FREECAB: A BENCHMARK FOR NEURAL ENCODING AND DECODING FROM FREELY MOVING HIPPOCAMPAL CALCIUM IMAGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural encoding and decoding are commonly evaluated in constrained or trial-structured settings with curated targets and relatively homogeneous recordings, leaving continuous neural–behavior modeling under freely moving conditions less systematically characterized. We introduce FreeCAB, a benchmark built from 58 curated session-level recordings that pair dorsal CA1 miniscope calcium imaging with synchronized multi-view behavior. FreeCAB defines matched bidirectional sequence tasks under a unified session-based interface, evaluating both 2D position and reconstructed 3D pose in single-session neural encoding and decoding, together with explicit tracking-quality control. Position-based extensions further evaluate behavioral-state dependence, held-out adaptation after multi-session pretraining, and joint encoding–decoding training. Across five general-purpose baselines, position decoding is consistently more reliable than position-to-neural encoding. Using 3D pose as the encoding input yields higher Pearson correlation than 2D position for all five baselines, although explained variance does not improve uniformly; conversely, 3D pose remains decodable from neural activity but is more difficult to decode than 2D position. Position decoding is also consistently stronger during locomotion across tested speed thresholds. Multi-session pretraining and joint training produce model-, metric-, and task-dependent effects rather than uniformly reliable gains. Together, FreeCAB provides a benchmark for studying how behavioral representation, behavioral state, and session heterogeneity shape neural–behavior modeling under continuous free behavior.

1 INTRODUCTION

Modeling the relationship between behavior, stimuli, and neural population activity is a central objective in systems neuroscience and neural machine learning Brown et al. (2004); Paninski et al. (2007); Cunningham & Yu (2014); Saxena & Cunningham (2019). This objective is commonly studied through two complementary paradigms: neural decoding, which predicts behavioral or experimental variables from neural activity Zhang et al. (1998); Brown et al. (1998); Glaser et al. (2020), and neural encoding, which predicts neural responses from behavioral or external variables Paninski et al. (2007); Pillow et al. (2008). Over the past decade, a growing set of standardized tasks and public datasets has accelerated progress in both directions by enabling systematic comparison of model architectures under controlled conditions Pei et al. (2021); Karpowicz et al. (2024); Willeke et al. (2022); Gifford et al. (2024); Lueckmann et al. (2025). These benchmarks have been highly valuable for refining decoding and encoding algorithms, particularly in settings with well-defined task structure, curated labels, and relatively homogeneous recordings.

However, existing benchmarks often rely on experimental conditions that impose explicit task structure and simplify natural behavior. Many widely used evaluations are conducted in head-fixed preparations or highly structured paradigms Pei et al. (2021); Karpowicz et al. (2024), where behavioral variability is restricted and target variables are cleanly defined. While such settings are essential for isolating specific neural computations, they do not fully capture the challenges of modeling neural activity in freely moving conditions. In freely moving exploration, behavior unfolds continuously without externally prescribed trial boundaries, while behavioral training targets are extracted from video-based tracking or pose-estimation pipelines and aligned with ongoing neural recordings. Dif-

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.

ferent targets can vary substantially in temporal stability and measurement noise, and recording heterogeneity across sessions can become a dominant source of modeling difficulty. These challenges are especially pronounced in freely moving calcium imaging, where population recordings enable naturalistic neural-behavior analysis but also introduce variability in signal quality, detected neuron count, and cross-session cell identity alignment Ghosh et al. (2011); Aharoni & Hoogland (2019); Wirtshafter & Disterhoft (2022); Sheintuch et al. (2017). This motivates standardized evaluation of neural–behavioral modeling under continuous free behavior, across behavioral representations, states, and recording sessions.

To address this gap, we introduce FreeCAB, a benchmark for neural encoding and decoding from freely moving hippocampal calcium imaging. FreeCAB is built from synchronized multi-view behavioral recordings and dorsal CA1 miniscope data collected across multiple animals and sessions. Rather than focusing on a single aggregate score, FreeCAB places neural decoding and neural encoding under a shared session-based interface. We instantiate this bidirectional framework around continuous neural–behavioral regression using two-dimensional position and richer three-dimensional pose: decoding predicts either representation from neural activity, whereas encoding predicts neural activity from the corresponding aligned behavioral covariates. This design enables systematic study of whether encoding and decoding exhibit similar difficulty, how richer 3D behavior affects neural prediction, whether position decoding depends on behavioral state, whether neural-behavioral models generalize across heterogeneous recording sessions, and whether joint encoding-decoding training provides complementary benefit.

Our contributions are summarized as follows:

- **A benchmark for bidirectional neural–behavioral modeling under continuous free behavior.** We introduce FreeCAB, combining freely moving hippocampal calcium imaging with synchronized multi-view behavioral recordings to support neural encoding and decoding using position and 3D pose from continuous neural–behavioral sequences.
- **A unified protocol across behavioral representations, states, and recording sessions.** FreeCAB evaluates encoding and decoding under a shared windowed session interface, enabling matched single-session evaluation, behavioral-state-conditioned analysis, cross-session adaptation, and supplementary joint encoding–decoding analysis.
- **Empirical findings on behavioral representation, state dependence, and transfer uncertainty.** We observe position encoding–decoding asymmetry, higher encoding correlation with 3D pose than with position alone, and more reliable position decoding during locomotion than stationary/low-mobility periods. Multi-session pretraining and joint training produce conditional, model-dependent effects rather than uniformly reliable improvements.

2 Related Work

Neural encoding and decoding paradigms. Neural encoding and decoding provide complementary views of the relationship between neural activity, behavior, stimuli, and other external variables Paninski et al. (2007); Brown et al. (2004); Pillow et al. (2008); Glaser et al. (2020). Decoding asks what behavioral or task-related information can be read out from neural population activity, whereas encoding asks how neural responses can be predicted from measured covariates. These paradigms are closely tied to population-level neural analysis Cunningham & Yu (2014); Saxena & Cunningham (2019). In hippocampal recordings, spatial position is a canonical behavioral variable linked to neural activity through place-cell coding, ensemble population codes, and position reconstruction studies O'Keefe & Dostrovsky (1971); Wilson & McNaughton (1993); Brown et al. (1998); Zhang et al. (1998); Jensen & Lisman (2000). At the same time, hippocampal activity is modulated by locomotion, running speed, and theta-related dynamics McFarland et al. (1975); Geisler et al. (2007); McClain et al. (2019), motivating evaluation beyond a single aggregate score. FreeCAB builds on this tradition by placing neural encoding and decoding across position and pose representations under a unified benchmark protocol for freely moving calcium imaging.

Benchmarks for neural encoding and decoding. Standardized benchmarks have become increasingly important for evaluating neural models across datasets, tasks, and recording conditions, with related efforts spanning brain-model comparison and standardized BCI algorithm benchmarking Schrimpf et al. (2018); Jayaram & Barachant (2018). The Neural Latents Benchmark

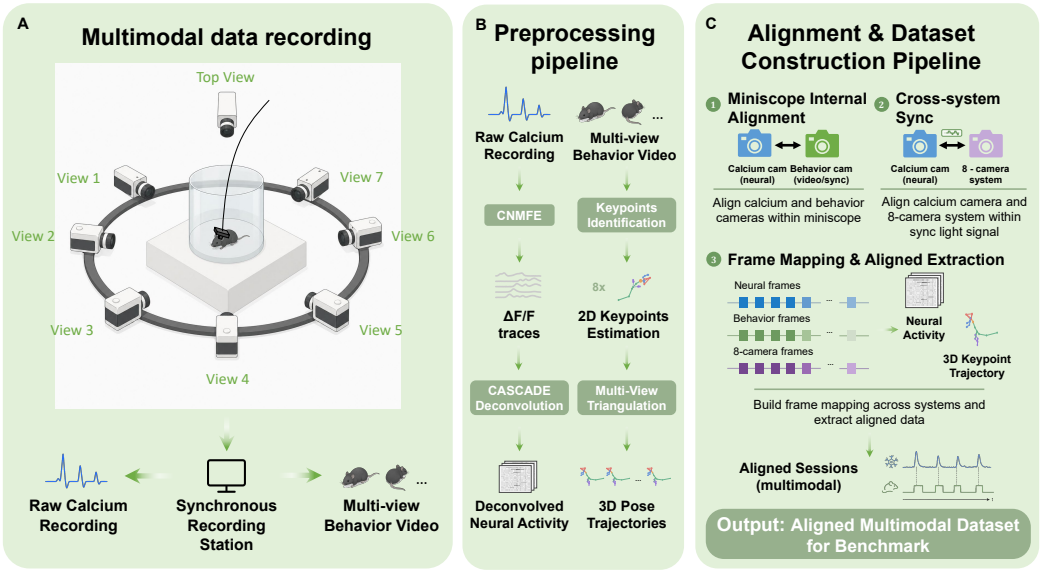

Figure 1: FreeCAB data acquisition and alignment pipeline. Synchronized multi-view behavior and dorsal CA1 calcium imaging are processed into aligned position, 3D pose, and neural activity sequences for benchmark construction.

(NLB'21) Pei et al. (2021) established a rigorous benchmark for neural population modeling, while FALCON Karpowicz et al. (2024) highlighted consistent decoding under realistic deployment constraints. In sensory neuroscience, Sensorium Willeke et al. (2022) and the Algonauts Project Gifford et al. (2024) advanced large-scale neural encoding under naturalistic stimuli, and ZAPBench Lueckmann et al. (2025) extended benchmark-oriented evaluation to whole-brain activity prediction in zebrafish. However, these benchmarks largely operate in trial-structured, clinically constrained, stimulus-driven, or non-hippocampal settings. FreeCAB instead targets freely moving hippocampal calcium imaging, where continuous behavior, cross-system alignment, and session heterogeneity jointly challenge neural-behavioral modeling.

Multi-session learning and held-out adaptation. Neural modeling has increasingly moved beyond single-session fitting toward reusable, adaptive, and data-efficient models. Recent work has explored representation learning, multi-context pretraining, scalable decoding, few-shot calibration, and multi-session training across recordings Azabou et al. (2023; 2025); Ye & Pandarinath (2021); Ye et al. (2023); Karpowicz et al. (2024); related foundation-style and multimodal approaches further study shared neural representations across tasks, animals, stimulus conditions, or modeling directions Schneider et al. (2023); Antoniades et al. (2023); Zhang et al. (2025); Wang et al. (2025); Willeke et al. (2026). Other recent work addresses heterogeneous input structure through population-level representation learning, topology-aware pretraining, and generative neural population models Chau et al. (2025); Yi et al. (2023); Kapoor et al. (2024). In freely moving calcium imaging, however, heterogeneity is especially pronounced: miniaturized calcium imaging enables population recordings during natural behavior Ghosh et al. (2011); Aharoni & Hoogland (2019); Wirtshafter & Disterhoft (2022), but sessions vary in neuron count, signal quality, behavioral coverage, and cell identity alignment Sheintuch et al. (2017); neural population representations may also vary or remain stable across environments, behaviors, and long timescales Muller & Kubie (1987); Ziv et al. (2013); Kinsky et al. (2018); Gallego et al. (2020). FreeCAB therefore evaluates multi-session learning through held-out adaptation: models are pretrained on held-in sessions and then adapted to unseen held-out sessions, with scratch adaptation as a matched target-session baseline.

3 DATASET

3.1 OVERVIEW

We study a freely moving hippocampal calcium imaging dataset collected from mice performing unconstrained exploration in a circular open-field arena. The dataset combines synchronized multi-view behavioral video with single-photon calcium imaging in dorsal CA1, enabling paired neural-behavior modeling from continuous naturalistic recordings. Unlike trial-based datasets with externally imposed structure, FreeCAB preserves ongoing behavioral variability and therefore provides a realistic setting for evaluating neural encoding and decoding under freely moving conditions.

The raw collection covered 19 mice across multiple recording days. Because the benchmark is defined at the session level rather than the animal level, different animals contribute different numbers of benchmark-eligible sessions after preprocessing and screening. After preprocessing, alignment, and quality control, the retained benchmark pool contains 58 session-level recordings, with 48 held-in sessions used for model development and pretraining and 10 held-out sessions reserved for final reporting. The complete anonymized session inventory is provided in Appendix A.5.

3.2 MULTIMODAL ACQUISITION, PREPROCESSING, AND TRACKING QUALITY

Behavioral data were acquired using a synchronized multi-view recording system, while neural activity was recorded from dorsal CA1 using a Miniscope v3. The behavioral pipeline converts multi-view videos into keypoints, 3D pose trajectories, and two-dimensional position traces, whereas the neural pipeline converts raw calcium recordings into extracted and deconvolved neural activity. FreeCAB uses 2D position as a canonical spatial representation and 3D pose as a richer description of ongoing behavior, enabling neural–behavioral analyses across complementary behavioral representations. The multi-view recordings additionally improve tracking robustness and support motion-derived variables used in downstream behavioral-state analyses. The neural and behavioral streams are aligned through a frame-mapping procedure to produce paired neural–behavior sequences at the session level.

Behavioral tracking quality. We assess 3D tracking consistency by reprojecting the final 3D keypoints into the camera views and comparing them with confidence-eligible 2D detections. Across all 58 retained sessions, covering 1,235,835 frames and 236,958,675 confidence-eligible observations across available camera views, the pooled reprojection residual has a median of 21.90 pixels (IQR: 12.61–36.52; P95: 78.80). Session-level statistics in Figure 4 show broadly consistent tracking quality across recordings. These residuals quantify multi-view consistency with the underlying 2D detections rather than error against independent 3D ground truth. Missing-view coordinates synthesized by projection are excluded from all QC statistics.

Figure 1 summarizes the multimodal acquisition, preprocessing, and alignment pipeline. Detailed acquisition, alignment, and tracking-quality procedures are provided in Appendix A.1, Appendix A.2, and Appendix A.3, respectively.

