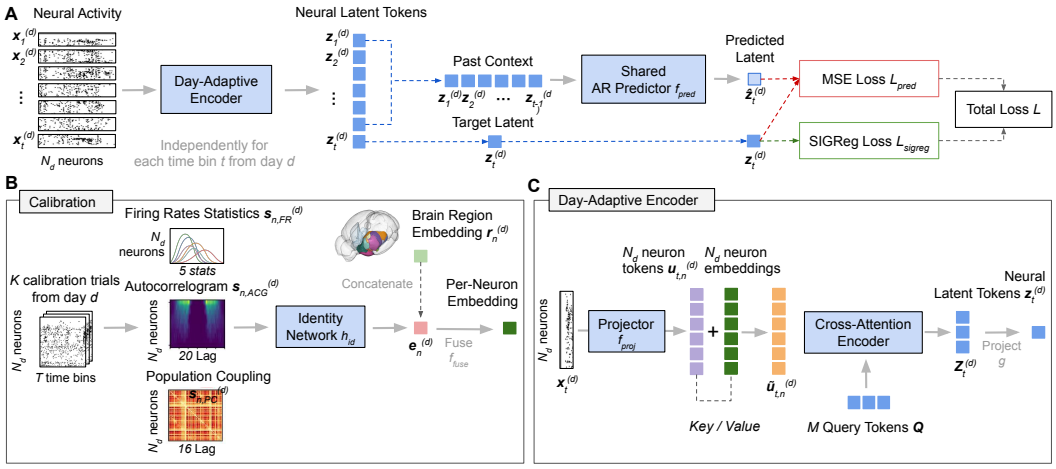

Figure 1: **NeuroLens framework.** (A) *Model overview.* A day-adaptive encoder maps spike counts at each time bin to latent representations, and an autoregressive (AR) predictor shared across days predicts future latents from past latents. The model is trained using a latent prediction loss together with SIGReg loss to prevent representation collapse. (B-C) *Day-adaptive cross-attention encoder based on calibration data.* At each time bin, an MLP independently projects each neuron's spike count into a neuron token. These tokens serve as the keys and values of the cross-attention module. Because attention is permutation-invariant, we distinguish neuron identities using calibration trials sampled from each recording day. From these trials, we compute firing-rate statistics, autocorrelograms (ACGs), and population coupling to generate a signature for each neuron. A second MLP maps these signatures to neuron embeddings, which are concatenated with brain-region embeddings and added to the corresponding neuron tokens. Learned queries then aggregate the variable-length sequence of neuron tokens through cross-attention, producing a fixed number of latent embeddings for the AR predictor. Processing each time bin independently prevents temporal leakage, while calibration-based neuron embeddings enable adaptation to previously unseen neural populations across recording days.

$\mathbf{U}_t^{(d)} = \{\mathbf{u}_{t,n}^{(d)}\}_{n=1}^{N_d} \in \mathbb{R}^{N_d \times H}$. We augment each spike token with neuron and region embeddings to obtain $\widetilde{\mathbf{U}}_t^{(d)}$ (defined below). Then, M learned queries $\mathbf{Q} \in \mathbb{R}^{M \times H}$ aggregate neuron tokens via cross-attention, and a linear projection g maps each latent embedding to a scalar:

$$\mathbf{Z}_t^{(d)} = \mathrm{CrossAttn}\left(\mathbf{Q}, \widetilde{\mathbf{U}}_t^{(d)}, \widetilde{\mathbf{U}}_t^{(d)}\right) \in \mathbb{R}^{M \times H}, \qquad \mathbf{z}_t^{(d)} = g(\mathbf{Z}_t^{(d)}) \in \mathbb{R}^{M}. \tag{1}$$

Each time bin is encoded independently: $\mathbf{z}_t^{(d)}$ depends only on $\mathbf{x}_t^{(d)}$. This prevents temporal leakage during encoding and leaves temporal modeling to the subsequent AR predictor.

Calibration-based neuron identification. Cross-attention is invariant to neuron-token order (Eq. 1), so spike-count tokens alone do not explicitly encode neuron identity. We therefore characterize each neuron n on day d using a small set of daily calibration trials: $\mathbf{s}_n^{(d)} = \mathrm{concat}\left(\mathbf{s}_{n,\mathrm{FR}}^{(d)}, \mathbf{s}_{n,\mathrm{ACG}}^{(d)}, \mathbf{s}_{n,\mathrm{PC}}^{(d)}\right)$, where FR, ACG, and PC summarize firing-rate statistics, autocorrelograms, and population coupling in 5, 20, and 16 dimensions, respectively (Appendix A). A two-layer network h_{id} with GELU and output LayerNorm maps each signature to a neuron embedding $\mathbf{e}_n^{(d)}$, which is concatenated with a learnable region embedding $\mathbf{r}_n^{(d)}$, projected into token space, and added to the spike token before cross-attention (Eq. 1):

$$\mathbf{e}_n^{(d)} = h_{\mathrm{id}}\left(\mathbf{s}_n^{(d)}\right) \in \mathbb{R}^{H'}, \quad \widetilde{\mathbf{u}}_{t,n}^{(d)} = \mathrm{LayerNorm}\left(\mathbf{u}_{t,n}^{(d)} + f_{\mathrm{fuse}}\left(\mathrm{concat}\left(\mathbf{e}_n^{(d)}, \mathbf{r}_n^{(d)}\right)\right)\right) \in \mathbb{R}^{H}. \tag{2}$$

The identity network uses physiological and population signatures instead of fixed neuron-ID embeddings (Azabou et al., 2023; Ryoo et al., 2026), enabling transfer to unseen neurons using few calibration trials. Appendix H shows that such signature-based identities perform comparably to learned ID embeddings.

For human Utah array recordings, however, channels capture mixtures of nearby neurons rather than spike-sorted units. Applying neuron signatures (Eq. 2) can therefore obscure channel identity and induce representation collapse. We instead forgo the cross-attention encoder in Eq. 1: channel features are mapped to the latent space by an MLP encoder, and a hypernetwork conditioned on calibration statistics generates day-specific FiLM (Perez et al., 2018) parameters for the encoder layers to adapt to channel drift (Appendix B).

Autoregressive predictor. We employ a causal transformer P_θ, shared across days, to model the temporal dynamics of encoder latents $\mathbf{z}_{1:T}^{(d)}$ (Eq. 1). After adding learned temporal position embeddings, we pass $\mathbf{z}_{1:T}^{(d)}$ through transformer blocks, each applying pre-normalized, causally masked multi-head self-attention followed by a feed-forward network. Each prediction uses only preceding latents: $\widehat{\mathbf{z}}_t^{(d)} = f_{\text{pred}}\Big(P_\theta(\mathbf{z}_{1:t-1}^{(d)})\Big)$, where f_{pred} is a two-layer MLP with intermediate GELU activation and batch normalization. Appendix K provides architectural details and hyperparameters.

4.2 Training Objectives

The training objective combines latent prediction loss and regularization loss: $\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda \mathcal{L}_{\text{sigreg}}$, where λ controls regularization strength.

Prediction loss. We minimize the mean-squared error between predicted and target encoder latents to encourage the predictor to capture temporally predictable structure in the latent space:

$$\mathcal{L}_{\text{pred}} = \frac{1}{(T-1)M} \sum_{t=2}^{T} \left\| \widehat{\mathbf{z}}_t^{(d)} - \mathbf{z}_t^{(d)} \right\|_2^2. \tag{3}$$

Regularization loss. Jointly training the encoder and predictor can yield trivial constant representations. To prevent representation collapse, we use the sketch isotropic Gaussian regularizer (SIGReg) (Balestriero & LeCun, 2025), which matches the characteristic functions of randomly projected encoder embeddings to that of an isotropic Gaussian distribution, encouraging nonzero variance across latent dimensions. For encoder embeddings $\mathbf{z}_{t,b}^{(d)} \in \mathbb{R}^M$, where b indexes the B samples in a batch, and random unit vectors $\mathbf{a}_j \in \mathbb{R}^M$, define the empirical characteristic function of each projection as $\widehat{\phi}_{t,j}(\xi) = \frac{1}{B} \sum_{b=1}^{B} \exp\Big(i\xi\, \mathbf{a}_j^\top \mathbf{z}_{t,b}^{(d)}\Big)$. Using J projections, the regularizer is

$$\mathcal{L}_{\text{sigreg}} = \frac{B}{TJ} \sum_{t=1}^{T} \sum_{j=1}^{J} \sum_{k=1}^{K} w_k \left\| \widehat{\phi}_{t,j}(\xi_k) - e^{-\xi_k^2/2} \right\|_2^2, \tag{4}$$

where the integral is approximated using Gaussian-weighted trapezoidal quadrature with $K = 17$ uniformly spaced points $\xi_k \in [0, 3]$ and corresponding weights w_k.

4.3 Generalization to unseen data after pretraining

We pretrain NeuroLens on recordings from multiple days, using calibration trials from each day to condition and update the encoder. We then evaluate three strategies for adapting a pretrained model to unseen neural populations. These strategies vary in their calibration data requirements, computational cost, and predictive performance.

Gradient-free (*GF*) few-shot adaptation. We compute neuron signatures $\mathbf{s}_n^{(d)}$ from calibration trials sampled from an unseen day and apply the pretrained identity network h_{id} (Eq. 2) to infer identities for previously unseen neurons and obtain neural embeddings (Fig. 1B–C). Cross-attention accommodates varying neuron counts, mapping the new population into the learned latent space without updating model parameters. For neural decoding, we train a linear decoder on the frozen predictor representations $\widehat{\mathbf{z}}_t^{(d)}$.

Gradient-based (*GB*) few-shot adaptation. We adapt to data from a new day using sampled calibration trials, updating only the read-in layers of the encoder, including the spike projector f_{proj}, the identity network h_{id}, and the fusion projector f_{fuse} (Figure 1). For neural decoding, we additionally update an attached decoder head. All other parameters, including the encoder cross-attention layers, region embeddings, and the AR predictor, are frozen to preserve the representations and dynamics learned during pretraining.

Full finetuning. With sufficient data from a new session, we fine-tune the entire model, including the decoder head for neural decoding, when few-shot adaptation may not perform well for tasks such as speech decoding (Willett et al., 2023).

