
Inpainting the Neural Picture: Inferring Unrecorded Brain Area Dynamics from Multi-Animal Datasets

Ji Xia

Center for Theoretical Neuroscience,
Zuckerman Mind Brain Behavior Institute,
Kavli Institute for Brain Science,
Columbia University
jx2484@columbia.edu

Yizi Zhang

Department of Statistics,
Center for Theoretical Neuroscience,
Zuckerman Mind Brain Behavior Institute,
Kavli Institute for Brain Science,
Grossman Center for the Statistics of Mind,
Columbia University

Shuqi Wang

Laboratory of Computational Neuroscience,
École polytechnique fédérale de Lausanne (EPFL)

Genevera I. Allen

Department of Statistics,
Irving Center for Cancer Dynamics,
Center for Theoretical Neuroscience,
Zuckerman Mind Brain Behavior Institute,
Kavli Institute for Brain Science,
Columbia University

Liam Paninski

Department of Statistics,
Center for Theoretical Neuroscience,
Zuckerman Mind Brain Behavior Institute,
Kavli Institute for Brain Science,
Grossman Center for the Statistics of Mind,
Columbia University

Cole Lincoln Hurwitz

Center for Theoretical Neuroscience,
Zuckerman Mind Brain Behavior Institute,
Kavli Institute for Brain Science,
Grossman Center for the Statistics of Mind,
Columbia University

Kenneth D. Miller

Center for Theoretical Neuroscience,
Zuckerman Mind Brain Behavior Institute,
Kavli Institute for Brain Science,
Columbia University

Abstract

Characterizing interactions between brain areas is a fundamental goal of systems neuroscience. While such analyses are possible when areas are recorded simultaneously, it is rare to observe all combinations of areas of interest within a single animal or recording session. How can we leverage multi-animal datasets to better understand multi-area interactions? Building on recent progress in large-scale, multi-animal models, we introduce **NeuroPaint**, a masked autoencoding approach for inferring the dynamics of unrecorded brain areas. By training across animals with overlapping subsets of recorded areas, NeuroPaint learns to reconstruct activity in missing areas based on shared structure across individuals. We train and evaluate our approach on synthetic data and two multi-animal, multi-area Neuropixels datasets. Our results demonstrate that models trained across animals with partial observations can successfully in-paint the dynamics of unrecorded areas, enabling

multi-area analyses that transcend the limitations of any single experiment. Code is available at the following github repository: NeuroPaint

1 Introduction

Understanding how brain areas coordinate their activity to support complex behaviors, such as memory-guided actions and perceptual decision-making, is a central challenge in systems neuroscience [17]. Advances in electrophysiological recording techniques now enable single-cell, single-spike resolution measurements across dozens of interconnected brain areas [25, 43, 44]. These developments have driven the collection of brain-wide datasets from hundreds of behaving mice, creating new opportunities to study distributed neural computation at scale [28, 10, 27, 30]. Capitalizing on these datasets requires computational methods capable of modeling inter-area interactions across animals and tasks.

Traditional approaches to modeling inter-area interactions rely on pairwise neuronal correlations [12] or low-dimensional linear methods such as communication subspace analysis [39]. More expressive models, such as DLAG [15], mDLA [16], and dCSFA [22], capture aspects of temporal and spatial dependencies but typically assume a fixed communication structure and can scale poorly to large datasets. Most recently, large-scale, multi-animal approaches like multi-task-masking (MtM) have shown that pretraining across animals with recordings from overlapping brain areas can improve cross-area prediction [61]. However, these methods are limited to modeling only the areas observed within a given recording session and do not take advantage of a key opportunity in multi-animal datasets: using data from multiple recording sessions across animals to infer activity in unrecorded areas of any given session.

In this paper, we introduce a new framework for analyzing multi-animal, multi-area recordings, NeuroPaint, that explicitly leverages shared anatomical structure across animals to model both recorded and unrecorded brain areas. Building on the masked transformer architecture introduced in [61], our approach incorporates two key innovations: (1) a masking strategy tailored for unrecorded areas, in which unrecorded brain areas are treated the same as masked areas during inference; (2) an architecture that models latent dynamics separately for each brain area, enabling interpretation of area-to-area interactions. To our knowledge, this is the first method specifically designed to infer the dynamics of unrecorded brain areas using data from both other animals and other sessions of the same animal.