3.3 ALIGNED BENCHMARK SESSIONS

We treat session inclusion as benchmark curation rather than model selection. Sessions are retained only when they provide usable neural-behavior alignment, sufficient behavioral coverage, and stable usable segments. Some sessions are truncated to a fixed usable prefix when later portions show reduced coverage or drift. These decisions are fixed before model training and are recorded in the public session inventory.

The retained session pool intentionally preserves realistic heterogeneity in neuron count, usable duration, behavioral coverage, and session identity. This heterogeneity is part of the benchmark challenge rather than a nuisance to be removed, and it motivates the held-out adaptation and session-level analyses used later in the paper. The full quality-control procedure and final retained inventory are described in Appendix A.4 and Appendix A.5.

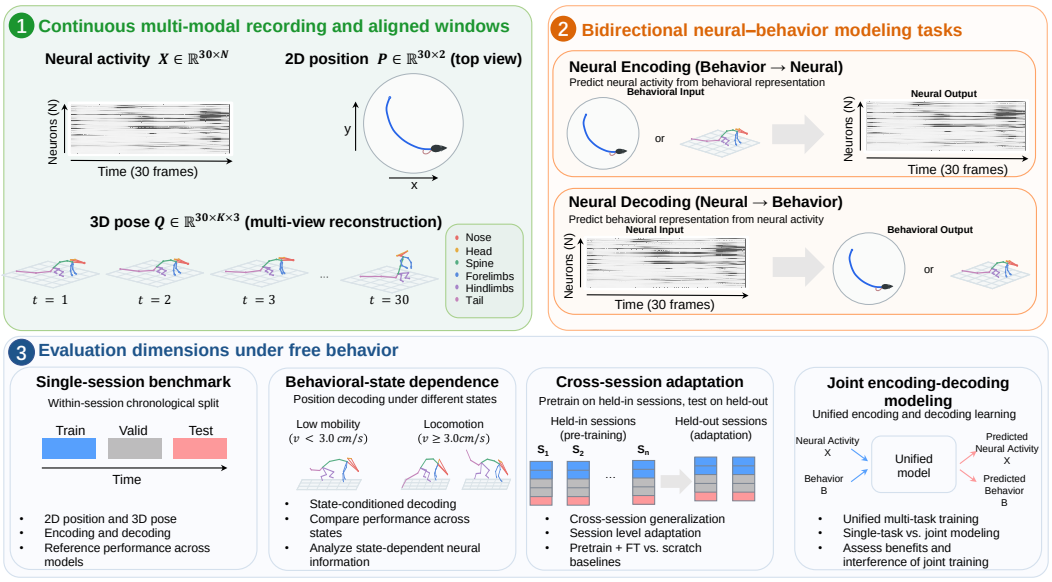

Figure 2: FreeCAB benchmark framework. (1) Temporally aligned 1-second windows contain neural activity, 2D position, and reconstructed 3D pose. (2) FreeCAB defines bidirectional neural encoding and decoding using either 2D position or 3D pose as the behavioral representation in the single-session setting. (3) Position-based tasks are further evaluated across behavioral states, cross-session adaptation, and joint encoding–decoding modeling.

4 FreeCAB Benchmark Framework

FreeCAB is designed as a benchmark for bidirectional neural-behavior modeling in freely moving hippocampal calcium imaging. Rather than evaluating a single model family or reporting only one aggregate score, the benchmark organizes comparison across modeling direction, behavioral representation, session setting, and behavioral state. In the current study, we evaluate continuous neural–behavioral regression using position and 3D pose under a shared single-session protocol, with held-out adaptation from multi-session pretraining further assessing position decoding. We next describe the task formulation, evaluation protocol, research questions, and baseline families considered in this work.

4.1 Task Formulation

Let $X_{t:t+L-1} \in \mathbb{R}^{L\times d_n}$ denote a window of neural activity over L time steps, where d_n is the number of neural features in a session. Let $B_{t:t+L-1} \in \mathbb{R}^{L\times d_b}$ denote the aligned behavioral covariates on the same temporal support, where d_b is task-dependent. We consider two behavioral representations: two-dimensional position as a canonical spatial variable, and three-dimensional pose as a richer description of ongoing behavior. The selected representation defines B, serving as the target for decoding or the input for encoding. All tasks are constructed from continuous freely moving recordings rather than from pre-segmented trials.

FreeCAB defines two complementary prediction directions for each behavioral representation. In the decoding direction, a model predicts behavior from neural activity:

$$\hat{B}_{t:t+L-1} = f_\theta(X_{t:t+L-1}), \tag{1}$$

where $\hat{B}_{t:t+L-1}$ denotes the predicted behavioral window. Depending on the task, $B_{t:t+L-1}$ is the aligned two-dimensional position trajectory or three-dimensional pose sequence over the same window.

In the encoding direction, a model predicts neural activity from behavioral inputs:

$$\hat{X}_{t:t+L-1} = g_\phi(B_{t:t+L-1}), \tag{2}$$

where $\hat{X}_{t:t+L-1}$ denotes the predicted neural activity window. This direction asks how well neural activity can be predicted from the selected behavioral representation under the same temporal windowing and evaluation protocol.

The two directions are evaluated under the same windowed data interface and metric set, with results reported separately for position and 3D pose. This provides a common evaluation protocol while preserving the continuous structure of free behavior. Rather than relying on externally segmented trials or repeated task events, FreeCAB samples windows from ongoing neural–behavioral activity.

This continuous regression formulation matches freely moving neural–behavioral data, where position, pose, and neural activity evolve continuously rather than as discrete trial labels.

4.2 Evaluation Protocol

FreeCAB uses two complementary evaluation settings. In the single-session setting, 2D-position and 3D-pose tasks are trained and evaluated independently within each session using chronological train, validation, and test splits. This provides a matched-session reference in which training and test data share the same recording context and neural population while remaining temporally separated.

For cross-session adaptation, evaluated for position decoding, models are pretrained on held-in sessions and adapted separately to each held-out session. We compare Pretrain+FT against a scratch model trained on the same target-session split and evaluate both on the same held-out test split. All splits are chronological without temporal shuffling, and model selection uses validation performance. Detailed split definitions, aggregation rules, and statistical reporting are provided in Appendix C.

4.3 Research Questions

FreeCAB is designed not only as an evaluation suite, but also as a benchmark for analyzing neural–behavior modeling under continuous freely moving conditions. We first establish a matched single-session reference for neural encoding and decoding, which provides a common basis for the analyses below. We then investigate four complementary questions concerning behavioral representation, behavioral state, session heterogeneity, and unified modeling.

RQ1: How does behavioral representation affect neural encoding and decoding? Two-dimensional position provides a canonical spatial description of hippocampal behavior, whereas 3D pose captures richer body configurations during free movement. We therefore ask whether neural activity supports prediction of this richer representation and whether richer behavioral inputs improve neural prediction. We evaluate neural-to-pose decoding and compare position-to-neural with pose-to-neural encoding under matched single-session settings.

RQ2: Does position decoding difficulty depend on behavioral state? Freely moving behavior spans substantially different locomotor regimes, which may not be equally represented by a single aggregate decoding score. We therefore stratify test windows into stationary/low-mobility and locomotion regimes and evaluate whether position decoding performance changes systematically across behavioral states.

RQ3: Does multi-session learning support cross-session adaptation under session heterogeneity? Freely moving calcium-imaging sessions vary in neural population size, recording quality, and behavioral coverage, making cross-session transfer nontrivial. We compare pretrain+fine-tune with scratch adaptation on the same held-out sessions to test whether multi-session learning captures transferable neural–behavioral structure rather than only session-specific mappings.

RQ4: Can joint encoding–decoding training benefit unified neural–behavioral modeling? Recent neural modeling approaches increasingly seek shared representations across tasks and modalities. Because FreeCAB evaluates encoding and decoding under the same aligned neural–behavioral interface, we ask whether jointly optimizing both directions can exploit complementary structure and improve over direction-specific training.

4.4 BASELINES

We evaluate FreeCAB with both general-purpose sequence models and neural-domain baselines under the continuous regression formulation. The general-purpose baselines include MLP, DLinear Zeng et al. (2023), TCN Bai et al. (2018), Transformer Vaswani et al. (2017), and PatchTST Nie et al. (2022). These models provide a stable benchmark backbone spanning feed-forward, lightweight linear, convolutional, attention-based, and patch-based sequence modeling.

To complement them, we include neural-domain baselines developed for neural decoding, representation learning, or joint neural-behavior modeling, including LFADS, NeuroFormer, CEBRA, NDT2, POYO+, and NEDS Sussillo et al. (2016); Pandarinath et al. (2018); Ye et al. (2023); Antoniades et al. (2023); Azabou et al. (2025); Zhang et al. (2025). These methods introduce stronger domain-specific inductive biases and allow us to test whether neural-specialized architectures offer advantages over general-purpose sequence models, particularly under session heterogeneity and bidirectional neural-behavior modeling. All baselines use the same benchmark data interface and evaluation metrics. The five general-purpose baselines additionally share the matched supervised regression protocol, whereas neural-domain baselines retain method-specific objectives or adaptations when required. Detailed model descriptions and implementation settings are provided in Appendix D.

Reproducibility. Anonymized review access to the processed benchmark data, predefined splits, session inventory, and evaluation code is provided with the submission.

5 DISCOVERY

We organize the empirical analysis around the benchmark questions introduced above: Section 5.1 establishes a single-session reference for encoding–decoding difficulty, Section 5.2 evaluates behavioral-state-conditioned position decoding, Section 5.3 tests cross-session adaptation from multi-session training, and Section 5.4 analyzes joint encoding–decoding training. As summarized in Figure 3, performance depends on both behavioral representation and modeling direction: 3D pose yields higher encoding correlation but is more difficult to decode than 2D position; position decoding improves during locomotion; and multi-session pretraining yields model- and metric-dependent effects rather than uniformly reliable transfer.

5.1 SINGLE-SESSION BENCHMARK ACROSS BEHAVIORAL REPRESENTATIONS

Table 1 summarizes matched single-session results for 2D position and 3D pose using the same five general-purpose baselines in both encoding and decoding directions. Because the prediction targets differ in dimensionality and units, cross-direction and cross-representation comparisons focus primarily on R^2 and Pearson correlation, while MAE and MSE are interpreted within each task.

For 2D position, decoding is consistently more reliable than encoding: all five baselines achieve positive R^2 and moderate Pearson correlation for decoding, whereas position-to-neural encoding remains weak, with negative R^2 and low correlation. This reflects the greater difficulty of predicting high-dimensional neural activity from limited behavioral covariates.

Behavioral representation further affects performance. Using 3D pose as the encoding input yields higher Pearson correlation than 2D position for all five baselines, although R^2 remains negative and does not improve consistently. Conversely, 3D pose is decodable from neural activity but is consistently harder than 2D position, with lower R^2 and Pearson correlation. Together, these results show that single-session performance depends on both modeling direction and behavioral representation. Additional neural-domain references and per-session results are provided in Appendix E.1.

5.2 LOCOMOTION FACILITATES POSITION DECODING

We next examine whether position-decoding difficulty depends on behavioral state. Test windows are stratified by mean speed, using 3.0 cm/s as the main threshold between low-mobility and locomotion periods. The threshold construction and window counts are described in Appendix B.2, with additional session-level composition and threshold-sensitivity analyses reported in Appendix E.2.

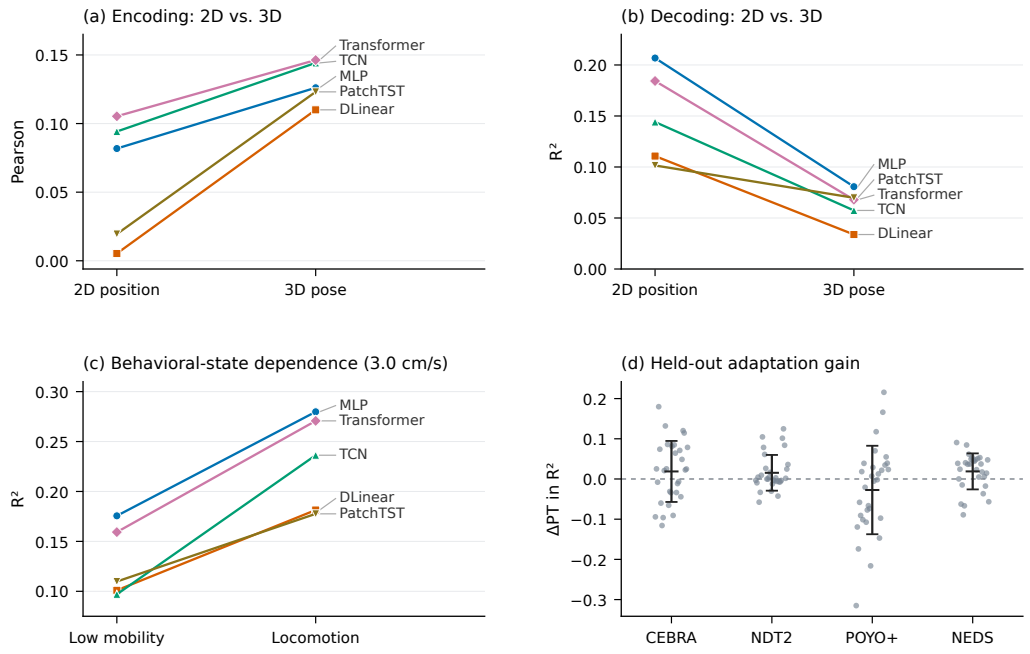

Figure 3: Summary of the main empirical findings in FreeCAB. (a) In single-session neural encoding, 3D pose yields higher Pearson correlation than 2D position across all five general-purpose baselines. (b) Both behavioral representations are decodable from neural activity, while 3D-pose decoding is more difficult than 2D-position decoding. (c) Position decoding improves consistently during locomotion relative to low-mobility periods. (d) Held-out adaptation gains from multi-session pretraining are model-dependent, where Δ_{PT} denotes Pretrain+FT minus Scratch performance in R^2.