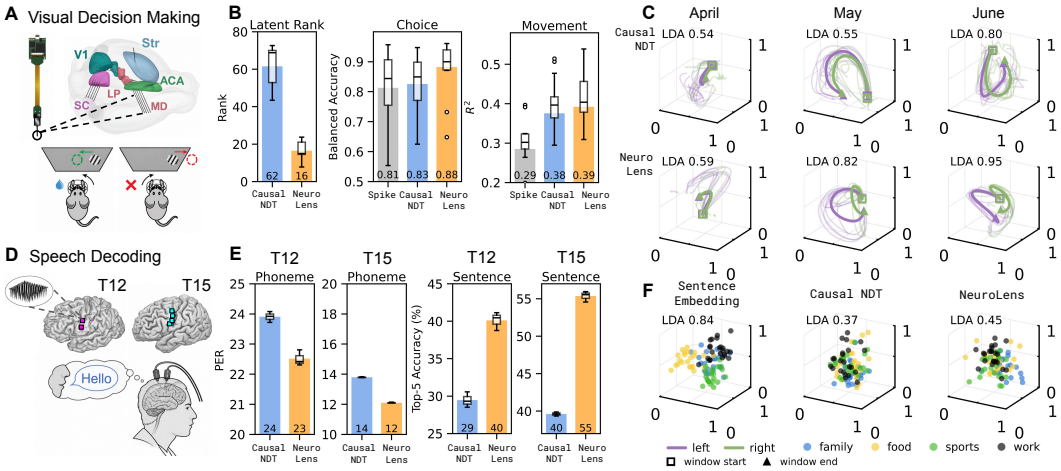

Figure 2: **NeuroLens improves decoding of decision-related and semantic information from mouse and human intracortical recordings.** (A) Chronic multi-area Neuropixels recordings from a mouse reporting left or right stimulus location by turning a wheel (*choice*). (B) Across 14 test sessions, NeuroLens achieves lower latent effective rank, better choice decoding, and comparable movement decoding relative to causal NDT. Both models outperform decoding using raw neural activity (*Spike*). (C) Latent trajectories for left- and right-choice trials, projected onto the first three principal components (PCs), from April to June. Dark (light) lines show trial averages (individual trials). As the animal learns the task, trajectories for the two choices become increasingly separated, with clearer separation in NeuroLens than in causal NDT. We apply linear discriminant analysis (LDA) to the first three PCs to quantify this separation with balanced classification accuracy. (D) Multi-area Utah array recordings from two human participants as they attempted to speak displayed sentences. (E) NeuroLens improves sentence-embedding decoding over causal NDT with comparable phoneme decoding. Lower phoneme error rate (PER) and higher top-5 sentence retrieval accuracy indicate better decoding performance. Boxplots summarize five random seeds. We report average top-5 accuracy across held-out days. Each day has a candidate pool of 288 $\sim$ 448 sentences, which keeps chance level low (1$\sim$2%). (F) For each trial, we average neural latents from causal NDT and NeuroLens across time bins, project them onto the first three PCs, and compare them with ground-truth sentence embeddings from all-MiniLM-L6-v2 (Reimers & Gurevych, 2019). Trials are colored by semantic topic; NeuroLens shows clearer semantic separation than causal NDT, quantified by LDA balanced accuracy.

5 Results

5.1 High-level vs. low-level neural decoding

Brain–computer interfaces (BCIs) can decode movement and speech and enable cursor control, but predicting high-level cognitive variables such as semantic content remains less explored. To better evaluate the neural representations learned by different models, we compare their performance on both high-level and low-level neural decoding tasks.

Baselines. For IBL, we compare NeuroLens with *Spike*, a linear model that predicts future behavior from past neural activity, and *causal NDT*, a self-supervised transformer-based autoencoder that predicts neural activity in the next time bin from past activity (Ye & Pandarinath, 2021). For human data, we use only causal NDT because using linear decoders with neural activity input perform poorly. For all baselines, we match latent embedding size, model capacity, and training setup to ensure a fair comparison (Appendix K). These comparisons test whether latent-space prediction better captures decisions and sentence semantics than observation-space prediction.

Evaluation on IBL data. We evaluate learned representations from single-session models on low- and high-level decoding tasks on the same 14 days that are later held out in Sec. 5.2, each modeled separately; see Appendix D for data splits. We train NeuroLens and causal NDT separately for each day, freeze their parameters, and extract latents $\widehat{\mathbf{z}}_t^{(d)}$ using sliding windows with stride 1 and up to 59 preceding bins. Both models use past neural activity to predict the neural activity (causal NDT) or neural latent (NeuroLens) at the current time step, which is then used for decoding. We evaluate representation quality with linear probing while keeping the pretrained models frozen: we fit linear decoders to predict behavior, select pretrained checkpoints by validation decoding performance, and

report test results. We report R^2 for five time-bin-level variables—wheel velocity, whisker motion energy from each camera, and left-camera paw and nose speeds—and balanced accuracy for choice after pooling representations across all time steps within each trial into a single vector.

Evaluation on human data. To evaluate representation quality, we use the provided data split (Appendix D), with the majority of recording days represented in both splits to avoid temporal distribution shifts. We pretrain NeuroLens and causal NDT on pooled training data from both participants in a self-supervised manner. During pretraining, we evaluate models on the validation set using a linear probe to predict sentence embeddings, measured by cosine similarity and top-5 closed-vocabulary sentence retrieval accuracy (Appendix E.3), and an MLP to decode phonemes, measured by phoneme error rate (PER) (Appendix E.2). We select checkpoints by aggregate validation performance across these metrics. Unlike the linear-probing-only IBL experiments, strong human decoding performance requires full-model fine-tuning. For each model, we initialize separate T12 and T15 models from its shared checkpoint and fine-tune on each participant's data. For each baseline, we train phoneme and sentence-embedding decoders separately, attaching a head to the predictor output with CTC and MSE loss, respectively (Appendix G).

Results. For IBL data, Figure 2B shows that both causal NDT and NeuroLens outperform raw-spike decoding, confirming the benefit of learning nonlinear neural latents via SSL. Notably, NeuroLens achieves higher choice decoding accuracy and comparable movement-related R^2 despite a lower effective latent rank than causal NDT (see Appendix E.1 for rank computation), indicating that latent-space models can learn more task-relevant representations while remaining far more compact than reconstruction-based models. This is reinforced by Figure 2C, where, as the animal transitions from novice to expert between April and June, left- and right-trial latents (choice condition) grow increasingly separable—more so for NeuroLens than for causal NDT, reflecting denoised, more task-relevant encoding. A parallel pattern emerges in the human data: Figure 2E shows that NeuroLens substantially outperforms causal NDT on sentence-embedding decoding while achieving comparable performance on phoneme decoding, and that NeuroLens's latents separate more cleanly by semantic class than those of causal NDT, pointing to more semantically meaningful representations. Relative to causal NDT, NeuroLens achieves larger gains on choice (6%) than on movement (2%) decoding, and on sentence-embedding (38%) than on phoneme (4-16%) decoding, suggesting that it captures richer global, semantic structure from neural activity overall.

5.2 Generalization Across Time

Representations that generalize across days help track learning-related neural changes despite recording drift and maintain stable BCI decoding performance over time. To this end, we evaluate models on unseen future data using the various adaptation regimes described in Sec. 4.3.

Baselines. For IBL, we compare NeuroLens with causal NDT (Section 5.1) and four additional baselines: (1) *CEBRA*: a self-supervised contrastive learning method that aligns neural samples with similar temporal contexts to capture temporally consistent neural dynamics (Schneider et al., 2023). (2) *SPINT*: a supervised sliding-window transformer decoder that learns context-dependent neuron identities from raw calibration activity to adapt to unseen neural populations (Le et al., 2026). (3) *POYO*: a supervised decoder based on Perceiver IO that maps spike tokens to latent representations via cross-attention, processes them with a transformer, and decodes behavior through a second cross-attention module (Azabou et al., 2023). (4) *POSSM*: replaces POYO's transformer with a recurrent state-space model (SSM) for efficient real-time decoding (Ryoo et al., 2026). For human data, we exclude SPINT because its fixed window size cannot accommodate variable-length trials, and POYO because it performs poorly at sentence-embedding decoding. We control for model capacity and training setup across all baselines except CEBRA, for which we adopt the original architecture to ensure a fair comparison (Appendix K and L).

Evaluation on IBL data. Using a chronological split (Figure 3A), we pretrain all baselines on 90% of the data from each of 39 past days, using behavioral labels only for supervised models. We evaluate on 14 held-out future days, splitting each into 20% adaptation, 10% validation, and 70% test data, under two few-shot settings (Section 4.3). In *gradient-free (GF) adaptation*, we use 64 calibration trials to infer neuron identities without updating pretrained parameters. Using sliding windows with stride 1, NeuroLens extracts and aggregates next-timestep representations for training linear decoders. In *gradient-based (GB) adaptation*, we fine-tune each model according to its

