PRETRAINED NEURON-LEVEL REPRESENTATIONS ENABLE ACROSS-SUBJECT ZERO-SHOT TRANSFER OF NEURAL DECODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretraining neural decoders across electrophysiology recordings offers a path toward models that generalize to unseen subjects, yet existing approaches typically require adaptation to each new neural population. A key obstacle is that spike-sorted recordings lack a shared input space. Each recording contains a new, unordered population of neurons, which existing decoders typically represent using recording-specific parameters. We introduce ZEN, a framework that replaces these parameters with transferable neuron representations inferred by a pretrained neuron-level encoder. A decoder trained over this space then operates directly on the new population, without target-specific gradient updates, alignment, or behavioral labels. We evaluate three neuron-level encoders and three decoder architectures, including our proposed Time-Patch decoder. Time-Patch preserves neuron-specific tokens throughout the Transformer backbone, whereas prior architectures aggregate activity across neurons before the Transformer backbone. Our strongest configuration, ZENITH, combines NuCLR representations with Time-Patch and achieves state-of-the-art cross-subject zero-shot decoding on the IBL Brain-Wide Map and Allen Visual Behavior Neuropixels datasets, reaching approximately 83% and 85% balanced accuracy on reward decoding, respectively. Under the finetuning setting, ZENITH surpasses the best fully finetuned baselines on IBL wheel speed and reward decoding with only 25% of the available target finetuning trials. These results establish transferable neuron representations as a practical basis for both zero-shot neural decoding and data-efficient supervised adaptation.

1 INTRODUCTION

Neural decoding predicts behavior or stimuli from population neural activity, providing a widely used tool in systems neuroscience for probing the information carried by neural populations (Quian Quiroga & Panzeri, 2009; Kriegeskorte & Douglas, 2019). Spike-sorted extracellular electrophysiology is especially well suited to this setting because it resolves the activity of individual neurons at sub-millisecond resolution, allowing neural activity to be modeled and decoded at single-cell resolution (Harris et al., 2016). Recent advances in high-density recording and spike sorting now make it possible to record hundreds or thousands of neurons simultaneously (Jun et al., 2017; Pachitariu et al., 2016). This increase in recording scale, together with the growing availability of large, multi-animal electrophysiology datasets (Steinmetz et al., 2019; Siegle et al., 2021; IBL et al., 2025a; Bennett et al., 2026), has created the opportunity to pretrain neural decoders across recordings and subjects with the goal of learning structure shared across neural populations rather than training a separate model for every experiment (Ye et al., 2023; Azabou et al., 2023; Zhang et al., 2024; Ye et al., 2025a).

Despite this progress, pretraining across recordings faces a challenge specific to spike-sorted data: recordings do not share a common input space, limiting their ability to serve as broadly reusable models across experiments, probe implants, and subjects (Dabagia et al., 2023). In modalities where a direct *zero-shot transfer*[1] across subjects/recordings has become possible, inputs are typically expressed in a representation that remains consistent across examples, datasets, or subjects. In

[1] By zero-shot transfer, we mean an application that does not rely on target supervision and gradient-based model updates. We precisely define our flavor of zero-shot in Section 3.

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.

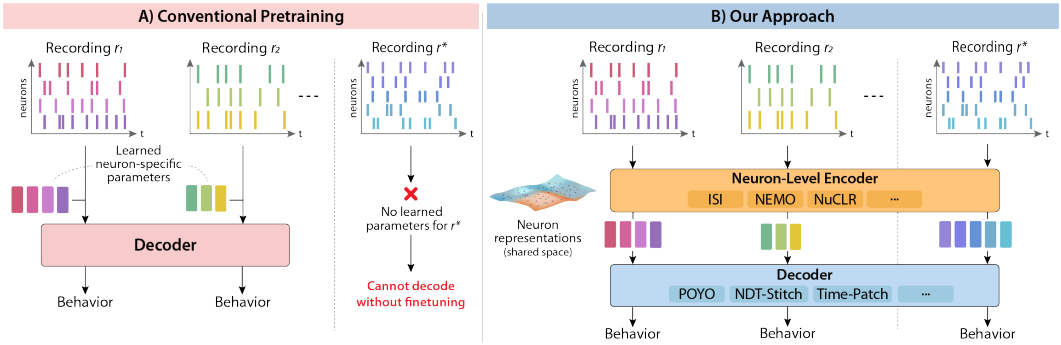

Figure 1: **Overview of our approach. (A)** Conventional neural decoder pretraining architecturally requires learning per-neuron parameters, which prevents them from being applied zero-shot to an unseen neural population without finetuning. **(B)** Our approach instead uses a Neuron-Level Encoder to infer neuron representations from activity itself, and trains a decoder on this space to generalize to unseen populations.

EEG and fMRI, shared spatial reference systems provide a basis for learning across recordings and subjects: EEG models can associate channels with standardized electrode locations (Jiang et al., 2024), while fMRI models can represent brain regions using coordinates in a common anatomical space (Ortega Caro et al., 2024). However, no analogous correspondence exists for spike-sorted electrophysiology. Each recording contains a new, unordered population of neurons, or "units", whose size and identities can differ substantially across recordings, so unit i in one session has no meaningful correspondence to unit i in another. As a result, the effective input vocabulary of a neural decoder changes with every recording.

State-of-the-art multi-recording models typically accommodate these changing neural populations by introducing recording-specific parameters, often as learned embeddings for individual units (like unit embeddings in Azabou et al. (2023)) or recording-specific projections (like stitching matrices in Pandarinath et al. (2018); Zhang et al. (2024)). These parameters allow different recordings to share a common decoder backbone, but the representation of each population remains tied to the recordings seen during training. As more recordings are incorporated, the model accumulates an ever-expanding set of these recording-specific representations. A previously unseen recording therefore introduces units for which no corresponding parameters exist in the pretrained model, preventing direct decoding without first learning new recording-specific parameters.

We hypothesize that, rather than requiring recording-specific parameters to be learned, the measured activity of neurons itself contains sufficient information to support a universal vocabulary upon which neural decoders can be trained. Recent work on neuron-level representation learning suggests that neural measurements can indeed provide the basis for such a universal vocabulary. NuCLR (Arora et al., 2025) and NEMO (Yu et al., 2025) show that representations derived from neural population activity and spike waveforms can capture properties related to cell type, anatomical location, and functional characteristics, while generalizing across recordings and subjects.

We make a key connection between neuron-level representation learning and neural decoder pretraining. In our framework, ZEN[2], a shared Neuron-Level Encoder (NLE) computes a representation for each neuron from activity, and a decoder is pretrained over this inferred representation space. On a new recording, the NLE provides representations for the unseen neurons from the subject's own recording, allowing the application of the pretrained decoder-backbone directly, without finetuning, alignment, or behavioral labels.

We evaluate our approach with a wide range of well-established NLEs and decoder formulations, and also introduce a new decoder tokenization scheme more suitable in the zero-shot setting. We evaluate ZEN on two large-scale electrophysiology datasets, the International Brain Laboratory Brain Wide Map (IBL et al., 2025a) and Allen Visual Behavior (Bennett et al., 2026), testing zero-shot transfer to completely unseen subjects. Our best-performing model configuration, which we call ZENITH, achieves the best zero-shot performance without target-specific finetuning or alignment, achieving 83% balanced accuracy for reward decoding on IBL BWM and 85% on Allen VBN. ZENITH also exhibits state-of-the-art data efficiency under the standard supervised finetuning regime: with only

[2]ZEN is an abbreviation for "**Ze**ro-shot on **n**ew neurons."

25% labels in the few-shot finetuning, ZENITH outperforms full-finetuning performance of most baselines.

Our core contributions are:

- We introduce **ZEN**, a framework that connects neuron representation-learning with neural decoder pretraining by replacing recording-specific parameters with inferred neuron representations. This enables zero-shot decoding from unseen spike-sorted neural populations without target behavioral labels, alignment, or parameter updates.
- We systematically compare three neuron-level encoders and three decoder architectures, including our proposed **Time-Patch** decoder, which preserves neuron-specific tokens throughout the backbone. Combining Time-Patch with NuCLR yields **ZENITH**, our strongest configuration, which outperforms other zero-shot baselines on two large-scale electrophysiology dataset.
- We show that ZENITH also supports data-efficient supervised adaptation: using only 25% of the available target finetuning trials, it surpasses the best fully finetuned baselines on IBL wheel speed and reward decoding, while matching or exceeding baseline performance across all three IBL tasks with full finetuning.

2 THE NEURON-IDENTITY PROBLEM

Transferring a decoder across recordings requires a shared basis for interpreting its inputs. EEG electrodes have standardized scalp locations (Jiang et al., 2024), and fMRI voxels can be registered to a common anatomical template (Ortega Caro et al., 2024). However, spike-sorted electrophysiology has no analogous shared input space. Each probe insertion produces a new population of neurons whose size, ordering, and identities differ across recordings. Consequently, unit n in one recording has no meaningful correspondence to unit n in another.

This creates the ***neuron-identity problem***: a decoder must represent not only *when* a neuron fired, but also *which neuron* produced that activity. Current multi-recording decoders typically establish this identity through recording-specific parameters. These appear explicitly as learned unit embeddings in spike-based models such as POYO (Azabou et al., 2023), or implicitly through recording-specific input projections in population-stitching methods (Pandarinath et al., 2018; Zhang et al., 2024). Such parameters allow a shared decoder to distinguish and interpret neurons encountered during training. However, a new recording introduces neurons for which no corresponding parameters have been learned, thus requiring finetuning. Even if the decoder has learned relationships between neural activity and behavior across many subjects, applying those relationships to a new population still requires establishing how its neurons should be represented.

We therefore ask whether the representation needed to establish neuron identity can instead be inferred directly from neural observations, allowing a pretrained decoder to interpret neurons from an unseen subject without learning new recording-specific parameters. This enables decoding that is *gradient-free*, i.e. requiring no parameter updates on the target recording, and *zero-shot*, i.e. requiring no target behavioral labels.

3 METHOD

We study cross-recording neural decoding in the setting where each recording is acquired with a different set of probe insertions and therefore samples a new population of neurons, whether across subjects or across *acute* probes from the same subject. For recording r, let $\mathcal{X}^{(r)}$ denote the activity of its N_r recorded neurons and $Y^{(r)}$ the corresponding behavioral variables. Because the probe and recorded population change across recordings, the number of neurons and their identities do not correspond between them.