Table 1: Matched single-session encoding and decoding performance for 2D position and 3D pose. Results are reported for five general-purpose baselines evaluated on all 58 retained sessions with three random seeds. Values are means ± standard deviations across seeds after averaging over sessions within each seed. Bold and underlined values indicate the best and second-best results, respectively, within each behavioral representation and metric.

Method	Encoding				Decoding			
	MAE ↓	MSE ↓	R^2 ↑	Pearson ↑	MAE ↓	MSE ↓	R^2 ↑	Pearson ↑
2D position								
MLP	**0.4411 ± 0.0002**	1.1203 ± 0.0003	−0.1488 ± 0.0004	0.0818 ± 0.0011	0.7186 ± 0.0016	**0.7694 ± 0.0049**	**0.2068 ± 0.0043**	0.4919 ± 0.0025
DLinear	0.4981 ± 0.0040	1.2004 ± 0.0054	−0.6178 ± 0.0592	0.0053 ± 0.0002	0.7710 ± 0.0060	0.8639 ± 0.0128	0.1107 ± 0.0133	0.3965 ± 0.0065
TCN	0.4461 ± 0.0015	1.1225 ± 0.0003	−0.2057 ± 0.0030	0.0942 ± 0.0010	0.7340 ± 0.0024	0.8259 ± 0.0086	0.1441 ± 0.0084	0.4680 ± 0.0038
Transformer	0.4454 ± 0.0010	**1.1193 ± 0.0004**	−0.2050 ± 0.0020	**0.1053 ± 0.0028**	**0.7122 ± 0.0010**	0.7838 ± 0.0011	0.1843 ± 0.0010	**0.5040 ± 0.0039**
PatchTST	0.4474 ± 0.0004	1.1281 ± 0.0020	**−0.1358 ± 0.0119**	0.0195 ± 0.0030	0.7645 ± 0.0014	0.8623 ± 0.0048	0.1015 ± 0.0058	0.4246 ± 0.0056
3D pose								
MLP	**0.4405 ± 0.0003**	**1.1123 ± 0.0014**	**−0.1839 ± 0.0121**	0.1261 ± 0.0009	0.7340 ± 0.0017	**1.0018 ± 0.0039**	**0.0807 ± 0.0039**	0.3601 ± 0.0011
DLinear	0.4557 ± 0.0004	1.1228 ± 0.0005	−0.2426 ± 0.0218	0.1100 ± 0.0033	0.7593 ± 0.0013	1.0396 ± 0.0020	0.0338 ± 0.0014	0.3249 ± 0.0039
TCN	0.4514 ± 0.0005	1.1152 ± 0.0002	−0.3094 ± 0.0103	0.1442 ± 0.0010	0.7321 ± 0.0011	1.0132 ± 0.0040	0.0574 ± 0.0040	0.3833 ± 0.0034
Transformer	0.4563 ± 0.0007	1.1163 ± 0.0008	−0.2742 ± 0.0047	**0.1463 ± 0.0009**	**0.7230 ± 0.0023**	1.0052 ± 0.0062	0.0679 ± 0.0067	**0.3955 ± 0.0014**
PatchTST	0.4503 ± 0.0012	1.1180 ± 0.0005	−0.2346 ± 0.0237	0.1231 ± 0.0052	0.7401 ± 0.0024	1.0046 ± 0.0057	0.0698 ± 0.0079	0.3737 ± 0.0025

As shown in Figure 3c and Table 2, all five general-purpose baselines achieve higher R^2 and Pearson correlation during locomotion, with positive locomotion-minus-low-mobility differences across models. The same direction is preserved under the alternative 2.0 and 4.5 cm/s thresholds, indicating that the effect is not specific to the main cutoff. These results show that aggregate decoding performance can mask substantial behavioral-state dependence.

Table 2: Behavioral-state-conditioned position decoding under the main threshold of 3.0 cm/s. Results are reported as mean ± std over 3 seeds. Δ denotes locomotion minus stationary/low-mobility performance.

Method	R^2 ↑			Pearson ↑		
	Stationary	Locomotion	Δ	Stationary	Locomotion	Δ
MLP	0.1756 ± 0.0058	0.2798 ± 0.0030	$+0.1042 \pm 0.0032$	0.4369 ± 0.0045	0.5300 ± 0.0026	$+0.0931 \pm 0.0021$
DLinear	0.1010 ± 0.0066	0.1814 ± 0.0125	$+0.0804 \pm 0.0061$	0.3692 ± 0.0061	0.4342 ± 0.0114	$+0.0651 \pm 0.0058$
TCN	0.0970 ± 0.0123	0.2363 ± 0.0079	$+0.1393 \pm 0.0082$	0.4044 ± 0.0085	0.5078 ± 0.0049	$+0.1034 \pm 0.0062$
Transformer	0.1593 ± 0.0043	0.2707 ± 0.0027	$+0.1114 \pm 0.0049$	0.4564 ± 0.0011	0.5402 ± 0.0013	$+0.0838 \pm 0.0020$
PatchTST	0.1098 ± 0.0098	0.1776 ± 0.0035	$+0.0677 \pm 0.0111$	0.4013 ± 0.0059	0.4526 ± 0.0033	$+0.0513 \pm 0.0089$

5.3 Held-out Adaptation Exhibits High Transfer Uncertainty

We next evaluate whether multi-session pretraining improves adaptation to unseen recording sessions. Figure 3d summarizes the adaptation gain Δ_{PT}, defined as Pretrain+FT minus Scratch performance on the same held-out session splits. Full aggregate and per-session results are provided in Appendix E.3.

The results show model- and metric-dependent transfer rather than a uniform pretraining benefit. Adaptation effects vary in both magnitude and sign across baselines, and improvements in R^2 do not necessarily coincide with improvements in Pearson correlation. Moreover, the variability of Δ_{PT} is often comparable to or larger than its mean effect, indicating substantial uncertainty across held-out adaptation runs.

Thus, under the evaluated baselines, multi-session pretraining does not yet provide uniformly reliable cross-session gains in freely moving calcium-imaging data. In FreeCAB, held-out adaptation therefore serves as a stress test for transferable neural–behavioral representations rather than a simple larger-data setting.

5.4 Joint Modeling Provides Task-Dependent Benefit

We finally evaluate whether encoding and decoding benefit from a unified bidirectional objective. Joint optimization does not provide a uniform advantage across tasks: decoding performance generally decreases under joint training, whereas the effect on encoding is smaller and varies across evaluation settings and metrics. Detailed comparisons between direction-specific and joint NEDS training are provided in Appendix E.4.

The training dynamics provide one possible explanation. Encoding and decoding differ substantially in difficulty and converge at different rates, creating a mismatch when both objectives are optimized simultaneously. A shared training trajectory and model-selection criterion may therefore favor one direction at the expense of the other, leading to objective interference rather than complementary learning. Appendix E.4 also provides the corresponding training and validation loss analyses.

6 Conclusions

We introduced FreeCAB, a benchmark for bidirectional neural–behavior modeling from continuous freely moving hippocampal calcium imaging. FreeCAB pairs dorsal CA1 miniscope activity with synchronized multi-view behavior and evaluates 2D position and 3D pose under a matched single-session protocol, with position-based analyses of behavioral state, cross-session adaptation, and joint training. Across matched baselines, 3D pose yields higher neural-encoding correlation but is more difficult to decode than 2D position; position decoding improves during locomotion, while multi-session pretraining and joint training provide model-, metric-, and task-dependent effects. Together, these results show that neural–behavior modeling under free behavior depends jointly on behavioral representation, behavioral state, session heterogeneity, and modeling direction.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and polishing the clarity and readability of parts of the manuscript. All AI-assisted edits were reviewed and revised by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

All animal procedures and data collection were approved by the relevant institutional animal ethics committee and were conducted in accordance with applicable institutional and ethical guidelines. Identifying approval details are withheld during double-blind review.

REFERENCES

Daniel Aharoni and Tycho M Hoogland. Circuit investigations with open-source miniaturized microscopes: past, present and future. *Frontiers in cellular neuroscience*, 13:141, 2019.

Antonis Antoniades, Yiyi Yu, Joseph Canzano, William Wang, and Spencer LaVere Smith. Neuroformer: Multimodal and multitask generative pretraining for brain data. *arXiv preprint arXiv:2311.00136*, 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.

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.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Emery N Brown, Loren M Frank, Dengda Tang, Michael C Quirk, and Matthew A Wilson. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *Journal of Neuroscience*, 18(18):7411–7425, 1998.

Emery N Brown, Robert E Kass, and Partha P Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature neuroscience*, 7(5):456–461, 2004.

Geeling Chau, Christopher Wang, Sabera Talukder, Vighnesh Subramaniam, Saraswati Soedarmadji, Yisong Yue, Boris Katz, and Andrei Barbu. Population transformer: Learning population-level representations of neural activity. *ArXiv*, pp. arXiv–2406, 2025.

John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500–1509, 2014.

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.

Caroline Geisler, David Robbe, Michaël Zugaro, Anton Sirota, and György Buzsáki. Hippocampal place cell assemblies are speed-controlled oscillators. *Proceedings of the National Academy of Sciences*, 104(19):8149–8154, 2007.

Kunal K Ghosh, Laurie D Burns, Eric D Cocker, Axel Nimmerjahn, Yaniv Ziv, Abbas El Gamal, and Mark J Schnitzer. Miniaturized integration of a fluorescence microscope. *Nature methods*, 8 (10):871–878, 2011.

Alessandro T Gifford, Domenic Bersch, Marie St-Laurent, Basile Pinsard, Julie Boyle, Lune Bellec, Aude Oliva, Gemma Roig, and Radoslaw M Cichy. The algonauts project 2025 challenge: How the human brain makes sense of multimodal movies. *arXiv preprint arXiv:2501.00504*, 2024.

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.

Vinay Jayaram and Alexandre Barachant. Moabb: trustworthy algorithm benchmarking for bcis. *Journal of neural engineering*, 15(6):066011, 2018.

Ole Jensen and John E Lisman. Position reconstruction from an ensemble of hippocampal place cells: contribution of theta phase coding. *Journal of neurophysiology*, 83(5):2602–2609, 2000.

Jaivardhan Kapoor, Auguste Schulz, Julius Vetter, Felix Pei, Richard Gao, and Jakob H Macke. Latent diffusion for neural spiking data. *Advances in Neural Information Processing Systems*, 37: 118119–118154, 2024.

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.

Nathaniel R Kinsky, David W Sullivan, William Mau, Michael E Hasselmo, and Howard B Eichenbaum. Hippocampal place fields maintain a coherent and flexible map across long timescales. *Current Biology*, 28(22):3578–3588, 2018.

Jan-Matthis Lueckmann, Alexander Immer, Alex Bo-Yuan Chen, Peter H Li, Mariela D Petkova, Nirmala A Iyer, Luuk Willem Hesselink, Aparna Dev, Gudrun Ihrke, Woohyun Park, et al. Zapbench: A benchmark for whole-brain activity prediction in zebrafish. *arXiv preprint arXiv:2503.02618*, 2025.

Kathryn McClain, David Tingley, David J Heeger, and György Buzsáki. Position–theta-phase model of hippocampal place cell activity applied to quantification of running speed modulation of firing rate. *Proceedings of the National Academy of Sciences*, 116(52):27035–27042, 2019.

Willard L McFarland, Herman Teitelbaum, and Elizabeth K Hedges. Relationship between hippocampal theta activity and running speed in the rat. *Journal of comparative and physiological psychology*, 88(1):324, 1975.

Robert U Muller and John L Kubie. The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. *Journal of Neuroscience*, 7(7):1951–1968, 1987.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 1971.

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.

Liam Paninski, Jonathan Pillow, and Jeremy Lewi. Statistical models for neural encoding, decoding, and optimal stimulus design. *Progress in brain research*, 165:493–507, 2007.

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.

Jonathan W Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, EJ Chichilnisky, and Eero P Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

Peter Rupprecht, Stefano Carta, Adrian Hoffmann, Mayumi Echizen, Antonin Blot, Alex C Kwan, Yang Dan, Sonja B Hofer, Kazuo Kitamura, Fritjof Helmchen, et al. A database and deep learning toolbox for noise-optimized, generalized spike inference from calcium imaging. *Nature neuroscience*, 24(9):1324–1337, 2021.

Shreya Saxena and John P Cunningham. Towards the neural population doctrine. *Current opinion in neurobiology*, 55:103–111, 2019.

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

Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, et al. Brain-score: Which artificial neural network for object recognition is most brain-like? *BioRxiv*, pp. 407007, 2018.

Andrew R Sedler and Chethan Pandarinath. lfads-torch: A modular and extensible implementation of latent factor analysis via dynamical systems. *arXiv preprint arXiv:2309.01230*, 2023.

Liron Sheintuch, Alon Rubin, Noa Brande-Eilat, Nitzan Geva, Noa Sadeh, Or Pinchasof, and Yaniv Ziv. Tracking the same neurons across multiple days in ca2+ imaging data. *Cell reports*, 21(4): 1102–1115, 2017.

David Sussillo, Rafal Jozefowicz, LF Abbott, and Chethan Pandarinath. Lfads-latent factor analysis via dynamical systems. *arXiv preprint arXiv:1608.06315*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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, 2025.

Konstantin F Willeke, Paul G Fahey, Mohammad Bashiri, Laura Pede, Max F Burg, Christoph Blessing, Santiago A Cadena, Zhiwei Ding, Konstantin-Klemens Lurz, Kayla Ponder, et al. The sensorium competition on predicting large-scale mouse primary visual cortex activity. *arXiv preprint arXiv:2206.08666*, 2022.

Konstantin Friedrich Willeke, Polina Turishcheva, Alex Gilbert, Goirik Chakrabarty, Hasan Atakan Bedel, Paul G. Fahey, Yongrong Qiu, Marissa A. Weis, Michaela Vystrčilová, Taliah Muhammad, Lydia Ntanavara, Rachel E Froebe, Kayla Ponder, Zheng Huan Tan, Emin Orhan, Erick Cobos, Sophia Sanborn, Katrin Franke, Fabian H. Sinz, Alexander S. Ecker, and Andreas S. Tolias. Omnimouse: Scaling properties of multi-modal, multi-task brain models on 150b neural tokens. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mEw4lhAn0F.