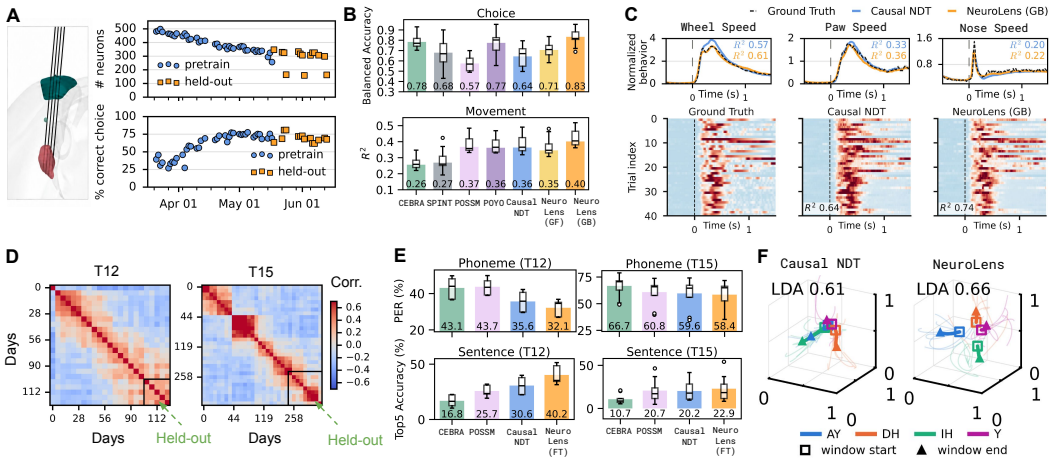

Figure 3: **NeuroLens learns neural representations that generalize across time, enabling rapid adaptation to unseen neural populations and more stable decoding than SOTA baselines.** (A) The mouse makes increasingly accurate decisions about the visual stimulus location from April to June, while the number of simultaneously recorded neurons decreases over time. We pretrain on the first 39 days and test generalization on the held-out last 14 days. (B) Across 14 held-out days, NeuroLens outperforms baselines in choice and movement decoding, demonstrating better generalization across time. (C) Comparison of predicted movement-related behavior averaged across all trials (top) and predicted behavior from single trials (bottom), shown to demonstrate that the baselines can predict trial-to-trial variability in behaviors. The R^2 values for both trial-average and single-trial behavior are shown. The dashed line shows the stimulus onset. (D) T12 and T15 show clear day-to-day distribution shifts in neural activity. We hold out the last 5 and 9 days, respectively, for testing and pretrain on the remaining days. (E) NeuroLens (FT, full fine-tuning) consistently outperforms baselines in phoneme and sentence embedding decoding across 5 (T12) and 9 (T15) held-out days, reflecting better generalization over time. Lower phoneme error rate (PER) and higher top-5 sentence retrieval accuracy indicate better decoding performance. (F) Latent trajectories for different phoneme onsets, projected onto the first three PCs, across held-out days in T15. Dark (light) lines show averaged (individual) time windows. NeuroLens shows clearer phoneme separation than causal NDT, indicating better held-out adaptation, quantified by LDA balanced classification accuracy.

architecture. For self-supervised models (NeuroLens, causal NDT, and CEBRA), we train separate decoder heads for trial-level choice using binary cross-entropy (BCE) and the five concatenated time-bin-level variables using MSE. We update the read-in layers of the encoders, as well as the decoder heads, while freezing the transformer backbones. SPINT similarly updates its read-in layers and behavior decoder without adding parameters, freezing its cross-attention module, transformer backbone, and learned queries instead. POSSM also updates only its read-in layers, unit embeddings, and decoder heads, freezing the rest, following its original setup (Section 2.3 in (Ryoo et al., 2026)). See Appendix K and L for decoder and training details.

Evaluation on human data. We split T12 and T15 chronologically (Appendix D), holding out future days for testing (Figure 3D). We pretrain NeuroLens and causal NDT without labels on pooled training data from both participants and select checkpoints by validation performance on phoneme and sentence-embedding decoding (Section 5.1). Because few-shot adaptation performs poorly, we initialize separate models for each participant and task from each model's shared checkpoint and fully fine-tune them on that participant's training data. We select final checkpoints by aggregate validation performance and evaluate on held-out test days.

Results. In Figure 3B, we benchmark multi-day pretrained NeuroLens against other pretrained baselines and find that it achieves the best choice and movement-related behavior decoding among all methods, demonstrating stronger generalization to unseen held-out days. Strikingly, NeuroLens (GF) needs only a handful of unlabeled calibration trials for few-shot adaptation (no gradient updates required) and still outperforms SPINT, POSSM and causal NDT on choice decoding, as well as CEBRA and SPINT on movement decoding. This suggests that its encoder can rapidly adapt to previously unseen neural populations. Figure 2C offers a qualitative comparison of decoded movement-related behavior between causal NDT and NeuroLens on held-out days, with the top panel showing trial-averaged ground truth (GT) versus predicted behavior and the bottom panel showing single-trial GT versus predicted behavior. Compared to causal NDT, NeuroLens more faithfully captures

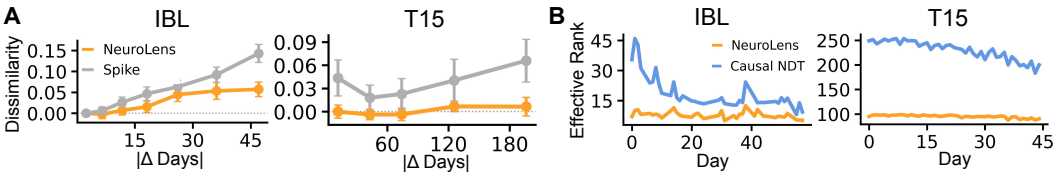

Figure 4: **NeuroLens learns denoised, stable latents with more consistent representations across days and stable effective rank over time.** (A) We measure representational similarity between each pair of recording days for NeuroLens latents and PCA latents of raw spikes using 1−RDM similarity (Appendix E.4). Error bars denote one standard error of the dissimilarity metrics across day pairs. As the temporal gap between recording days increases, NeuroLens latents exhibit greater temporal consistency than raw spikes in both the IBL and human (T15) datasets. (B) Figure 3A shows that the number of recorded neurons decreases over time, as reflected by the decreasing effective rank of the causal NDT latents. In contrast, NeuroLens maintains a more compact and stable latent rank over time in both the IBL and T15 datasets.

both trial-averaged patterns and trial-to-trial variability in wheel, paw, and nose speed, pointing to better generalization over time. Human data similarly show better generalization over time, with NeuroLens outperforming baselines in phoneme and sentence embedding decoding across held-out days (Figure 2E). Finally, Figure 2F shows that on held-out days, NeuroLens's latents separate more clearly across the four selected phoneme classes than causal NDT's, offering further evidence of its stronger temporal generalization.

5.3 Task-Related Representation Drift vs. Recording Nonstationarity

Having established that NeuroLens captures task-relevant representations that generalize across temporal distribution shifts, we next investigate how they evolve in chronic recordings.

Representation drift. In chronic data, changes in neural activity can reflect learning as well as intrinsic changes unrelated to behavior. Here, we loosely refer to these changes as representation drift. We first examine how representational similarity changes with the temporal separation between recording days. For each day pair, we compare trial-by-trial representations using representational similarity analysis (RSA; Appendix E.4) and define representational dissimilarity as 1 - RDM similarity, with higher values indicating more dissimilar representational geometry. As the temporal separation increases, representations become increasingly dissimilar in the raw spike space, whereas NeuroLens latents remain more similar. Thus, NeuroLens learns temporally consistent representations that are robust to distribution shifts in chronic neural activity.

Latent dimensionality. We next quantify how representation dimensionality changes over time by computing the effective rank of the causal NDT and NeuroLens latents for each recording day. As the number of recorded neurons declines across the IBL chronic recordings (Figure 3A), the effective rank of causal NDT decreases substantially over time, tracking the loss of recorded neurons. In contrast, NeuroLens's effective rank remains stable across days in both datasets (Figure 4B).

These results show that NeuroLens learns stable, compact representations that separate task-related changes from recording nonstationarity, enabling studies of learning over long timescales.

6 Discussion

In this work, we introduce NeuroLens, a JEPA-based self-supervised model that learns denoised, semantically informative representations from chronic neural recordings. Across mouse and human intracortical data, NeuroLens improves decoding of higher-level task variables, generalizes across recording days, and enables rapid few-shot adaptation to unseen neural populations. By extracting stable, task-relevant structure from nonstationary neural activity, NeuroLens provides a framework for studying neural representations over long timescales. Despite its promise, this work has several limitations. Although we analyze representation drift across days, we do not disentangle changes due to learning and behavioral changes from intrinsic neural changes, nor do we model cross-area interactions or region-specific representation drift. Separating these sources of variability through joint behavioral modeling and cross-area analyses is an important direction for future work.

AI USE STATEMENT

The authors used generative AI tools for limited assistance with model implementation and debugging, as well as manuscript refinement. All AI-assisted content was reviewed by the authors, who take responsibility for the final work.

REFERENCES

Alexandre Andre, Shivashriganesh P. Mahato, Vinam Arora, Keshav Balaji, Divyansha Lachi, Nanda H. Krishna, Jingyun Xiao, Yizi Zhang, Ximeng Mao, Wenrui Ma, Han Yu, International Brain Laboratory, et al. Brainwidebench: Benchmarking large-scale pretraining and across-animal transfer in multi-region neural recordings. *arXiv preprint arXiv:2609.22064*, 2026. URL https://arxiv.org/abs/2609.22064.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, 36:44937–44956, 2023.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Dan Biderman, Matthew R Whiteway, Cole Hurwitz, Nicholas Greenspan, Robert S Lee, Ankit Vishnubhotla, Richard Warren, Federico Pedraja, Dillon Noone, Michael M Schartner, et al. Lightning pose: improved animal pose estimation via semi-supervised learning, bayesian ensembling and cloud-native open-source tools. *Nature methods*, 21(7):1316–1328, 2024.

Célian Bimbard, Flóra Takács, Joana A Catarino, Julie MJ Fabre, Sukriti Gupta, Stephen C Lenzi, Maxwell D Melin, Nathanael O'Neill, Ivana Orsolic, Magdalena Robacha, et al. An adaptable, reusable, and light implant for chronic neuropixels probes. *Elife*, 13:RP98522, 2025.

Nicholas S Card, Maitreyee Wairagkar, Carrina Iacobacci, Xianda Hou, Tyler Singer-Clark, Francis R Willett, Erin M Kunz, Chaofei Fan, Maryam Vahdati Nia, Darrel R Deo, et al. An accurate and rapidly calibrating speech neuroprosthesis. *New England Journal of Medicine*, 391(7):609–618, 2024.

Mark M Churchland, Byron M Yu, John P Cunningham, Leo P Sugrue, Marlene R Cohen, Greg S Corrado, William T Newsome, Andrew M Clark, Paymon Hosseini, Benjamin B Scott, et al. Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature neuroscience*, 13(3):369–378, 2010.

Chaofei Fan, Nick Hahn, Foram Kamdar, Donald Avansino, Guy Wilson, Leigh Hochberg, Krishna V Shenoy, Jaimie Henderson, and Francis Willett. Plug-and-play stability for intracortical brain-computer interfaces: a one-year demonstration of seamless brain-to-text communication. *Advances in neural information processing systems*, 36:42258–42270, 2023.

Ebrahim Feghhi, Shreyas Kaasyap, Nima Hadidi, and Jonathan Kao. Time-masked transformers with lightweight test-time adaptation for neural speech decoding. *Advances in Neural Information Processing Systems*, 38:93047–93072, 2026.