We evaluate our method on synthetic data and two large-scale, brain-wide mouse datasets [9, 28]. We find that training across animals with overlapping subsets of recorded brain areas enables NeuroPaint to reliably infer the latent dynamics of missing areas. We compare our approach to LFADS [46] and generalized linear regression, in which we directly predict activity in missing areas from observed activity, demonstrating that our inferred latents provide significantly more predictive information about unrecorded areas. Taken together, this work introduces a new paradigm for analyzing brain-wide activity, demonstrating how shared structure across animals can be exploited to in-paint dynamics from missing brain areas and opening new possibilities for multi-area analyses that transcend the limitations of single-subject recordings.

2 Related work

Multi-area models. Advances in electrophysiology now enable simultaneous recordings across multiple brain areas [25, 44, 59, 50]. To study concurrent signaling among distributed neuronal populations [1, 24, 28], generative models such as DLAG and mDLA have been developed to uncover shared latent dynamics across brain areas [22, 15, 16]. While these models offer interpretability, they typically assume fixed bidirectional communication structures and suffer from limited scalability. Multi-area recurrent dynamical models have also been proposed to capture inter-area interactions more flexibly [20, 35, 34, 14]. However, none of these approaches can infer the dynamics of brain areas that are unrecorded during a given session.

Stitching or quilting multi-session recordings. Recently, several groups have developed so-called “stitching” [42] or “quilting” [53] approaches to infer functional connectivity or latent dynamics from multi-session recordings with partially overlapping sets of neurons. Some of these methods

directly estimate the covariance [5, 6, 63], noise correlations [41, 55], or functional connectivity using generalized linear models [42] and graphical models [53, 7], while others infer latent factors or neural dynamics across sessions [51, 31]. Among these, LFADS [33] is most relevant to our work, as it infers latent dynamics using a nonlinear state space model and can be extended to multi-session settings via linear stitchers. However, state space approaches process time points sequentially and impose strong assumptions on temporal dynamics and noise structure, limiting their scalability and flexibility in large-scale, multi-animal datasets. Moreover, these approaches cannot directly infer neural dynamics in unrecorded brain areas.

Large-scale models for neural analysis. Recent work suggests that scaling models across animals and brain areas can be beneficial, as neural activity exhibits shared structure across individuals and regions [37, 2, 58, 60–62]. Transformer-based models have emerged as powerful tools for modeling these multi-animal neural datasets, including methods such as POYO+ [3], a multi-task decoder across animals; NDT [58], which uses masked modeling for neural prediction; and MtM [61] and NEDS [62], which utilize multi-task-masking to learn spatiotemporal structure in neural activity and the bidirectional relationship between neural activity and behavior. However, these models overlook a key opportunity in multi-animal, multi-area datasets: inferring neural dynamics in unrecorded brain areas of one animal by leveraging shared structure across animals.

3 Methods

In this work, we present **NeuroPaint**, a transformer-based masked modeling approach that predicts neural dynamics in both recorded and unrecorded brain areas using observed activity. In multi-animal extracellular recordings, each session samples neurons from only a subset of brain areas, leaving others unrecorded (Fig. 1A, 3A, E). With sufficient overlap across animals, shared structure can be leveraged to infer missing activity. To model this, we randomly mask recorded areas and train the model to reconstruct them, using a set of low-dimensional latents for each brain area (even when unrecorded), to enable inference of missing brain area dynamics across sessions.

3.1 Architecture

The architecture of NeuroPaint has four main components (Fig. 1C): (1) a cross-attention [23] stitcher that maps neural activity from each unmasked brain area to area-specific embedding factors; (2) a tokenizer that transforms these embedding factors into tokens and adds mask tokens for masked and unrecorded areas; (3) a transformer encoder that processes all tokens to produce latent factors for each brain area; and (4) a generalized linear stitcher that reconstructs neural activity from latent factors via a linear readout followed by an exponential nonlinearity. Most parameters are shared across sessions, with a few session-specific parameters noted below.