Across-subject, gradient-free, zero-shot decoding. Let $\mathcal{R}_{\text{train}}$ denote the training recordings and let $r^* \notin \mathcal{R}_{\text{train}}$ be a previously unseen target recording. We define *zero-shot decoding* as predicting behavior from r^* without using behavioral labels from that recording. The model f may use neural activity from r^*, including to infer neuron representations. We call an evaluation *gradient-free* when no model parameters are updated on the target recording. Thus,

$$\hat{Y}^{(r^*)} = f\left(\mathcal{X}^{(r^*)}; \theta_{\text{train}}\right), \tag{1}$$

where θ_{train} remains fixed after training. In particular, no neuron-level parameters, alignment mappings, input projections, or decoder parameters are optimized on r^*.

3.1 ZEN: ZERO-SHOT DECODING THROUGH PRETRAINED NEURON EMBEDDINGS

Our central hypothesis is that neuron representations useful for decoding can be inferred from neural activity rather than learned as recording-specific parameters. Neurons with similar functional properties may exhibit recurring patterns of activity and population interactions across recordings, providing a shared basis for representing unseen neurons.

This motivates a modular architecture that separates neuron representation from behavioral decoding. We formalize ZEN as consisting of two components: a *Neuron-Level Encoder* (NLE), which infers one representation per recorded neuron from population activity, and a *decoder*, which combines trial activity with these representations to predict behavior.

Formally, the NLE maps the recorded population activity to a set of per-neuron embeddings:

$$E^{(r)} = \left[\mathbf{e}_1^{(r)}, \ldots, \mathbf{e}_{N_r}^{(r)}\right] = f_{\text{NLE}}\left(\mathcal{X}_1^{(r)}, \ldots, \mathcal{X}_{N_r}^{(r)}; \theta_{\text{NLE}}\right), \tag{2}$$

where $\mathcal{X}_n^{(r)}$ is the activity of unit n in recording r, and $\mathbf{e}_n$ is its produced representation. The encoder f_{NLE} may be a pretrained model with parameters θ_{NLE} shared across recordings, or a fixed feature extractor with no learned parameters.

The decoder is pretrained to operate on top of NLE's representation space. It receives the activity over the interval to be decoded together with the representation of each neuron that generated it:

$$\hat{\mathbf{y}} = f_{\text{dec}}\left(\left[(\mathbf{x}_n, \mathbf{e}_n^{(r)})\right]_{n=1}^{N_r}; \theta_{\text{dec}}\right), \tag{3}$$

where $\mathbf{x}_n$ denotes the activity of neuron n over the decoding interval and N is the number of recorded neurons. The embeddings provide neuron-level context through which the decoder learns to interpret activity across populations with different sizes and compositions.

At inference, we apply the frozen NLE to neural observations from an unseen subject and pass the resulting embeddings, together with the activity to be decoded, to the pretrained decoder. Both sets of model parameters ($\theta_{\text{NLE}}, \theta_{\text{dec}}$) remain fixed. The target recording supplies neural observations from which neuron representations are computed, but no behavioral labels or fitting steps are required. Zero-shot performance thus indicates whether the shared representation space supports generalization of the learned decoding relationship to a previously unseen population.

This formulation lets us examine both what information is useful for representing unseen neurons and how decoders should incorporate those representations. Below, we describe the neuron-level encoders and decoder architectures used to explore these two questions.

Neuron-Level Encoders. Our framework accommodates a wide range of NLEs and several existing models fit our formulation (Eq. 2). In this work, we evaluate two pretrained approaches that characterize neurons using different sources of information:

- **NEMO** (Yu et al., 2025) is a pretrained model that combines extracellular waveforms with univariate spiking statistics. This method captures both spike shape and individual firing characteristics.
- **NuCLR** (Arora et al., 2025) is a pretrained model that represents each neuron's spiking activity in the context of the simultaneously recorded population, allowing its representation to reflect relationships with other neurons.

We also evaluate fixed **ISI** features that represent each neuron by its inter-spike-interval histogram, capturing univariate spiking statistics. Regardless of how they are obtained, the neuron representations serve the same role of contextualizing each neuron's activity for a shared decoder. We next describe how different decoder architectures combine these two inputs to predict behavior.

Neural Decoders. Decoder architectures differ in how they combine neural activity with neuron-level representations to predict behavior. A range of existing neural population decoding models can be adapted to this setting (Ye et al., 2023; Ryoo et al., 2025). We consider two established

Transformer-based decoders whose architectural components have also been adopted in subsequent neural population models:

- **POYO** (Azabou et al., 2023) operates on raw spike times. Each spike is represented by a neuron identity embedding (which we replace with the NLE representations) and spike timing. POYO uses cross-attention as its first layer, attending from a fixed set of learnable latent tokens over the spike tokens followed by multiple layers of attention to produce the behavior prediction.
- **NDT-Stitch** (Ye & Pandarinath, 2021) is a variant of the Neural Data Transformer that incorporates neural-stitching (Pandarinath et al., 2018). Each neural population is assigned a stitching matrix to project the population into a shared latent space over which a transformer backbone can operate. We replace these stitching matrices with our NLE representations.

We provide further details of both decoders and their integration with NLE representations in Appendix C.

Both POYO and NDT-Stitch introduce a population-level bottleneck by aggregating activity across neurons into shared latent representations before the Transformer backbone. As an alternative, we introduce **Time-Patch**, which preserves separate neuron-level tokens at the input to the backbone, as described in Section 3.2. In our experiments, Time-Patch achieves stronger zero-shot decoding performance than these alternatives.

3.2 Time-Patch Decoder

Our Time-Patch decoder constructs separate tokens for each neuron by combining temporal patches of its activity with its NLE representation. By preserving neuron-level tokens, Time-Patch allows the backbone to further refine each neuron's representation through interactions with other neurons and across time. This flexibility may be particularly useful in the gradient-free zero-shot setting, where neuron embeddings cannot be optimized for the target population. In contrast, models such as POYO learn recording-specific embeddings jointly with the decoder, allowing those embeddings to adapt to the population-level aggregation at its input.

Let $X \in \mathbb{R}^{N \times T}$ denote binned spiking activity from N neurons over T time bins. We divide each neuron's activity into P non-overlapping patches of $B = T/P$ bins, with $\mathbf{x}_{np} \in \mathbb{R}^B$ denoting the patch for neuron n at temporal position p. Each token is constructed as:

$$\mathbf{z}_{np} = W_{\text{patch}} \mathbf{x}_{np} + f_{\text{MLP}} \left(\mathbf{e}_n^{(r)} \right), \tag{4}$$

where $W_{\text{patch}} \in \mathbb{R}^{d_{\text{model}} \times B}$ projects the activity patch into the decoder's latent space, and f_{MLP} maps the NLE representation into the same space. Both mappings are shared across neurons and recordings. This produces $N \times P$ tokens, each combining a local spiking pattern with information about the neuron that generated it.

We pass these tokens to the same transformer backbone architecture used for the other decoder variants. Self-attention blocks model interactions across neurons and temporal patches, and the backbone retains a separate token for each neuron and temporal patch while allowing information to flow between tokens. Behavioral queries at the desired prediction timesteps then attend to the resulting neural tokens through cross-attention blocks, and the resulting behavior latents are linearly projected into the target space. Please see Section C for further implementation details.

3.3 Pre-Training Procedure

We follow a two-step pretraining procedure. First, since we use SSL-based NLEs (NEMO and NuCLR), we pretrain them independently using their own loss and training procedures. Then, we freeze the NLEs and pretrain the decoders on the supervised neural decoding task. This second step uses the neuron representations produced by the frozen NLEs. In the case of multiple decoding tasks, we use the same NLE weights. We also use two regularization techniques during decoder training: an exponential moving average (EMA) of the decoder and random unit dropout, and provide further implementation details in Section C.4.

4 RESULTS

We evaluate ZEN on two large-scale electrophysiology datasets. First, we consider the standard zero-shot setting and evaluate whether a pretrained behavioral decoder can transfer to previously unseen subjects without target behavioral labels or parameter updates. We use **ZENITH**, our strongest configuration combining NuCLR with the Time-Patch decoder, as our main comparison point against external baselines. We then examine how the choice of neuron-level encoder and decoder architecture affects zero-shot transfer. Next, we study how zero-shot performance scales with the amount of source data. Finally, we allow target-session supervision and evaluate adaptation from the few-shot regime to standard full finetuning, where we compare against state-of-the-art pretrained decoders.

4.1 DATASETS AND EVALUATION SETUP

We evaluate on the IBL Brain-Wide Map (BWM) and Allen Visual Behavior Neuropixels (VBN) datasets. IBL BWM (IBL et al., 2025a) contains recordings from 119 mice across 357 sessions and 540 probe insertions. Mice perform a visual decision-making task by rotating a wheel to recenter a stimulus. Following prior work (Zhang et al., 2025a; 2024), we evaluate whisker motion energy, wheel speed, and reward decoding.

Allen VBN (Bennett et al., 2026) contains recordings from 54 mice across 103 sessions and 609 probe insertions. We use active task blocks, during which mice report changes in a sequence of natural images by licking for reward. We evaluate running speed and reward decoding from 0.8s trials.

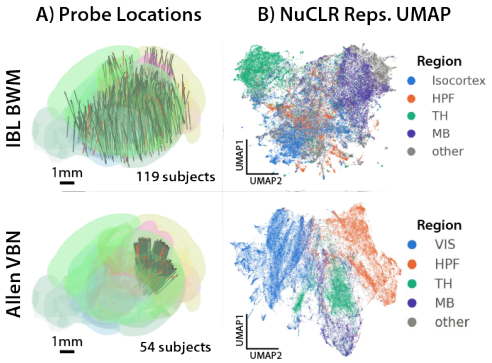

Figure 2: **(A)** A visualization of Neuropixel probe locations (gray lines) in the IBL BWM and Allen VBN datasets. **(B)** UMAP visualizations of NuCLR's representations colored by brain region.

The two datasets provide complementary anatomical coverage (Fig. 2A). IBL BWM samples neurons across widely distributed brain regions, whereas Allen VBN concentrates recordings in visual cortex and selected subcortical regions, including visual thalamus, hippocampus, and midbrain. Together, they allow us to evaluate decoding across distinct behavioral tasks and patterns of anatomical sampling. Additional details are provided in Section A.2.

