NeuroHorizon: Long-Horizon Forecasting of Neural Population Activity with Hierarchical Memory

Peng Jiang[1,2,3]
jiang-p21@mails.tsinghua.edu.cn

Xiaoxuan Jia[1,2,3]*
jiaxx@mail.tsinghua.edu.cn

Abstract

Long-horizon forecasting of neural population activity tests whether a model has captured dynamics that remain predictive well beyond the observed history. Prediction quality typically declines as the horizon grows, yet neither short-horizon accuracy nor teacher-forced evaluation, which supplies the true activity preceding each predicted step, reveals how well models sustain predictions from an initial history alone. We compare direct multi-step prediction and autoregressive rollout on spiking recordings across horizons and introduce NEUROHORIZON, which pairs event-based spike encoding with a causal decoder for population firing rates. Its hierarchical prediction memory keeps recent predictions at full temporal resolution and compresses earlier ones into segment summaries, giving the decoder both local detail and longer-range context. Under teacher-forced training this memory holds ground-truth activity, whereas during rollout it holds the model's own predictions. To reduce this mismatch, we train with scheduled sampling, progressively exposing the decoder to its own predictions. From 500 ms of macaque motor-cortex activity, NEUROHORIZON achieves the highest forward-prediction score among the evaluated autonomous models at all four horizons from 100 to 1000 ms, and its per-bin predictions remain above a mean-firing-rate baseline across nearly the full 1 s window. Complementary experiments on macaque sensorimotor and mouse multi-region recordings show that performance generally increases with model and data size, alongside improved prediction after cross-subject pretraining and fine-tuning. Together, these results clarify how forecasting performance changes across horizons and offer an effective approach to long-horizon forecasting of neural population activity.

1 Introduction

Predicting future neural population activity from past spikes provides a quantitative test of learned neural dynamics [1, 2]. Long-horizon prediction extends this test by asking how long those dynamics remain informative without new observations, with potential relevance to anticipatory neural control [3]. Because predictive performance typically declines as the forecast horizon increases, models should be evaluated both on their overall performance and on how well they sustain predictions over time.

Autonomous forecasting can use direct multi-step prediction, defined here as jointly predicting the future window, or autoregressive rollout, which advances through generated outputs or latent states [4]. In both cases, predictions depend only on the initial neural history. Existing neural population models

*Corresponding author.
[1]Center for Life Sciences and Artificial Intelligence, School of Life Sciences, Tsinghua University, Beijing, China.
[2]IDG/McGovern Institute for Brain Research, Tsinghua University, Beijing, China.
[3]Tsinghua-Peking Center for Life Sciences, Tsinghua University, Beijing, China.
Code: https://github.com/Jp-17/NeuroHorizon

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

include LFADS, masked transformers such as IBL-MTM and NDT2, and autoregressive event generators such as Neuroformer [1, 5–7]. Forward prediction has been evaluated over short fixed windows, including 200 ms in NLB and IBL-MTM [2, 5]. Teacher-forced evaluation instead supplies true future prefixes, so its scores do not measure forecasting performance from the initial history alone [8]. The resulting question is how these forecasting strategies compare as the horizon increases when no further neural observations are available.

Calcium-imaging studies have already examined forecasting over longer intervals. POCO and ZAP-Bench evaluate future neural activity across extended windows [9, 10], and Lu et al. report informative forecasts of mouse cortical activity up to approximately 1.5 s [3]. However, calcium fluorescence reflects temporally filtered spiking activity, with calcium and indicator kinetics contributing persistence to the measured signal [11, 12]. These findings do not directly establish how far into the future neural activity can be predicted from spike recordings alone.

To assess long-horizon forecasting from spike recordings, we compare direct multi-step prediction and autoregressive rollout across sub-second to second-scale horizons using only an initial observation window. Direct prediction avoids feeding earlier prediction errors back into subsequent steps, but must infer the entire future trajectory from the observed history. Autoregressive rollout instead constructs this trajectory sequentially, allowing later predictions to depend on earlier generated activity while also permitting errors to propagate through that feedback [4]. When a model is trained only on true prefixes, replacing them with predicted prefixes during rollout introduces an additional training–inference mismatch, commonly termed exposure bias [8, 13]. This motivates an autoregressive approach that combines a structured representation of the predicted trajectory with training that exposes the model to its own predictions.

We introduce NEUROHORIZON, an encoder–decoder model for long-horizon forecasting of neural population activity. A POYO+-style event encoder represents the observed spike history, and a causal decoder predicts future population firing rates one bin at a time [14]. The decoder uses a hierarchical prediction memory that retains recent predicted bins at full temporal resolution and summarizes the preceding trajectory at a coarser resolution, drawing on the use of compressed memory for long-range sequence modeling [15]. This organization preserves local temporal detail while providing compact access to longer-range context. During autonomous rollout, the memory is constructed from model predictions, while the observed history remains available through the encoder latents. Scheduled sampling progressively increases the probability of feeding predictions rather than ground-truth counts into memory during training, exposing the decoder to the inputs it will encounter during rollout [8].

We compare NEUROHORIZON with direct multi-step predictors and rollout-based models on the Perich–Miller dataset of macaque motor-cortex recordings [16], using 500 ms of history and four prediction horizons from 100 to 1000 ms. NEUROHORIZON achieves the highest mean forward-prediction score among the evaluated autonomous models at every horizon. Beyond whole-window scores, per-bin evaluation shows predictions above a mean-firing-rate baseline across nearly the full 1 s window. In ablations, the combined memory yields higher mean rollout scores than either memory component alone, and scheduled sampling improves long-horizon rollout performance. Complementary experiments examine model-size and data-size scaling on macaque sensorimotor and mouse multi-region recordings, and show improved cross-subject rollout prediction at 500 ms after pretraining and fine-tuning.

Our main contributions are summarized below.

- **Multi-horizon evaluation of spike-based forecasting.** We compare direct multi-step prediction and autoregressive rollout using the same initial history across models, evaluating whole-window performance and per-bin prediction decay while reporting teacher-forced evaluation separately.
- **Hierarchical memory for long-horizon forecasting.** We introduce NEUROHORIZON, combining prediction memory at two temporal resolutions with scheduled-sampling training to improve autonomous forecasting. The model achieves the highest mean scores at all four evaluated horizons in the comparison on the Perich–Miller dataset.
- **Scaling and cross-subject transfer.** We examine model-size and data-size scaling on macaque and mouse recordings, and observe improved 500 ms rollout prediction after cross-subject pretraining and fine-tuning.

2 Related Work

Neural population modeling. Neural population models capture dependencies across neurons and time through different representations and learning objectives. Point-process generalized linear models describe history-dependent spiking and coupling between neurons, while LFADS learns latent dynamics underlying population activity [1, 17]. Transformer-based approaches such as NDT2 and IBL-MTM learn population representations through masked modeling across sessions [5, 6]. At the event level, Neuroformer generates neural activity autoregressively, including extended stimulus-conditioned simulations of events inferred from calcium imaging [7]. Event-based representations also support behavioral decoding, as demonstrated by POYO and POYO+ across heterogeneous neural populations [14, 18]. These advances in neural representation learning also provide a basis for studying how well population activity can be predicted across different horizons. Building on the POYO+ encoder, we represent the observed spike history and use a causal decoder to predict future population firing rates.

Neural activity forecasting and evaluation. Neural activity forecasting has been evaluated under varied observation conditions, prediction horizons, and performance measures. For spike recordings, the Neural Latents Benchmark evaluates held-out future activity, XFADS reports 500 ms forecasts from a 200 ms initial trial segment on MC_Maze, and SpikeProphecy benchmarks next-bin spike-count prediction from recorded history and analyzes its degradation during multi-step rollout [2, 19, 20]. For calcium recordings, POCO conditions neuron-specific forecasts on population activity, ZAPBench and Lu et al. benchmark whole-brain and probabilistic forecasting models, and OmniMouse includes a one-second forecasting task on calcium-derived activity [3, 9, 10, 21]. Building on these settings, we focus on spike recordings and ask how forecasting performance degrades as models predict further ahead without new observations, and whether this degradation differs between direct multi-step prediction and autoregressive rollout.