Matthew A Wilson and Bruce L McNaughton. Dynamics of the hippocampal ensemble code for space. *Science*, 261(5124):1055–1058, 1993.

Hannah S Wirtshafter and John F Disterhoft. In vivo multi-day calcium imaging of ca1 hippocampus in freely moving rats reveals a high preponderance of place cells with consistent place fields. *Journal of Neuroscience*, 42(22):4538–4554, 2022.

Teng Xu, Taotao Zhou, Youjia Wang, Peng Yang, Simin Tang, Kuixiang Shao, Zifeng Tang, Yifei Liu, Xinyuan Chen, Hongshuang Wang, et al. Mousegpt: A large-scale vision-language model for mouse behavior analysis. *arXiv preprint arXiv:2503.10212*, 2025.

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*, 36:80352–80374, 2023.

Ke Yi, Yansen Wang, Kan Ren, and Dongsheng Li. Learning topology-agnostic eeg representations with geometry-aware modeling. *Advances in Neural Information Processing Systems*, 36:53875–53891, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Kechen Zhang, Iris Ginzburg, Bruce L McNaughton, and Terrence J Sejnowski. Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. *Journal of neurophysiology*, 79(2):1017–1044, 1998.

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.

Pengcheng Zhou, Shanna L Resendez, Jose Rodriguez-Romaguera, Jessica C Jimenez, Shay Q Neufeld, Andrea Giovannucci, Johannes Friedrich, Eftychios A Pnevmatikakis, Garret D Stuber, Rene Hen, et al. Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data. *elife*, 7:e28728, 2018.

Yaniv Ziv, Laurie D Burns, Eric D Cocker, Elizabeth O Hamel, Kunal K Ghosh, Lacey J Kitch, Abbas El Gamal, and Mark J Schnitzer. Long-term dynamics of ca1 hippocampal place codes. *Nature neuroscience*, 16(3):264–266, 2013.

A DATA DETAILS

A.1 DETAILED DATA ACQUISITION

We used a synchronized neural-behavioral recording setup combining multi-view behavioral video acquisition with single-photon calcium imaging. Behavioral videos were acquired using an '8' - camera synchronized platform, consisting of '7' surrounding cameras and '1' overhead camera. The system used 'Z-Cam E2' cameras equipped with 'Olympus 14-42mm f/3.5-5.6 EZ' lenses, and videos were recorded at '4K' resolution and '30 FPS' in a circular open-field arena with a diameter of approximately '25 cm'.

Simultaneous calcium imaging was performed using 'Miniscope v3' in hippocampal 'dorsal CA1' at '30 Hz'. Neural activity was labeled via viral expression of 'GCaMP6f', and recordings were conducted after post-surgical recovery. Each recording session lasted approximately '15 min', and sessions were initiated at approximately the same time each day to reduce time-of-day variability. The full raw acquisition covered '19' mice, and each animal was, in principle, recorded for at least '7' days.

After acquisition, behavioral and neural signals entered separate preprocessing pipelines. On the neural side, standard miniscope preprocessing was applied to obtain denoised, registered, and 'CNMFE'-extracted neural activity matrices Zhou et al. (2018). These calcium traces were further processed with 'CASCADE' Rupprecht et al. (2021) to obtain deconvolved event estimates used in downstream analysis. On the behavioral side, synchronized multi-view videos were used to derive 2D position, 3D pose trajectories, and motion-related variables. At a high level, this behavioral processing followed the same experimental setup as in MouseGPT Xu et al. (2025); specifically, animal keypoints were detected using a YOLOv5-based keypoint model matching the one used there, and the resulting multi-view observations were used for 2D position estimation, 3D pose reconstruction, and downstream motion attribute extraction.

A.2 ALIGNMENT ACROSS RECORDING SYSTEMS

The miniscope system and the external multi-camera behavioral system were recorded independently and did not share a common timestamp stream. We therefore performed an explicit post hoc alignment procedure to place neural activity and behavioral trajectories onto a common session-level time base.

Stage 1: Intra-miniscope alignment. Within the miniscope system, frames from the calcium-imaging camera were first aligned to the miniscope-associated behavioral camera using the raw miniscope timestamp table. This step established a frame-level correspondence between neural frames and the miniscope behavioral stream.

Stage 2: Cross-system synchronization. We then identified synchronized clip intervals between the miniscope behavioral stream and the external multi-camera behavioral system using an external synchronization workflow. This step provided the temporal correspondence needed to relate the miniscope-side behavioral stream to the independently recorded 8-camera system.

Stage 3: Unified frame mapping and data extraction. The two mappings above were combined to construct a unified frame correspondence across the external behavioral video, the miniscope behavioral camera, and the miniscope calcium camera. Using this mapping, we extracted aligned behavioral trajectories and aligned neural signals for each session. Behavioral trajectories were resampled onto the aligned time base to produce session-level 2D position and 3D pose sequences, while neural activity was sampled using the corresponding calcium-camera frame indices. These aligned session representations were then used for benchmark task construction.

At the session level, the synchronized intervals extracted from the two recording systems were not always exactly identical in length, and residual mismatches of less than 1s can occur. Rather than assuming perfect raw synchronization, we treated alignment as an explicit frame-mapping problem and retained only sessions that satisfied downstream preprocessing completeness and task-viability criteria. This step is important because synchronization quality directly affects the validity of neural-behavior correspondence in downstream decoding evaluation.

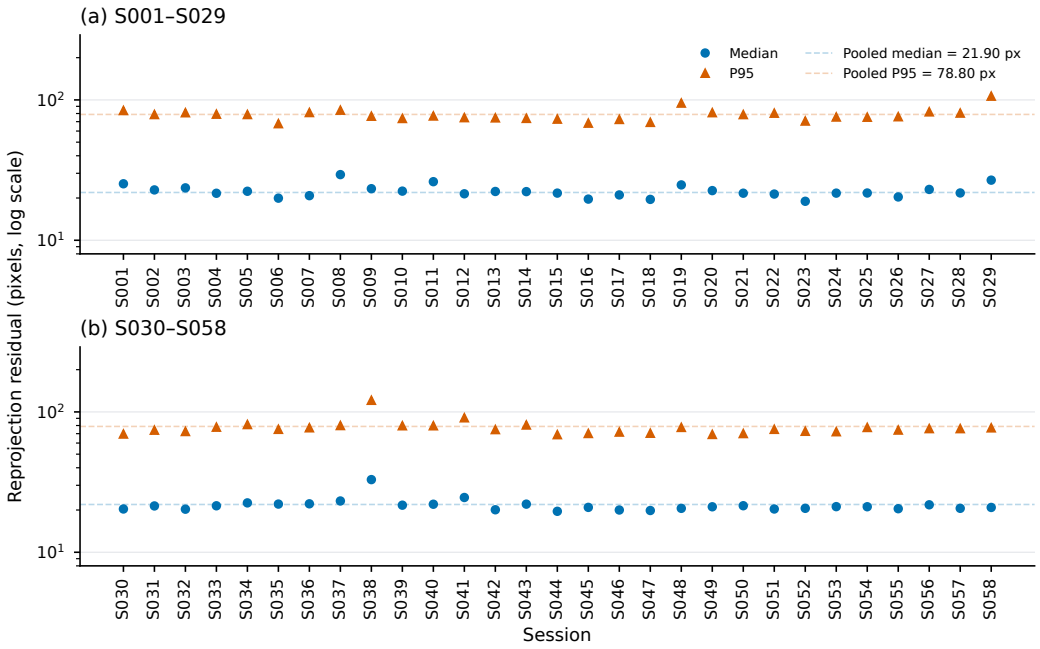

Figure 4: Session-level tracking quality measured by reprojection residuals. Markers show the median and P95 for each of the 58 retained sessions on a logarithmic scale. Across 1,235,835 frames and 236,958,675 confidence-eligible observations from available camera views, the pooled median is 21.90 px (IQR: 12.61–36.52; P95: 78.80). Residuals quantify multi-view consistency with 2D detections rather than error against independent 3D ground truth; synthesized missing-view coordinates are excluded from QC statistics.

A.3 Behavioral Tracking Quality Control

Before quality assessment, reconstructed 3D trajectories are smoothed using a Savitzky–Golay filter with a 15-frame window and a fifth-order polynomial. Smoothing is applied independently within contiguous valid segments of each data split and never crosses missing observations or train/validation/test boundaries.

Tracking consistency is evaluated by reprojecting the final smoothed 3D keypoints into the original camera views and measuring pixel-space residuals relative to 2D detections with confidence ≥ 0.1. We include all confidence-eligible observations rather than only observations retained as triangulation inliers, providing a direct measure of consistency between the final reconstructed trajectories and the underlying multi-view detections.

The tracking-quality analysis covers all 58 retained sessions, comprising 1,235,835 frames and 236,958,675 confidence-eligible observations across available camera views. The pooled reprojection residual has a median of 21.90 pixels, an interquartile range of 12.61–36.52 pixels, and a 95th percentile of 78.80 pixels. Figure 4 summarizes the session-level medians and 95th percentiles. These residuals measure multi-view detection consistency rather than absolute 3D reconstruction error, since independent 3D ground-truth trajectories are not available. Missing-view coordinates synthesized by projection are excluded from all QC statistics.

A.4 Filtering and Session Selection

Session inclusion was treated as a benchmark curation problem rather than as a model-selection problem. After preprocessing and cross-system alignment, sessions entered a quality-control stage designed to determine whether they supported stable and interpretable neural-behavior evaluation under the benchmark protocol. This quality-control stage considered three factors: behavioral ade-

quacy, usable-segment stability, and neural viability. Algorithm 1 summarizes the overall session-level quality-control and usable-prefix selection procedure.

To assess behavioral adequacy, we summarized arena sampling from the aligned position trajectory using an occupancy-based coverage distribution. After mapping the circular arena coordinates into a square occupancy representation, we partitioned the resulting two-dimensional support into spatial cells $\{B_i\}_{i=1}^{K}$ and defined

$$p_i = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}\{x_t \in B_i\}. \tag{3}$$

where x_t denotes the position at time t and T is the session length. We then computed the occupancy entropy

$$H_{\mathrm{occ}} = -\sum_{i=1}^{K} p_i \log p_i, \tag{4}$$

which increases when arena sampling is broader and more uniform. In the benchmark, H_{occ} was used as an informative quality-control signal rather than as a standalone inclusion rule.

To assess usable-segment stability, we compared the occupancy distribution of the early usable portion of a session with that of the later portion. Let $p^{(E)}$ and $p^{(L)}$ denote the occupancy distributions estimated from the early and late segments, respectively. We summarize their discrepancy using the Jensen–Shannon divergence

$$\begin{aligned} D_{\mathrm{JS}}\left(p^{(E)} \,\|\, p^{(L)}\right) &= \frac{1}{2} D_{\mathrm{KL}}\left(p^{(E)} \,\|\, m\right) + \frac{1}{2} D_{\mathrm{KL}}\left(p^{(L)} \,\|\, m\right), \\ m &= \frac{1}{2}\left(p^{(E)} + p^{(L)}\right), \end{aligned} \tag{5}$$

where larger values indicate stronger within-session behavioral drift. For some retained sessions, the later portion no longer remained behaviorally consistent with the earlier usable portion of the same recording, for example because of reduced spatial coverage, prolonged low-mobility periods, or stronger behavioral drift. In such cases, we retained only a fixed usable prefix rather than the full session. Once assigned during curation, `prefix_ratio` was fixed in the benchmark inventory and applied during preprocessing, so that the released benchmark sessions already correspond to the selected usable segment.

Algorithm 1 Session-level quality control and usable-prefix selection

Require: Aligned session-level neural-behavior candidates
for each session **do**
  compute occupancy distribution $\{p_i\}_{i=1}^{K}$ and occupancy entropy H_{occ}
  evaluate whether behavioral coverage is sufficient for benchmark use
  split the session into early and late portions and compute $D_{\mathrm{JS}}\left(p^{(E)} \,\|\, p^{(L)}\right)$
  evaluate whether the later portion remains behaviorally consistent with the earlier usable portion
  evaluate neural viability using session-level neural quality indicators
  if behavioral adequacy and neural viability are both sufficient, and within-session drift is limited **then**
    retain the full session
  else if the early portion is usable but the later portion shows substantial drift or reduced informativeness **then**
    retain a fixed usable prefix and record its `prefix_ratio`
  else
    exclude the session from the benchmark inventory
  end if
end for

A.5 Final Session Inventory

After applying the filtering and session-selection protocol described above, we define the final benchmark inventory at the session level. Main-text experiments are conducted on curated benchmark pools rather than on the full aligned candidate set, and each retained session is treated as an independent unit of analysis. The resulting public session inventory is summarized in Table 3.

For manuscript presentation and benchmark reporting, retained sessions are referenced only through anonymized benchmark identifiers rather than raw acquisition names. In the current benchmark release, the retained pool contains 58 sessions, which are subsequently partitioned into 48 held-in sessions for pretraining and 10 held-out sessions for final reporting. Each inventory entry records the benchmark mouse identifier, benchmark session identifier, held-in status, fixed usable prefix ratio, and neuron count. These fields make explicit both the split membership of each retained session and key sources of heterogeneity that remain in the benchmark after curation.

Because the benchmark is defined at the session level, animals do not contribute equal numbers of retained sessions. This imbalance is preserved as part of the realistic structure of the dataset rather than removed through additional balancing. The complete retained session list is provided in the appendix for transparency and reproducibility, while the private mapping from benchmark identifiers to raw acquisition records is maintained separately for internal traceability and data management.

Table 3: Public session-level inventory of the retained FreeCAB benchmark pool. Each retained session is reported using anonymized benchmark identifiers, together with its held-in/held-out split membership, fixed usable-prefix ratio, and neuron count.