Juan A Gallego, Matthew G Perich, Raeed H Chowdhury, Sara A Solla, and Lee E Miller. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature neuroscience*, 23 (2):260–270, 2020.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

Brianna M Karpowicz, Joel Ye, Chaofei Fan, Pablo Tostado-Marcos, Fabio Rizzoglio, Clay Washington, Thiago Scodeler, Diogo de Lucena, Samuel R Nason-Tomaszewski, Matthew J Mender, et al. Few-shot algorithms for consistent neural decoding (falcon) benchmark. *Advances in Neural Information Processing Systems*, 37:76578–76615, 2024.

Brianna M Karpowicz, Yahia H Ali, Lahiru N Wimalasena, Andrew R Sedler, Mohammad Reza Keshtkaran, Kevin Bodkin, Xuan Ma, Daniel B Rubin, Ziv M Williams, Sydney S Cash, et al. Stabilizing brain-computer interfaces through alignment of latent dynamics. *Nature Communications*, 16(1):4662, 2025.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008.

Trung Le, Hao Fang, Jingyuan Li, Tung Nguyen, Lu Mi, Amy L Orsborn, Uygar Sümbül, and Eli Shlizerman. Spint: Spatial permutation-invariant neural transformer for consistent intracortical motor decoding. *Advances in Neural Information Processing Systems*, 38:25815–25835, 2026.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710, 1966.

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

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Maxwell D Melin, Anup Khanal, Marvin Vasquez, Michael B Ryan, Anne K Churchland, and Joao Couto. Large scale, simultaneous, chronic neural recordings from multiple brain areas. *bioRxiv*, pp. 2023–12, 2024.

Marius Pachitariu, Nicholas Steinmetz, Shabnam Kadir, Matteo Carandini, and Harris Kenneth D. Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. *BioRxiv*, pp. 061481, 2016.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Andrew J Peters, Simon X Chen, and Takaki Komiyama. Emergence of reproducible spatiotemporal activity during motor learning. *Nature*, 510(7504):263–267, 2014.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Michael E Rule, Adrianna R Loback, Dhruva V Raman, Laura N Driscoll, Christopher D Harvey, and Timothy O'Leary. Stable task information from an unstable neural population. *elife*, 9: e51121, 2020.

Avery Hee-Woon Ryoo, Nanda H Krishna, Ximeng Mao, Mehdi Azabou, Eva L Dyer, Matthew G Perich, and Guillaume Lajoie. Generalizable, real-time neural decoding with hybrid state-space models. *Advances in Neural Information Processing Systems*, 38:51764–51791, 2026.

Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, 2023.

Sofia Skromne Carrasco, Guillaume Viejo, and Adrien Peyrache. Months-long stability of the head-direction system. *Nature*, 652(8108):167–173, 2026.

Tian-Yi Wang, Chengcong Feng, Chengyao Wang, Chi Ren, and Zhengtuo Zhao. Dynamics of mesoscale brain network during visual discrimination learning revealed by chronic, large-scale single-unit recording. *Elife*, 14:RP108083, 2026.

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

Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *bioRxiv*, pp. 2021–01, 2021.

Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. Neural data transformer 2: multi-context pretraining for neural spiking activity. *Advances in Neural Information Processing Systems*, 36:80352–80374, 2023.

Yizi Zhang, Yanchen Wang, Donato M Jiménez-Benetó, Zixuan Wang, Mehdi Azabou, Blake Richards, Renee Tung, Olivier Winter, Eva Dyer, Liam Paninski, et al. Towards a" universal translator" for neural dynamics at single-cell, single-spike resolution. *Advances in Neural Information Processing Systems*, 37:80495–80521, 2024.

Yizi Zhang, Yanchen Wang, Mehdi Azabou, Alexandre Andre, Zixuan Wang, Hanrui Lyu, The International Brain Laboratory, Eva Dyer, Liam Paninski, and Cole Hurwitz. Neural encoding and decoding at scale. *ArXiv*, pp. arXiv–2504, 2025.

A Calibration-Based Neuron Identification

For each recording day d, we compute a fixed physiological and functional signature for every neuron using only a small set of calibration trials. Let $\mathbf{x}^{(d,k)} \in \mathbb{R}^{T\times N_d}, k = 1,\dots,K$, denote the binned spike-count matrix from calibration trial k on day d, where N_d is the number of recorded neurons and $x_{t,n}^{(d,k)}$ is the spike count of neuron n in time bin t. By default, $K = 64$ training trials are selected once and held fixed. The resulting signature $\mathbf{s}_n^{(d)}$ is computed without gradients and frozen during model training. As described in the main text, the learnable network h_{id} maps this signature to the neuron embedding $\mathbf{e}_n^{(d)}$. We compute the following statistics independently for each calibration trial and then average them across trials, and use $\epsilon = 10^{-6}$ for numerical stability.

Firing Rate Statistics. For each neuron n and calibration trial k, we define the mean and variance as follows:

$$\mu_n^{(d,k)} = \frac{1}{T}\sum_{t=1}^{T} x_{t,n}^{(d,k)}, \qquad \left(\sigma_n^{(d,k)}\right)^2 = \frac{1}{T-1}\sum_{t=1}^{T}\left(x_{t,n}^{(d,k)} - \mu_n^{(d,k)}\right)^2.$$

The firing-rate-based neuron signature is computed as follows:

$$\mathbf{s}_{n,\mathrm{FR}}^{(d)} = \left\langle \left[\log\!\left(1+\mu_n^{(d,k)}\right),\ \log\!\left(1+\sigma_n^{(d,k)}\right),\ \frac{\left(\sigma_n^{(d,k)}\right)^2}{\mu_n^{(d,k)}+\epsilon},\ \frac{\sigma_n^{(d,k)}}{\mu_n^{(d,k)}+\epsilon},\ \frac{1}{T}\sum_{t=1}^{T}\mathbf{1}\{x_{t,n}^{(d,k)} > 0\} \right] \right\rangle_k,$$

where $\langle\cdot\rangle_k$ denotes averaging over the K calibration trials. These five statistics are the log mean rate, log standard deviation, Fano factor (Churchland et al., 2010), coefficient of variation, and fraction of active time bins.

Autocorrelogram. Let $\widetilde{x}_{t,n}^{(d,k)} = x_{t,n}^{(d,k)} - \mu_n^{(d,k)}$ be the mean-centered spike count. For lags $l = 1,\dots,L$, we define the l-lag autocorrelation-based neuron signature $\mathbf{s}_{n,\mathrm{ACG}}^{(d)} \in \mathbb{R}^L$ as follows:

$$\mathbf{s}_{n,\mathrm{ACG}}^{(d)}(l) = \left\langle \frac{\frac{1}{T-l}\sum_{t=1}^{T-l}\widetilde{x}_{t,n}^{(d,k)}\widetilde{x}_{t+l,n}^{(d,k)}}{\frac{1}{T}\sum_{t=1}^{T}\left(\widetilde{x}_{t,n}^{(d,k)}\right)^2 + \epsilon} \right\rangle_k.$$

This normalized autocorrelogram captures each neuron's intrinsic timescale and its tendency toward bursting or refractoriness. We omit the 0-lag term because normalization makes it approximately one by construction. For the IBL data, we use $L = 20$ lags.

Population Coupling. Within each calibration trial k, we z-score each neuron's spike counts: $z_{t,n}^{(d,k)} = \frac{\widetilde{x}_{t,n}^{(d,k)}}{\sigma_n^{(d,k)}+\epsilon}$. We then define the leave-one-neuron-out population signal as $p_{t,n}^{(d,k)} = \sum_{\substack{j=1 \\ j\neq n}}^{N_d} z_{t,j}^{(d,k)}$ and normalize it to obtain $\overline{p}_{t,n}^{(d,k)}$. For lags $l = 0, \ldots, L-1$, the l-lag population-coupling-based neuron signature $\mathbf{s}_{n,\mathrm{PC}}^{(d)} \in \mathbb{R}^L$ is defined as follows:

$$\mathbf{s}_{n,\mathrm{PC}}^{(d)}(l) = \left\langle \frac{1}{T-l} \sum_{t=1}^{T-l} z_{t,n}^{(d,k)} \, \overline{p}_{t+l,n}^{(d,k)} \right\rangle_k .$$

This is the Pearson cross-correlation between neuron n and the activity of the remaining population, with the population shifted by l bins. , Positive lags indicate whether activity in neuron n precedes activity in the rest of the population. Since the population signal is formed by summing over all other neurons, the feature does not depend on neuron order and can be computed for data with different numbers of neurons. For the IBL data, we use $L = 16$ lags.

B Calibration-Based Channel Drift Adaptation

Channel Statistics. For each recording day d, we use a fixed set of calibration trials to compute five summary statistics for each channel and then average these statistics across trials. The choice of statistics depends on the input data. For T12 spike count data (Willett et al., 2023), which are non-negative and count-valued, we use the firing-rate statistics described in Appendix A: mean activity, standard deviation, Fano factor (Churchland et al., 2010), coefficient of variation, and the fraction of nonzero time bins. In contrast, spike-band power (SBP) is not count-valued, and the T15 dataset provides pre-z-scored features without raw spike counts (Card et al., 2024). For these inputs, we instead use statistics that describe the distribution's location, spread, skewness, tail weight, and magnitude.

Per-Day Drift Context Vector. We normalize each statistic using constants estimated from the training days, so that no individual statistic dominates because of its numerical scale. For each channel, we first convert its five normalized calibration statistics into a learned channel embedding using the same MLP layer. This shared mapping lets the model represent different patterns of activity and recording drift in a common feature space. We then average the channel embeddings separately within each input modality (spike counts and SBP). The resulting modality-level vectors summarize the typical calibration distribution for that modality on a given day. Finally, we concatenate these modality summaries and project them into a single day context vector $\mathbf{c}^{(d)}$. Because the channel embeddings are averaged, the context vector $\mathbf{c}^{(d)}$ does not depend on the order of channels.