Memory and exposure bias in autoregressive forecasting. Long-horizon autoregressive forecasting raises two problems, namely how to represent an expanding prediction history and how to train models to condition on their own outputs. Transformer-XL represents long sequences by reusing hidden states across segments, while Compressive Transformer retains recent memories alongside compressed older representations [15, 22]. Exposure bias arises when a model trained on true prefixes must instead condition on its own outputs during inference, which can amplify errors over successive prediction steps [8, 13]. Scheduled sampling addresses this mismatch by introducing model predictions as inputs during training [8]. We adapt these ideas in NEUROHORIZON by combining hierarchical prediction memory with scheduled sampling for neural population forecasting.

3 Method

NEUROHORIZON encodes observed spike events into fixed history latents and predicts future population activity autoregressively (Figure 1). Within this framework, a hierarchical memory represents the growing prediction prefix, while scheduled sampling exposes the decoder to predicted inputs during training. Together, these components condition each prediction on the observed history and the preceding forecast.

3.1 Forecasting Task and Objective

We observe spike events within $[-T_{\text{obs}}\Delta t, 0)$ and forecast the next T_{pred} bins without further observations. Here T_{obs} and T_{pred} specify window lengths in units of $\Delta t = 20\,\text{ms}$, while the observed events retain their continuous timestamps. Prediction targets are spike counts in these future bins. Let $\boldsymbol{y}_\tau \in \mathbb{N}_0^N$ denote the counts for N recorded neurons at prediction bin $\tau = 1, \ldots, T_{\text{pred}}$. We use a conditional Poisson observation model $y_{\tau,n} \sim \text{Poisson}(\mu_{\tau,n})$ [17], where $\mu_{\tau,n} = \lambda_{\tau,n}\Delta t$ for firing rate $\lambda_{\tau,n}$. The decoder outputs the log expected count $\ell_{\tau,n} = \log \hat{\mu}_{\tau,n}$.

Training objective. To fit the predicted means to observed spike counts, the training objective is the Poisson negative log-likelihood averaged over prediction bins and recorded neurons,

$$\mathcal{L}_{\text{train}} = \frac{1}{NT_{\text{pred}}} \sum_{\tau=1}^{T_{\text{pred}}} \sum_{n=1}^{N} \left[\hat{\mu}_{\tau,n} - y_{\tau,n} \log \hat{\mu}_{\tau,n} + \log(y_{\tau,n}!)\right]. \tag{1}$$

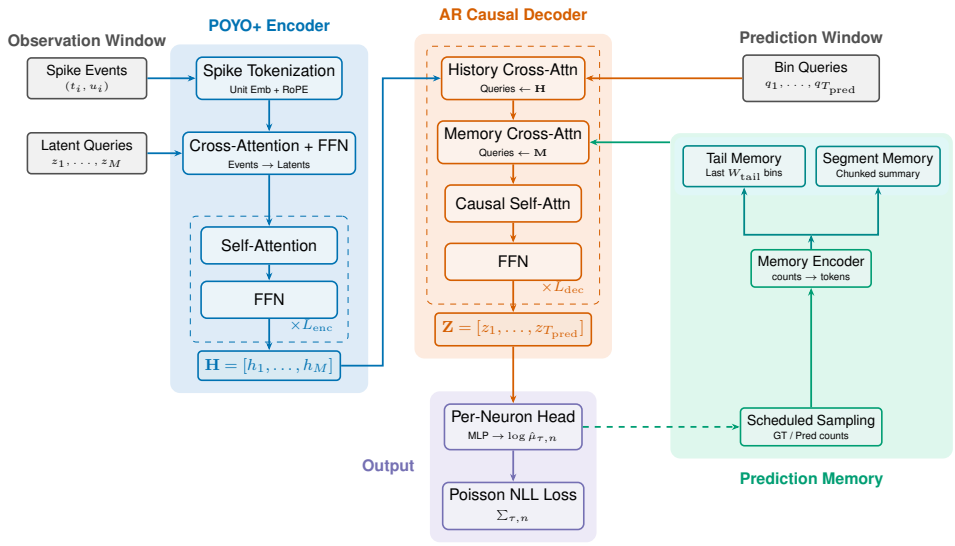

Figure 1: **Overview of the NEUROHORIZON architecture.** A POYO+ event encoder maps observed spike events to history latents. A causal decoder predicts future firing rates bin by bin, attending to both history latents and a hierarchical prediction memory. During training, scheduled sampling chooses whether memory receives ground-truth counts or model predictions.

The term $\log(y_{\tau,n}!)$ depends only on the observed counts and is omitted during optimization. For minibatches with different population sizes, we average the loss over all valid neuron-bin entries, excluding padded neurons.

3.2 Event Encoder and Causal Decoder

Event encoder. To represent recordings with different neural populations in a common form, we encode spike timing and neuron identity into a fixed set of latent tokens. Following POYO+ [14], we represent observed spikes using learned neuron identity embeddings and continuous-time rotary positional encoding (RoPE). Perceiver-style cross-attention [23] and a feed-forward network (FFN), followed by L_{enc} self-attention–FFN blocks [24], map the events to history latents $\boldsymbol{H} \in \mathbb{R}^{M \times d}$.

Causal decoder. The decoder conditions each future bin on both the observed activity and the preceding forecast through two cross-attention pathways. History cross-attention provides a fixed reference to the observations, while memory cross-attention incorporates context accumulated over preceding prediction bins. Causal self-attention then relates current and earlier bin representations without accessing future bins. Each prediction bin starts from a learned query with RoPE evaluated at its center timestamp. Each decoder layer updates its input states $\boldsymbol{Q}$ through history cross-attention, memory cross-attention, causal self-attention, and an FFN, with residual connections,

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{Q} + \mathrm{HistCrossAttn}(\boldsymbol{Q}, \boldsymbol{H}), \\ \boldsymbol{V} &= \boldsymbol{C} + \mathrm{MemCrossAttn}(\boldsymbol{C}, \{\boldsymbol{M}_\tau\}_\tau), \\ \boldsymbol{U} &= \boldsymbol{V} + \mathrm{CausalSelfAttn}(\boldsymbol{V}), \\ \boldsymbol{Z} &= \boldsymbol{U} + \mathrm{FFN}(\boldsymbol{U}). \end{aligned} \tag{2}$$

To translate the shared population context encoded in each final bin state into neuron-specific predictions, a shared MLP combines that state with the target neuron's identity embedding and outputs its log expected spike count.

3.3 Hierarchical Prediction Memory

The prediction prefix contains both recent changes and activity patterns spanning multiple bins. We represent these at two temporal resolutions so that the decoder can use local detail together with broader population context. Let $\boldsymbol{x}_\tau$ denote the population vector written to memory after predicting

bin τ. It contains ground-truth counts or predicted expected counts during training, and predicted expected counts during rollout. The training feedback is selected by scheduled sampling (Section 3.4).

Memory encoder. The decoder needs a memory representation whose size does not depend on the number of recorded neurons. We therefore combine each neuron's count with its identity embedding and pool the resulting features using learned summary queries. The queries attend to neuron features as keys and values, producing a fixed number of tokens for each population vector. The tail and segment encoders, ϕ_{tail} and ϕ_{seg}, use this construction with independent parameters and produce K_{tail} tokens per tail bin and K_{seg} tokens per segment, respectively.

Tail and segment memory. To preserve recent temporal detail, the tail keeps a separate representation for each of the most recent W_{tail} bins. At prediction step τ, these representations are concatenated as

$$\boldsymbol{M}_{\tau}^{\text{tail}} = \text{Concat}_{u=\max(1,\tau-W_{\text{tail}})}^{\tau-1} \phi_{\text{tail}}(\boldsymbol{x}_u). \tag{3}$$

To provide context beyond the recent window, the segment pathway summarizes the full available prefix using averages over consecutive groups of up to S bins. These summaries preserve a broader view of population activity without keeping a separate token group for every older bin. Let $\bar{\boldsymbol{x}}_{j,\tau}$ denote the mean population vector of segment j among the J_τ segments available before step τ. The segment tokens and the combined memory are

$$\boldsymbol{M}_{\tau}^{\text{seg}} = \text{Concat}_{j=1}^{J_\tau} \phi_{\text{seg}}(\bar{\boldsymbol{x}}_{j,\tau}), \qquad \boldsymbol{M}_{\tau} = [\boldsymbol{M}_{\tau}^{\text{seg}}; \boldsymbol{M}_{\tau}^{\text{tail}}]. \tag{4}$$

Memory regularization. Because generated feedback can be inaccurate, we train the decoder with perturbed or masked prediction memory. During training, we perturb memory-encoder input features with noise and random zeroing. We also apply block dropout to memory cross-attention, masking all tail and segment tokens for selected queries. This trains the decoder to use observed history and causal bin representations when prediction memory is unavailable.