Mouse ID	Session ID	Held-in	Prefix ratio	Neurons
M01	S001	Yes	0.70	80
M01	S002	Yes	0.80	64
M01	S003	No	0.80	68
M01	S004	Yes	0.90	69
M02	S005	Yes	0.90	60
M03	S006	Yes	0.70	74
M03	S007	Yes	0.60	97
M03	S008	Yes	0.90	60
M03	S009	Yes	0.70	35
M03	S010	Yes	0.70	40
M03	S011	Yes	0.60	60
M03	S012	Yes	0.90	43
M03	S013	Yes	0.90	38
M03	S014	Yes	0.60	83
M03	S015	Yes	0.70	58
M03	S016	Yes	0.80	95
M03	S017	No	0.70	93
M03	S018	Yes	0.90	83
M03	S019	Yes	0.60	85
M04	S020	Yes	0.80	43
M04	S021	No	0.60	58
M04	S022	Yes	0.90	41
M05	S023	Yes	0.80	40
M05	S024	Yes	0.60	82
M05	S025	Yes	0.80	96
M05	S026	Yes	0.80	95
M05	S027	Yes	0.80	77
M05	S028	Yes	0.60	71
M05	S029	No	0.90	64
M06	S030	Yes	0.80	42
M06	S031	Yes	0.80	31
M06	S032	Yes	0.80	56
M06	S033	Yes	0.90	46
M07	S034	Yes	0.80	88
M07	S035	Yes	1.00	32
M07	S036	Yes	0.70	106
M07	S037	Yes	0.70	127
M07	S038	No	0.90	168

Continued on next page

Table 3 continued.

Mouse ID	Session ID	Held-in	Prefix ratio	Neurons
M08	S039	No	0.60	59
M09	S040	Yes	0.90	64
M09	S041	Yes	0.80	116
M10	S042	Yes	0.70	86
M10	S043	No	0.80	104
M10	S044	Yes	1.00	81
M11	S045	No	0.60	93
M11	S046	Yes	1.00	168
M11	S047	Yes	0.90	137
M11	S048	Yes	0.70	161
M12	S049	Yes	0.80	145
M13	S050	Yes	0.60	50
M13	S051	Yes	0.60	49
M13	S052	No	0.90	46
M13	S053	Yes	0.90	96
M13	S054	Yes	0.60	105
M14	S055	No	0.70	64
M14	S056	Yes	0.60	95
M15	S057	Yes	0.60	32
M15	S058	Yes	0.70	36

B Benchmark Task Construction

B.1 Windowed Sample Construction and Normalization

FreeCAB constructs samples from continuous aligned neural–behavioral recordings. For each retained session, let $X_{1:T} \in \mathbb{R}^{T \times d_n}$ denote the aligned neural activity sequence and $B_{1:T} \in \mathbb{R}^{T \times d_b}$ denote the aligned behavioral sequence, following the notation in Section 4.1. The behavioral dimension d_b depends on the task configuration: B denotes either two-dimensional position or three-dimensional pose, which serves as the decoding target or encoding input under the corresponding single-session task. Before sample construction, each retained session is optionally truncated by its fixed usable-prefix ratio determined during session curation. The remaining sequence is split chronologically into train, validation, and test segments using a $70\%/10\%/20\%$ split, without temporal shuffling.

Because the aligned neural and behavioral streams are sampled at 30 Hz, a 30-frame window corresponds to approximately one second. We use a window length of $L = 30$ frames and extract paired windows:

$$(X_{t:t+L-1}, B_{t:t+L-1}), \quad L = 30. \tag{6}$$

The benchmark uses an aligned sequence-to-sequence formulation rather than future forecasting: the prediction window is identical to the input window, with prediction length equal to context length and zero lag. Thus, each 1-second neural window is paired with the corresponding 1-second behavioral window on the same temporal support.

As defined in Section 4.1, decoding estimates $\hat{B}_{t:t+L-1}$ from $X_{t:t+L-1}$, while encoding estimates $\hat{X}_{t:t+L-1}$ from $B_{t:t+L-1}$. Appendix B focuses on implementation details of sample construction, normalization, and metric computation rather than redefining the task objectives.

Continuous variables are normalized using statistics computed from the training split of each session. The same fitted transformation is then applied to the validation and test splits. Neural activity is standardized per session, and behavioral variables are normalized using the same training-split-only rule. This avoids information leakage from validation or test data while reducing scale differences across sessions and variables. Velocity and speed are derived from the aligned position trajectory in physical units before behavioral normalization, and are reported in cm/s for behavioral-state stratification.

B.2 BEHAVIORAL-STATE-CONDITIONED EVALUATION

To evaluate how behavioral state affects position decoding difficulty, we stratify test windows by the animal's average speed within each 1-second window. This stratification is applied only at evaluation time and does not change the training objective, sample construction, or model architecture. For compactness, we refer to below-threshold windows as stationary, although they should be interpreted as stationary or low-mobility windows under the speed-based definition.

As shown in Figure 5, we use 3.0 cm/s as the main threshold and use 2.0 cm/s and 4.5 cm/s as alternative thresholds for sensitivity analysis. The corresponding window counts are summarized in Table 4. Additional held-out session state-composition statistics and threshold sensitivity results are reported in Appendix E.2.

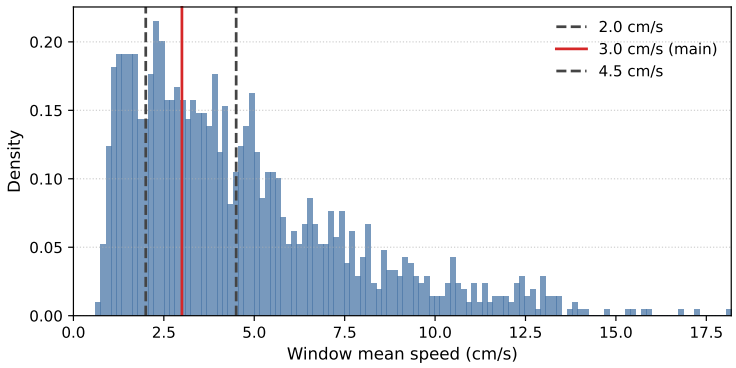

Figure 5: Distribution of window-level speed on held-out test windows. Window speed is computed as the mean speed within each 1-second window. Vertical lines indicate the candidate thresholds used for state-conditioned evaluation, with 3.0 cm/s used as the main threshold.

Table 4: Behavioral-state thresholds used for state-conditioned evaluation. The main analysis uses 3.0 cm/s, which provides a balance between behavioral separation and sample coverage.

Threshold	Stationary windows	Locomotion windows	Locomotion ratio
2.0 cm/s	5,472 (21.3%)	20,268 (78.7%)	78.7%
3.0 cm/s	10,008 (38.9%)	15,732 (61.1%)	61.1%
4.5 cm/s	15,624 (60.7%)	10,116 (39.3%)	39.3%

B.3 EVALUATION METRICS

FreeCAB evaluates continuous prediction tasks using mean absolute error (MAE), mean squared error (MSE), coefficient of determination (R^2), and Pearson correlation. For sequence-to-sequence prediction with batch size N, window length L, and output dimension d, predictions and targets are flattened from $\mathbb{R}^{N\times L\times d}$ into $\mathbb{R}^{M\times d}$, where $M = NL$. Let $Y \in \mathbb{R}^{M\times d}$ denote the target values and $\hat{Y} \in \mathbb{R}^{M\times d}$ denote the corresponding predictions.

$$\mathrm{MAE} = \frac{1}{Md}\sum_{i=1}^{M}\sum_{j=1}^{d}\left|Y_{ij} - \hat{Y}_{ij}\right|. \tag{7}$$

$$\mathrm{MSE} = \frac{1}{Md}\sum_{i=1}^{M}\sum_{j=1}^{d}\left(Y_{ij} - \hat{Y}_{ij}\right)^2. \tag{8}$$

For R^2, we use the standard coefficient of determination:

$$R^2 = 1 - \frac{\sum_{i=1}^{M}\sum_{j=1}^{d}\left(Y_{ij} - \hat{Y}_{ij}\right)^2}{\sum_{i=1}^{M}\sum_{j=1}^{d}\left(Y_{ij} - \bar{Y}\right)^2}, \tag{9}$$

where $\bar{Y}$ denotes the mean of all target values in the evaluated set.

Pearson correlation is computed for each output dimension and then averaged across dimensions:

$$\rho_j = \frac{\sum_{i=1}^{M}\left(Y_{ij} - \bar{Y}_j\right)\left(\hat{Y}_{ij} - \bar{\hat{Y}}_j\right)}{\sqrt{\sum_{i=1}^{M}\left(Y_{ij} - \bar{Y}_j\right)^2}\sqrt{\sum_{i=1}^{M}\left(\hat{Y}_{ij} - \bar{\hat{Y}}_j\right)^2}}, \tag{10}$$

$$\text{Pearson} = \frac{1}{d}\sum_{j=1}^{d}\rho_j. \tag{11}$$

All metrics are computed on the same flattened sequence outputs for all models and evaluation settings.

C Evaluation Protocol

C.1 Held-in / held-out session split

The retained sessions are divided into held-in and held-out pools at the session level. The held-in pool is used for model development and multi-session pretraining, whereas the held-out pool is reserved for final reporting and adaptation-based evaluation. Held-out sessions are not used during pretraining; they are only used through their own train/validation/test splits during single-session or adaptation experiments.

Table 5: Summary of FreeCAB evaluation protocols. The held-in and held-out pools are defined in the session inventory in Appendix A.5.

Protocol	Initial training	Adaptation	Final evaluation
Single-session	Same-session train split	–	Same-session test split
Scratch adaptation	Held-out session train split	–	Same held-out session test split
Pretrain + fine-tune	Held-in session train splits	Held-out session train split	Same held-out session test split

In the single-session protocol, each session is trained and evaluated independently, providing a matched-distribution reference for task difficulty. Scratch adaptation is the held-out-session instance of this protocol and serves as the from-scratch baseline for adaptation. In the pretrain + fine-tune setting, the model is first pretrained on held-in sessions and then fine-tuned on each held-out session before evaluation. The comparison between scratch adaptation and pretrain + fine-tune therefore measures whether multi-session pretraining improves held-out session adaptation.

C.2 Single-session and scratch adaptation

In the single-session protocol, each session is trained and evaluated independently. For a given session, the model is initialized from scratch, trained only on that session's training split, selected using the validation split from the same session, and evaluated on the test split of the same session. No samples from other sessions are used in this protocol. This setting therefore provides a matched-distribution reference for estimating the intrinsic difficulty of the encoding and decoding tasks within a session.

When the single-session protocol is applied to held-out sessions, it also serves as the scratch adaptation baseline. In this setting, the model is trained from random initialization using only the training split of the target held-out session and evaluated on the same session's test split. This baseline answers how well a model can adapt to a new session without any multi-session pretraining, and is used as the reference point for evaluating the benefit of pretrain + fine-tune adaptation.

C.3 PRETRAIN + FINE-TUNE ADAPTATION

The pretrain + fine-tune protocol evaluates whether multi-session pretraining improves adaptation to unseen sessions. In the pretraining stage, a model is trained on the training splits of the held-in sessions to obtain a pretrained checkpoint. Held-out sessions are not used during this stage. In the adaptation stage, the pretrained checkpoint is fine-tuned separately on the training split of each held-out session and then evaluated on the test split of the same held-out session.

We compare pretrain + fine-tune against scratch adaptation using the same held-out session splits. The adaptation gain is defined as

$$\Delta_{\mathrm{PT}} = S_{\mathrm{pretrain+FT}} - S_{\mathrm{scratch}}, \tag{12}$$

where S denotes an evaluation score such as R^2 or Pearson correlation. Positive values indicate that multi-session pretraining improves held-out adaptation, whereas values near zero or below zero indicate limited or negative transfer. This comparison isolates the contribution of the pretrained initialization, since both scratch adaptation and pretrain + fine-tune use the same target-session training and test splits.

C.4 STATISTICAL REPORTING

Unless otherwise stated, supervised training experiments are repeated with three random seeds, $\{0, 7, 42\}$, and results are reported as mean $\pm$ standard deviation across seeds. Each seed corresponds to an independent model initialization and training run under the same fixed chronological data split. For state-conditioned evaluation, the same trained models are evaluated separately on stationary/low-mobility and locomotion windows, and state differences are computed from the corresponding state-conditioned metrics.

For multi-session pretraining, the pretraining stage is performed once per model due to its higher computational cost. When reporting pretrain–fine-tune adaptation, we explicitly distinguish this setting from fully repeated pretraining: variability reflects repeated adaptation runs when applicable, not independent pretraining runs. Scratch adaptation uses the same held-out session splits and serves as the from-random-initialization baseline for evaluating the benefit of the pretrained checkpoint.

D BASELINE DETAILS

D.1 COMPUTE RESOURCES

All experiments were conducted on an internal GPU cluster using NVIDIA TITAN X and L20 GPUs. General-purpose supervised baselines (MLP, DLinear, TCN, Transformer, and PatchTST) were trained on TITAN X GPUs, whereas neural-domain models and multi-session pretraining experiments were run on L20 GPUs due to their higher memory capacity.

A typical single-session training run for general-purpose supervised baselines required approximately 1–3 minutes per session, while neural-domain baselines typically required approximately 5–20 minutes depending on model complexity and training protocol. Multi-session pretraining was substantially more expensive and was therefore performed once per model. Held-out fine-tuning and evaluation were repeated across random seeds when applicable.

In addition to the reported benchmark runs, the project also required preliminary experiments for debugging, implementation verification, hyperparameter tuning, and unsuccessful adaptation attempts. These auxiliary runs are not included in the benchmark tables.

D.2 BASELINES

D.2.1 MLP

Description. We include a multilayer perceptron (MLP) as a simple feedforward baseline for neural decoding and neural encoding. The MLP operates on fixed-length windows without explicit recurrent, convolutional, or attention-based temporal structure, providing a reference for how much performance can be obtained from direct supervised mapping alone.