Cross-attention read-in stitcher. We use a cross-attention stitcher (shown in Fig. 1B) to map neural activity into a shared latent space, producing area-specific “embedding factors” that are consistent across sessions. This module builds on the Perceiver-IO architecture [23, 2], using latent tokens to reduce input length. Neural activity is tokenized at the neuron level, with each neuron providing keys and values. A fixed set of learnable latent tokens act as queries, ensuring that the output embedding factors have the same size as the latent tokens. To construct each neuron token, we concatenate the neural activity embedding with three learnable embeddings: an area embedding (encoding the neuron’s brain area), a hemisphere embedding (indicating left or right hemisphere), and a unit embedding (unique to each neuron). The cross-attention stitcher then outputs a distinct set of embedding factors for each brain area. For a more detailed description of the cross-attention stitcher, see Appendix A.2.

We adopt a cross-attention stitcher [2, 3] in place of a linear stitcher [33, 61, 57], as the input-side transformation must be highly expressive. Neural activity can vary substantially across sessions, and aligning it into consistent across-session dynamics requires non-linear transformations. At the same time, minimizing the number of parameters is critical to avoid overfitting. The cross-attention stitcher supports parameter sharing across sessions and brain areas, improving generalization. Only the unit embeddings remain session-specific, as each session includes a distinct set of recorded neurons.

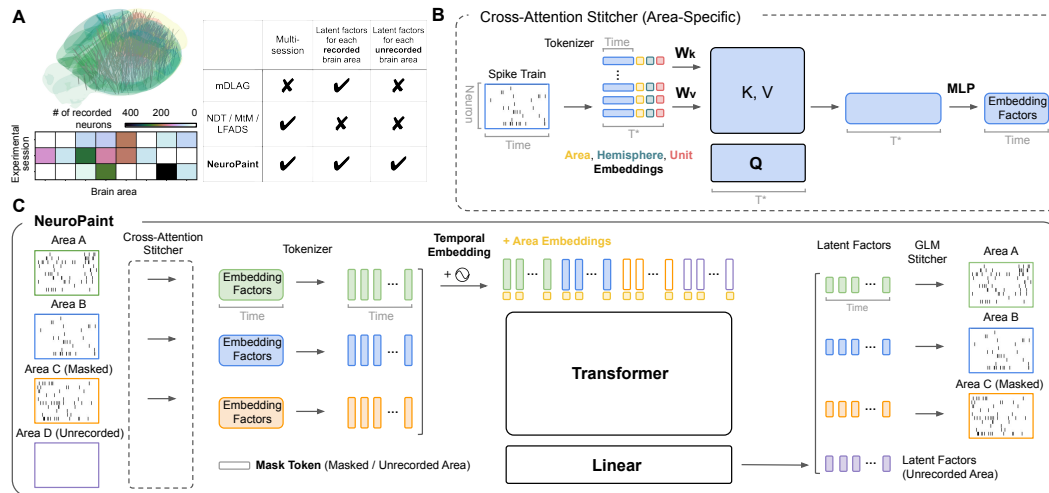


Figure 1: Schematic illustration of NeuroPaint. (A) Neuropixels probes only record a subset of brain areas simultaneously. The brain schematic summarizes probe locations across IBL sessions, and the table below shows variability in recorded neuron counts by area and session. In contrast to previous approaches (e.g., mDLAG [16], NDT [57, 58], MtM [61], LFADS [33]), NeuroPaint is the first method that can infer latent dynamics for each brain area, including unrecorded areas, using large-scale multi-session, multi-animal datasets. (B) We use a cross-attention stitcher to convert spike data into tokens, and concatenate area, hemisphere, and unit embeddings to form the keys and values. We use learnable latent tokens as queries to reduce input length and produce area-specific embedding factors. Bold symbols indicate parameters shared across sessions, while non-bold symbols denote area-specific parameters. (C) NeuroPaint uses a transformer-based architecture with cross-attention stitchers (as shown in B) to encode spike counts into area-specific embedding factors that are then tokenized into temporal tokens. We add temporal and area embeddings to the tokens, which are passed through the transformer and a linear layer to produce area-specific latent factors. During training, we mask tokens from sampled brain areas, and predict them using area-specific GLM stitchers.