3.4 Scheduled Sampling

Training exclusively on ground-truth prefixes does not expose the decoder to the predicted inputs used during rollout. Scheduled sampling [8] reduces this mismatch by selecting the memory source independently for each example and prediction bin. At training epoch e, predicted feedback is selected with probability ϵ_e,

$$\boldsymbol{x}_\tau = \begin{cases} \boldsymbol{y}_\tau & \text{with probability } 1-\epsilon_e, \\ \text{sg}(\hat{\boldsymbol{\mu}}_\tau) & \text{with probability } \epsilon_e. \end{cases} \tag{5}$$

The choice applies to the entire population vector. The stop-gradient operator sg prevents later losses from propagating through the feedback values into earlier predictions, while the memory encoders remain trainable. To introduce predicted feedback gradually, we increase its probability over E_{ramp} training epochs to a maximum $\epsilon_{\max}$,

$$\epsilon_e = \epsilon_{\max} \min(e/E_{\text{ramp}}, 1). \tag{6}$$

The probability increases linearly from zero and remains at $\epsilon_{\max}$ after the ramp.

4 Experiments

4.1 Datasets

We use three neural population datasets for the principal experiments (Table 1). The **Perich–Miller dataset** [16] supports our comparison across prediction horizons, with 10 sessions of structured center-out reaching from three macaques. We access these recordings through brainsets [18]. The **O'Doherty–Sabes dataset** [25] comprises 47 sessions from two macaques performing continuous 2-D cursor reaching controlled by hand or finger movements, with recordings from M1 and S1. We use 32 sessions from 32 mice in the **Allen Brain Observatory Visual Coding Neuropixels dataset** [26] during passive natural-movie viewing. The recordings span multiple brain regions, including visual cortical, thalamic, and hippocampal areas. We use the O'Doherty–Sabes dataset and the Allen Neuropixels dataset to examine the effects of model size and training data size. We additionally evaluate 500 ms forecasting on MC_Maze from the Neural Latents Benchmark (NLB) [2].

Table 1: **Dataset overview.** Neural population spiking datasets used in the main experiments. Session and subject counts refer to the recordings used in this study.

Dataset	Sessions	Subjects	Region(s)	Task
Perich–Miller dataset [16]	10	3	M1 + PMd	Instructed-delay center-out reaching
O'Doherty–Sabes dataset [25]	47	2	M1 + S1	Continuous hand-controlled 2D cursor reaching
Allen Neuropixels dataset [26]	32	32	VIS, CA1, etc.	Passive natural movie viewing

For comparisons on the Perich–Miller dataset, each window contains a 500 ms observation period followed by a prediction horizon of 100, 240, 500, or 1000 ms. Training windows are sampled with randomized offsets within the training domains and shuffled each epoch. Validation and test windows are sampled sequentially within their assigned trial intervals. At each prediction horizon, all models share the same data splits, evaluation windows, null model, and metrics. Dataset-specific splits are described in Appendix A.

4.2 Evaluation Modes and Metrics

We distinguish three prediction modes. **Direct** forecasting predicts the entire future window from the observed history without feeding predictions back. **Rollout (RO)** advances through generated predictions or an autonomously evolving latent state. Direct forecasting and rollout are autonomous because they use only the initial neural history, without further observations. **Teacher-forced (TF)** evaluation instead supplies the true activity preceding each prediction step. We therefore report TF results separately from autonomous forecasting results.

The primary metric is forward-prediction bits-per-spike (fp-bps) [2]. For each session, we compute

$$\text{fp-bps} = \frac{\mathcal{L}_{\text{null}} - \mathcal{L}_{\text{model}}}{\log 2 \cdot \sum_{w,\tau,n} y_{w,\tau,n}}. \tag{7}$$

Here w indexes evaluation windows, τ prediction bins, and n recorded neurons within the session. $\mathcal{L}_{\text{model}}$ and $\mathcal{L}_{\text{null}}$ are the summed Poisson negative log-likelihoods over the same evaluation entries for the model predictions and a homogeneous Poisson null using each neuron's training-set mean firing rate, respectively. The denominator normalizes the likelihood improvement by the total observed spike count and converts it to bits. Zero indicates the same Poisson score as this mean-firing-rate baseline. We report the mean session score with SEM across sessions. We additionally report the coefficient of determination (R^2) between predicted and observed binned counts as a supplementary metric.

4.3 Baseline Models

We compare NEUROHORIZON with models that predict the future window in parallel or generate future activity sequentially from the observed neural history. **IBL-MTM** [5] masks the future bins and reconstructs them in parallel using the observed history as context. **POCO** [9] combines neuron-specific forecasting with population conditioning to predict the entire future window in a single forward pass. **LFADS** [1] infers an initial latent state from the observed history and evolves its generator autonomously over the future window, with controller and external inputs disabled. **Causal NDT2** [6] is adapted for next-bin count prediction and advances through the future window by feeding back counts sampled from its predictions. **Neuroformer** [7] generates neural events autoregressively, which we convert to binned counts for evaluation.

In the current implementations, IBL-MTM and POCO are evaluated in Direct mode, and LFADS in RO mode. NEUROHORIZON, causal NDT2, and Neuroformer are evaluated in both TF and RO modes, conditioning on ground-truth prefixes in TF and model-generated outputs in RO. Model adaptations, feedback mechanisms, and scoring details are described in Appendix B.

Table 2: **Forecasting performance across horizons.** Forward-prediction bits per spike (fp-bps) on the Perich–Miller dataset using 500 ms of observed neural history. Direct and rollout (RO) forecasts use only this initial history, while teacher-forced (TF) evaluation receives true preceding activity. Values are session means with SEM. The highest mean in each column within each evaluation block is bold.

Model (mode)	100 ms	240 ms	500 ms	1000 ms
Autonomous forecasting				
IBL-MTM (Direct)	0.226 (0.019)	0.200 (0.013)	0.120 (0.011)	-0.018 (0.025)
POCO (Direct)	0.177 (0.020)	0.136 (0.019)	0.070 (0.017)	0.035 (0.007)
Neuroformer (RO)	-0.707 (0.059)	-0.779 (0.066)	-2.478 (0.241)	-5.473 (0.117)
LFADS (RO)	0.212 (0.017)	0.204 (0.014)	0.172 (0.014)	0.139 (0.011)
Causal NDT2 (RO)	0.235 (0.019)	0.156 (0.015)	0.095 (0.017)	0.035 (0.021)
NEUROHORIZON (RO)	**0.254** (0.020)	**0.224** (0.016)	**0.188** (0.015)	**0.143** (0.011)
TF evaluation				
Neuroformer (TF)	-0.735 (0.053)	-0.641 (0.049)	-0.498 (0.068)	-1.676 (0.389)
Causal NDT2 (TF)	**0.296** (0.022)	**0.299** (0.024)	**0.289** (0.022)	0.291 (0.025)
NEUROHORIZON (TF)	0.281 (0.024)	0.285 (0.024)	0.281 (0.025)	**0.295** (0.027)

4.4 Training Configuration

The default NEUROHORIZON model uses a 6-layer encoder with hidden size 128 and a 2-layer causal decoder. Its prediction memory combines bin-level representations of the 4 most recent bins with summaries of the prediction prefix in segments of up to 5 bins. We train NEUROHORIZON for 300 epochs using SparseLamb, bfloat16 mixed precision, and a Poisson negative log-likelihood objective. Scheduled sampling linearly increases the probability of feeding model predictions into memory from 0 to 0.5 over the first 150 epochs. Table S1 lists the training hyperparameters; Appendix C provides training and compute details.

5 Results

5.1 Autonomous Forecasting Across Horizons

Longer prediction horizons test how well models sustain forecasts using only an initial neural history. We compare models on the Perich–Miller dataset using 500 ms of observed activity and prediction horizons of 100, 240, 500, and 1000 ms (Table 2).