Implementation. For both encoding and decoding, the MLP baseline uses the benchmark's aligned single-session $30 \rightarrow 30$ regression setting: each sample contains a 30-step neural input window and a matched 30-step target window. The model flattens the neural input and applies a three-layer feedforward network with hidden dimensions 128 and 64, ReLU activations, and dropout 0.3. Training uses Adam with learning rate 10^{-4}, weight decay 10^{-4}, batch size 32, and 100 epochs.

D.2.2 DLINEAR

Description DLinear Zeng et al. (2023) is a lightweight linear forecasting architecture designed for efficient time-series prediction. Despite its simplicity, it has proven competitive in many sequence modeling settings. We include DLinear as a low-complexity temporal baseline to test whether simple linear trend modeling is sufficient for neural-behavior prediction tasks in freely moving calcium imaging data.

Implementation. DLinear is evaluated under the same aligned single-session $30 \rightarrow 30$ regression protocol as the other simple baselines. The model uses a moving-average decomposition with kernel size 25 and a shared linear projection across channels. Training uses Adam with learning rate 10^{-4}, weight decay 10^{-4}, batch size 32, and 100 epochs.

D.2.3 TCN

Description Temporal convolutional networks (TCNs) Bai et al. (2018) provide a convolutional alternative to recurrent and attention-based sequence models for time-series modeling. They are widely used because they can efficiently capture local temporal dependencies while remaining straightforward to optimize. We include TCN as a general-purpose temporal baseline for both neural encoding and neural decoding under the same benchmark-native protocol as the other simple models.

Implementation. The TCN baseline is trained under the same aligned single-session $30 \rightarrow 30$ protocol. It uses hidden dimension 128, 4 temporal blocks, kernel size 3, and dropout 0.2. Optimization uses Adam with learning rate 10^{-4}, weight decay 10^{-4}, batch size 32, and 100 epochs.

D.2.4 TRANSFORMER

Description Transformer-based sequence models Vaswani et al. (2017) provide a flexible architecture for learning long-range temporal dependencies and have become a standard baseline in time-series modeling. We include a generic Transformer as a strong sequence-modeling baseline for both encoding and decoding tasks. Its role in our benchmark is to test whether increased temporal modeling capacity consistently improves performance relative to simpler feedforward or convolutional baselines.

Implementation. The Transformer baseline follows the aligned single-session $30 \rightarrow 30$ regression setting used throughout the benchmark. The model uses embedding dimension 128, 4 attention heads, feedforward dimension 256, 2 encoder layers, dropout 0.1, and sinusoidal positional encoding. Training uses Adam with learning rate 10^{-4}, weight decay 10^{-4}, batch size 32, and 100 epochs.

D.2.5 PATCHTST

Description PatchTST Nie et al. (2022) is a patch-based transformer for time-series forecasting that improves temporal modeling by operating on local tokenized segments rather than raw time steps. We include PatchTST as a strong benchmark-native sequence baseline under the same single-session protocol as the other simple models. In our implementation, PatchTST uses overlapping temporal patches and is evaluated under the same benchmark-side task definitions and metrics as the other simple baselines.

Implementation. PatchTST is evaluated under the same aligned single-session $30 \rightarrow 30$ benchmark protocol. The model uses embedding dimension 128, 4 attention heads, 2 encoder layers, feedforward dimension 256, and dropout 0.2. To better match the short 30-step context window,

we use overlapping temporal patches with patch length 10 and stride 3. Training uses Adam with learning rate 5×10^{-4}, weight decay 10^{-4}, batch size 32, and 100 epochs.

D.2.6 NEUROFORMER

Description. NeuroFormer (Antoniades et al., 2023) is a multimodal and multitask generative framework for modeling neural activity together with aligned behavioral variables. We include NeuroFormer as a neural-domain reference for single-session 2D-position decoding, complementing both general-purpose supervised sequence models and other neural-specific baselines.

Implementation. NeuroFormer is adapted to single-session neural-to-behavior decoding under the aligned $30 \rightarrow 30$ benchmark protocol. The model uses an embedding dimension of 32, one attention head, one layer in each state, state-history, behavior, and self-attention module, and dropout 0.1. Neural activity is binarized using a threshold of 0.5 for tokenization, with the current and previous token sequences each capped at 1024 tokens. For the reported benchmark experiment, the task-specific regression head predicts the two coordinates of 2D position. Training uses AdamW with learning rate 10^{-4}, weight decay 1.0, $\beta = (0.9, 0.95)$, batch size 1.

D.2.7 LFADS

Description. LFADS Sussillo et al. (2016); Pandarinath et al. (2018) is a latent variable model for inferring low-dimensional neural population dynamics from single-trial activity. We include LFADS as a transferred neural-domain baseline for single-session position decoding. Unlike the benchmark-native supervised baselines, LFADS retains a method-specific training objective and native training backend. It should therefore be interpreted as a latent-dynamics representation model followed by a benchmark-side decoder, rather than as a directly supervised end-to-end regressor.

Implementation. Our LFADS implementation is based on the modular `lfads-torch` codebase Sedler & Pandarinath (2023). We evaluate LFADS under the aligned $30 \rightarrow 30$ single-session decoding setting. LFADS is trained on the neural activity windows using its native objective, with factor dimension 64, generator dimension 128, initial-condition dimension 64, controller dimension 4, dropout rate 0.3, and learning rate 10^{-3}. After training, we use the inferred LFADS factors as latent representations and evaluate position decoding with a downstream ridge-regression readout. The ridge regularization coefficient is set to 1.0.

D.2.8 POYO+

Description POYO+ Azabou et al. (2025) is a neural modeling framework designed for multi-session and multi-task decoding across heterogeneous neural recordings. Its formulation is particularly relevant to calcium imaging and other settings in which recordings differ across sessions. We include POYO+ as a strong neural-domain baseline for evaluating whether architectures explicitly designed for heterogeneous neural data provide advantages over general-purpose sequence models in freely moving hippocampal calcium imaging.

Implementation. For calcium-imaging data, our POYO+ experiments use the CaPOYO variant under the aligned single-session $30 \rightarrow 30$ position decoding setting. The reported scratch single-session configuration uses latent dimension 128, depth 6, attention head dimension 64, 8 latents per step, latent step 0.02, feedforward dropout 0.2, linear dropout 0.4, attention dropout 0.0, learning rate 5×10^{-4}, and weight decay 10^{-4}. For transfer experiments, we additionally study a pretrain-then-finetune protocol in which a model first undergoes multi-session pretraining on held-in sessions and is then fine-tuned separately on each held-out session.

D.2.9 NDT2

Description Neural Data Transformer 2 (NDT2) Ye et al. (2023) is a transformer-based neural modeling framework originally developed for large-scale neural sequence modeling and pretraining across contexts. We include NDT2 as a neural-domain transformer baseline because it provides a strong sequence-modeling architecture for population activity and is well suited for testing whether such models remain effective under session heterogeneity and freely moving behavioral variability.

Implementation. Our NDT2 experiments use the benchmark-aligned single-session $30 \rightarrow 30$ regression setting, in which each sample contains a 30-step input window and a matched 30-step target window. The current configuration uses hidden size 128, 4 attention heads, 4 transformer layers, feedforward expansion factor 2.0, dropout 0.1, input dropout 0.05, batch size 8, learning rate 2×10^{-4}, and 10 training epochs. In our implementation, we reuse the upstream NDT2-style SpaceTimeTransformer backbone while keeping data loading, window construction, task semantics, output shaping, and metric computation inside the benchmark framework, so that NDT2 is evaluated under the same benchmark interface as the other baselines.

D.2.10 CEBRA

Description CEBRA Schneider et al. (2023) is a representation-learning framework for joint analysis of neural and behavioral data. It learns latent embeddings that align neural activity with behavioral or experimental variables and has shown strong performance in behavior-related neural analysis. We include CEBRA as a neural-domain baseline that emphasizes learned joint representations rather than purely supervised end-to-end prediction.

Implementation For CEBRA, we evaluate position decoding on aligned $30 \rightarrow 30$ neural-behavior samples. In addition to the single-session setting, we evaluate a transfer setting based on CEBRA's unified-session interface, where the model is first pretrained on held-in sessions and then fine-tuned on each held-out target session.

For single-session training, we use time offset 1, output dimension 8, 100 training iterations, batch size 128, learning rate 5×10^{-4}, and hidden dimension 32. For multi-session pretraining, we reduce the batch size to 16 and use learning rate 3×10^{-5} to improve training stability. During held-out fine-tuning, we keep the batch size at 16 and increase the learning rate to 1×10^{-4} to adapt more quickly to the target session.

D.2.11 NEDS

Description Neural Encoding and Decoding at Scale (NEDS) Zhang et al. (2025) is a unified framework for jointly modeling neural encoding and decoding across large neural datasets. We include NEDS as a highly relevant neural-domain baseline because it directly aligns with the bidirectional evaluation setting of our benchmark. In particular, NEDS provides a reference point for testing whether jointly structured neural-behavior modeling approaches offer advantages over task-specific baselines in freely moving hippocampal calcium imaging.

Implementation We made adaptability modifications to the original model structure of NEDS to match the data modalities of our study. In particular, we adjusted the input and output modality into neural signals and position trajectories in continuous time series. For the loss function, we replace the original Poisson distribution for spike data with MSE and remove the corresponding Poisson hypothesis to better fit the continuous data.

Based on this model , we carry out experiments in three training settings: encoding-only, decoding-only, and joint under the training paradigms of single-session and pretrain-then-finetune. The model structure is consistent across expressions, including: weight decay of 0.0001, batch size of 16, number of transformer layers of 5, and hidden layer dimension of 256. In single-session training, different tasks use different learning rates: 0.00001 for decoding tasks, 0.0005 for encoding tasks, and 0.0001 for joint training. Under the pretrain-then-finetune setting, the learning rate is 2.010^{-5} in the pretraining stage, and the mask ratio is set to 0.3, and in the fine-tuning stage, the learning rate is also 2.010^{-5}, and the mask ratio is reduced to 0.1 to facilitate the model's adaptation on the target session.

E Additional Results

E.1 Additional Single-Session Results

The main text reports the matched single-session comparison of five general-purpose baselines across 2D position and 3D pose. Here, we provide two complementary views of the single-session

results. First, we report additional neural-domain reference models for 2D position. Second, we provide session-level results for the matched general-purpose baselines on the 10 held-out reporting sessions for both behavioral representations.

Neural-domain reference models. Table 6 reports additional single-session 2D-position results for neural-domain baselines that require method-specific adaptations to the FreeCAB interface. These results are provided as complementary references and are not included in the matched 2D–3D representation comparison in the main text.

Table 6: Additional neural-domain reference results for single-session 2D-position encoding and decoding. Dashes indicate directions not evaluated for the corresponding method.

Method	Encoding				Decoding			
	MAE ↓	MSE ↓	R^2 ↑	Pearson ↑	MAE ↓	MSE ↓	R^2 ↑	Pearson ↑
LFADS	–	–	–	–	0.8544 ± 0.0007	1.1709 ± 0.0084	-0.2161 ± 0.0086	0.3400 ± 0.0009
NeuroFormer	–	–	–	–	0.7883 ± 0.0984	0.9089 ± 0.2027	0.0603 ± 0.1568	0.3363 ± 0.1831
NDT2	–	–	–	–	0.8746 ± 0.0010	0.9971 ± 0.0016	-0.0277 ± 0.0023	0.0926 ± 0.0133
CEBRA	–	–	–	–	0.7038 ± 0.0248	0.8323 ± 0.0496	0.1370 ± 0.0499	0.4960 ± 0.0415
POYO+	–	–	–	–	0.7866 ± 0.1436	0.8800 ± 0.2145	0.0923 ± 0.2070	0.2839 ± 0.2636
NEDS	0.4403 ± 0.0031	0.8980 ± 0.0017	-0.0733 ± 0.0136	0.0576 ± 0.0121	0.7286 ± 0.0317	0.7718 ± 0.0489	0.2042 ± 0.0501	0.5070 ± 0.0463

Matched per-session results. Tables 7 and 8 report the session-level 2D-position results for the five general-purpose baselines used in the matched representation comparison. Tables 9 and 10 report the corresponding 3D-pose results. Each entry is formatted as R^2 / Pearson and averaged over three random seeds; seed-level standard deviations are omitted for readability.

Table 7: Per-session single-session 2D-position decoding results on the 10 held-out reporting sessions. Each entry reports mean R^2 / Pearson over three seeds.

Session	MLP	DLinear	TCN	Transformer	PatchTST
S003	0.449 / 0.683	0.410 / 0.661	0.364 / 0.638	0.413 / 0.671	0.296 / 0.619
S017	0.513 / 0.734	0.396 / 0.669	0.546 / 0.767	0.478 / 0.716	0.420 / 0.689
S021	0.309 / 0.579	0.045 / 0.209	0.187 / 0.501	0.209 / 0.549	0.154 / 0.529
S029	0.363 / 0.615	0.272 / 0.568	0.285 / 0.600	0.312 / 0.609	0.256 / 0.561
S038	0.587 / 0.780	0.535 / 0.747	0.542 / 0.741	0.625 / 0.797	0.508 / 0.737
S039	0.102 / 0.342	0.003 / 0.219	0.060 / 0.269	0.032 / 0.285	0.100 / 0.356
S043	0.140 / 0.416	0.040 / 0.327	0.061 / 0.397	0.048 / 0.389	0.088 / 0.391
S045	0.239 / 0.503	0.122 / 0.361	0.219 / 0.522	0.212 / 0.546	0.007 / 0.262
S052	0.191 / 0.472	0.115 / 0.391	0.108 / 0.420	0.160 / 0.475	0.034 / 0.382
S055	0.136 / 0.419	0.040 / 0.244	0.126 / 0.394	0.170 / 0.446	0.090 / 0.339

Table 8: Per-session single-session 2D-position neural encoding results on the 10 held-out reporting sessions. Each entry reports mean R^2 / Pearson over three seeds.