Per-Day FiLM Conditioning Conditioned on $\mathbf{c}^{(d)}$, a shared hypernetwork generates the scale and shift parameters of a day-specific feature-wise linear modulation (FiLM) layer (Perez et al., 2018):

$$\left(\boldsymbol{\gamma}_\ell^{(d)}, \boldsymbol{\beta}_\ell^{(d)}\right) = h_{\mathrm{hyper},\ell}\left(\mathbf{c}^{(d)}\right), \qquad \widetilde{\mathbf{h}}_{t,\ell}^{(d)} = \left(1 + \boldsymbol{\gamma}_\ell^{(d)}\right) \odot \mathbf{h}_{t,\ell}^{(d)} + \boldsymbol{\beta}_\ell^{(d)},$$

where $\mathbf{h}_{t,\ell}^{(d)}$ is the output at time t in encoder layer ℓ. The FiLM layer therefore applies a day-specific, feature-wise affine transformation to the encoder layers. The hypernetwork parameters are shared across days and learned during training. For an unseen day, calibration trials are used only to compute $\mathbf{c}^{(d)}$, which generates the corresponding FiLM parameters without updating the model weights.

C Data Preprocessing

C.1 IBL chronic Neuropixels data

For the IBL chronic Neuropixels dataset, we analyzed neural and video recordings collected over 3 months from a single mouse (*MM008*) (Melin et al., 2024). We derived binned spike counts from the neural recordings and behavioral variables from the video recordings, as described below.

Neural Activity (`spikes`). For each recording session, we combined spike-sorted units using *Kilosort* (Pachitariu et al., 2016) from all probes into a single neural population. We retained units with a mean firing rate greater than 1 Hz, computed using only the training portion of each recording; thus, unit selection did not use validation or test data. Recorded units were located mainly in primary and secondary motor cortex and dorsal striatum, with smaller numbers in prelimbic and dorsal anterior cingulate cortex. Spike times were binned into 16.7 ms intervals ($\sim$60 Hz), and raw spike counts were used as model inputs.

We divided each session into trial and non-trial periods. For *trial* data, each trial was represented by a 2 s window spanning 0.8 s before to 1.2 s after stimulus onset (120 bins). For *non-trial* data, we excluded the interval from 0.2 s before to 0.8 s after stimulus onset for every valid trial. We segmented the remaining recording into consecutive, non-overlapping 2 s windows (120 bins).

Trial-Level Behavior. We used the mouse's binary choice in the visual decision-making task as a discrete behavioral variable. On each trial, the mouse reported whether it perceived the stimulus on the left or right side. Choice labels were therefore available only for trial windows; non-trial windows were unlabeled.

Time-Bin-Level Behavior. From video recordings, we extracted wheel speed, whisker motion energy from the left and right cameras, and paw and nose speeds from the left camera using keypoint estimation (Biderman et al., 2024). Each signal was resampled to the same 16.7 ms bins used for spike counts. Because these measures are available during both trial and non-trial periods, we included them in both window types. Frames with missing keypoints were assigned a speed of zero and were not interpolated. Each behavioral variable was scaled to the range [0, 10] using its minimum and maximum values from the training portion of the recording only. The resulting scaling parameters were then applied consistently across all data splits and window types.

C.2 Human chronic Utah array data.

Neural Activity (`spikes` and `SBP`). We used channel-level features rather than spike-sorted units. Both T12 (Willett et al., 2023) and T15 (Card et al., 2024) provide threshold-crossing counts and spike-band power (SBP) for each electrode in 20 ms bins. We smoothed each feature separately within each trial using a Gaussian kernel with a standard deviation of 40 ms, then combined every five consecutive bins into a 100 ms input token. Because the sentences differ in length, trials have variable durations.

For T12, we used the 128 electrodes in ventral premotor cortex (area 6v) and excluded electrodes in area 44. The two features from each electrode yielded 256 input dimensions per time bin. We first z-scored each feature within each recording session, then z-scored it again within each day using statistics computed only from that day's training trials. For T15, we used all 256 electrodes in the precentral gyrus, which yielded 512 input dimensions per time bin. We used the publicly released, normalized T15 features without further normalization.

Phoneme Labels. We derived phoneme labels following the procedures of Willett et al. (2023); Card et al. (2024). For T12, we converted each sentence to a phoneme sequence with the *g2p-en* grapheme-to-phoneme package. Before conversion, we lowercased the text and removed punctuation other than apostrophes. We then removed stress markers and added a silence token at word boundaries and at the end of each sentence. This produced a vocabulary of 39 phonemes and one silence token; we added a blank token for CTC training. For T15, we used the phoneme sequences provided with the dataset, which use the same 40 non-blank classes. When training supervised baselines jointly across T12 and T15 datasets, we remapped T15 labels to the T12 class order to ensure consistent labeling across participants.

Sentence Embeddings. We encoded each sentence with the pretrained `all-MiniLM-L6-v2` model (Reimers & Gurevych, 2019). We averaged its contextual token embeddings and L2-normalized the result to obtain one 384-dimensional vector per sentence. The shared encoder placed sentence representations from both participants in the same embedding space.

D Data Splits

For both the IBL and human chronic datasets, we considered two experimental settings: (1) *high- vs. low-level neural decoding* (Section 5.1) and (2) *generalization across time* (Section 5.2). The two settings use the same model architectures and training procedures (Appendices K and L) and differ only in how the data are split. In the first setting, training and test data were drawn from the same recording days, allowing us to evaluate representation quality without requiring generalization across days. In the second setting, we split the data chronologically: models were trained on earlier recording days and evaluated on later, held-out days, creating a temporal distribution shift between training and test data.

IBL chronic Neuropixels data. For *high- vs. low-level neural decoding*, we focused on single-session models. Within each session, we split the data into 70% training, 10% validation and 20% test sets. Because all splits came from the same recording day, this setting does not introduce a substantial temporal distribution shift. For *generalization across time*, we split the full *MM008* dataset chronologically to introduce an across-day distribution shift. We used 39 days collected between *2023-03-21* and *2023-05-17* for multi-day pretraining, and held out 14 subsequent days, collected between *2023-05-18* and *2023-06-13* for evaluation. When adapting the model to each held-out day, we used the first 20% of trials as the adaptation set, the next 10% as the validation set for checkpoint selection, and the remaining 70% to evaluate decoding performance.

Human chronic Utah array data. For *high- vs. low-level neural decoding*, all data splits were drawn from the same recording days. For T12 (Willett et al., 2023), we use the provided training partition for training, hold out the last block of each training day for validation, and use the provided evaluation partition as the test set, resulting in 7,920 training, 880 validation, and 880 test trials across 24 recording days. For T15 (Card et al., 2024), we use the provided training partition and divide the provided evaluation partition equally into validation and test sets, resulting in 8,072 training, 704 validation, and 722 test trials across 45 days. For *generalization across time*, we split each participant's full dataset chronologically, using the earliest 80% of recording days for training and holding out the final 20% for evaluation. For T12, we trained on 18 past days and evaluated on the subsequent 5 days (excluding 2022-06-14 due to data quality). For T15, we trained on 36 past days and evaluated on the following 9 days. Within each training day, we used the first 90% for training and the remaining 10% for validation. For each held-out day, the first 20% served only as unlabeled calibration data, and we reported decoding performance on the remaining 80%.

E Metrics

E.1 Effective Rank of Neural Latents

We use effective rank to quantify the dimensionality of a latent representation. Intuitively, it measures the number of dimensions that contribute substantially to the representation. For each recording day, we estimated the within-trial covariance Σ across the trial-aligned time windows (Appendix C). We sampled the same number of trials from each day so that differences in the computed effective rank were not driven by differences in trial count. For a covariance matrix Σ with eigenvalues $\lambda_1, \ldots, \lambda_D \geq 0$, where D is the latent dimension, let $p_i = \frac{\lambda_i}{\sum_{j=1}^{D} \lambda_j}$. The effective rank is the exponential of the Shannon entropy of this normalized eigenvalue spectrum:

$$\mathrm{erank}(\Sigma) = \exp\left(-\sum_{i=1}^{D} p_i \log p_i\right),$$

which ranges from 1, when variance is concentrated in a single direction (representation collapse), to D, when variance is distributed equally across all dimensions. We computed the effective rank of

the PCs of raw neural activity $\mathbf{x}_t^{(d)}$, the causal NDT neural latents, and the NeuroLens latents $\hat{\mathbf{z}}_t^{(d)}$. Figure 4 shows how these values change across recording days.

E.2 Phoneme Error Rate

We used a 41-token phoneme vocabulary comprising all phonemes, silence, and the CTC blank token. We computed the phoneme error count by first removing repeated tokens and blanks from the decoded phoneme sequences and then calculating the Levenshtein edit distance (Levenshtein, 1966) between the decoded and ground-truth sequences, counting substitutions, insertions, and deletions. We then normalized the error count by the total number of ground-truth phonemes to obtain the phoneme error rate.

E.3 Top-5 Sentence Retrieval Accuracy

Top-5 sentence retrieval accuracy measures whether a model's predicted sentence embedding is close enough to the ground-truth (GT) sentence to pick it out of a candidate pool: a trial counts as correct if the GT sentence is among the 5 most similar to the prediction. We use this metric on the T12 and T15 datasets to evaluate representation quality via the semantic decoding task, where the target is the 384-dimensional `all-MiniLM-L6-v2` sentence embedding (Reimers & Gurevych, 2019). We report average top-5 accuracy across held-out days. Each day has a candidate pool of 288 $\sim$ 448 sentences, which keeps chance level low (1$\sim$2%). Predicted and GT candidate embeddings are first centered by subtracting the mean GT embedding across test trials, so we can focus on trial-to-trial variability; candidates are then ranked by cosine similarity to the centered prediction.

E.4 Representation Similarity Analysis

Representational similarity analysis. Representational Similarity Analysis (RSA) (Kriegeskorte et al., 2008) compares representations through their second-order structure. Rather than aligning representations point by point, RSA first converts each representation into a representational dissimilarity matrix (RDM), which quantifies the dissimilarity between every pair of conditions (time bins within a trial). It then compares two representations by correlating their RDMs. Because this comparison depends on pairwise relationships rather than the coordinate system itself, RSA is invariant to rotations of the underlying representation and can be applied across models or recording days without a shared basis.

Cross-day representational dissimilarity. For each day d and representation (PCs of raw neural activity $\mathbf{x}_t^{(d)}$, causal NDT latents, or NeuroLens latents $\hat{\mathbf{z}}_t^{(d)}$), we constructed a temporal RDM, $\mathrm{RDM}^{(d)} = \mathbf{1} - \mathbf{P}^{(d)}$, where $\mathbf{P}^{(d)} \in \mathbb{R}^{T\times T}$ contains Pearson correlations between time bins. For a day pair d_i and d_j, we quantified their dissimilarity as

$$\tau_{i,j} = 1 - \rho_{\text{spearman}}\left(\mathrm{RDM}^{(d_i)}, \mathrm{RDM}^{(d_j)}\right),$$

where ρ_{spearman} denotes the Spearman correlation. This measure is zero when the two days have identical temporal geometries and increases as their geometries diverge.