Autonomous forecasting scores generally decrease as the horizon grows. NEUROHORIZON achieves the highest mean at all four horizons, with rollout fp-bps of 0.254, 0.224, 0.188, and 0.143, respectively. Session-level comparisons show that NEUROHORIZON remains above the mean-firing-rate baseline across all recordings and horizons, while differences between models vary across sessions (Table S2). NEUROHORIZON also achieves the highest mean autonomous R^2 at each horizon, supporting its performance beyond the likelihood-based metric (Table S3). By contrast, TF scores for NEUROHORIZON and causal NDT2 remain relatively stable across horizons despite their declining rollout scores. On the MC_Maze forecasting adaptation, NEUROHORIZON also achieves higher autonomous fp-bps than IBL-MTM and POCO at 500 ms (Table S4). The example forecasts broadly reflect the observed differences in activity levels across neurons (Figure S1). A complementary Lorenz experiment examines information about known latent trajectories in the predicted activity. NEUROHORIZON achieves slightly higher aligned latent R^2 and trajectory correlation than causal NDT2 (Table S5 and Figure S2).

5.2 Per-Bin Temporal Decay

Whole-window scores do not reveal how prediction quality changes within a forecast. We therefore examine per-bin fp-bps to determine how long predictions remain informative (Figure 2). Scores generally decline as prediction proceeds within each horizon. In the later portion of the 1000 ms window, IBL-MTM, POCO, and causal NDT2 reach near-zero or negative scores, indicating performance close to or below the mean-firing-rate baseline.

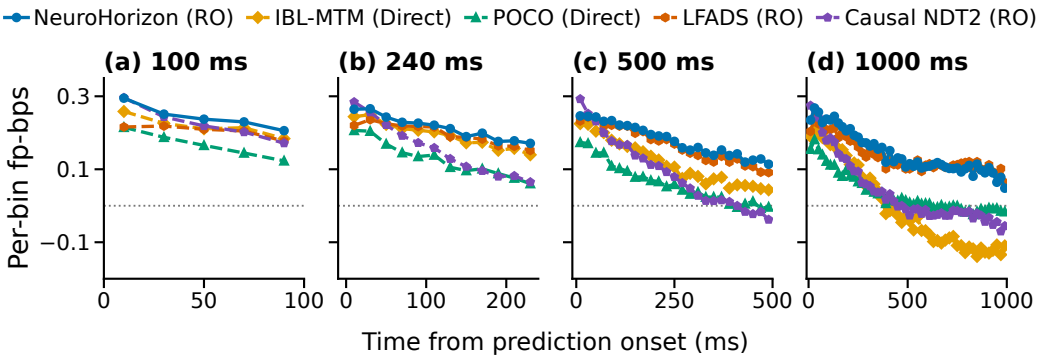

Figure 2: **Per-bin temporal decay of forecasting performance.** Panels (a–d) show per-bin fp-bps for prediction horizons of 100, 240, 500, and 1000 ms, respectively, using a 500 ms observation window. Scores use global spike weighting across sessions, neurons, and evaluation windows. All models forecast without additional neural observations, with prediction modes indicated in the legend.

Table 3: **Effects of prediction memory and scheduled sampling.** Values are session means of fp-bps with SEM. (a) Four configurations compare no prediction memory, tail memory alone, segment memory alone, and their combination. Configurations with prediction memory use the same scheduled-sampling schedule with $\epsilon_{\max} = 0.50$. TF and RO yield identical predictions without prediction memory, since neither true nor predicted counts are fed back. (b) The combined tail and segment memory is retained while $\epsilon_{\max}$ is varied. The highest mean in each column within each section is bold.

	500 ms		1000 ms	
Configuration	TF	RO	TF	RO
(a) Prediction memory				
No memory	0.174 (0.014)	0.174 (0.014)	0.139 (0.012)	0.139 (0.012)
Only tail memory	0.278 (0.024)	0.182 (0.015)	0.291 (0.026)	0.126 (0.013)
Only segment memory	0.268 (0.025)	0.169 (0.015)	0.274 (0.027)	0.126 (0.012)
NEUROHORIZON (tail and segment memory)	**0.281** (0.025)	**0.188** (0.015)	**0.295** (0.027)	**0.143** (0.011)
(b) Scheduled sampling				
No SS ($\epsilon = 0$)	**0.293** (0.024)	0.073 (0.025)	**0.296** (0.023)	−0.123 (0.022)
Low SS ($\epsilon_{\max} = 0.25$)	0.291 (0.023)	0.180 (0.013)	**0.296** (0.025)	0.128 (0.012)
NEUROHORIZON ($\epsilon_{\max} = 0.50$)	0.281 (0.025)	**0.188** (0.015)	0.295 (0.027)	**0.143** (0.011)

LFADS and NEUROHORIZON retain positive scores throughout that window, demonstrating predictive value beyond this baseline. NEUROHORIZON has higher per-bin scores than LFADS throughout the 100, 240, and 500 ms windows, complementing its higher whole-window means in Table 2. Nevertheless, the positive scores of both models late in the 1000 ms window are much lower than their early scores, indicating that the predictive gains captured by these models remain limited at longer delays.

5.3 Ablation Study

Long-horizon rollout depends on how the decoder represents its prediction history and how training exposes it to predicted inputs. We assess these two aspects by comparing prediction-memory configurations and scheduled-sampling settings at 500 and 1000 ms under TF and RO evaluation (Table 3).

Prediction memory. To examine the role of temporal resolution, we compare no memory, tail memory alone, segment memory alone, and their combination (Table 3(a)). The combined memory achieves the highest rollout means at both horizons, exceeding either single-memory configuration and the no-memory scores of 0.174 and 0.139. Its advantage over tail memory alone is larger at 1000 ms. These results support combining recent bin-level detail with segment summaries of the prediction prefix.

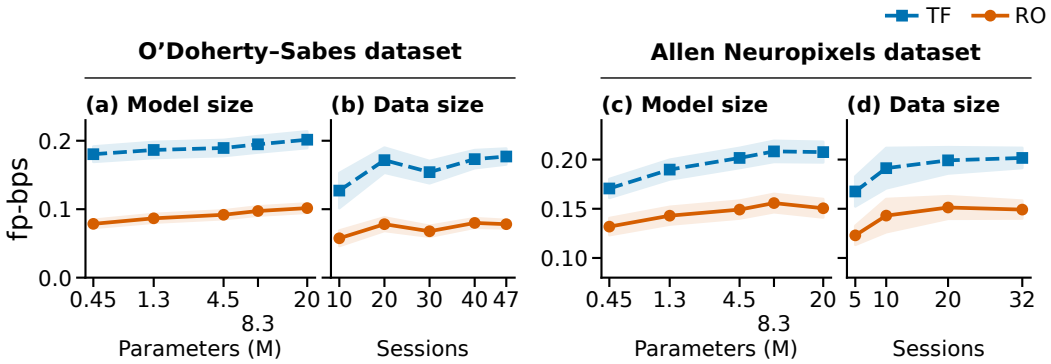

Figure 3: **Scaling analysis.** Panels (a,b) show results on the O'Doherty–Sabes dataset, and panels (c,d) show results on the Allen Neuropixels dataset. Model-size experiments use 47 and 32 recording sessions, respectively. Data-size experiments vary the number of training sessions and evaluate the sessions included at each size. All panels report session means with ± 1 SEM for TF and RO fp-bps, using 500 ms observation and prediction windows. Parameter counts are shown in millions and exclude neuron and session embeddings.

Table 4: **Cross-subject transfer with full fine-tuning.** Results on the O'Doherty–Sabes dataset of macaque sensorimotor recordings, with Indy as the source subject and Loco as the target subject. Pretraining on 37 Indy sessions followed by full fine-tuning on the training splits of five Loco sessions is compared with training from scratch on the same target data. Values are test-session means of fp-bps with SEM across the five Loco sessions. The highest mean in each column is bold.

	500 ms		**1000 ms**	
Setting	TF	RO	TF	RO
From scratch (5 Loco)	0.058 (0.008)	0.026 (0.007)	0.060 (0.010)	**0.018** (0.008)
Transfer full fine-tune (37 Indy $\rightarrow$ 5 Loco)	**0.082** (0.008)	**0.035** (0.009)	**0.076** (0.010)	0.015 (0.010)

Separate sensitivity analyses examine observed history and generated-history tail capacity at a 1000 ms prediction horizon. Increasing observation history from 0.5 to 1 and 2 s does not yield monotonic gains, so additional observations do not automatically improve rollout under these settings (Table S6). Reducing W_{tail} from 4 to 1 lowers performance, while increasing it to 8 does not improve it, indicating that a longer tail is not necessarily more effective (Table S7).