Session	MLP	DLinear	TCN	Transformer	PatchTST
S003	0.008 / 0.129	-0.088 / 0.002	0.009 / 0.133	0.006 / 0.155	-0.012 / 0.022
S017	0.017 / 0.160	-0.105 / 0.006	0.014 / 0.168	0.027 / 0.213	-0.012 / 0.029
S021	-0.029 / 0.045	-0.119 / 0.004	-0.030 / 0.099	-0.032 / 0.105	-0.033 / 0.028
S029	-0.007 / 0.094	-0.071 / 0.001	-0.011 / 0.097	-0.006 / 0.137	-0.014 / 0.050
S038	0.005 / 0.168	-0.137 / 0.003	-0.004 / 0.174	0.006 / 0.183	-0.019 / 0.044
S039	-0.073 / 0.093	-0.201 / 0.004	-0.082 / 0.096	-0.083 / 0.104	-0.089 / 0.021
S043	-0.053 / 0.021	-0.525 / -0.000	-0.089 / 0.047	-0.094 / 0.039	-0.059 / 0.003
S045	-0.062 / 0.076	-0.849 / -0.000	-0.087 / 0.076	-0.106 / 0.076	-0.052 / 0.000
S052	-0.032 / 0.069	-0.367 / 0.002	-0.050 / 0.072	-0.050 / 0.078	-0.042 / 0.012
S055	-0.904 / 0.071	-4.294 / 0.001	-1.340 / 0.070	-1.066 / 0.074	-0.507 / -0.005

Table 9: Per-session single-session 3D-pose decoding results on the 10 held-out reporting sessions. Each entry reports mean R^2 / Pearson over three seeds.

Session	MLP	DLinear	TCN	Transformer	PatchTST
S003	0.2147 / 0.5125	0.2180 / 0.5249	0.1919 / 0.5389	0.2174 / 0.5504	0.2093 / 0.5385
S017	0.3292 / 0.5907	0.1592 / 0.5425	0.3451 / 0.6427	0.2492 / 0.5799	0.2243 / 0.5785
S021	0.1233 / 0.3843	-0.0443 / 0.1035	0.0647 / 0.3553	0.0396 / 0.3801	0.0672 / 0.4018
S029	0.1976 / 0.4662	0.1508 / 0.4467	0.1223 / 0.4764	0.1776 / 0.5075	0.1276 / 0.4121
S038	0.1630 / 0.6076	0.1389 / 0.6188	0.0700 / 0.6073	0.1457 / 0.6388	0.2367 / 0.6393
S039	0.0122 / 0.2286	-0.0074 / 0.2249	0.0131 / 0.2339	-0.0229 / 0.2100	0.0079 / 0.2047
S043	-0.0155 / 0.2817	-0.0684 / 0.2730	-0.0217 / 0.3244	-0.0467 / 0.2903	0.0031 / 0.3207
S045	0.0980 / 0.2964	0.0258 / 0.2187	0.0640 / 0.3530	0.0883 / 0.3533	0.0061 / 0.2043
S052	0.1236 / 0.3688	0.0944 / 0.3400	0.0407 / 0.3425	0.1078 / 0.3741	0.1079 / 0.3683
S055	0.1435 / 0.4008	0.0899 / 0.3555	0.1259 / 0.4027	0.1146 / 0.3938	0.0692 / 0.2612

Table 10: Per-session single-session 3D-pose neural encoding results on the 10 held-out reporting sessions. Each entry reports mean R^2 / Pearson over three seeds.

Session	MLP	DLinear	TCN	Transformer	PatchTST
S003	0.0165 / 0.1899	-0.0126 / 0.1888	0.0034 / 0.2086	-0.0151 / 0.2094	0.0042 / 0.2013
S017	0.0779 / 0.2742	0.0378 / 0.2355	0.0878 / 0.3111	0.0708 / 0.3025	0.0545 / 0.2591
S021	-0.0118 / 0.1611	-0.0552 / 0.1390	-0.0377 / 0.1180	-0.0576 / 0.1768	-0.0300 / 0.0559
S029	0.0220 / 0.1872	-0.0234 / 0.0545	-0.0115 / 0.1170	0.0025 / 0.2144	-0.0147 / 0.0545
S038	0.0346 / 0.2460	0.0097 / 0.1889	0.0198 / 0.2802	0.0401 / 0.2793	0.0238 / 0.2573
S039	-0.1130 / 0.0947	-0.1057 / 0.0923	-0.1089 / 0.1124	-0.1181 / 0.1142	-0.1078 / 0.0799
S043	-0.1438 / 0.0517	-0.1514 / 0.0306	-0.2311 / 0.0613	-0.2241 / 0.0665	-0.1598 / 0.0509
S045	-0.0864 / 0.0725	-0.1334 / 0.0870	-0.1347 / 0.0792	-0.1381 / 0.0849	-0.0934 / 0.0844
S052	-0.0460 / 0.1087	-0.0720 / 0.0910	-0.0648 / 0.1119	-0.0910 / 0.0985	-0.0576 / 0.0807
S055	-2.4738 / 0.1386	-1.4825 / 0.1340	-2.1444 / 0.1651	-1.9169 / 0.1441	-1.4708 / 0.1425

E.2 Stationary / locomotion statistics

We further summarize the behavioral-state composition of the held-out test split used for state-conditioned evaluation. Under the main threshold of 3.0 cm/s, test windows are divided into stationary/low-mobility and locomotion regimes based on their 1-second mean speed. This anal-

ysis is applied only at evaluation time and does not change the training data, model architecture, or optimization objective.

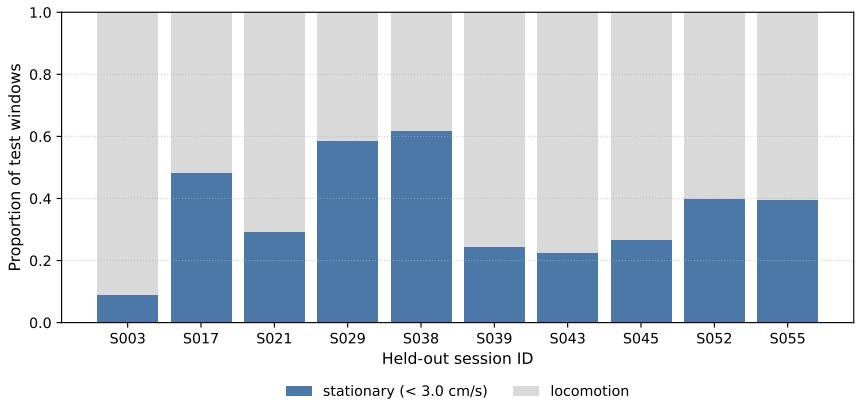

Figure 6: State composition across held-out sessions under the main threshold of 3.0 cm/s. Stationary/low-mobility windows are defined as windows with mean speed below 3.0 cm/s, and locomotion windows as those at or above 3.0 cm/s. The proportion of stationary and locomotion windows varies across held-out sessions, reflecting session-level behavioral variability.

Figure 6 shows that the behavioral-state composition differs across held-out sessions. Some sessions contain mostly locomotion windows, whereas others contain a larger stationary/low-mobility fraction. This session-level variability motivates reporting state-conditioned results together with threshold sensitivity analysis.

Table 11 shows that the locomotion advantage remains positive across all tested thresholds and all general-purpose sequence baselines. The gap is largest under the stricter 2.0 cm/s threshold and smallest under the more permissive 4.5 cm/s threshold, which is expected because higher thresholds include more slow-moving windows in the stationary/low-mobility group. These results indicate that the state-conditioned decoding difference is not specific to the main 3.0 cm/s threshold.

Table 11: Threshold sensitivity of behavioral-state-conditioned position decoding. Values report locomotion minus stationary/low-mobility performance as mean ± std over 3 seeds.

Threshold	Method	$\Delta R^2 \uparrow$	Δ Pearson $\uparrow$
2.0 cm/s	MLP	$+0.1175 \pm 0.0026$	$+0.1041 \pm 0.0027$
2.0 cm/s	DLinear	$+0.0835 \pm 0.0120$	$+0.0702 \pm 0.0100$
2.0 cm/s	TCN	$+0.1641 \pm 0.0102$	$+0.1210 \pm 0.0059$
2.0 cm/s	Transformer	$+0.1254 \pm 0.0189$	$+0.0954 \pm 0.0109$
2.0 cm/s	PatchTST	$+0.0786 \pm 0.0094$	$+0.0616 \pm 0.0049$
3.0 cm/s	MLP	$+0.1042 \pm 0.0032$	$+0.0931 \pm 0.0021$
3.0 cm/s	DLinear	$+0.0804 \pm 0.0061$	$+0.0651 \pm 0.0058$
3.0 cm/s	TCN	$+0.1393 \pm 0.0082$	$+0.1034 \pm 0.0062$
3.0 cm/s	Transformer	$+0.1114 \pm 0.0049$	$+0.0838 \pm 0.0020$
3.0 cm/s	PatchTST	$+0.0677 \pm 0.0111$	$+0.0513 \pm 0.0089$
4.5 cm/s	MLP	$+0.0700 \pm 0.0026$	$+0.0615 \pm 0.0014$
4.5 cm/s	DLinear	$+0.0552 \pm 0.0096$	$+0.0378 \pm 0.0092$
4.5 cm/s	TCN	$+0.0969 \pm 0.0044$	$+0.0670 \pm 0.0023$
4.5 cm/s	Transformer	$+0.0693 \pm 0.0033$	$+0.0499 \pm 0.0042$
4.5 cm/s	PatchTST	$+0.0370 \pm 0.0151$	$+0.0240 \pm 0.0134$

Table 12: Held-out adaptation results for pretrainable neural-domain baselines. Δ_{PT} denotes Pretrain+FT minus Scratch performance for the corresponding metric.

Method	$R^2 \uparrow$			Pearson ↑		
	Scratch	Pretrain+FT	Δ_{PT}	Scratch	Pretrain+FT	Δ_{PT}
CEBRA	0.1370 ± 0.0499	0.1764 ± 0.0394	0.0394 ± 0.0636	0.4960 ± 0.0415	0.4521 ± 0.0453	-0.0439 ± 0.0614
NDT2	-0.0277 ± 0.0023	-0.0123 ± 0.0020	0.0154 ± 0.0036	0.0926 ± 0.0133	0.0948 ± 0.0046	0.0022 ± 0.0122
POYO+	0.0923 ± 0.2070	0.0649 ± 0.1497	-0.0274 ± 0.1082	0.2839 ± 0.2636	0.2376 ± 0.2183	-0.0463 ± 0.1852
NEDS	0.2042 ± 0.0501	0.2095 ± 0.0279	0.0053 ± 0.0573	0.5070 ± 0.0463	0.4716 ± 0.0255	-0.0354 ± 0.0529

E.3 Held-out Adaptation: Pretrain + Fine-tune vs. Scratch

This section provides detailed results for the held-out adaptation analysis summarized in Section 5.3 and Figure 3d. Table 12 reports the aggregate Scratch, Pretrain+FT, and Δ_{PT} results across held-out sessions, while Tables 13 and 14 report the corresponding session-level results in R^2 and Pearson correlation. The session-level results help determine whether aggregate adaptation effects are consistent across recordings or driven by a subset of sessions.

Table 13: Per-session held-out adaptation performance in R^2. Each entry reports the mean across seeds of Scratch / Pretrain+FT / Δ_{PT}, where $\Delta_{\mathrm{PT}} = (\text{Pretrain+FT}) - (\text{Scratch})$.

Session / Model	CEBRA	NDT2	POYO+	NEDS
S003	0.285 / 0.315 / 0.030	0.019 / 0.017 / -0.003	0.342 / 0.139 / -0.203	0.416 / 0.513 / 0.097
S017	0.380 / 0.465 / 0.085	0.016 / -0.011 / -0.027	0.373 / 0.329 / -0.044	0.365 / 0.504 / 0.139
S021	0.049 / 0.100 / 0.051	-0.108 / -0.026 / 0.081	-0.048 / -0.076 / -0.028	0.080 / -0.231 / -0.311
S029	0.083 / 0.144 / 0.061	-0.015 / -0.007 / 0.009	0.080 / -0.004 / -0.084	0.205 / 0.216 / 0.011
S038	0.591 / 0.548 / -0.043	0.037 / 0.034 / -0.003	0.430 / 0.336 / -0.094	0.571 / 0.540 / -0.031
S039	-0.078 / 0.042 / 0.120	0.014 / 0.008 / -0.006	-0.006 / 0.025 / 0.031	0.087 / 0.112 / 0.025
S043	-0.017 / -0.072 / -0.055	-0.044 / -0.059 / -0.015	-0.065 / -0.022 / 0.043	0.016 / 0.082 / 0.066
S045	0.056 / 0.089 / 0.033	0.002 / -0.004 / -0.006	0.075 / 0.001 / -0.074	0.126 / 0.213 / 0.087
S052	0.022 / 0.101 / 0.079	-0.156 / -0.052 / 0.103	-0.161 / -0.022 / 0.139	0.124 / 0.121 / -0.003
S055	-0.001 / 0.031 / 0.034	-0.043 / -0.022 / 0.021	-0.097 / -0.058 / 0.039	0.052 / 0.026 / -0.026

Table 14: Per-session held-out adaptation results in Pearson correlation. Entries report the seed-averaged Scratch, Pretrain+FT, and adaptation gain Δ_{PT}, defined as Pretrain+FT minus Scratch.