Evaluation protocol. In both datasets, probes are inserted acutely for each session, so every session samples a new, unaligned neural population. Session-level and subject-level splits therefore pose the same neuron-identity problem (Section 2). We split by subject, as it is the stricter of the two settings and additionally holds out subject-specific variables such as individual anatomy and behavioral strategy. We partition subjects into disjoint training, validation, and test sets. IBL BWM uses 94 training, 12 validation, and 13 test subjects, while Allen VBN uses 38, 8, and 8, respectively. All reported results are obtained on held-out test subjects, and all methods are evaluated on the same held-out test trials. Zero-shot evaluation is gradient-free and uses neural activity from the test recordings to infer neuron representations at inference time, without access to target behavioral labels. Further evaluation details are provided in Appendix B.

Baselines (Gradient-Free Zero-Shot). Most pretrained neural decoders require target-specific finetuning, leaving relatively few direct comparisons for gradient-free, zero-shot transfer to unseen subjects. We therefore construct several reference baselines that operate without learned identities for target neurons. Population-level baselines apply a Linear model or GRU to the population-mean firing-rate time series. *Unsorted POYO* applies the POYO architecture to unsorted threshold-crossing events rather than spike-sorted units. We also compare against SPINT (Le et al., 2025), a supervised pretrained model originally designed for cross-session transfer of long-term implants. Further details on these baselines are provided in Appendix D.

Baselines (Finetuning Setting). For the finetuning setting, we compare ZENITH against recent state-of-the-art neural decoder pretraining approaches including POYO (Azabou et al., 2023), NDT-Stitch (Ye & Pandarinath, 2021; Zhang et al., 2024), POSSM (Ryoo et al., 2025), and MTM (Zhang et al., 2024). We also compare with a Time-Patch decoder variant that uses learned neuron identity embeddings similar to POYO. Further implementation and finetuning details are provided in Appendix E.

Table 1: **Comparison with baselines under gradient-free zero-shot evaluation.** ZENITH is our best configuration, using NUCLR as NLE with the Time-Patch decoder. All rows are evaluated zero-shot on held-out subjects. Results are mean ± SEM over 3 seeds.

Method	IBL BWM			Allen VBN	
	Whisker ME R^2	Wheel Speed R^2	Reward (c=2) Bal. Acc.	Running Speed R^2	Reward (c=2) Bal. Acc.
Linear / pop. avg.	-0.029 ± 0.000	0.156 ± 0.001	0.502 ± 0.003	0.051 ± 0.000	0.531 ± 0.019
GRU / pop. avg.	-0.008 ± 0.001	0.162 ± 0.002	0.507 ± 0.003	0.077 ± 0.033	0.616 ± 0.046
POYO / unsorted	-0.054 ± 0.015	0.143 ± 0.003	0.514 ± 0.003	-0.600 ± 0.034	0.631 ± 0.006
Set Transformer	0.024 ± 0.001	0.208 ± 0.000	0.656 ± 0.003	-0.270 ± 0.024	0.612 ± 0.030
SPINT	0.050 ± 0.008	0.223 ± 0.001	0.708 ± 0.007	0.311 ± 0.022	0.792 ± 0.012
ZENITH (Ours, NuCLR + Time-Patch)	**0.163** ± 0.004	**0.262** ± 0.002	**0.828** ± 0.006	**0.611** ± 0.013	**0.851** ± 0.008

Table 2: **Performance of ZEN with various combinations of neuron representations and decoder architectures.** Decoders are pretrained on one set of subjects and evaluated on held-out subjects, with no supervised calibration on the target sessions. Results are mean ± SEM over 3 seeds. **Top-1** and top-2 results are highlighted.

NLE	Decoder Architecture	IBL BWM			Allen VBN	
		Whisker ME R^2	Wheel Speed R^2	Reward (c=2) Bal. Acc.	Running Speed R^2	Reward (c=2) Bal. Acc.
ISI Single-neuron spiking	NDT-Stitch	-0.020 ± 0.002	0.171 ± 0.001	0.591 ± 0.003	-0.292 ± 0.026	0.700 ± 0.004
	POYO	-0.034 ± 0.002	0.158 ± 0.003	0.621 ± 0.002	-0.329 ± 0.010	0.774 ± 0.007
	Time-Patch	0.013 ± 0.002	0.180 ± 0.002	0.693 ± 0.005	0.272 ± 0.014	0.816 ± 0.001
NEMO Single-neuron spiking and waveform	NDT-Stitch	0.012 ± 0.005	0.155 ± 0.003	0.619 ± 0.011	0.132 ± 0.014	0.750 ± 0.005
	POYO	-0.003 ± 0.004	0.146 ± 0.009	0.607 ± 0.004	0.200 ± 0.019	0.747 ± 0.006
	Time-Patch	0.090 ± 0.008	0.198 ± 0.005	0.702 ± 0.009	0.317 ± 0.012	0.790 ± 0.005
NUCLR Population context	NDT-Stitch	0.124 ± 0.005	0.240 ± 0.003	0.812 ± 0.007	0.440 ± 0.053	0.815 ± 0.014
	POYO	0.161 ± 0.003	0.244 ± 0.005	0.801 ± 0.006	**0.616** ± 0.016	0.825 ± 0.007
	Time-Patch (ZENITH)	**0.163** ± 0.004	**0.262** ± 0.002	**0.828** ± 0.006	0.611 ± 0.013	**0.851** ± 0.008

4.2 ZEN GENERALIZES TO PREVIOUSLY UNSEEN SUBJECTS

We first evaluate whether ZEN can generalize to previously unseen subjects without target behavioral labels and parameter updates. We use ZENITH, our strongest configuration based on the analysis in Section 4.3, for comparison against external zero-shot baselines on both datasets (Table 1).

As shown in Table 1, ZENITH achieves the strongest zero-shot performance across all five tasks. Compared with SPINT, the best baseline, ZENITH shows the largest gains on IBL BWM for whisker motion energy and reward, increasing reward balanced accuracy from 0.708 to 0.828. The gains are even larger on Allen VBN, where running-speed decoding improves from $R^2 = 0.311$ to 0.611, alongside a further improvement in reward decoding. Overall, ZENITH outperforms SPINT across both datasets and all evaluated behaviors.

Furthermore, despite using no target behavioral labels, zero-shot ZENITH approaches the performance of fully-finetuned external baselines on several tasks. On IBL, it reaches $R^2 = 0.262$ on wheel speed compared with 0.328 for the strongest fully finetuned baseline, and 0.828 balanced accuracy on reward compared with 0.852. We compare these settings in more detail in Section 4.5.

4.3 HOW DO NLE AND DECODER ARCHITECTURE AFFECT ZERO-SHOT TRANSFER?

Having established ZENITH's strong zero-shot transfer relative to baseline methods, we next examine how its two main components, neuron representation and decoder architecture, contribute to performance. The modularity of ZEN allows us to vary each independently. Table 2 compares all combinations of three NLEs and three decoder architectures.

The Effect of Neuron Representations. The three NLEs differ in the information used to represent each neuron. ISI captures only univariate spiking statistics, while NEMO additionally incorporates extracellular waveform features. However, NEMO provides mixed gains over ISI across tasks and decoder architectures. Thus, incorporating waveform information does not consistently improve zero-shot transfer. NuCLR instead uses raw firing rates and population context, and outperforms both ISI and NEMO on every task with every decoder across both datasets. This suggests that NuCLR's richer input supports more transferable neuron representations.

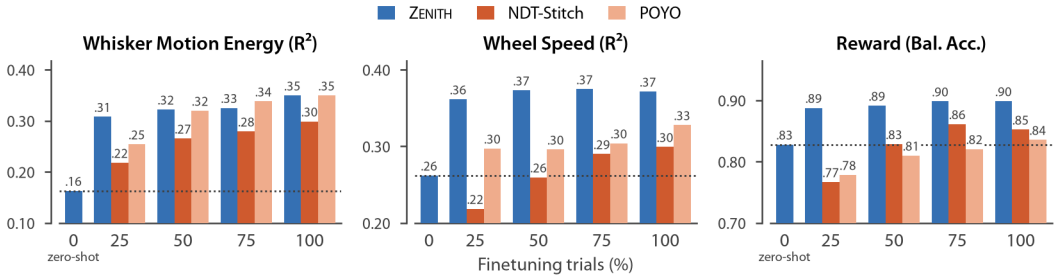

Figure 3: **ZENITH provides data-efficient supervised few-shot finetuning.** Decoding performance of ZENITH, NDT-Stitch, and POYO across increasing fractions of labeled target-session trials on three IBL tasks. The 0% bars and dotted horizontal lines indicate ZENITH's zero-shot performance.

Preserving Neuron-Specific Information During Tokenization. Decoder architecture also influences zero-shot performance. Across NLEs, Time-Patch (introduced in Section 3.2) consistently achieves higher performance and, importantly, remains relatively effective even when paired with relatively simple neuron representations. All three decoders share the same transformer backbone and are parameter-matched, ruling out model scale as an explanation. As described in Section 3.2, POYO and NDT-Stitch aggregate information across neurons before the transformer backbone, whereas Time-Patch preserves separate neuron-level latents. These results are consistent with the hypothesis that preserving neuron-specific information during tokenization is useful for zero-shot generalization.

4.4 ZERO-SHOT PERFORMANCE OF ZENITH SCALES WITH PRETRAINING SUBJECTS

Zero-shot performance of ZENITH improves with the number of pretraining subjects (Fig. 4) Scaling from 11 to 94 subjects yields approximately log-linear improvement in whisker motion energy and wheel-speed decoding, suggesting that larger pretraining datasets could further improve transfer to unseen subjects. Reward decoding also improves with the same trend from 11 to 47 subjects before beginning to plateau, closely approaching its performance after full-finetuning (Section 4.5).

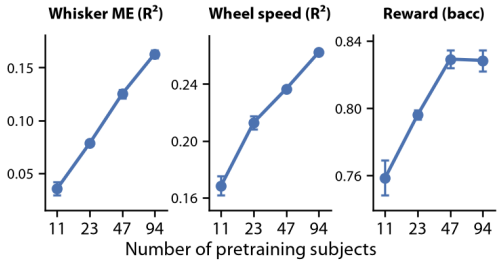

Figure 4: Performance of ZENITH on the IBL Brain-WideMap with varying number of pretraining subjects.