Scheduled sampling. Training on true prefixes differs from conditioning on predictions during rollout. We examine the effect of this mismatch by varying $\epsilon_{\max} \in \{0, 0.25, 0.50\}$ with the combined memory fixed (Table 3(b)). Without scheduled sampling, rollout fp-bps is 0.073 at 500 ms and -0.123 at 1000 ms. Scheduled sampling improves rollout at both horizons, with $\epsilon_{\max} = 0.50$ yielding the highest means of 0.188 and 0.143. TF scores remain comparatively similar across settings. The greater improvement in rollout supports exposing the decoder to its own predictions during training to sustain autonomous forecasting.

5.4 Scaling and Cross-Subject Transfer

Extending forecasting to broader recording settings raises questions about model capacity, training data size, and initialization for new subjects. We examine scaling on two datasets and transfer through cross-subject pretraining and fine-tuning.

Capacity and data scaling. To assess how forecasting changes with capacity and available recordings, we vary model size and the number of training sessions on the O'Doherty–Sabes dataset and the Allen Neuropixels dataset, using 500 ms observation and prediction windows (Figure 3). The five model configurations span 0.45–20.00 million backbone parameters (Table S8). Both datasets show overall improvements in TF and RO performance as model and data size increase, with smaller gains at

larger scales on the Allen Neuropixels dataset. These trends motivate further study of model capacity and recording scale as ways to improve forecasting.

Cross-subject transfer. Pretraining on another subject may provide a useful initialization for a new neural population. We test this by comparing pretraining on 37 Indy sessions followed by full fine-tuning on five Loco sessions with training from scratch on the same target data (Table 4). The architecture and target training settings are shared. Pretraining improves the TF and RO point estimates at 500 ms, and the TF improvement persists at 1000 ms. The 500 ms results indicate that source-subject pretraining can benefit autonomous prediction after adaptation to the target subject.

6 Discussion

We investigated how long neural population activity remains predictable from an initial spike history without further observations, comparing direct multi-step prediction and autoregressive rollout across horizons through whole-window scores and per-bin prediction decay. NEUROHORIZON, which combines hierarchical prediction memory with scheduled sampling, achieves the highest mean autonomous forecasting scores across the evaluated horizons on the Perich–Miller dataset and retains positive per-bin scores throughout the one-second window (Table 2; Figure 2). Its advantage narrows at 1000 ms, and late-window scores remain well below early-window scores. Ablations show that rollout benefits from combining tail and segment memory and degrades sharply without scheduled sampling (Table 3). Together with the scaling and transfer results, these findings indicate that motor-cortical population activity remains partially predictable for up to one second from initial spike history, and that prediction memory and scheduled sampling improve autonomous rollout.

Future work could examine how far autonomous forecasting extends and what information supports it at longer horizons. Evaluating larger models and datasets over longer horizons, and over a broader range of observation durations, would test whether the observed scaling trends hold beyond the evaluated windows. Analyzing learned representations, incorporating stimulus inputs, and relating predicted activity to behavior could further reveal which aspects of population dynamics remain predictive and whether forecasts benefit from information beyond spike history.

Acknowledgments and Disclosure of Funding

This work was supported by a grant from the National Natural Science Foundation of China (No. 92370116), the Beijing Major Science and Technology Project (Contract No. Z251100008125058), and the Tsinghua University Dushi Program. The authors declare no competing interests.

References

[1] Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15(10):805–815, 2018. doi: 10.1038/s41592-018-0109-9.

[2] Felix Pei, Joel Ye, David M. Zoltowski, Anqi Wu, Raeed H. Chowdhury, Hansem Sohn, et al. Neural Latents Benchmark '21: Evaluating latent variable models of neural population activity. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/979d472a84804b9f647bc185a877a8b5-Abstract-round2.html.

[3] Ziyu Lu, Anna J. Li, Alexander E. Ladd, Pascha Matveev, Aditya Deole, et al. Benchmarking probabilistic time series forecasting models on neural activity. arXiv:2510.18037, 2025. URL https://arxiv.org/abs/2510.18037. NeurIPS 2025 Workshop: Data on the Brain & Mind.

[4] Souhaib Ben Taieb, Gianluca Bontempi, Amir F. Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, 39(8):7067–7083, 2012. doi: 10.1016/j.eswa.2012.01.039.

[5] Yizi Zhang, Yanchen Wang, Donato M. Jiménez-Benetó, Zixuan Wang, Mehdi Azabou, Blake Richards, et al. Towards a "universal translator" for neural dynamics at single-cell, single-spike resolution. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2559. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/934eb45b99eff8f16b5cb8e4d3cb5641-Abstract-Conference.html.

[6] Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. Neural Data Transformer 2: Multi-context pretraining for neural spiking activity. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/fe51de4e7baf52e743b679e3bdba7905-Abstract-Conference.html.

[7] Antonis Antoniades, Yiyi Yu, Joe S. Canzano, William Yang Wang, and Spencer Smith. Neuroformer: Multimodal and multitask generative pretraining for brain data. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/b16e55ac49f93d33a4b87747d5121b64-Abstract-Conference.html.

[8] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/e995f98d56967d946471af29d7bf99f1-Abstract.html.

[9] Yu Duan, Hamza Tahir Chaudhry, Misha B. Ahrens, Christopher D. Harvey, Matthew G. Perich, Karl Deisseroth, et al. POCO: Scalable neural forecasting through population conditioning. In *Advances in Neural Information Processing Systems*, volume 38, pages 23631–23669, 2025. doi: 10.52202/085713-0798. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/21fbf654963f9e455e8eec48e45bdea0-Abstract-Conference.html.

[10] Jan-Matthis Lueckmann, Alexander Immer, Alex Bo-Yuan Chen, Peter H. Li, Mariela D. Petkova, Nirmala A. Iyer, et al. ZAPBench: A benchmark for whole-brain activity prediction in zebrafish. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/668563ef18fbfef0b66af491ea334d5f-Abstract-Conference.html.

[11] Tsai-Wen Chen, Trevor J. Wardill, Yi Sun, Stefan R. Pulver, Sabine L. Renninger, Amy Baohan, et al. Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458): 295–300, 2013. doi: 10.1038/nature12354.

[12] Thomas Deneux, Attila Kaszás, Gergely Szalay, Gergely Katona, Tamás Lakner, Amiram Grinvald, et al. Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo. *Nature Communications*, 7:12190, 2016. doi: 10.1038/ncomms12190.

[13] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.06732.

[14] 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 *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/953390c834451505703c9da45de634d8-Abstract-Conference.html.

[15] Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylKikSYDH.

[16] Matthew G. Perich, Lee E. Miller, Mehdi Azabou, and Eva L. Dyer. Long-term recordings of motor and premotor cortical spiking activity during reaching in monkeys. DANDI Archive, 2025. URL https://doi.org/10.48324/dandi.000688/0.250122.1735. Version 0.250122.1735.

[17] Wilson Truccolo, Uri T. Eden, Matthew R. Fellows, John P. Donoghue, and Emery N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, 93(2):1074–1089, 2005. doi: 10.1152/jn.00697.2004.

[18] Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael J. Mendelson, et al. A unified, scalable framework for neural population decoding. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8ca113d122584f12a6727341aaf58887-Abstract-Conference.html.

[19] Matthew Dowling, Yuan Zhao, and Il Memming Park. eXponential FAmily Dynamical Systems (XFADS): Large-scale nonlinear Gaussian state-space modeling. In *Advances in Neural Information Processing Systems*, volume 37, pages 13458–13488, 2024. doi: 10.52202/079017-0430. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/18595bc3e802a3b11035927fd928eb9c-Abstract-Conference.html.

[20] John R. Minnick, Jinghui Geng, Kamran Hussain, Jesus Gonzalez-Ferrer, Ash Robbins, Mohammed A. Mostajo-Radji, et al. SpikeProphecy: A large-scale benchmark for autoregressive neural population forecasting. arXiv:2605.12992, 2026. URL https://arxiv.org/abs/2605.12992.