Session / Model	CEBRA	NDT2	POYO+	NEDS
S003	0.619 / 0.583 / -0.036	0.164 / 0.169 / 0.005	0.606 / 0.291 / -0.314	0.667 / 0.719 / 0.052
S017	0.691 / 0.731 / 0.040	0.170 / 0.208 / 0.038	0.630 / 0.586 / -0.044	0.733 / 0.729 / -0.004
S021	0.405 / 0.353 / -0.052	0.010 / -0.021 / -0.031	0.032 / 0.107 / 0.075	0.477 / 0.004 / -0.473
S029	0.553 / 0.426 / -0.127	0.101 / 0.078 / -0.023	0.323 / 0.155 / -0.168	0.573 / 0.529 / -0.044
S038	0.788 / 0.762 / -0.026	0.209 / 0.171 / -0.037	0.691 / 0.616 / -0.075	0.794 / 0.780 / -0.014
S039	0.356 / 0.290 / -0.066	0.209 / 0.218 / 0.010	0.242 / 0.171 / -0.072	0.330 / 0.387 / 0.057
S043	0.333 / 0.232 / -0.101	-0.076 / -0.029 / 0.048	0.049 / 0.032 / -0.018	0.226 / 0.307 / 0.081
S045	0.456 / 0.445 / -0.011	0.097 / 0.028 / -0.069	0.205 / 0.072 / -0.133	0.430 / 0.501 / 0.071
S052	0.379 / 0.420 / 0.041	0.009 / 0.011 / 0.003	0.085 / 0.234 / 0.149	0.460 / 0.456 / -0.004
S055	0.379 / 0.279 / -0.100	0.035 / 0.114 / 0.079	-0.025 / 0.112 / 0.137	0.379 / 0.303 / -0.076

E.4 Joint Modeling and Convergence Analysis

Joint-modeling results. Table 15 reports the detailed comparison between direction-specific and joint NEDS training. Joint optimization does not produce uniform improvements across prediction directions and evaluation settings, consistent with the task-dependent effects summarized in Section 5.4.

Convergence analysis. Figure 7 compares the training and validation dynamics of encoding, decoding, and joint training. Decoding converges substantially faster than encoding during both pretraining and fine-tuning, leading to different preferred model-selection epochs across prediction

Table 15: Comparison of NEDS under direction-specific and joint encoding–decoding training in single-session and multi-session settings. Δ denotes Joint minus Single-task performance for the corresponding metric. Arrows indicate the preferred direction of each metric (↑ higher is better; ↓ lower is better).

Task		Encoding			Decoding		
Setting	**Metric**	Single-task	Joint	Δ	Single-task	Joint	Δ
Single-session	MAE↓	0.4303 ± 0.0028	0.4403 ± 0.0032	0.0099 ± 0.0043	0.7002 ± 0.0073	0.7286 ± 0.0317	0.0284 ± 0.0325
	MSE↓	0.8884 ± 0.0036	0.8980 ± 0.0017	0.0096 ± 0.0040	0.7105 ± 0.0085	0.7718 ± 0.0489	0.0613 ± 0.0496
	R^2 ↑	-0.0976 ± 0.0286	-0.0734 ± 0.0136	0.0242 ± 0.0317	0.2681 ± 0.0101	0.2042 ± 0.0501	-0.0639 ± 0.0511
	Pearson↑	0.1109 ± 0.0102	0.0576 ± 0.0121	-0.0533 ± 0.0158	0.5304 ± 0.0118	0.5070 ± 0.0464	-0.0234 ± 0.0479
Multi-session	MAE↓	0.4361 ± 0.0011	0.4306 ± 0.0079	-0.0055 ± 0.0080	0.6946 ± 0.0114	0.7250 ± 0.0142	0.0304 ± 0.0182
	MSE↓	0.8918 ± 0.0008	0.9033 ± 0.0016	0.0115 ± 0.0018	0.7008 ± 0.0194	0.7802 ± 0.0285	0.0794 ± 0.0345
	R^2 ↑	0.0070 ± 0.0008	-0.0056 ± 0.0018	-0.0126 ± 0.0020	0.2870 ± 0.0197	0.2095 ± 0.0279	-0.0775 ± 0.0342
	Pearson↑	0.0895 ± 0.0055	0.0446 ± 0.0064	-0.0449 ± 0.0084	0.5481 ± 0.0185	0.4716 ± 0.0255	-0.0765 ± 0.0315

directions. This mismatch provides a possible explanation for the conditional benefit of joint optimization.

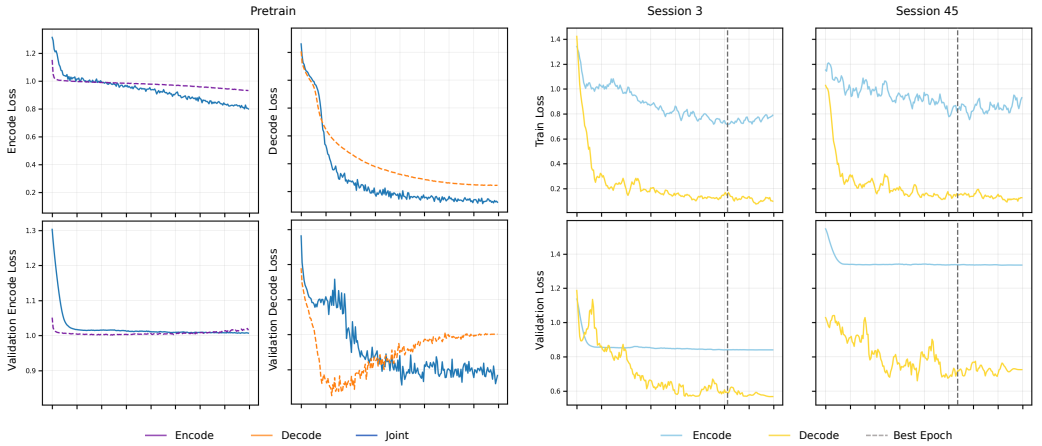

Figure 7: Training and validation loss curves for Encode, Decode, and Joint tasks across pretraining and sessions fine-tuning. Top rows are training loss and bottom rows are validation loss. We compare the convergence rate and corresponding validation loss of different targets under different task settings.

E.5 Qualitative prediction examples

Figure 8 presents representative qualitative examples of position decoding on held-out sessions. We show one stationary/low-mobility window and one locomotion window under the main threshold of 3.0 cm/s, and compare the ground-truth position coordinates with the predicted x- and y-coordinates over time. The illustrated windows are selected as representative examples from held-out sessions rather than as extreme best-case predictions.

Figure 9 complements the decoding examples by visualizing neural encoding predictions. Because the encoding target is high-dimensional neural activity, we use heatmaps over selected neurons and time frames rather than individual coordinate traces. Neurons are selected by ground-truth activity variance rather than prediction quality, and the error maps show the difference between predicted and ground-truth activity.

These qualitative examples are intended to provide an intuitive view of the benchmark tasks and typical model behavior. They should be interpreted as visual complements to the aggregate quantitative results, rather than as standalone evidence.

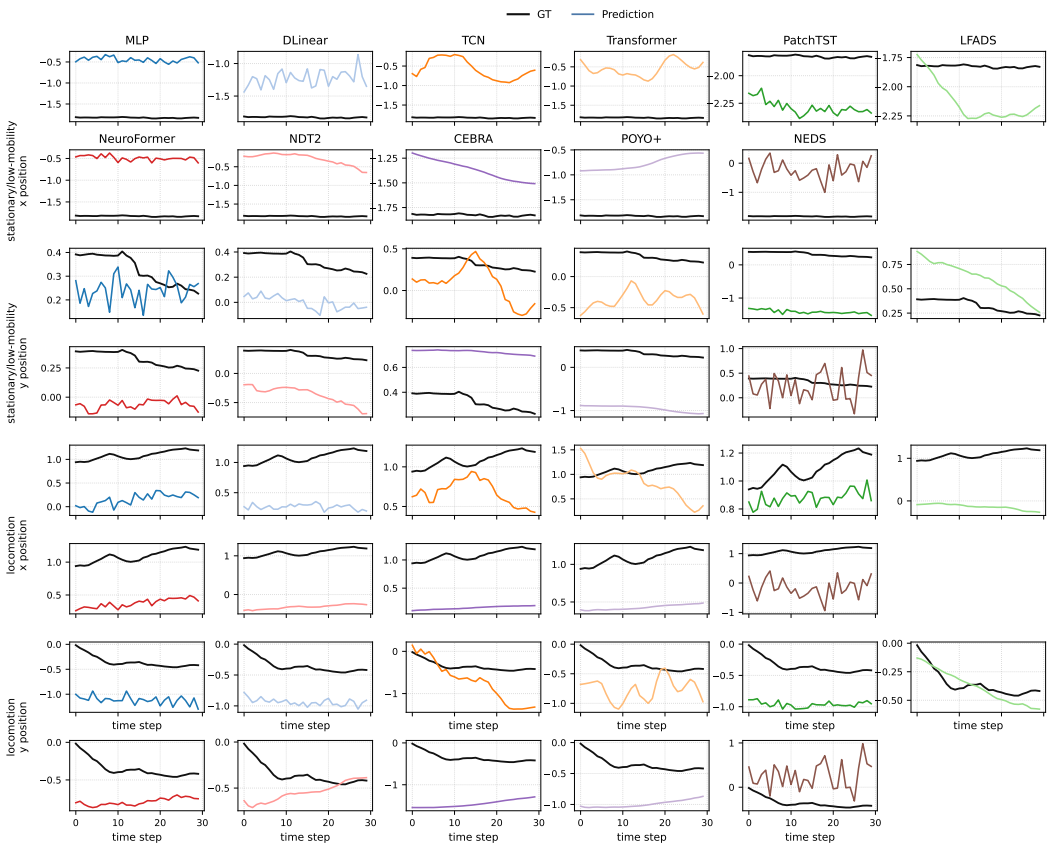

Figure 8: Qualitative position decoding examples on held-out sessions. We show representative stationary/low-mobility and locomotion windows under the main threshold of 3.0 cm/s. For each window, ground-truth and predicted x- and y-coordinates are compared over time across the evaluated models. These examples provide an intuitive view of how decoding quality varies across behavioral regimes.

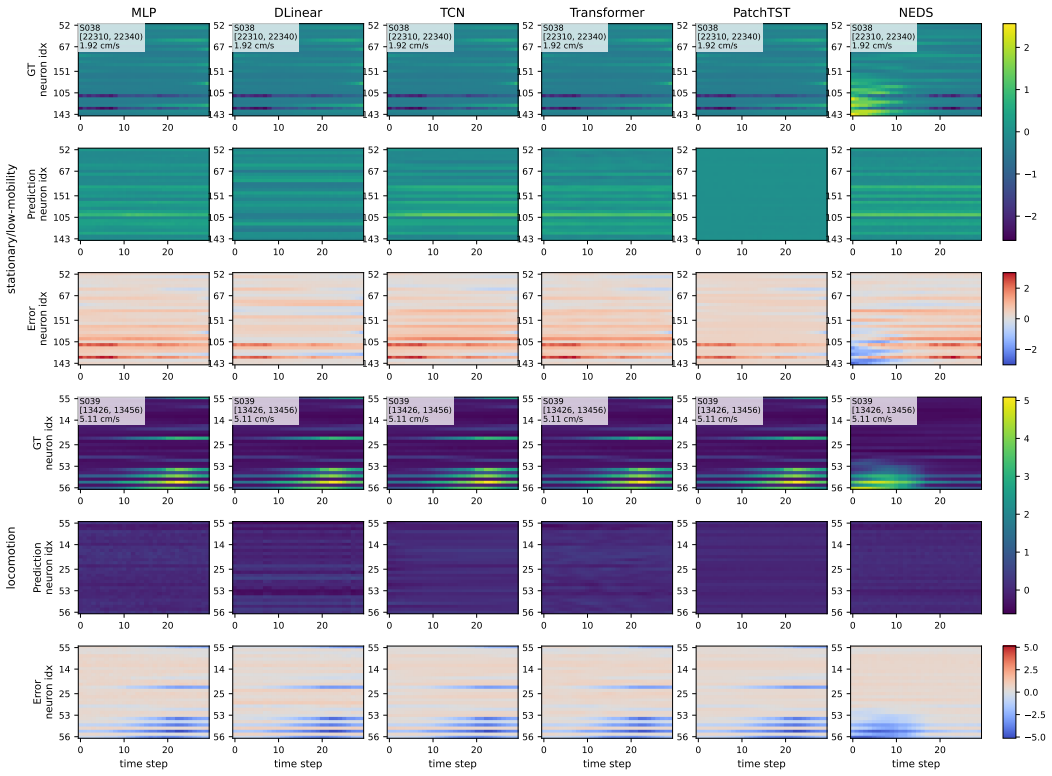

Figure 9: Qualitative neural encoding examples on held-out sessions. We compare ground-truth and predicted neural activity for representative stationary/low-mobility and locomotion windows. Heatmaps show selected neurons over time, with neurons selected by ground-truth activity variance rather than prediction quality. Error maps show prediction minus ground truth.

F Limitations

FreeCAB focuses on dorsal CA1 calcium imaging during freely moving open-field exploration. Although the matched single-session benchmark evaluates both 2D position and 3D pose, the behavioral-state, cross-session adaptation, and joint-modeling analyses currently focus on position. Conclusions about encoding–decoding asymmetry, behavioral representation, and state dependence should therefore be interpreted within this experimental setting rather than generalized to other brain regions, recording modalities, or behavioral paradigms.

The 3D behavioral representation is reconstructed from synchronized multi-view video without independent 3D ground truth. Our reprojection-based quality control measures consistency with the underlying 2D detections rather than absolute reconstruction accuracy. In addition, several neural-domain baselines require method-specific adaptations to the FreeCAB interface, so these results should be interpreted as benchmark reference points rather than definitive rankings of the corresponding model families.

Finally, cross-session conclusions remain limited by the number and diversity of retained animals and sessions. Larger multi-animal datasets, additional brain regions and behavioral paradigms, and broader evaluation of 3D behavior under cross-session and joint-modeling settings would provide stronger tests of transferable neural–behavioral representations.

G Broader Impact

FreeCAB is intended to support reproducible research on neural–behavioral modeling under freely moving conditions. By providing aligned multimodal recordings, documented session splits, bench-

mark protocols, and reference baselines, it enables systematic evaluation across behavioral representations, behavioral states, and recording sessions.

The benchmark is intended for research use rather than clinical or operational deployment. Because the current release is based on dorsal CA1 calcium imaging from mice during open-field exploration, benchmark results should not be interpreted as general conclusions about hippocampal coding or transferred directly to other species, brain regions, behavioral settings, or recording modalities. To reduce risks of misinterpretation, FreeCAB documents preprocessing, alignment, session curation, evaluation protocols, and benchmark scope explicitly.