4.5 DATA-EFFICIENT FINETUNING OF ZENITH

We next examine how efficiently ZENITH can be finetuned once target labels are available. Starting from pretrained ZENITH, we vary the fraction of labeled target trials used for supervised finetuning and compare against POYO and NDT-Stitch, which are also pretrained on the same dataset and evaluated on the same held-out test trials (Fig. 3).

Table 3: **Full finetuning performance** across different pretrained baselines on the IBL BWM dataset.

Method	Whisker ME	Wheel Speed	Reward
NDT-Stitch	0.299 ± 0.003	0.299 ± 0.002	0.852 ± 0.001
POYO	**0.350 ± 0.002**	0.328 ± 0.012	0.836 ± 0.004
POSSM	0.266 ± 0.010	0.255 ± 0.029	0.843 ± 0.013
MTM	0.309 ± 0.005	0.270 ± 0.012	0.849 ± 0.002
Time-Patch	0.296 ± 0.006	0.314 ± 0.005	0.879 ± 0.008
ZENITH (Ours)	**0.350 ± 0.009**	**0.372 ± 0.003**	**0.899 ± 0.010**

ZENITH adapts efficiently with only a small amount of target supervision. With only 25% of the available finetuning trials, it reaches approximately $R^2 = 0.36$ on wheel speed and 89% balanced accuracy on reward. These values already surpass the performance of the strongest fully finetuned baselines at 0.328 and 88% respectively (Table 3). Fig. 3 also shows that ZENITH achieves peak performance with much less data than POYO and NDT-Stitch.

With full finetuning, ZENITH matches the best baseline on whisker motion energy and outperforms all baselines on wheel speed and reward (Table 3). Together, these results suggest that adding zero-shot capability does not require sacrificing performance after supervised adaptation. In fact, the zero-shot solution provides a substantially stronger initialization for finetuning.

5 Related Work

Generalization in neural decoding. Neural decoding has traditionally predicted behavior from population activity within a single recording (Pei et al., 2021; Glaser et al., 2020), treating neurons as fixed, indexed inputs. Enabled by large-scale datasets from high-density and brain-wide recordings (Steinmetz et al., 2019; 2021; Ye et al., 2025b), recent foundation models pretrain across sessions and subjects (Azabou et al., 2023; Ye et al., 2023; Azabou et al., 2025; Zhang et al., 2024; Wang et al., 2025a; Ye et al., 2025a), but still associate each population to recording-specific parameters so transfer to a new recording requires alignment or finetuning. Cross-recording generalization has also been studied in brain-computer interfaces (BCIs), where neural drift degrades decoders over time (Dyer et al., 2017; Sussillo et al., 2016; Karpowicz et al., 2025), and benchmarks such as FALCON (Karpowicz et al., 2024) highlight the need for adaptation strategies. However, BCI decoders often operate on unsorted threshold crossings, and chronic implants provide electrodes that remain comparable across sessions (Gallego et al., 2020). These methods therefore do not address the neuron-identity problem under spike sorting, which is essential for transferring decoders across the acute recordings central to systems neuroscience (Harris et al., 2016).

Neuron-level representation learning. A parallel line of work learns neuron-level representations from activity to predict properties such as cell type and brain region (Mi et al., 2023; Wu et al., 2025; Arora et al., 2025; Tolossa et al., 2025; Schneider et al., 2023). Early approaches rely on intrinsic features such as waveform shape or firing statistics (Schneider et al., 2023; Tolossa et al., 2025), while more recent methods also incorporate population context (Mi et al., 2023; Arora et al., 2025). However, these representations have not been used so far as input spaces for behavioral decoding.

Across-subject and zero-shot neural decoding. Across-subject decoding has been most extensively studied in modalities with a shared coordinate system. fMRI recordings are registered to a common anatomical template, allowing methods such as MindEye2, NeuroPictor, and ZEBRA to learn shared latent spaces across individuals, with ZEBRA enabling zero-shot reconstruction (Scotti et al., 2024; Huo et al., 2024; Wang et al., 2025b). EEG models associate channels with standardized positions such as the 10–20 system (Yang et al., 2023; Jiang et al., 2024), and intracranial EEG electrodes can be mapped to MNI coordinates or brain regions (Zahorodnii et al., 2025). Because these signals already reflect the aggregate activity of many neurons (Buzsáki et al., 2012), spatial location can suffice to establish correspondence across subjects. This does not hold for spike-sorted recordings, where neighboring neurons can differ in cell type and tuning (Ohki et al., 2005; Harris & Shepherd, 2015), so anatomical coordinates alone do not determine a neuron's functional identity. ZEN instead infers neuron representations from activity, providing a shared input space for zero-shot transfer to entirely new populations of neurons.

6 Conclusion

We introduced ZEN, a framework that connects neuron-level representation learning with neural decoder pretraining to enable zero-shot transfer across unseen neural populations. By replacing recording-specific parameters with inferred neuron representations, ZEN allows pretrained decoders to operate on new subjects without target behavioral labels, alignment, or parameter updates. To our knowledge, this is the first demonstration of gradient-free, cross-subject zero-shot decoding from spike-sorted electrophysiology. Our work also provides an additional evaluation setting for neuron representation learning: beyond assessing whether representations predict anatomical location or functional characteristics, cross-recording decoding tests whether they capture information that a shared model can use to interpret activity from unseen neurons. Evaluations on two large-scale datasets shows that our approach also provides data-efficient pretrained models: with only 25% of the available finetuning trials, our model surpasses the best fully finetuned baselines on the IBL wheel speed and reward tasks. This suggests a path toward reusable neural decoding systems that can provide high performance in both zero-shot and finetuning regimes.

Limitations and Future Work. Our evaluation focuses on transfer to unseen subjects within two datasets, with stereotyped tasks. Generalization across experimental settings, species, and behavioral tasks remains to be established. Additionally, while we use well-established pretrained NLEs, future work could explore encoders designed specifically to support neural decoding within the ZEN framework.

AI USE STATEMENT

Generative AI tools were used to aid and polish the writing and to draft sections of the paper. The authors take responsibility for the final content of this work.

ETHICS STATEMENT

This work uses publicly available neural datasets and does not involve any new data collection. The study is methodological in nature and introduces no additional human- or animal-subject procedures.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the datasets, evaluation protocols, model architectures, training procedures, baseline implementations, and computational setup in the Appendix. Hyperparameters and implementation details are reported alongside the corresponding methods, and the code is included with the submission.

REFERENCES

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

Vinam Arora, Divyansha Lachi, Ian J Knight, Mehdi Azabou, Blake Richards, Cole Hurwitz, Joshua H Siegle, and Eva L Dyer. Know thyself by knowing others: Learning neuron identity from population context. In *Thirty-ninth Conference on Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2512.01199.

Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh B Nachimuthu, Michael Jacob Mendelson, Blake Aaron Richards, Matthew G Perich, Guillaume Lajoie, and Eva L Dyer. A unified, scalable framework for neural population decoding. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=sw2Y0sirtM.

Mehdi Azabou, Krystal Xuejing Pan, Vinam Arora, Ian Jarratt Knight, Eva L Dyer, and Blake Aaron Richards. Multi-session, multi-task neural decoding from distinct cell-types and brain regions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IuU0wcO0mo.

Corbett Bennett, Samuel D. Gale, Greggory Heller, Tamina K. Ramirez, Hannah Belski, Alex Piet, Omid Zobeiri, Adam Amster, Anton Arkhipov, Alex Cahoon, Shiella Caldejon, Mikayla Carlson, Linzy Casal, Scott F. Daniel, Colin Farrell, Marina Garrett, Ryan Gillis, Conor Grasso, Ben J. Hardcastle, Ross Hytnen, Tye Johnson, Peter Ledochowitsch, Quinn L'Heureux, Dana Mastrovito, Ethan G. McBride, Stefan Mihalas, Chris Mochizuki, Christopher B. Morrison, Chelsea Nayan, Nhan-Kiet Ngo, Kat North, Douglas R. Ollerenshaw, Ben Ouellette, Paul Rhoads, Kara Ronellenfitch, Martin Schroedter, Joshua H. Siegle, Cliff Slaughterbeck, David Sullivan, Jackie Swapp, Michael Taormina, Wayne Wakeman, Xana Waughman, Allison Williford, John W. Phillips, Peter A. Groblewski, Séverine Durand, Christof Koch, and Shawn R. Olsen. Map of spiking activity underlying change detection in the mouse visual system. *Cell*, 189(15):4775–4791.e13, 2026. doi: 10.1016/j.cell.2026.06.025.

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

Max Dabagia, Konrad P Kording, and Eva L Dyer. Aligning latent representations of neural activity. *Nature Biomedical Engineering*, 7(4):337–343, 2023.

Eva L Dyer, Mohammad Gheshlaghi Azar, Matthew G Perich, Hugo L Fernandes, Stephanie Naufel, Lee E Miller, and Konrad P Körding. A cryptography-based approach for movement decoding. *Nature biomedical engineering*, 1(12):967–976, 2017.

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.

Joshua I Glaser, Ari S Benjamin, Raeed H Chowdhury, Matthew G Perich, Lee E Miller, and Konrad P Kording. Machine learning for neural decoding. *eneuro*, 7(4), 2020.

Kenneth D Harris and Gordon MG Shepherd. The neocortical circuit: themes and variations. *Nature neuroscience*, 18(2):170–181, 2015.

Kenneth D Harris, Rodrigo Quian Quiroga, Jeremy Freeman, and Spencer L Smith. Improving data quality in neuronal population recordings. *Nature neuroscience*, 19(9):1165–1174, 2016.

Jingyang Huo, Yikai Wang, Yun Wang, Xuelin Qian, Chong Li, Yanwei Fu, and Jianfeng Feng. Neuropictor: Refining fmri-to-image reconstruction via multi-individual pretraining and multi-level modulation. In *European Conference on Computer Vision*, pp. 56–73. Springer, 2024.

IBL, Dora Angelaki, Brandon Benson, Julius Benson, Daniel Birman, Niccolò Bonacchi, Kcénia Bougrova, Sebastian A Bruijns, Matteo Carandini, Joana A Catarino, et al. A brain-wide map of neural activity during complex behaviour. *Nature*, 645(8079):177–191, 2025a.

IBL, Kush Banga, Julius Benson, Jai Bhagat, Dan Biderman, Daniel Birman, Niccolò Bonacchi, Sebastian A Bruijns, Kelly Buchanan, Robert AA Campbell, et al. Reproducibility of in vivo electrophysiological measurements in mice. *Elife*, 13:RP100840, 2025b.