[21] Konstantin F. Willeke, Polina Turishcheva, Alex Gilbert, Goirik Chakrabarty, Hasan A. Bedel, Paul G. Fahey, et al. OmniMouse: Scaling properties of multi-modal, multi-task brain models on 150B neural tokens. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d97122f599bdc6250c2334fa89c71606-Abstract-Conference.html.

[22] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019. doi: 10.18653/v1/P19-1285.

[23] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR, 2021. URL https://proceedings.mlr.press/v139/jaegle21a.html.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[25] Joseph E. O'Doherty, Mariana M. B. Cardoso, Joseph G. Makin, and Philip N. Sabes. Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology. Zenodo, 2017. URL https://doi.org/10.5281/zenodo.583331.

[26] Joshua H. Siegle, Xiaoxuan Jia, Séverine Durand, Sam Gale, Corbett Bennett, Nile Graddis, et al. Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature*, 592 (7852):86–92, 2021. doi: 10.1038/s41586-020-03171-x.

A Dataset Splits

We retain the training, validation, and test domains generated by the brainsets preprocessing pipelines for the Perich–Miller and O'Doherty–Sabes datasets. Perich–Miller uses trial-based partitions with buffers around held-out intervals, while O'Doherty–Sabes assigns ten temporal blocks to eight training blocks and one block each for validation and test, following the pipeline's movement-based allocation and filtering rules. For the Allen Neuropixels dataset, complete repeats of natural movie one and natural movie three are assigned chronologically to the three splits. We allocate 14 repeats of natural movie one to training, two to validation, and four to testing. For natural movie three, the corresponding numbers are seven, two, and one.

B Baseline Model Adaptations

We adapt the baseline models to forecasting from an initial neural history and evaluate their binned count predictions with the shared Poisson score and training-set null model.

IBL-MTM. We retain the self-supervised masking objective of IBL-MTM [5], alternating between neuron masks and causal masks with a base mask ratio of 0.3. Neuron masks hide subsets of neurons across the input window, while causal masks hide contiguous time spans. For Direct evaluation, the observed history provides context and all future target bins are masked and reconstructed jointly.

POCO. We retain the upstream POCO model core [9] and adapt its inputs and outputs to binned spike-count forecasting with a Poisson negative log-likelihood objective. The model receives only the observed history and predicts the entire future window in one forward pass, so we report it in Direct mode.

LFADS. We use the `lfads-torch` implementation of LFADS [1] in a multisession model with session-specific input and output mappings. Its encoder receives only the 500 ms observation counts, and the generator evolves autonomously over the future window with controller and external inputs disabled. The initial-condition latent and factors are 64-dimensional, and the generator state is 128-dimensional. Training combines a Poisson reconstruction objective over the history and future with initial-condition KL and recurrent-weight regularization, with checkpoints selected by validation ELBO. Future counts serve only as training targets, and forecasting is evaluated in RO mode.

Causal NDT2. We adapt the official multisession NDT2 model [6] from masked reconstruction to next-bin prediction under causal attention. Counts from bin $t-1$ are encoded at query time t, where a learned mask token queries bin t and the true target counts enter only the Poisson loss. The model is trained from scratch for 300 epochs with an effective batch size of 32 and a 500 ms observation window. Training and TF evaluation condition on true preceding counts, whereas RO feeds back counts sampled from the predicted Poisson distributions without access to true future activity.

Neuroformer. We retain the causal event-transformer components of Neuroformer [7]. Its feature, history, and state attention modules each contain six blocks. The model uses eight attention heads with hidden size 256 and dropout 0.2. History event blocks contain up to 512 events and current blocks contain up to 256, with 10 ms time tokens. Because the model generates events rather than a matrix of expected bin counts, we sample event sequences and assign their events to bins using the generated timestamps. For each evaluation window, we average the binned counts from 20 sampled sequences and apply a 10^{-4} floor before computing the Poisson count score. In TF evaluation, each event prediction conditions on the true event prefix, whereas RO incorporates generated events into the context for subsequent predictions.

C Training Configuration and Compute

The training settings for NEUROHORIZON on the Perich–Miller dataset are summarized in Table S1. Training uses bfloat16 mixed precision on a single NVIDIA RTX 4090 or RTX 4080 SUPER GPU. The 300-epoch runs take roughly 2–3 hours at 100 ms, 3.5 hours at 240 ms, 8–9 hours at 500 ms, and 17–18 hours at 1000 ms. On the RTX 4090 with 24 GB of memory, training at batch size 32 peaks at approximately 12–16 GB. TF and RO evaluation over the corresponding test set each complete in less than one hour.

Table S1: **Training hyperparameters.** Values correspond to the NeuroHorizon configuration used for experiments on the Perich–Miller dataset.

Hyperparameter	Value	Hyperparameter	Value
Encoder (POYO+)		*Training*	
Number of layers (L_{enc})	6	Optimizer	SparseLamb
Hidden dimension (d)	128	Peak learning rate	$3.125 \times 10^{-5} \times$ batch size
Attention heads	8 self, 2 cross	LR schedule	OneCycleLR with cosine annealing
Head dimension (d_{head})	64	Weight decay	10^{-4}
Dropout (attention, FFN)	0.2	Batch size	32
Dropout (input)	0.4	Epochs	300
Latent step	50 ms		
Latents per step	32		
Decoder		*Scheduled Sampling*	
Number of layers (L_{dec})	2	Schedule	Linear ramp, $\epsilon : 0 \rightarrow 0.5$
Hidden dimension (d)	128	Ramp duration	150 epochs
Tokens per tail bin (K_{tail})	4	*Memory Robustness*	
Prediction memory attention heads	4	Input noise (σ_{noise})	0.03
Tail memory window (W_{tail})	4 bins (80 ms)	Input dropout (p_{in})	0.05
Segment chunk size (S)	5 bins (100 ms)	Block dropout (p_{block})	0.15
Segment tokens per chunk (K_{seg})	1		

D Session-Level Forecasting Results

To examine how the overall forecasting scores reflect individual recordings, we compare session-level performance across the four prediction horizons on the Perich–Miller dataset in Table S2. NEUROHORIZON maintains positive rollout fp-bps in every session and at every horizon, and exceeds both IBL-MTM and POCO in all 40 session–horizon comparisons. These results indicate that the higher mean scores relative to the direct baselines reflect improvements across recordings and prediction horizons.

E Evaluation with R^2

To complement the likelihood-based score, we evaluate how closely predicted expected counts match observed binned counts on the same Perich–Miller test recordings. For each session s, we pool all valid evaluation windows, prediction bins, and neurons to compute

$$R_s^2 = 1 - \frac{\sum_{w,\tau,n}(\hat{\mu}_{w,\tau,n} - y_{w,\tau,n})^2}{\sum_{w,\tau,n}(y_{w,\tau,n} - \bar{y}_s)^2}, \tag{8}$$

where $\bar{y}_s$ is the mean observed count over those entries within session s. We then report the mean and SEM across sessions, rather than averaging separate neuron-level R^2 values. NEUROHORIZON achieves the highest mean autonomous R^2 at each horizon in Table S3, complementing the comparison based on fp-bps. TF results use additional true-prefix information and are reported separately.

F Forecasting on the MC_Maze Dataset

To assess forecasting beyond the Perich–Miller recordings, we compare NEUROHORIZON, IBL-MTM, and POCO on the single MC_Maze recording from the Neural Latents Benchmark [2]. Each model observes 500 ms of neural activity and forecasts the next 500 ms using the same data splits and evaluation windows. The official training trials are divided into local training and validation sets, while the official validation trials form the local test set. This is a forecasting adaptation of MC_Maze, not an official NLB benchmark score. NEUROHORIZON achieves higher autonomous fp-bps than both direct baselines, providing complementary evidence of forecasting performance under a different recording condition.

Table S2: **Per-session forward-prediction results (fp-bps).** C, J, and M denote Chewie, Jango, and Mihili, the three monkeys in the Perich–Miller dataset. Sessions are numbered chronologically within each monkey. Each horizon separates autonomous forecasts (Direct and RO) from TF evaluations. The final column reports the session mean (SEM).