Within-day correction for sampling variability. To adjust for within-day sampling variability, we estimated the within-day split-half drift ($\tau_{i,i}$), by computing RSA between RDMs estimated from two randomly selected halves of trials from day d_i. Our primary measure was the within-day-drift-corrected dissimilarity:

$$\widetilde{\tau}_{i,j} = \tau_{i,j} - \frac{1}{2}\left(\tau_{i,i} + \tau_{j,j}\right).$$

We visualized $\widetilde{\tau}_{i,j}$ as a function of the lag between each pair of days in Figure 4, which provides a model-free characterization of representational drift.

F Pretraining

Multi-day pretraining on prior sessions is beneficial only if it outperforms training directly on the target session. We evaluated this on the IBL dataset while holding the target-session data budget

fixed: a single-session model was trained from scratch on the same training split used to adapt the pretrained model. Thus, the two baselines differed only in exposure to the 39 prior sessions, with identical model architectures and training procedures.

Setup. We randomly selected 5 of the 14 held-out sessions using a fixed seed. For each session, a NeuroLens model was trained from random initialization on the session's 20% adaptation split, selected using the same 10% validation split, and evaluated on the same held-out 70% test split as its pretrained counterpart (Appendix D). The architecture, optimizer, learning-rate schedule, and epoch budget matched those of the pretrained counterpart (Appendix L); the two models were parameter-matched.

Result. In Table 1, pretraining improves movement-related behavior decoding on every session tested by 0.046 to 0.094 R^2 (mean 0.076), which is large relative to the spread across baselines in Section 5.2. For choice decoding, the pretrained model is ahead on four of the five sessions.

Table 1: Pretraining on 39 days vs. training on the target day alone, on 5 held-out sessions drawn at random. Both baselines under comparison use the same 20% / 10% / 70% adaptation/validation/test split for few-shot finetuning and the same architecture; they differ in whether NeuroLens is initialized with a 39-day pretrained backbone. Movement behavior R^2 is the mean over the 5 time-bin-level variables.

	Movement Behavior (R^2) ↑		Choice (Balanced Accuracy) ↑	
Session	Single-day	39-day pretrained	Single-day	39-day pretrained
2023-05-18	0.42	**0.47**	0.72	**0.76**
2023-05-22	0.41	**0.48**	0.63	**0.76**
2023-05-24	0.35	**0.44**	0.83	0.61
2023-05-30	0.38	**0.47**	0.63	**0.69**
2023-06-09	0.33	**0.42**	0.67	**0.82**
Mean	0.38	**0.45**	0.69	**0.73**

G Phoneme and Semantic Decoding

For each baseline, once its Transformer-, GRU-, or MLP-based backbone was pretrained, we attached a task-specific decoder and fine-tuned the backbone end-to-end to decode the target. For pretraining, data from T12 and T15 were pooled; the resulting model was then fine-tuned separately for phoneme and sentence-embedding decoding.

Phoneme decoding. A linear decoder head mapped the sequence of backbone outputs (e.g., $\widehat{\mathbf{z}}^{(d)}_{1:T}$ in Eq. 3) to phoneme logits, including the CTC blank token. We trained the model with CTC loss, which does not require time-bin-level alignment between neural activity and phoneme labels. At inference, we applied standard CTC decoding and evaluated performance using phoneme error rate (PER; Appendix E.2).

Semantic decoding. We pooled the sequence of backbone outputs (e.g., $\widehat{\mathbf{z}}^{(d)}_{1:T}$ in Eq. 3) by averaging across the T time bins to obtain a single sentence token, which we mapped to the target sentence embedding space with a task-specific decoder head. The decoder was trained with a cosine similarity objective to encourage alignment in representation space. At inference, we compared predicted embeddings with candidate sentence embeddings from a closed-book vocabulary and ranked candidates by cosine similarity. We report top-5 sentence retrieval accuracy; higher values indicate more accurate sentence retrieval and, therefore, better semantic decoding.

H Ablation Study of Neuron Identity Embeddings

We conducted an ablation study of the calibration-based neuron-identification method (Section 4.1 and Appendix A) on the IBL chronic dataset. Specifically, we pretrained NeuroLens without neuron-identity embeddings on data from the 39 pretraining days and fine-tuned it on 3 held-out sessions. We compared this model with the original NeuroLens model, which included calibration-based neuron-identity embeddings; all other architectural components and training procedures were

held constant. Table 2 reports decoding performance for movement-related behavior and choice. Calibration-based neuron-identity embeddings improved decoding performance across all held-out sessions and both targets.

Table 2: Ablation of calibration-based neuron-identity embeddings on 3 held-out IBL sessions. For both behavioral variables, higher R^2 and balanced accuracy correspond to better decoding performance.

	Movement Behavior (R^2) ↑		Choice (Balanced Accuracy) ↑	
Session (Day) / Neuron ID	✓	×	✓	×
2023-05-24	**0.44**	0.43	**0.88**	0.82
2023-06-05	**0.38**	0.36	**0.95**	0.93
2023-06-12	**0.38**	0.36	**0.94**	0.93

I Ablation Study of Channel Drift Adaptation

We conducted an ablation study of the calibration-based channel drift adaptation module (Section 4.1 and Appendix I) on the T12 dataset under the *generalization across time* setting described in Appendix D. Specifically, we pretrained NeuroLens without calibration-based channel drift adaptation (*Per-Day FiLM*) on T12's 18 past recording days and fine-tuned it on 5 held-out future days. We compared this model with the original NeuroLens model, which included channel-drift adaptation; all other architectural components and training procedures were held constant. Table 3 reports phoneme and sentence decoding performance. Channel drift adaptation yielded a slight improvement in phoneme decoding and a clear improvement in sentence retrieval.

Table 3: Ablation of calibration-based channel drift adaptation (*per-day FiLM*) on the T12 dataset. Values are reported as mean ± 1 standard error across 5 held-out days. Lower PER and higher sentence retrieval accuracy indicate better decoding performance.

	PER (Phoneme) (%) ↓		Top-5 Accuracy (Sentence) (%) ↑	
Held-Out Day / Per-Day FiLM	✓	×	✓	×
2022-08-11	**27.23**	27.50	**51.39**	40.62
2022-08-13	**26.90**	27.51	**48.61**	44.10
2022-08-18	35.19	**32.39**	**34.38**	32.81
2022-08-23	**34.60**	36.48	**35.27**	30.80
2022-08-25	**36.83**	38.78	**31.47**	30.13

J Two-Stage Model Training

For *generalization across time* (Figure 3), all baselines are trained in 2 stages: (1) In **stage 1**, all models are trained on the same pretraining data corpus: SSL models are not trained on any labels, whereas supervised baselines are pretrained with labels; (2) **stage 2** adapts the pretrained model to a target day or participant with supervision via finetuning, which is the same procedure for all baselines. Table 4 summarizes the detailed comparisons.

Table 4: Two-stage training setup used for both the IBL and human chronic datasets, across all baselines, for evaluating generalization across time.

	IBL	**Human**
Stage 1	39 training days, with labels used only for supervised baselines	Pooled T12 and T15 training data, with labels used only for supervised baselines
Stage 2	Backbone frozen; only update the read-in layers required to map new neural populations to the latent space, and the task-specific decoder heads	Full end-to-end finetuning (updating the backbone, read-in, and decoder heads)

K Model and Hyperparameter Details

K.1 Model Capacity Control via Shared Backbone

In our experiments, we controlled backbone capacity across all baselines. Transformer-based methods (e.g., NeuroLens, causal NDT, SPINT, POYO) used a common backbone configuration, matched in depth, width, and number of attention heads (Table 5). Therefore, differences in downstream performance among these methods reflect their training objectives and model-specific design rather than backbone capacity. For non-transformer baselines (e.g., POSSM, CEBRA), for which a shared backbone architecture was not applicable, we matched the total number of backbone parameters to that of the transformer backbone (Table 5). Model backbone hyperparameters were selected independently for the IBL and human datasets.

Table 5: Shared Transformer backbone architecture and backbone size for the IBL and human chronic datasets. Backbone size includes the causal self-attention blocks but excludes model-specific encoders that project spikes to latents before the backbone, decoder heads, and other model-specific components.

Shared Transformer Backbone			**Backbone Parameters**		
Hyperparameter	**IBL**	**Human**	**Model**	**IBL**	**Human**
Layers	6	6	NeuroLens	3.58M	7.29M
Embedding dimension	192	384	Causal NDT	3.28M	7.29M
Attention heads	8	6	POSSM	3.57M	7.27M
FFN dim	1024	768	SPINT	3.55M	—
Activation	GELU	GELU	CEBRA	0.31M	0.44M
Normalization	Pre-LN	Pre-LN	POYO+	3.55M	—
Dropout	0.2	0.2			

K.2 Model-Specific Details

This subsection describes the components that distinguish each model from the shared backbone (Table 5): how neural activity is encoded, the pretraining objective, and the latent representation used for downstream decoder heads.

Linear baselines. We use Ridge regression to decode time-bin-level behavior and logistic regression for trial-level choice, using *scikit-learn* packages, and choose hyperparameters, e.g., the L_2 penalty, based on cross-validation. The same approach is used both for decoding with the raw spike baselines and for linear probing with the learned neural latents (Figure 2). For time-bin-level behavior, the neural activity and behavior are time-bin aligned. Neural activity inputs are standardized on the training split, and the Ridge regression is fit independently per behavioral variable. For the trial-level choice variable, the neural latents are averaged over all time bins and pooled into a single vector for decoding. For the raw spike baseline, the decoder takes in neural activity across all timesteps and concatenates them as input to the logistic regression.

Table 6: Model-specific hyperparameters used for NeuroLens.