Wei-Bang Jiang, Li-Ming Zhao, and Bao-Liang Lu. Large brain model for learning generic representations with tremendous EEG data in BCI. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=QzTpTRVtrP.

James J Jun, Nicholas A Steinmetz, Joshua H Siegle, Daniel J Denman, Marius Bauza, Brian Barbarits, Albert K Lee, Costas A Anastassiou, Alexandru Andrei, Çağatay Aydın, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017.

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 and Pamela K Douglas. Interpreting encoding and decoding models. *Current opinion in neurobiology*, 55:167–179, 2019.

Trung Le, Hao Fang, Jingyuan Li, Tung Nguyen, Lu Mi, Amy Orsborn, Uygar Sümbül, and Eli Shlizerman. Spint: Spatial permutation-invariant neural transformer for consistent intracortical motor decoding. *arXiv preprint arXiv:2507.08402*, 2025.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3744–3753, 2019.

Timothy Paul Lillicrap, Jonathan James Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daniel Pieter Wierstra. Continuous control with deep reinforcement learning, 2016.

Lu Mi, Trung Le, Tianxing He, Eli Shlizerman, and Uygar Sümbül. Learning time-invariant representations for individual neurons from population dynamics. *Advances in Neural Information Processing Systems*, 36, 2023.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. Exponential moving average of weights in deep learning: Dynamics and benefits. *arXiv preprint arXiv:2411.18704*, 2024.

Kenichi Ohki, Sooyoung Chung, Yeang H Ch'ng, Prakash Kara, and R Clay Reid. Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. *Nature*, 433(7026): 597–603, 2005.

Josue Ortega Caro, Antonio Henrique Oliveira Fonseca, Syed A Rizvi, Matteo Rosati, Christopher Averill, James L Cross, Prateek Mittal, Emanuele Zappala, Rahul Madhav Dhodapkar, Chadi Abdallah, and David van Dijk. BrainLM: A foundation model for brain activity recordings. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RwI7ZEfR27.

Marius Pachitariu, Nicholas A Steinmetz, Shabnam N Kadir, Matteo Carandini, and Kenneth D Harris. Fast and accurate spike sorting of high-channel count probes with kilosort. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/1145a30ff80745b56fb0cecf65305017-Paper.pdf.

Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

Felix Pei, Joel Ye, David Zoltowski, Anqi Wu, Raeed H Chowdhury, Hansem Sohn, Joseph E O'Doherty, Krishna V Shenoy, Matthew T Kaufman, Mark Churchland, et al. Neural latents benchmark'21: evaluating latent variable models of neural population activity. *arXiv preprint arXiv:2109.04463*, 2021.

Rodrigo Quian Quiroga and Stefano Panzeri. Extracting information from neuronal populations: information theory and decoding approaches. *Nature Reviews Neuroscience*, 10(3):173–185, 2009.

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. *arXiv preprint arXiv:2506.05320*, 2025.

Aidan Schneider, Mehdi Azabou, Louis McDougall-Vigier, David F Parks, Sahara Ensley, Kiran Bhaskaran-Nair, Tomasz Nowakowski, Eva L Dyer, and Keith B Hengen. Transcriptomic cell type structures in vivo neuronal activity across multiple timescales. *Cell Reports*, 42(4):112318, 2023.

Paul S Scotti, Mihir Tripathy, Cesar Kadir Torrico Villanueva, Reese Kneeland, Tong Chen, Ashutosh Narang, Charan Santhirasegaran, Jonathan Xu, Thomas Naselaris, Kenneth A Norman, et al. Mindeye2: Shared-subject models enable fmri-to-image with 1 hour of data. *arXiv preprint arXiv:2403.11207*, 2024.

Joshua H Siegle, Xiaoxuan Jia, Séverine Durand, Sam Gale, Corbett Bennett, Nile Graddis, Greggory Heller, Tamina K Ramirez, Hannah Choi, Jennifer A Luviano, et al. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592(7852):86–92, 2021.

Nicholas A Steinmetz, Peter Zatka-Haas, Matteo Carandini, and Kenneth D Harris. Distributed coding of choice, action and engagement across the mouse brain. *Nature*, 576(7786):266–273, 2019.

Nicholas A Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539): eabf4588, 2021.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

David Sussillo, Sergey D Stavisky, Jonathan C Kao, Stephen I Ryu, and Krishna V Shenoy. Making brain–machine interfaces robust to future neural variability. *Nature communications*, 7(1):13749, 2016.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Gemechu Bekele Tolossa, Aidan M Schneider, Eva Dyer, and Keith B Hengen. Neurons throughout the brain embed robust signatures of their anatomical location into spike trains. *eLife*, 13:RP101506, 2025.

Eric Y Wang, Paul G Fahey, Zhuokun Ding, Stelios Papadopoulos, Kayla Ponder, Marissa A Weis, Andersen Chang, Taliah Muhammad, Saumil Patel, Zhiwei Ding, et al. Foundation model of neural activity predicts response to new stimulus types. *Nature*, 640(8058):470–477, 2025a.

Haonan Wang, Jingyu Lu, Hongrui Li, and Xiaomeng Li. Zebra: Towards zero-shot cross-subject generalization for universal brain visual decoding. *arXiv preprint arXiv:2510.27128*, 2025b.

Wei Wu, Can Liao, Zizhen Deng, Zhengrui Guo, and Jinzhuo Wang. Neuron platonic intrinsic representation from dynamics using contrastive learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Chaoqi Yang, M Westover, and Jimeng Sun. Biot: Biosignal transformer for cross-data learning in the wild. *Advances in Neural Information Processing Systems*, 36:78240–78260, 2023.

Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 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, 2023.

Joel Ye, Fabio Bahmani, Zhengyu Zhou, Jennifer L Collinger, Robert A Gaunt, Leila Wehbe, et al. A generalist intracortical motor decoder. *bioRxiv*, 2025a. doi:10.1101/2025.02.02.634313.

Zhiwen Ye, Andrew M Shelton, Jordan R Shaker, Julien Boussard, Jennifer Colonell, Daniel Birman, Sahar Manavi, Susu Chen, Charlie Windolf, Cole Hurwitz, et al. Ultra-high-density neuropixels probes improve detection and identification in neuronal recordings. *Neuron*, 113(23):3966–3982, 2025b.

Han Yu, Hanrui Lyu, YiXun Xu, Charlie Windolf, Eric Kenji Lee, Fan Yang, Andrew M Shelton, Olivier Winter, International Brain Laboratory, Eva L Dyer, Chandramouli Chandrasekaran, Nicholas A. Steinmetz, Liam Paninski, and Cole Lincoln Hurwitz. In vivo cell-type and brain region classification via multimodal contrastive learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=10JOlFIPjt.

Andrii Zahorodnii, Christopher Wang, Geeling Chau, Bennett Stankovits, Charikleia Moraitaki, Eli Gross, Alexander Brady, Andrei Barbu, Boris Katz, and Ila R Fiete. Neuroprobe: Evaluating intracranial brain responses to naturalistic stimuli. *arXiv preprint arXiv:2509.21671*, 2025.

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, International Brain Laboratory, Eva L Dyer, Liam Paninski, and Cole Lincoln Hurwitz. Neural encoding and decoding at scale. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=vOdz3zhSCj.

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. In *International Conference on Machine Learning*, 2025b.

APPENDIX

A DATASETS, PREPROCESSING, AND QUALITY CONTROL

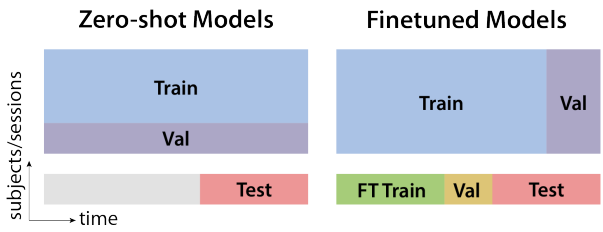

Figure A1: **Dataset splits.** Training, validation, and testing splits used for different model regimes. All models are evaluated on the same test set (marked in red).

We evaluate on two large-scale Neuropixels datasets that differ in behavioral paradigm and anatomical coverage (Fig. 2A). Table A1 summarizes both datasets after quality control (QC) and across subject splits, and Fig. A1 shows the coverage of each split for both the zero-shot and finetuning regimes.

Table A1: **Dataset statistics.** Counts before and after QC, and by subject split.

	IBL BWM	Allen VBN
Subjects (released / after QC)	139 / 119	54 / 54
Sessions (released / after QC)	452 / 357	103 / 103
Probe insertions (released / after QC)	679 / 540	609 / 609
Units after QC (total; median per session)	52,377; 128	72,370; 700
Trials per session (median)	605	282
Train / val / test subjects	94 / 12 / 13	38 / 8 / 8
Trial window	1.0 s	0.8 s
Tasks	Whisker ME, wheel speed, reward	Running speed, reward

A.1 IBL BRAIN-WIDE MAP

Dataset. The IBL BrainWide Map (BWM) (IBL et al., 2025a) contains Neuropixels 1.0 (Jun et al., 2017) recordings collected across 12 laboratories using a standardized visual decision-making task, in which mice report the location of a Gabor stimulus of varying contrast by turning a wheel and receive a water reward for correct choices. Data are accessed through the `ONE` API, using the spike-sorted units from the Brain-Wide Map release.

Trials and behavioral targets. Our choice of tasks and trial windowing follows prior multi-session decoding work on this dataset (Zhang et al., 2024; 2025b). Each sample is chosen to be a 1.0 s window aligned to different task events within each trial (e.g., movement onset, feedback time). Behavioral targets are defined as follows:

- **Whisker motion energy:** motion energy of the whisker-pad region of interest from the side camera, resampled to the bin grid and normalized.
- **Wheel speed:** angular speed of the wheel from the rotary encoder, interpolated to the bin grid and normalized.
- **Reward:** binary trial-level label derived from trial feedback (rewarded vs. unrewarded). Because classes are imbalanced, we report balanced accuracy.

Whisker motion energy and wheel speed are continuous variables sampled at regular intervals within each window (20 ms bins), and we decode them as per-timestep regression targets evaluated with R^2. Reward, in contrast, is a single discrete label per trial, which we decode as a binary classification from the full window.