Horizon (ms)	Model (mode)	C1	C2	C3	C4	J1	J2	J3	M1	M2	M3	Mean (SEM)
100	IBL-MTM (Direct)	0.171	0.147	0.139	0.198	0.266	0.290	0.299	0.273	0.243	0.237	0.226 (0.019)
	POCO (Direct)	0.133	0.104	0.091	0.147	0.245	0.254	0.268	0.163	0.168	0.199	0.177 (0.020)
	LFADS (RO)	0.163	0.159	0.121	0.186	0.264	0.275	0.262	0.253	0.207	0.225	0.212 (0.017)
	Causal NDT2 (RO)	0.176	0.165	0.144	0.193	0.270	0.297	0.303	0.269	0.270	0.259	0.235 (0.019)
	NEUROHORIZON (RO)	0.192	0.185	0.156	0.222	0.294	0.345	0.331	0.283	0.267	0.262	0.254 (0.020)
	Causal NDT2 (TF)	0.222	0.219	0.204	0.237	0.364	0.385	0.376	0.322	0.324	0.310	0.296 (0.022)
	NEUROHORIZON (TF)	0.212	0.199	0.170	0.234	0.358	0.379	0.378	0.299	0.291	0.288	0.281 (0.024)
240	IBL-MTM (Direct)	0.185	0.153	0.121	0.184	0.224	0.251	0.248	0.229	0.207	0.194	0.200 (0.013)
	POCO (Direct)	0.140	0.110	−0.016	0.143	0.170	0.170	0.216	0.127	0.159	0.142	0.136 (0.019)
	LFADS (RO)	0.180	0.148	0.132	0.201	0.207	0.269	0.256	0.238	0.211	0.200	0.204 (0.014)
	Causal NDT2 (RO)	0.127	0.123	0.067	0.134	0.192	0.138	0.197	0.222	0.186	0.178	0.156 (0.015)
	NEUROHORIZON (RO)	0.189	0.173	0.144	0.201	0.241	0.299	0.299	0.240	0.249	0.208	0.224 (0.016)
	Causal NDT2 (TF)	0.228	0.235	0.198	0.242	0.373	0.427	0.370	0.320	0.302	0.297	0.299 (0.024)
	NEUROHORIZON (TF)	0.226	0.216	0.176	0.236	0.369	0.399	0.383	0.288	0.292	0.267	0.285 (0.024)
500	IBL-MTM (Direct)	0.128	0.096	0.075	0.132	0.125	0.142	0.183	0.155	0.092	0.077	0.120 (0.011)
	POCO (Direct)	0.079	0.066	0.028	0.082	0.063	0.169	0.134	0.039	−0.028	0.066	0.070 (0.017)
	LFADS (RO)	0.145	0.121	0.102	0.152	0.202	0.204	0.235	0.234	0.171	0.157	0.172 (0.014)
	Causal NDT2 (RO)	0.080	0.043	0.002	0.057	0.092	0.097	0.160	0.172	0.137	0.109	0.095 (0.017)
	NEUROHORIZON (RO)	0.151	0.140	0.115	0.157	0.205	0.247	0.258	0.234	0.191	0.177	0.188 (0.015)
	Causal NDT2 (TF)	0.220	0.210	0.191	0.231	0.368	0.363	0.364	0.329	0.310	0.304	0.289 (0.022)
	NEUROHORIZON (TF)	0.211	0.202	0.182	0.218	0.367	0.392	0.380	0.298	0.277	0.285	0.281 (0.025)
1000	IBL-MTM (Direct)	0.116	0.028	0.025	0.022	−0.079	−0.072	−0.029	0.043	−0.081	−0.154	−0.018 (0.025)
	POCO (Direct)	0.063	0.037	0.002	0.050	0.044	0.049	0.066	0.017	0.017	0.004	0.035 (0.007)
	LFADS (RO)	0.137	0.112	0.087	0.124	0.121	0.151	0.208	0.174	0.142	0.138	0.139 (0.011)
	Causal NDT2 (RO)	0.049	0.008	−0.012	0.058	0.075	−0.113	0.121	0.015	0.087	0.063	0.035 (0.021)
	NEUROHORIZON (RO)	0.165	0.129	0.105	0.140	0.118	0.130	0.191	0.207	0.128	0.114	0.143 (0.011)
	Causal NDT2 (TF)	0.219	0.220	0.177	0.229	0.354	0.389	0.406	0.294	0.317	0.307	0.291 (0.025)
	NEUROHORIZON (TF)	0.225	0.216	0.183	0.231	0.356	0.420	0.421	0.312	0.296	0.292	0.295 (0.027)

Table S3: **Spike-count prediction accuracy.** Session means of R^2 with SEM on the Perich–Miller dataset. Autonomous forecasts use only the initial history, while TF evaluation receives true prefixes.

Model (mode)	100 ms	240 ms	500 ms	1000 ms
Autonomous forecasting				
IBL-MTM (Direct)	0.245 (0.017)	0.237 (0.018)	0.211 (0.019)	0.175 (0.018)
POCO (Direct)	0.230 (0.017)	0.217 (0.018)	0.200 (0.019)	0.183 (0.017)
Neuroformer (RO)	0.128 (0.018)	0.096 (0.014)	-2.722 (1.341)	-1.079 (0.449)
LFADS (RO)	0.242 (0.017)	0.238 (0.018)	0.226 (0.019)	0.209 (0.016)
Causal NDT2 (RO)	0.248 (0.017)	0.217 (0.018)	0.197 (0.015)	0.174 (0.021)
NEUROHORIZON (RO)	0.255 (0.018)	0.244 (0.018)	0.232 (0.018)	0.210 (0.017)
TF evaluation				
Neuroformer (TF)	0.142 (0.014)	0.145 (0.016)	0.173 (0.022)	0.116 (0.015)
Causal NDT2 (TF)	0.274 (0.018)	0.274 (0.020)	0.270 (0.021)	0.270 (0.018)
NEUROHORIZON (TF)	0.266 (0.019)	0.268 (0.021)	0.267 (0.021)	0.271 (0.018)

G Qualitative Population Forecasts

To visualize population activity beyond the aggregate scores, Figure S1 compares observed counts with NEUROHORIZON predictions under RO and TF across the four horizons. For display, each predicted expected count $\hat{\mu}$ is converted to the mean of five samples from $\mathrm{Poisson}(\hat{\mu})$. This sampling is used only for visualization, while autonomous rollout feeds back predicted expected counts. The

Table S4: **Forecasting on MC_Maze.** Global fp-bps and R^2 for 500 ms forecasts from 500 ms of observed activity. Autonomous forecasts and TF evaluation are reported separately.

Model (mode)	**Global fp-bps**	R^2
Autonomous forecasting		
IBL-MTM (Direct)	0.2475	0.1067
POCO (Direct)	0.2162	0.1036
NEUROHORIZON (RO)	0.2722	0.1114
TF evaluation		
NEUROHORIZON (TF)	0.3357	0.1226

examples show broad correspondence in activity levels across neurons alongside differences in individual spike counts. Quantitative comparisons rely on the forecasting scores and per-bin analysis.