Tokenizer for embedding factors. The cross-attention stitcher produces area-specific embedding factors, which are then tokenized by treating each time step as a separate token. We replace tokens corresponding to masked areas with a learnable mask token. For each unrecorded area, we also include learnable mask tokens, which allows the model to infer latent factors for unrecorded areas at test time. Inspired by masked autoencoders, which use positional embeddings to preserve the spatial location of image patches [21], we add a brain-area embedding to each token to encode its anatomical location. Although area information is already added in the cross-attention stitcher, adding it to mask tokens is crucial to ensure that the model can distinguish which brain area each token represents. Temporal information is encoded using rotary positional embeddings (RoPE) [45, 2].

Transformer encoder. To process the sequence of tokens from multiple areas and animals, we utilize an encoder-only transformer architecture composed of standard transformer blocks followed by a linear layer [52]. By computing interactions between all pairs of tokens in the input sequence, the self-attention mechanism of the transformer encoder allows the model to capture dependencies across brain areas and time steps. The encoder outputs latent factors for all brain areas of interest, including those that were unrecorded in a given session.

Generalized linear read-out stitcher. A generalized linear read-out stitcher maps the latent factors back to neural activity using a linear layer followed by an exponential activation. This component is kept intentionally simple to preserve the interpretability of latent factors, ensuring that those for unrecorded areas closely reflect actual neural activity. Its parameters are specific to each session and brain area. The number of latent factors per area is chosen based on the maximum linear dimensionality of that area's smoothed neural activity across sessions (see Appendix A.5).

3.2 Masking scheme

To enable the model to infer missing activity from unrecorded areas, we utilize an inter-area masking scheme: in each training batch, we randomly select a masking percentage between 0% and 60%,

and mask out that proportion of the recorded brain areas. We then train the model to reconstruct their activity. Since our goal is to predict activity in unrecorded brain areas, inter-area masking alone is insufficient for generalization. To address this, we introduce additional loss terms designed to improve the model's ability to infer activity in unrecorded areas.

3.3 Loss function

The loss function consists of three components: the reconstruction loss, consistency loss, and regularization loss. The **reconstruction loss** is a standard Poisson negative log-likelihood and is used to evaluate the accuracy of predicted firing rates against observed spike counts. The **consistency loss** constrains each brain area's embedding factors to preserve a stable correlation structure between factors across sessions by penalizing deviations from a session-averaged target correlation (see Appendix A.3 and A.4 for details). This constraint helps the model generalize to unrecorded areas in a given session by encouraging the embedding factors to encode information predictive of all observed areas across sessions. Importantly, the consistency loss is applied to embedding factors (see Fig. 1C for definition), which are related to the neural activity through learned nonlinear transformations. That is, we do not assume stable correlation between areas at the level of raw spike data or latent factors. Finally, the **regularization loss** penalizes rapid temporal fluctuations in the latent space, promoting smoother latent dynamics. Formal definitions of the consistency and regularization losses can be found in Appendix A.3.

4 Experiments

Our approach builds upon two core assumptions: (1) neural activity in each brain area lies on a **low-dimensional** manifold, allowing a fixed set of low-dimensional latent factors to explain most of the variance in the high-dimensional neural data; and (2) a **consistent and potentially nonlinear mapping** exists between the latent dynamics of different brain areas. To test our proposed model, we first apply it to a synthetic dataset constructed to satisfy both assumptions. We then evaluate its performance on two large-scale Neuropixels datasets spanning multiple animals and brain areas, where these assumptions are expected to hold approximately.