Quality control. Large-scale electrophysiology datasets exhibit substantial variability in recording quality across animals, sessions, probes, and behavioral modalities. Differences in electrode placement, recording stability, spike-sorting performance, animal engagement, and video tracking quality can all substantially affect downstream decoding performance and the reliability of learned neural representations. To account for this variability, we use session-, probe-, unit-, and behavior-level QC annotations released with BrainWideBench (Andre et al., 2026) to apply a multi-stage filtering criteria:

1. **Session level.** Recordings must satisfy the baseline inclusion criteria established by the IBL Brain-Wide Map release (IBL et al., 2025a), including sufficient task engagement, high behavioral accuracy on easy trials, and successful completion of hardware diagnostic checks.
2. **Probe level.** Insertions are annotated according to raw electrophysiology quality, spike-sorting consistency, and probe alignment reliability.
3. **Unit level.** Units are assessed on waveform amplitude, refractory-period violations, and firing-rate statistics, following the IBL single-unit criteria. Units with extremely low firing rates or severe spike-sorting artifacts are excluded.
4. **Behavior level.** Behavioral measurements derived from video recordings, such as whisker motion energy and tracked movements, are assessed for signal quality and temporal consistency.

Neural and behavioral QC are evaluated independently, and each session is assigned to a group according to whether it passes neural QC, behavioral QC, or both. Evaluation animals are selected from sessions passing both neural and behavioral QC criteria while maintaining broad anatomical coverage across held-out subjects. Units with extremely low firing rates or severe spike-sorting artifacts are excluded from evaluation. After filtering, the dataset comprises 119 subjects, 357 sessions, and 540 probe insertions, reduced from 139 subjects, 452 sessions, and 679 insertions in the original release. Data of lower quality is included in pretraining, where it increases the diversity of neural populations seen by the NLE and decoder without affecting the reliability of evaluation.

Subject splits. Subjects are partitioned into 94 training, 12 validation, and 13 test animals. Following BrainWideBench, splits are balanced by brain-region coverage using the Cosmos atlas parcellation, so that held-out animals sample regions that are represented in training. Within each held-out session, trials are further divided into finetuning, validation, and test portions, which are used for the finetuning experiments (Sec. 4.5). For consistency, zero-shot results are also reported on the same test trials.

Repeated-site vs. Brain-Wide Map sessions. Prior multi-session decoding work on IBL data has focused on the repeated-site sessions (Zhang et al., 2024; 2025b), in which probes across laboratories target a common set of brain regions (IBL et al., 2025b). We instead use the full Brain-Wide Map, which offers a substantially larger and more diverse set of recordings for both pretraining and evaluation. It includes more subjects and sessions, and probe insertions that span a wider range of brain regions. Evaluating on held-out animals from this broader set tests whether decoders transfer across neural populations that vary in both size and anatomical composition, which more closely reflects how pretrained models would be applied to new experiments in practice. Repeated-site insertions that pass our QC criteria are included in this set and assigned to splits alongside all other sessions.

A.2 Allen Visual Behavior Neuropixels

Dataset. The Allen Visual Behavior Neuropixels dataset (Bennett et al., 2026) contains Neuropixels 1.0 (Jun et al., 2017) recordings with up to six probes per session, concentrated in visual cortex and selected subcortical regions, including visual thalamus, hippocampus, and midbrain. Mice perform a change-detection task in which natural images are flashed in sequence and mice lick to report a change in image identity to receive a water reward. The dataset is processed using the `AllenSDK`.

Trials and behavioral targets. For both NLE pretraining and behavior decoding, only neural activity from the active task portion of the recordings is kept. For behavior decoding, windows

of 0.8 seconds are kept, where running-speed windows terminate at stimulus-presentation offsets, and reward-decoding windows extend from 0.45 seconds before to 0.35 seconds after the stimulus change time. Eligible reward trials are non-catch, non-aborted, non-auto-rewarded change trials labeled as either hits or misses. Hits are assigned a target of 1 and misses a target of 0. For each task, overlapping windows are removed, and the remaining windows are ordered chronologically and divided into training (40%), validation (20%), and test (40%) splits.

Quality control. Units are filtered using quality metrics provided by the `AllenSDK`, and the released dataset comprises 54 subjects, 103 sessions, and 609 probe insertions. We retain units labeled as "good" with ISI violations < 0.5, amplitude cutoff < 0.1, presence ratio > 0.95, mean firing rate > 0.1 Hz, and signal-to-noise ratio > 1.0.

Subject splits. Subjects are partitioned into 38 training, 8 validation, and 8 test animals. Zero-shot results are reported on the test portion of held-out subjects' sessions, and the training and validation portions of those sessions are used only for the finetuning experiments.

B Evaluation Protocol Details

B.1 Target-Session Evaluation Protocol

For each held-out animal, we use the standard within-session train, validation, and test splits used for supervised finetuning. Zero-shot models are evaluated directly on the test split without using behavioral labels from the target-session train or validation splits and without any target-specific optimization. Target-session neural activity is available at inference time to compute neuron representations, as described in Section 3.

This protocol allows direct comparison with pretrained decoders that require supervised adaptation to the target session. These models are finetuned on the target-session training data and evaluated on the same held-out test trials used for zero-shot evaluation. Thus, differences in performance arise from whether and how the model adapts to the target recording rather than from differences in the evaluation data.

B.2 Finetuning Protocol

Finetuning protocol. We finetune each pretrained model independently on every held-out target session. The adaptation procedure differs between the baselines and ZEN because the baselines introduce target-specific parameters, whereas ZEN infers neuron representations directly from neural activity.

Baselines. For POYO, NDT, POSSM, and the patched-transformer oracle, adapting to a new recording requires learning target-session-specific parameters. We use gradual unfreezing, first optimizing only the newly initialized unit- and session-specific parameters and the readout for 30 epochs while keeping the pretrained backbone fixed. We then unfreeze the backbone and jointly optimize all parameters for an additional 100 epochs.

ZEN. ZEN does not introduce new neuron- or session-specific parameters for the target recording. The decoder is therefore finetuned end-to-end from the first epoch, while the neuron-level encoder remains fixed throughout finetuning, as during pretraining. We finetune ZEN for 100 epochs.

Optimization. All models are finetuned with AdamW using a batch size of 16, linear learning-rate warmup followed by cosine decay, and gradient clipping with a global ℓ_2 norm of 1.0. We use a peak learning rate of 10^{-5} for ZEN and select method-specific learning rates for the baselines on the validation split. We use the target-session validation metric for early stopping and report test performance from the checkpoint with the best validation performance.

C MODEL DETAILS

C.1 NLE HYPERPARAMETERS

NuCLR NLE. We use NuCLR (Arora et al., 2025) as our activity-based NLE, with the architecture specified in Table A2. NuCLR operates on binned population activity and produces one representation for each recorded neuron. For each neuron, its activity is divided into non-overlapping temporal patches and linearly projected into a sequence of tokens. These tokens are first processed independently for each neuron by temporal transformer layers. They are then passed through spatiotemporal layers that alternate temporal attention with attention across neurons recorded on the same probe. Temporal attention uses rotary position embeddings, while spatial attention does not use positional information. The final tokens are averaged across time to obtain a single neuron representation. Since the same encoder is shared across neurons and recordings, representations can be computed for neurons in recordings not observed during pretraining.

Contrastive pretraining. NuCLR is pretrained using two temporal views sampled from the same recording. The same neuron across the two views forms a positive pair, while other neurons recorded on the same probe provide negatives. Neuron dropout is applied independently to the two views so that the encoder cannot rely on a fixed population. The resulting neuron representations are passed through a one-hidden-layer projection MLP, and the model is optimized using the sample-wise decoupled contrastive objective of NuCLR. The loss is computed independently for each recorded population rather than using neurons from other recordings in the minibatch as negatives. The projection head is discarded after pretraining.

Training. We pretrain the NLE for 50 epochs with AdamW using a peak learning rate of 5×10^{-4}, weight decay of 10^{-2}, and temperature $\tau = 0.2$. Unit dropout is applied during pretraining by uniformly sampling a retention fraction between 50% and 100% for each sample. The pretrained NLE is kept fixed for all subsequent decoder training.

Table A2: Architecture hyperparameters for NuCLR.

HYPERPARAMETER	VALUE
ARCHITECTURE	
BIN SIZE (MS)	20
LATENT STEP (S)	1.0
CONTEXT DURATION (S)	20.0
DIMENSION	256
ATTENTION HEADS	4
DIMENSION PER HEAD	64
TEMPORAL LAYERS	2
SPATIOTEMPORAL LAYERS	4
REGULARIZATION	
ATTENTION DROPOUT	0.0
LINEAR DROPOUT	0.3

NEMO NLE. As an alternative NLE, we use NEMO (Yu et al., 2025), which represents each neuron using its extracellular waveform and firing statistics. For the waveform view, we use the template waveform from the peak-amplitude channel, aligned to the trough and normalized by its peak absolute amplitude. The second view is a rate-conditioned autocorrelogram (ACG), which captures spike-timing structure across firing-rate bins and temporal lags. The waveform and ACG views are processed by separate one- and two-dimensional residual networks, respectively. Architecture and input specifications are given in Table A3.

Contrastive pretraining. NEMO is pretrained with a multimodal contrastive objective that aligns the waveform and ACG representations of the same neuron while separating representations from different neurons. Let $\mathbf{z}^{\mathrm{wvf}}n$ and $\mathbf{z}^{\mathrm{acg}}n$ denote the normalized projected representations of neuron n,

and define

$$s_{nm} = \left\langle \mathbf{z}_n^{\text{wvf}}, \mathbf{z}_m^{\text{acg}} \right\rangle . \tag{5}$$

We use the symmetric contrastive objective

$$\mathcal{L} = -\frac{1}{2N} \sum_{n=1}^{N} \left[\log \frac{\exp(s_{nn}/\tau)}{\sum m = 1^N \exp(s_{nm}/\tau)} + \log \frac{\exp(s_{nn}/\tau)}{\sum_{m=1}^{N} \exp(s_{mn}/\tau)} \right], \tag{6}$$

where N is the batch size and τ is the temperature. Both modalities are augmented during pretraining. Waveforms receive amplitude jitter and additive Gaussian noise, while ACGs are augmented with temporal smoothing and jitter, amplitude scaling, additive Gaussian noise, and pepper noise. After pretraining, the projection heads are discarded and the pre-projection waveform and ACG representations are concatenated to form the final neuron embedding.