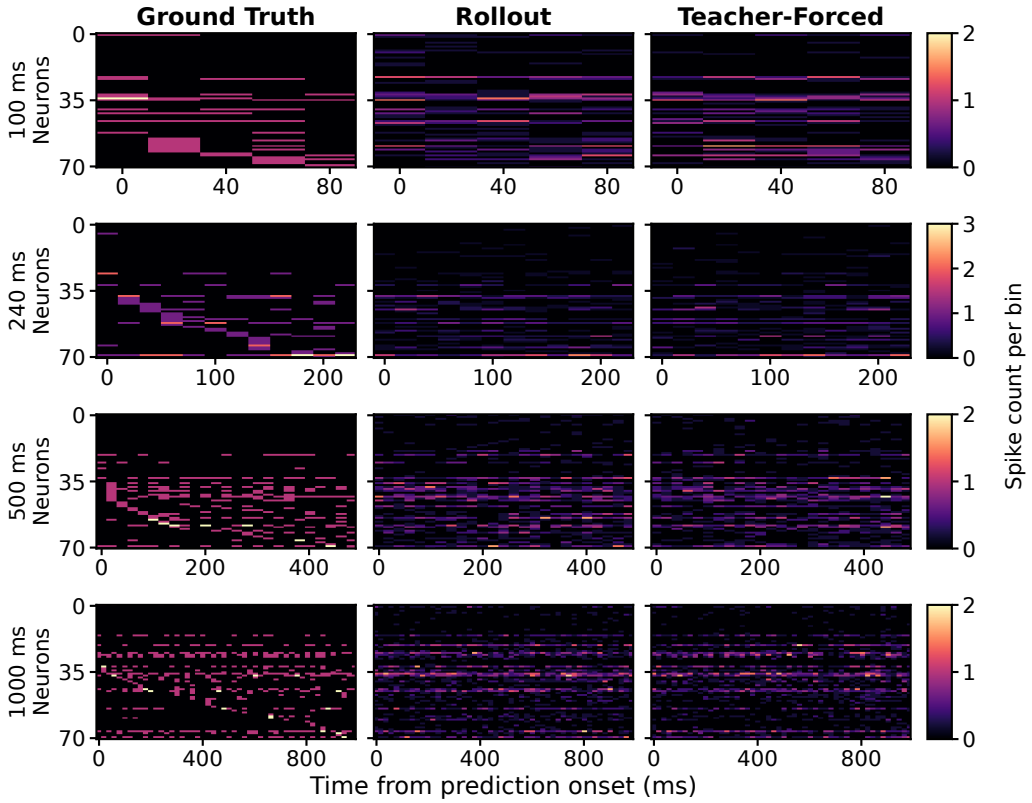

Figure S1: **Population forecasts across horizons.** Rows show 100, 240, 500, and 1000 ms forecasts for 71 neurons. Columns show observed counts, rollout predictions, and teacher-forced predictions. Colors indicate counts per bin.

H Forecasting on Lorenz-Driven Spike Trains

We use a controlled Lorenz system to test whether autonomous neural forecasts retain information about known latent trajectories. We discard the initial 2 s of each independent three-dimensional trajectory to reduce initial transients. The latent coordinates are standardized using training-set statistics and mapped through a fixed nonlinear readout to the firing rates of a 64-unit Poisson population sampled in 20 ms bins. We compare NEUROHORIZON with causal NDT2 on this synthetic dataset. Both models use the same 4,200 trajectories for training, 900 for validation, and 900 for test. Each observes 500 ms of spikes and autonomously forecasts the next 500 ms.

Table S5 reports spike forecasting relative to a training-split Poisson null (fp-bps), recovery of the known generating rates (Rate R^2), and two latent-trajectory measures. For each model, a ridge-regularized affine map from predicted log firing rates to ground-truth Lorenz coordinates is fitted on the validation set and frozen for test evaluation. Latent R^2 compares these aligned predictions with the test latents. Trajectory correlation is the Pearson correlation after flattening the test trajectories and their coordinates. Figure S2 illustrates the aligned trajectories. NEUROHORIZON has slightly higher latent R^2 and trajectory correlation, while causal NDT2 has higher fp-bps and Rate R^2. These results indicate that the forecasts retain information about the underlying latent trajectories, with neither model leading on all four metrics.

Table S5: **Forecasting on Lorenz-driven spike trains.** Autonomous 500 ms forecasts after a 500 ms observation. Metrics are evaluated on held-out trajectories, with latent alignment fitted on validation predictions.

Model	fp-bps	Rate R^2	Latent R^2	Trajectory corr.
NEUROHORIZON	0.1368	0.5535	0.7225	0.9517
Causal NDT2	0.1595	0.6835	0.7078	0.9491

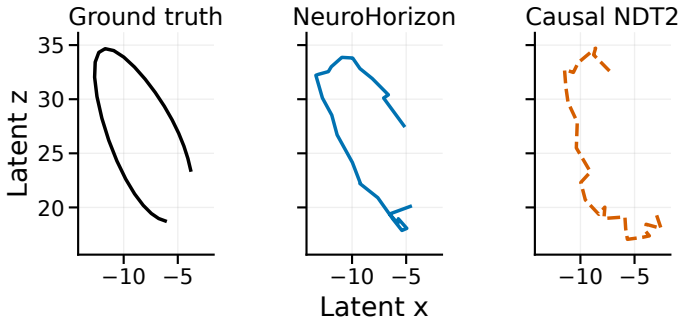

Figure S2: **Example Lorenz trajectories over a 500 ms forecast.** A held-out trajectory and validation-aligned predictions from NEUROHORIZON and causal NDT2, projected onto the x–z plane.

I Sensitivity to Observation History and Tail Length

Observation history. To examine whether additional observations improve long-horizon forecasting, we compare 0.5, 1, and 2 s of observed neural history on the Perich–Miller dataset with a 1000 ms prediction horizon (Table S6). The default setting uses 0.5 s of history, and models for 1 s and 2 s are trained from scratch. All three settings use 300 training epochs. Extending observation history does not produce a monotonic rollout improvement under these settings.

Table S6: **Observation-history sensitivity.** Autonomous 1000 ms prediction on the Perich–Miller dataset. Rollout fp-bps is reported as mean ± SEM across sessions. Δ is the mean difference from the 0.5 s reference.

Observation history	Rollout fp-bps	Δ vs. 0.5 s
0.5 s	0.1427 ± 0.0107	—
1.0 s	0.1442 ± 0.0172	$+0.0015$
2.0 s	0.1381 ± 0.0317	-0.0046

Tail length. To assess how much recent prediction history to retain, we use 500 ms of observed activity and vary the number W_{tail} of recent generated bins retained at high resolution while keeping the segment-memory pathway fixed (Table S7). Each tail bin contributes four memory tokens. Segment summaries use one token per five-bin group. At the final prediction step, the decoder sees $4W_{\text{tail}} + 10$ prediction-history memory tokens, including the tail and segment summaries. Reducing

the tail from $W_{\text{tail}} = 4$ to $W_{\text{tail}} = 1$ lowers rollout fp-bps by 0.0148 on average across sessions. Increasing it to $W_{\text{tail}} = 8$ does not improve performance. Among the tested configurations, $W_{\text{tail}} = 4$ gives the highest mean rollout and last-five-bin scores.

Table S7: **Generated-history tail sensitivity.** Perich–Miller dataset, 500 ms observation and 1000 ms autonomous prediction, with the segment-memory configuration fixed. Visible tokens count prediction-history memory tokens available at the final query. Rollout fp-bps is the session mean ± SEM. Last-five-bin fp-bps is the arithmetic mean of the five global per-bin scores covering 900–1000 ms (20 ms bins).

Tail W_{tail}	**Visible tokens**	**Rollout fp-bps**	**Last-five-bin fp-bps**
1	14	0.1279 ± 0.0117	0.0561
4	26	0.1427 ± 0.0107	0.0770
8	42	0.1297 ± 0.0112	0.0720

J Model Configurations for Scaling Experiments

Table S8 defines the five model sizes used in Figure 3 for the O'Doherty–Sabes and Allen Neuropixels datasets. The configurations vary hidden width, encoder and decoder depth, and attention configuration together.

Table S8: **Parameter-scaling model definitions.** Configurations used for model-size experiments on the O'Doherty–Sabes dataset and the Allen Neuropixels dataset. Parameter counts exclude neuron and session embeddings.

Size	**Hidden dim**	**Encoder depth**	**Decoder depth**	**Backbone params**
Tiny	64	3	1	0.45M
Small	96	4	2	1.30M
Base	128	6	2	4.52M
Large	192	8	3	8.33M
XLarge	256	12	4	20.00M

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are supported by the method and experimental results in Sections 3 and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses the scope of the current evaluation and directions for future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present theoretical results requiring proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 3–4 and Appendices A–C describe the model, data splits, baseline adaptations, training configuration, and evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The neural recording datasets used in this work are publicly available from their original sources. The NEUROHORIZON source code is publicly available at https://github.com/Jp-17/NeuroHorizon.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Section 4 describes the experimental settings, with dataset splits in Appendix A and training hyperparameters for the Perich–Miller experiments in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The main comparison, ablation, scaling, and transfer results report session means with standard errors of the mean (SEM) across sessions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C reports the hardware, GPU memory usage, and training and evaluation times for the main experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is foundational research on neural population forecasting for scientific study. It is not tied to societal applications or deployments, and we do not foresee direct societal impacts from the work presented.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released asset is research code for offline neural forecasting and does not include sensitive data or high-risk pretrained models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets, software, and baseline models used in this work are appropriately credited through citations to their original sources.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new NEUROHORIZON code repository organizes the model implementation, configurations, and training and evaluation scripts with accompanying installation and usage documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing or human-subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research does not involve human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not involved in the core method development of this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.