Hyperparameter	**IBL**	**Human**
Input representation	Binned spike counts	5-bin time patches
Input dimension	N neurons	1280 (T12); 2560 (T15)
Latent dimension	192	384
Attention heads	8 (predictor) & 1 (encoder)	6
FFN dimension	1024 (predictor) & 256 (encoder)	768
Dropout	0.1	0.1
Day adaptation	Neuron identity embeddings	Per-Day FiLM
SIGReg loss weight	0.6	1.0
Context window size	59 preceding bins	All preceding patches
Backbone # of parameters	3.58M	7.29M

NeuroLens. NeuroLens is our self-supervised framework for learning transferable neural representations across recording sessions. For calibration-based adaptation, we sample 64 calibration

time windows to infer neuron embeddings on the IBL dataset. On human chronic datasets, the number of calibration time windows used vary across days as recording sessions from some days are fewer than 64. For trial-level decoding (choice and sentence embedding), we average the neural latents from the predictor outputs $\widehat{\mathbf{z}}_{1:T}^{(d)}$ across the T time bins. On IBL, a $\mathrm{LayerNorm} \to \mathrm{Dropout} \to \mathrm{Linear}(192, 2)$ head predicts choice with a binary cross-entropy loss; on the human chronic datasets, a $\mathrm{LayerNorm} \to \mathrm{Linear}(384, 384)$ head predicts the sentence embedding with a cosine similarity loss. For time-bin-level variables, predictions are made from $\mathbf{h}_t$ at each time bin: on IBL, a $\mathrm{LayerNorm} \to \mathrm{Linear}(192, 192) \to \mathrm{GELU} \to \mathrm{Dropout} \to \mathrm{Linear}(192, 6)$ head regresses the movement-related behaviors with a mean-squared-error loss; on the human chronic datasets, a $\mathrm{LayerNorm} \to \mathrm{Linear}(384, 41)$ head produces phoneme logits, trained with a CTC loss.

Table 7: Model-specific hyperparameters used for Causal NDT.

Hyperparameter	**IBL**	**Human**
Input representation	Binned spike counts	5-bin time patches
Latent dimension	192	384
# of attention heads	8	6
FFN dimension	1024	768
Dropout	0.2	0.2
Context window size	59 preceding bins	All preceding patches
Backbone # of parameters	3.28M	7.29M

Causal NDT. Causal NDT is a self-supervised causal Transformer baseline built on (Ye & Pandarinath, 2021). A 2-layer MLP read-in maps neural activity at each time step to the latent space, a causal transformer backbone (Table 5) processes the sequence, and a linear output layer maps each hidden state back to observation space to predict the next step's activity. For IBL, inputs are raw spike counts. For multi-day pretraining, each session (day) has its own read-in and output layer during stage-1 training, since each day has a distinct number of neurons. The output layer predicts per-neuron log firing rates, trained to predict the next bin's spike counts under a Poisson negative log-likelihood, with no behavioral labels used. In stage 2, only a newly initialized per-day read-in and the decoding heads are updated, with the rest of the model frozen. For decoding, the decoder heads and losses are the same as other baselines (binary cross-entropy for choice, MSE for time-bin-level behavior). For the human chronic dataset, 5 consecutive 20 ms time bins are concatenated along the channel dimension to form 1 input time patch. The spike data are projected with a 2-layer MLP followed by a learned temporal positional embedding, and we employ separate participant-specific output layers to reconstruct their own spike data, while keeping the Transformer backbone shared. In Stage 1, the training objective is next-step prediction of the following time patch's neural features using an MSE loss; no behavioral labels are required for this stage, and the spike reconstruction output layer is discarded after Stage 1. In Stage 2, the model is fine-tuned end-to-end on a single participant with the task-specific decoding losses, as in Appendix G.

Table 8: Model-specific hyperparameters used for POSSM.

Hyperparameter	**IBL**	**Human**
Input representation	Spike events	5-bin time patch
Latent dimension	256	192
# of latents per time step	1	1
# of cross-attention heads	1	1
Cross-attention head dimension	32	64
GRU depth	2	8
GRU hidden dimension	588	395
GRU backbone # of parameters	3.57M	7.27M
Is current input time step included?	No	No
Context window size	59 preceding states	All preceding states

POSSM. We implement POSSM (Ryoo et al., 2026) as a supervised multi-session baseline with an event-based neural read-in, a unidirectional GRU temporal backbone, and a causal cross-attention decoder. Unlike the shared IBL input representation, which consists of vectors of binned spike counts, POSSM retains individual spike events within each 16.7 ms bin. Each spike is represented by a learned unit embedding, and one learned latent token cross-attends to the spike-event tokens

within each bin. The resulting bin representation is processed by the GRU, after which an output query cross-attends to preceding GRU hidden states to produce the downstream latent.

During multi-day IBL pretraining, each recording day has its own unit and day embeddings, while the input cross-attention, GRU, output cross-attention, and supervised task heads are shared across recording days. The stage-1 training objective is the equally weighted sum of time-bin-level (movement) behavior MSE and choice binary cross entropy losses. Both the time-bin-level movement-related behavior output and the choice decoder use linear projections of the 192-dimensional decoder representation (Table 8).

To align POSSM with the strictly causal setting adopted by NeuroLens, each recurrent chunk contains exactly one 16.7 ms time bin, and the decoder excludes the hidden state corresponding to the current bin. When producing the latent for time bin t, the decoder explicitly attends to time bins up to time bin $t-1$. The unidirectional GRU may carry information from earlier states into this explicit context, but neither the GRU nor the decoder receives neural activity from the current or future bins to prevent data leakage. For an unseen IBL day, the recurrent and cross-attention modules remain frozen during target day adaptation, while the new day-specific unit embeddings and the fresh decoder head are updated during few-shot finetuning.

For BCI, 5 consecutive 20 ms feature vectors are concatenated along the channel dimension to produce one 100 ms input step. T12 and T15 use separate participant-specific 2-layer MLP read-in layers, followed by a shared POSSM input cross-attention, GRU, causal decoder and decoder head. The decoder head attends to all preceding recurrent states while excluding the current 100 ms input step. The phoneme decoding model applies $\mathrm{LayerNorm} \to \mathrm{Linear}(256, 41)$ to every decoded step. The sentence embedding decoding model averages the decoded representations over all time steps and applies $\mathrm{LayerNorm} \to \mathrm{Linear}(256, 384)$. We separately train the model to decode phonemes and sentence embeddings because joint optimization caused substantial interference between the two tasks.

We perform a targeted architectural search rather than a large random hyperparameter sweep. For IBL, we compare explicit decoder contexts of 9 and 59 states and multiple GRU depth, including width near the shared backbone budget. The reported model uses a four-layer GRU with hidden size 588. For T12 and T15, we compare multiple GRU depths while matching the full 7.29-million-parameter transformer backbone budget. The reported model-capacity-aligned model uses 8 GRU layers with hidden size 395. The remaining architectural settings are summarized in Table 8.

Table 9: Model-specific hyperparameters used for SPINT, which is only applied to IBL data.

Hyperparameter	**IBL**
Input representation	Binned spike counts
Latent dimension	256
Cross-attention depth	2
# of attention heads & head dimension	4 & 64
FFN dimension	2944
Activation	ReLU
Normalization	Pre-LN
Attention & FFN & residual dropout	0.1
Dynamic neuron dropout	$p \sim \mathrm{Uniform}(0.1, 0.5)$
# of latent queries	6
# of calibration trials per day	64
Is current input time step included?	No
Context window size	59 preceding bins
Backbone # of parameters	3.55M

SPINT. We implement SPINT (*Spatial Permutation-Invariant Neural Transformer*) (Le et al., 2026) as a supervised multi-session decoding baseline on the IBL chronic dataset, with a calibration-based neural read-in module and spatial cross-attention over recorded units. For each neuron, the model takes its spike count sequence across consecutive time bins and represents that sequence as a single token. A shared identity network encodes 64 unlabeled calibration windows from a session's training span, averages the identity network outputs across windows, and maps them to a per-neuron identity vector. This neuron identity vector is added to each neuron's projected spike count token before the shared MLP read-in. Learned latent queries, one per behavior output, cross-attend to

the resulting neural tokens to produce fixed-dimensional query latents. This read-in module supports variable-sized populations without a learnable neuron-identity lookup table. Dynamic neuron dropout is also employed to improve robustness to changes in the recorded population.

During multi-day pretraining, all model parameters are shared across days, while a calibration set is sampled from each day. For IBL, the stage-1 training objective is an equally weighted sum of mean-squared error for time-bin-level, movement-related behavior and binary cross-entropy for choice. The 5 time-bin-level behaviors comprise 6 scalar outputs because left-paw motion has 2 dimensions. A shared linear decoder head maps each query latent to a behavior sequence. For choice decoding, query latents are averaged across time steps within each trial and passed through a 2-layer MLP. Choice supervision is applied only to trials containing a stimulus onset and a valid choice label; all other time windows contribute only to time-bin-level behavior decoding. To align SPINT with the strictly causal decoding setup, for a window ending at time bin t, the prediction uses neural activity only up to time $t-1$, together with a fixed calibration set.

For an unseen day, neuron embeddings are inferred from calibration sets sampled from its adaptation split. The cross-attention backbone and learned queries remain frozen, whereas the identity encoder, shared spike read-in, and task-specific decoder heads are adapted. For time-bin-level behavior and choice decoding, models are initialized from the same pretrained checkpoint but fine-tuned separately for their respective targets. For trial-level choice decoding, query latents are averaged across all time steps within a trial to obtain a per-trial vector. The model hyperparameters are summarized in Table 9. SPINT is evaluated only on the IBL dataset.

Table 10: Model-specific settings used for CEBRA. Parameter counts refer to the encoder, which serves as the backbone for this model.

Hyperparameter	**IBL**	**Human**
Input representation	Binned spike counts	5-bin time patch
Encoder	`offset10-model`	MLP read-in + 3 conv blocks
Embedding dimension	192	192
Conv block latent dimension	128	192
Receptive field	10 time bins	10 time patches
Positional embedding	None	None
Distance & temperature	Cosine & auto (floor at 0.1)	Cosine & fixed $\tau = 1$
Pretraining batch size	512	64
Finetuning batch size	16	64
Finetuning learning rate	1×10^{-3}	5×10^{-4}
Is current input time step included?	No	No
Backbone # of parameters	0.31M	0.44M

CEBRA. CEBRA (Schneider et al., 2023) is a self-supervised contrastive learning framework that learns embeddings preserving temporal proximity, trained via an InfoNCE objective, wherein each anchor is paired with a positive sample from the same trial at a small temporal offset and contrasted against negatives sampled from the remaining anchors in the batch using cosine similarity, without any behavioral labels.