Training. We pretrain NEMO for 400 epochs with AdamW using a peak learning rate of 5×10^{-4}, weight decay of 10^{-4}, and a batch size of 1024. Gradients are clipped to a global ℓ_2 norm of 1.0, and training uses bfloat16 mixed precision. For held-out recordings, ACGs are computed only from spikes preceding the evaluated windows so that activity from the evaluation period does not enter the neuron representation. Model selection is based on brain-region decoding performance on the validation split. The pretrained NEMO encoder is kept fixed during subsequent decoder training.

Table A3: Architecture hyperparameters for NEMO.

HYPERPARAMETER	VALUE
INPUTS	
WAVEFORM SAMPLES (PRE/POST TROUGH)	20/40
ACG WINDOW (MS)	66.7
ACG FIRING-RATE BINS	10
WAVEFORM ENCODER (1D RESNET)	
BASE CHANNELS	64
BLOCKS PER STAGE	2, 3, 3, 3
REPRESENTATION DIM	512
DROPOUT	0.1
ACG ENCODER (2D RESNET)	
BASE CHANNELS	32
BLOCKS PER STAGE	2, 2, 2, 3
REPRESENTATION DIM	256
DROPOUT	0.2
PROJECTION AND OUTPUT	
PROJECTION DIM	512
EMBEDDING DIM	768
TEMPERATURE	0.5

C.2 DECODER TOKENIZATION SCHEMES

POYO integration with NLE representations POYO (Azabou et al., 2023) represents neural activity as a sequence of spike tokens, where each spike is associated with its timestamp and a learned embedding for the neuron that emitted it. These spike tokens are compressed into a fixed set of latent tokens through cross-attention and then processed by the Transformer backbone for behavioral decoding.

In the standard POYO formulation, the unit embeddings are learned separately for every neuron observed during training and therefore cannot be directly assigned to neurons in a new recording. We replace this learned embedding table with neuron representations produced by the NLE. For each neuron n, the NLE representation $\mathbf{u}_n$ is projected to the POYO embedding dimension using a shallow MLP,

$$\tilde{\mathbf{e}}_n = f_{\text{MLP}}(\mathbf{e}_n), \tag{7}$$

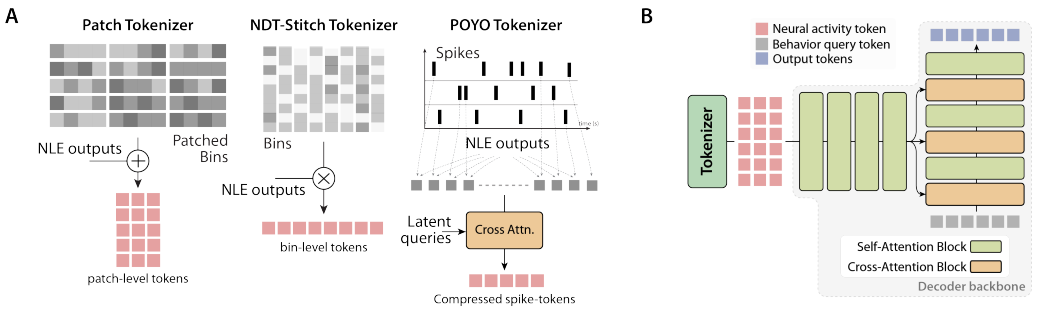

Figure A2: **Decoder architectures**. **(A)** Overview of the three tokenization schemes. **(B)** A schematic of the common transformer decoder backbone.

and $\mathbf{e}_n$ is used as the identity embedding for every spike emitted by neuron n. The spike token compression using a cross-attention layer remains the same. The resulting latents are then processed by the decoder backbone. For fair comparison across ZEN decoders, we use a common decoder backbone architecture described in Section C.3.

NDT-Stitch integration with NLE representations NDT-Stitch (Ye & Pandarinath, 2021; Pandarinath et al., 2018) operates on binned population activity and maps each time bin to a latent token using a recording-specific stitching matrix. For a population of N neurons, let $\mathbf{x}_t \in \mathbb{R}^N$ denote the spike counts at time bin t, and let $W \in \mathbb{R}^{d\times N}$ be the learned stitching matrix. The input token is

$$\mathbf{z}_t = W\mathbf{x}_t = \sum_{n=1}^{N} \mathbf{w}_n x_{tn}, \tag{8}$$

where $\mathbf{w}_n$ is the nth column of W. This form makes explicit that stitching assigns a learned vector $\mathbf{w}_n$ to each neuron in the recording.

We replace these recording-specific vectors with neuron representations produced by the NLE. Each NLE representation $\mathbf{u}n$ is projected to the stitching dimension using a shallow MLP, giving

$$\mathbf{z}_t = \sum_{n=1}^{N} f_{\text{MLP}}(\mathbf{e}_n), x_{tn} = \Big[f_{\text{MLP}}(\mathbf{e}_1), \ldots, f_{\text{MLP}}(\mathbf{e}_N)\Big]\mathbf{x}_t. \tag{9}$$

Thus, the stitching matrix is no longer learned separately for each recording, but is constructed from NLE representations at inference time. The resulting token sequence is then processed by the common decoder backbone architecture described in Section C.3.

Time-Patch integration with NLE representations. Time-Patch directly combines each neuron's binned activity with its NLE representation, as described in Section 3.2. Let $\mathbf{x}_{np} \in \mathbb{R}^B$ denote the pth temporal patch of neuron n, and let $\mathbf{u}n$ denote its NLE representation. Each neural token is constructed as

$$\mathbf{z}_{np} = W_{\text{patch}}\mathbf{x}_{np} + f_{\text{MLP}}(\mathbf{e}_n), \tag{10}$$

where W_{patch} projects the activity patch into the decoder dimension and f_{MLP} projects the NLE representation into the same space. Both mappings are shared across neurons and recordings.

This produces one token for each neuron and temporal patch, allowing the preservation neuron-level information through most of the transformer backbone.

C.3 DECODER TRANSFORMER BACKBONE

We use the same transformer backbone for all our main models (Fig. A2B). Given a set of tokenized latents $Z = (\mathbf{z}_1, \ldots, \mathbf{z}_L)$ produced by the tokenizer, the decoder produces behavioral predictions at a set of queried timesteps. The tokenized latents are first processed through L layers of Self-Attention blocks, allowing each token to integrate information across the full neural population and across time. These contextualized latents are then queried for behavior latents using H layers of Cross-Attention

Table A4: Architecture of our neural decoder backbones

HYPERPARAMETER	VALUE
ARCHITECTURE	
CONTEXT DURATION (S)	1.0
NUMBER OF SA LAYERS	8
NUMBER OF CA LAYERS	4
HIDDEN DIMENSION	256
ATTENTION HEADS	4
DIMENSION PER HEAD	64
REGULARIZATION	
ATTENTION DROPOUT	0.0
LINEAR DROPOUT	0.2

Transformer Blocks. Finally, the output latents are linearly projected into the behavior space to produce the predictions. We use RoPE (Su et al., 2024) for incorporating all temporal information into the network.

Hyperparameters of our the backbone are listed in Table A4.

C.4 PRE-TRAINING PROCEDURE AND TECHNIQUES

We train our decoders with a linear-rise-cosine-decay learning rate schedule, with a max learning rate of $5e^{-4}$. We train with batch size 512 trials, with the AdamW optimizer, and set a max epoch limit to 30 epochs (sufficient for convergence on these large datasets), and do early stopping with patience of 10 epochs.

While training the decoder, we also employ two regularization techniques to promote across-subject generalization.

EMA Decoder. Individual gradient updates to the decoder can be noisy, making the weights at any particular training step especially less stable for zero-shot inference. To obtain smoother parameters, we maintain an exponential moving average (EMA) of the decoder weights (Morales-Brotons et al., 2024), a technique widely used in reinforcement learning (Mnih et al., 2015; Lillicrap et al., 2016) and semi-supervised learning (Tarvainen & Valpola, 2017). A copy of the decoder with weights θ_{ema}, initialized identically to the online decoder, is maintained throughout training. While the *online* decoder parameters θ_{gd} are updated via gradient descent, the EMA decoder is updated after each step as

$$\theta_{\text{ema}} \leftarrow \beta, \theta_{\text{ema}} + (1 - \beta), \theta_{\text{gd}}, \tag{11}$$

where $\beta \in (0, 1)$ controls the smoothing rate. At the end of training, the online decoder is discarded and the EMA checkpoint is used for zero-shot evaluation and as the initialization for finetuning. This is ablated in Section G.

Unit Dropout. We find that aggressively dropping neurons during decoder pretraining improves zero-shot generalization. For each training sample, we uniformly sample a dropout fraction between 0 and 75% and randomly drop the corresponding subset of neurons.

D ZERO-SHOT BASELINES

We compare ZEN against five baselines that can be evaluated directly on unseen recordings without fitting unit- or session-specific parameters at test time. The linear and GRU baselines operate on population-averaged firing rates and therefore discard neuron identity entirely. Set Transformer retains separate unit activity traces but does not assign identities to them. Unsorted POYO replaces unit identity with electrode identity effectively using threshold-crossing events, while SPINT infers an identity representation directly from neural activity. We describe the implementation and training details for each baseline below.

D.1 LINEAR

The linear baseline uses only the population-averaged firing rate. For each decoding window, spikes are binned at 20,ms and averaged across the N recorded neurons,

$$\bar{x}_t = \frac{1}{N} \sum_{n=1}^{N} x_{nt}, \qquad \bar{\mathbf{x}} \in \mathbb{R}^T, \tag{12}$$

where x_{nt} is the spike count of neuron n in time bin t. A linear readout maps the resulting population-rate sequence to the behavioral prediction. Since the input does not depend on neuron identity or population size, the same model can be evaluated directly on held-out recordings. The model is trained on the pretraining recordings using the same optimization procedure as the other decoders. We use a bin size of 20,ms and introduce no additional architectural hyperparameters.

D.2 GATED RECURRENT UNIT (GRU)

The GRU baseline uses the same population-averaged firing-rate sequence defined in Eq. 12. The sequence is processed by a bidirectional GRU, followed by a linear readout from the hidden state at each time bin. Bidirectional recurrence allows each prediction to use activity from the full decoding window. As with the linear baseline, the input contains no unit- or session-specific information, so the model can be evaluated directly on held-out recordings. The model is trained on the pretraining recordings, with architecture hyperparameters given in Table A5.

Table A5: Hyperparameters for the GRU decoding baseline.

HYPERPARAMETER	VALUE
BIN SIZE (MS)	20
DEPTH	4
HIDDEN DIMENSION	128
DROPOUT	0.2
BIDIRECTIONAL	TRUE

D.3 UNSORTED POYO

POYO (Azabou et al., 2023) represents neural activity as a sequence of discrete events, with each event associated with a learned identity embedding and its time of occurrence. In the standard spike-sorted setting, this identity corresponds to a sorted unit and is therefore recording-specific. For Unsorted POYO, we instead use threshold-crossing events and represent each event by the identity of the electrode on which it was detected. The electrode embedding table is shared across recordings, allowing the model to operate on new sessions without introducing unit-specific parameters.

The decoder follows the POYO architecture, using cross-attention to map the variable-length event sequence to latent tokens, transformer layers to process the latent representation, and behavior queries for readout. For recordings with multiple probes, each probe is processed independently and the resulting predictions are averaged.

Training. Unsorted POYO is trained end-to-end on the pretraining recordings and evaluated zero-shot on held-out recordings. We use a hidden dimension of 256, four attention heads, 16 latent tokens per 125,ms interval, eight self-attention layers in the encoder, four cross-attention layers in the decoder, rotary time embeddings, and dropout of 0.2.

D.4 SET TRANSFORMER

We use a Set Transformer (Lee et al., 2019) as an identity-free, session-agnostic baseline. For each unit, we bin its spike counts at 20,ms over the decoding window and map the resulting activity vector to a latent token using an MLP shared across all units and recordings. No unit- or session-specific embeddings are added. The resulting set of unit tokens is processed by self-attention without positional encodings and mean pooled before the behavioral readout. This allows the model to operate directly on unseen populations with different numbers and orderings of units.

Training. The model is trained end-to-end on the pretraining recordings and evaluated zero-shot on held-out recordings. We use a hidden dimension of 256, eight self-attention layers with four heads, dropout of 0.2, and unit dropout during training.

D.5 SPINT

SPINT (Le et al., 2025) is a permutation-invariant neural decoder designed to transfer across recordings with changes in the number and identity of recorded inputs. Rather than assigning a learned embedding to each input, SPINT infers its identity directly from neural activity and uses these inferred identities to condition the decoder.

SPINT was originally designed for intracortical motor decoding from threshold-crossing activity recorded with fixed electrode arrays. Our benchmarks instead use spike-sorted Neuropixels recordings. We adapt SPINT by treating each sorted unit as an input channel and using its binned spike train in place of the corresponding threshold-crossing trace. The rest of the architecture remains unchanged.

Neural identity encoder. For unit i, SPINT infers an identity embedding from a collection of unlabeled activity windows. Let $\mathbf{c}_i^{(m)}$ denote the activity of unit i in the mth window. Its identity is computed as

$$\mathbf{e}i = \psi\left(\frac{1}{M}\sum_{m=1}^{M}\phi\left(\mathbf{c}_i^{(m)}\right)\right), \tag{13}$$

where ϕ and ψ are MLPs shared across all units and recordings. The inferred identity is then added to the activity of the unit in the decoding window,

$$\mathbf{z}_i = \mathbf{x}_i + \mathbf{e}_i, \tag{14}$$

where $\mathbf{x}_i$ denotes its binned activity.

Decoder. Each identity-conditioned trace $\mathbf{z}_i$ is mapped to a latent unit token by a shared read-in MLP. Learnable behavior queries aggregate information across the set of unit tokens using cross-attention, followed by a feedforward block and linear readout. Since the same transformations are applied to every unit and cross-attention operates over an unordered set, SPINT can accommodate different numbers and orderings of recorded units.

Training. We train SPINT end-to-end, jointly optimizing the identity encoder and decoder. We use 20,ms bins, a hidden dimension of 512, three-layer MLPs for the identity encoder, a single cross-attention layer, and unit dropout during training. In contrast to ZEN, whose neuron-level encoder is pretrained separately and kept fixed during decoder training, SPINT learns its activity-derived unit representations jointly with the decoding objective.

E FINETUNED BASELINES

We compare against several pretrained neural decoders that span commonly used approaches for modeling spike-sorted population activity, including spike-token models, stitching-based models, recurrent architectures, and masked-modeling objectives. For each baseline, we follow the original architecture and pretraining procedure as closely as possible while using a common finetuning protocol for adaptation to target recordings. We describe the implementation and training details for each model below.

We also attempted to include NEDS (Zhang et al., 2025a) as an additional baseline using the publicly available implementation. However, we were unable to train the model successfully on our datasets or reproduce the results reported in the original work, and therefore do not include NEDS in our comparisons.

E.1 POYO

POYO (Azabou et al., 2023) was pretrained using the Adam optimizer with a base learning rate of 5×10^{-5}, weight decay of 10^{-4}, and batch size of 256 in `bf16` precision. We use a one-cycle

learning rate schedule with a warmup fraction of 0.5 and divisor factor of 1.0. We also apply unit dropout during training for better generalization. Architecture hyperparameters are listed in Table A6.

Table A6: Architecture hyperparameters for POYO.

Hyperparameter	Value
Architecture	
Latent step (s)	0.125
Latents per step	32
Dimension	256
Depth	6
Dimension per head	64
Cross-attention heads	2
Self-attention heads	8
Regularization	
FFN dropout	0.2
Linear dropout	0.4
Attention dropout	0.2

E.2 POSSM

POSSM (Ryoo et al., 2025) shares POYO's spike tokenization and input/output cross-attention, but replaces the latent self-attention layers with a recurrent state-space backbone. Spikes are grouped into contiguous, non-overlapping 50 ms chunks. Within each chunk, a cross-attention layer compresses the spike tokens into a single latent vector, using the same unit embeddings and rotary time encodings as POYO. The sequence of chunk latents is processed by the recurrent backbone. Output queries attend over the hidden states of the $k = 3$ most recent chunks to produce behavioral predictions. Following Andre et al. (2026), we use a bidirectional recurrent backbone in place of the original causal one to match the non-causal decoders used elsewhere. Optimization, precision, learning rate schedule, and unit dropout follow POYO. Architecture hyperparameters are listed in Table A7.

Table A7: Architecture hyperparameters for POSSM. Shared hyperparameters follow POYO (Table A6).

Hyperparameter	Value
Architecture	
Chunk size (s)	0.05
Latents per chunk	1
Dimension	256
Dimension per head	64
Cross-attention heads	1
Recurrent backbone	Bidirectional
Readout context (chunks)	3
Regularization	
Linear dropout	0.4
Attention dropout	0.2

E.3 NDT-Stitch

NDT-Stitch (Ye & Pandarinath, 2021) is pretrained in `bf16` precision for 100 epochs using a masked token modeling objective. Input spikes are masked with a ratio of 0.5, using a mixture of random token masking and block masking (block mask probability 0.5, maximum block size 7 bins).

E.4 MtM

MtM (Zhang et al., 2024) uses the same Transformer backbone as NDT-Stitch, but replaces temporal masking with a multi-task masking objective that alternates among four schemes. *Causal* masking

Table A8: Architecture hyperparameters for NDT.

HYPERPARAMETER	VALUE
ARCHITECTURE	
BIN SIZE (MS)	20
HIDDEN DIMENSION	256
ENCODER LAYERS	13
FEEDFORWARD DIMENSION	1028
ATTENTION HEADS	8
ACTIVATION	RELU
REGULARIZATION	
PRE-ENCODER DROPOUT	0.2
POST-ENCODER DROPOUT	0.2
ATTENTION DROPOUT	0.2
INITIALIZATION	
T-FIXUP BASE SCALE	0.67
T-FIXUP VALUE SCALE	$\sqrt{2}$

hides future time bins, *meuron* masking hides random subsets of units, *intra-region* masking hides random units within a randomly selected brain region, and *inter-region* masking hides all units in a brain region. The region-based schemes use the anatomical annotation of each unit, and masked inputs are zeroed out. A learnable prompt token identifying the active masking scheme is prepended to the input sequence, along with a session identity token. Each recording is mapped into the hidden dimension by a session-specific linear read-in followed by a shared MLP, and mapped back to its units by a session-specific read-out stitcher. Architecture hyperparameters match NDT-Stitch, as well as training hyperparamters where relevant.

Fewshot Finetuning. We finetune all pretrained baselines using gradual unfreezing. For the first 30 epochs, only the newly initialized session-specific parameters (unit embeddings, session embeddings, and readout layers) are trained while the pretrained backbone remains frozen. The backbone is then unfrozen and all parameters are trained jointly, with early stopping patience of 10 epochs, up to a maximum of 130 epochs total.

F RUNTIME DETAILS

Hardware. All models are pretrained on 1 NVIDIA RTX B200 GPU and finetuning was done on 1 NVIDIA RTX 5090 GPU.

Runtime Pretraining ZEN and the corresponding base architectures requires approximately 5 hours, depending on the amount of pretraining data used. This includes pretraining NuCLR ($\sim$4hr) and the neural decoder ($\sim$1hr). Fine-tuning a pretrained model across all 29 evaluation sessions requires an additional $\sim$1 hour.

G ABLATION STUDIES

We evaluate the contribution of each component of ZENITH through the ablations summarized in Table A9. The largest degradation occurs when neuron embeddings are removed entirely (last row). Removing the EMA decoder also produces a more modest but consistent degradation, particularly on the regression tasks, while leaving reward decoding largely unchanged. This suggests that EMA improves the stability of zero-shot decoding, but is less important than the neuron representations themselves.

Table A9: **Ablations.** Base model corresponds to the ZENITH configuration.

Model Variant	Whisker Motion Energy		Wheel Speed		Reward (c=2)	
	R^2	Pearson r	R^2	Pearson r	Bal. Acc.	Bal. AP
Base	0.163 ± 0.004	0.459 ± 0.005	0.262 ± 0.002	0.558 ± 0.003	0.828 ± 0.006	0.876 ± 0.003
w/o EMA model	0.148 ± 0.006	0.444 ± 0.011	0.246 ± 0.002	0.534 ± 0.009	0.830 ± 0.004	0.875 ± 0.002
w/o Neuron Embed	0.042 ± 0.001	0.305 ± 0.001	0.185 ± 0.001	0.499 ± 0.001	0.626 ± 0.001	0.747 ± 0.002