CEBRA is not transformer-based, so it does not use the shared backbone of Table 5; its encoder is a convolutional network with a 10-step receptive field and no positional embedding. We adopt the architecture provided by the authors (Schneider et al., 2023) rather than widening it to match the transformer parameter count. On IBL data, we use the provided model config `offset10-model` (128 hidden units, 192 latent dimensions). On T12 and T15, we employ the same 2-layer MLP read-in as for the other baselines, followed by 3 causal convolutions (kernel 4, receptive field 10 patched steps, width 192), so that it shares the participant-specific read-in and the decoder heads with the other baselines. Because the encoder has no positional embedding and no cross-time mixing beyond its receptive field, its per-time-bin embedding is window-invariant, which the other models' positional embeddings do not permit.

The provided encoder centers its receptive field, so the embedding labelling time bin t would read time bins $[t-5, t+5)$ (4 bins into the future). We shift the receptive field to end 1 bin early, so that we use neural activity only up to $t-1$, to prevent data leakage and match the strictly causal setup under which the other models are scored. Downstream, the per-time-bin embedding is the latent: time-bin-level behaviors are decoded per time bin, and the trial-level target is decoded from

the embedding averaged over the time bins within a trial, then classified with a linear decoder. On an unseen day, the pretrained backbone is frozen, and that day's own read-in encoder is trained together with a new read-in layer to accommodate the new data distribution, which aligns with CEBRA's own few-shot update setting (Schneider et al., 2023).

For few-shot finetuning adaptation, we searched the learning rate (1×10^{-3}, 3×10^{-4}, 1×10^{-4}) and batch size (16, 64) on 3 held-out sessions spanning the range of population sizes; during pretraining, we searched the embedding dimension (32, 192). In both cases, we selected the best checkpoint based on validation decoding performance. The other settings are summarized in Table 10.

Table 11: Model-specific settings used for POYO.

Hyperparameter	**IBL**
Input representation	Spike events
Latent dimension	192
# of latents per time step	2
Latent grid step	1 bin (16.7 ms)
# of cross-attention heads & head dimension	2 & 32
Time positional embedding	RoPE
Decoder head	2-layer MLP (width 192, dropout 0.2)
Unit dropout	0.1
Finetuning learning rate & weight decay	3×10^{-4} & 0.05
Finetuning batch size	64
Is current input time step included?	No
Context window size	59 preceding bins
Backbone # of parameters	3.55M

POYO. POYO (Azabou et al., 2023) is a supervised decoder trained to decode behavioral targets. Unlike models based on binned spike counts, it tokenizes individual spikes, each with a per-neuron embedding and a rotary time embedding (RoPE); since timestamps enter the tokenizer for spikes rather than via learnable time embeddings, the backbone has no positional embedding, so its Table 5 count reflects only the self-attention blocks. A cross-attention read-in compresses spikes within each 16.7 ms step into latent tokens, the shared causal Transformer backbone processes them, and a cross-attention read-out queries the result at the output times before per-target MLP decoder heads. The objective equally weights behavior MSE and choice binary cross entropy.

Attention is masked on three levels so that no model output depends on neural activity after the time bin it is queried at (no temporal leakage): (1) the cross-attention read-in latent attends only to spikes preceding the end of its own time step, (2) latents attend only to earlier latents, (3) and the cross-attention read-out query attends only to latents whose time step has ended. For an unseen day, the read-in, backbone, and read-out are frozen, and only that session's new unit and day embedding rows are updated together with a fresh decoder head, following POYO's own few-shot recipe (Azabou et al., 2023), as also done for POSSM (Ryoo et al., 2026). For hyperparameter tuning, we searched the finetuning learning rate, batch size, window stride, unit dropout ratio, and different initializations for new rows in the neuron embedding lookup table, selecting the best checkpoint based on validation decoding performance. The other settings are summarized in Table 11. POYO is evaluated on IBL data only.

L Training Details

All baselines are trained with the AdamW optimizer (Loshchilov & Hutter, 2017) and employ the same training setup and budgets. The checkpoint selection approach, based on validation metrics, is also the same across all baselines to ensure fair comparisons. We describe training details for each baseline below.

NeuroLens. We train both the single-day and multi-day NeuroLens using a learning rate of 1×10^{-4} and weight decay 0.05. For IBL, the 39-day model is trained on a single Nvidia A40 GPU in 21.1 hours for a total of 1000 epochs. Single-day models can be trained on a single Nvidia A40 or A100 GPU in $1.0 \sim 2.7$ hours, also for 1000 epochs. Stage-2 fine-tuning (behavior and choice, 300 epochs each) requires $1.1 \sim 4.0$ hours per held-out session on one A40 or A100 GPU. For T12 and T15 datasets, NeuroLens is trained under all experimental settings on a single Nvidia A100 GPU

(40 GB) for a total of 1000 epochs: approximately 20 hours for the high- vs. low-level decoding setting and 16 hours for the generalization-across-time setting. Fine-tuning can be performed on a single A100 or A40 GPU for 500 epochs, requiring approximately $2 \sim 3$ hours per run for T12 and $6 \sim 9$ hours for T15. For the ablation experiments, we use one A100 or A40 GPU and complete each run in $2 \sim 9$ hours.

Causal NDT. We train both the single-day and multi-day Causal NDT using the same optimizer settings as NeuroLens. For IBL, the 39-day model is trained on a single Nvidia A100 GPU in 8.7 hours for a total of 1000 epochs. Single-day models can be trained on a single Nvidia A40 or A100 GPU in $0.5 \sim 1.4$ hours, also for 1000 epochs. Stage-2 fine-tuning (behavior and choice, 300 epochs each) requires $1.2 \sim 3.8$ hours per held-out session on one A40 or A100 GPU. For T12 and T15 datasets, causal NDT is trained under all experimental settings on a single Nvidia A100 GPU (40 GB) for a total of 1000 epochs: approximately 16 hours for the high- vs. low-level decoding setting and 14 hours for the generalization-across-time setting. Fine-tuning can be performed on a single A100 or A40 GPU for 500 epochs, requiring approximately $2.4 \sim 3$ hours per run for T12 and $5.5 \sim 8$ hours for T15; fine-tuning repeats over random seeds fall in the same range ($2.4 \sim 2.6$ hours for T12, $5.5 \sim 6.2$ hours for T15).

POSSM. We train both the single-day and multi-day POSSM using the AdamW optimizer. On IBL, the 39-day model is pretrained on a 96GB H100 GPU in $\sim$9.8 hours for a total of 1000 epochs with early stopping, using batch size 64, peak learning rate 0.001, weight decay 0.05, unit dropout 0.15, gradient clipping at 1, and 15% warmup for the learning rate scheduler. Stage-2 finetuning is run on a single GPU in $\sim$1 hour using two sub-stages: (1) unit identification (learning rate 3×10^{-4}, at most 100 epochs) followed by (2) read-in and readout fine-tuning (learning rate 1×10^{-5}, at most 400 epochs), both with batch size 16 and weight decay 0.1. On T12 and T15 data, POSSM is trained on a single H100 GPU in $\sim$2.5 hours for a total of 500 epochs, using AdamW with batch size 32, peak learning rate 5×10^{-4}, minimum learning rate 1×10^{-5}, weight decay 10^{-3}, cosine decay with 2% warmup, and gradient clipping at 10.

SPINT. On IBL data, SPINT is pretrained for 1000 epochs on the 39 pooled days. Training uses AdamW optimizer with a OneCycle learning-rate scheduler, with each batch drawn from a single day. The Stage-1 checkpoint maximizes the mean per-behavior validation R^2, computed from predictions pooled across the pretraining days' validation set. Choice cross-entropy contributes to pretraining but does not enter this model selection criterion. For finetuning, we employ training batches of 16 trials. The reported choice decoding runs use AdamW optimizer with peak learning rate 1×10^{-4}, weight decay 0.05, a OneCycle schedule with 15% warmup, and gradient clipping at 1.0, for a fixed budget of 500 epochs. Training runs for the full epoch budget without early stopping.

CEBRA. We train CEBRA using its own solver for Stage 1 and AdamW for Stage 2. On IBL, Stage 1 is trained on one A40 or A100 GPU in $\sim$1 hour for a total of 12,000 iterations, on the 39 pooled days, with batch size 512 and learning rate 3×10^{-4}. Stage 2 is trained on a single GPU in 2$\sim$14 minutes per test day per behavior target, attaching a decoder head to the frozen pretrained encoder and training it together with the held-out day's own read-in encoder, using learning rate 1×10^{-3}, weight decay 0.05, batch size 16, with early stopping at up to 1000 epochs. Time-bin-level behavior and choice decoding are independent runs from the same Stage-1 checkpoint, selected on validation mean R^2 and validation balanced accuracy respectively. On T12 ad T15 datasets, CEBRA follows the shared two-stage training protocol in Table 4. Stage 1 is trained on one A40 or A100 GPU in $\sim$7.3 hours for a total of 1000 epochs, pooling T12 and T15 with learning rate 5×10^{-4}, weight decay 1×10^{-5}, cosine schedule without warmup, batch size 64, using a time-contrastive loss with a fixed temperature of 1. Stage 2 is trained on a single GPU for a total of 1000 epochs, once per participant and behavioral variable, fine-tuning end to end and taking $\sim$3.7 hours per run for T12 and 7 hours for T15.

POYO. We train POYO using the AdamW optimizer. On IBL data, Stage 1 is trained on one A40 or A100 GPU in $2.8 \sim 5.2$ hours across runs, for 1000 epochs with early stopping on the 39 pooled days, using learning rate 3×10^{-4}, weight decay 0.05, batch size 64, gradient clipping at max-norm 1.0, and a cosine schedule with 5% warmup. The training objective equally weights time-bin-level behavior MSE and choice binary cross-entropy. Stage 2 is trained on a single GPU in $\sim$25 seconds to 5 minutes per test day per behavior target; finetuning runs are short since they early-stop well within the epoch budget. We freeze the read-in, backbone, and read-out, training only the held-out

day's new unit and session embedding rows with a new decoder head, using learning rate 3×10^{-4}, weight decay 0.05, and batch size 16, for at most 500 epochs. Time-bin-level and choice decoding are independent runs, selected by validation mean R^2 and balanced accuracy, respectively.