In the synthetic data, ground truth firing rates from unrecorded areas are available. For the real data, *we treat one recorded area per session as unrecorded* by holding it out during training, allowing evaluation during test with ground-truth data. We measure how well the inferred latent factors capture dynamics in the held-out area by training, during the test phase, a generalized linear model (GLM) to predict firing rates for that area from its inferred factors. Note that these GLMs cannot be learned during training for the held-out areas because their spike data are not made available during training.

All trials used for evaluating the latent factors come from a test set that was excluded during NeuroPaint training. We split this test set into 60% training and 40% testing trials for cross-validation of the GLM performance. Performance is quantified using deviance fraction explained (DFE), a normalized goodness-of-fit metric (see Appendix A.7). We use DFE as an intuitive, bounded metric where 1 indicates perfect prediction, 0 matches a null model using the average firing rate, and negative values indicate worse-than-null performance. However, when the null model closely matches the ground truth, DFE can be low even if the model performs well. Conversely, DFE values near 1 may reflect overfitting—not in the conventional sense of poor generalization across data splits, but in that the model may capture high-frequency spike noise rather than meaningful structure in the firing rates.

4.1 Datasets

Synthetic dataset. To generate synthetic neural data, we simulate an autonomous recurrent neural network (RNN) comprising five brain areas, each with 200 RNN units (see Appendix A.6). Units within each area are densely connected (all-to-all), while inter-area connections are sparse (1%). The recurrent weights induce chaotic dynamics, resulting in trial-to-trial variability driven by different initial conditions. Counterintuitively, the chaotic dynamics generated by such randomly connected networks are often low-dimensional—occupying a subspace whose dimensionality is much smaller than the number of RNN units [11]. For each synthetic session, spike trains are produced using a session-specific GLM with sparse weights (2%), mapping RNN activity to spikes. This setup yields distinct neuron populations per session, each arising from partially overlapping subsets of RNN units.

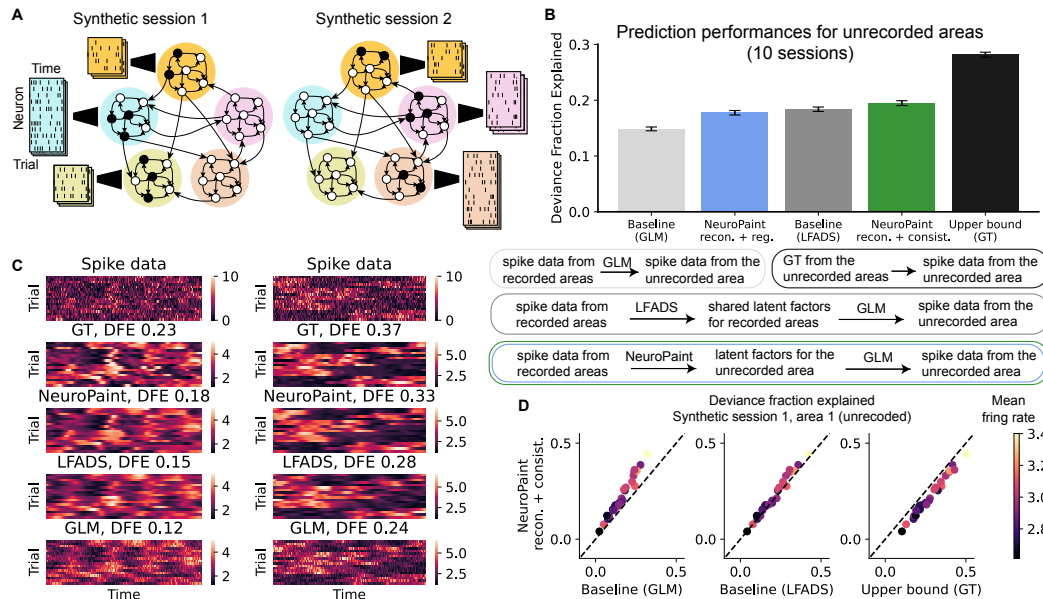


Figure 2: Quantitative and qualitative comparison of NeuroPaint with baseline models on synthetic datasets. (A) Synthetic spike trains are generated across 10 sessions using a shared underlying RNN, with five brain areas of 200 units each. Spike trains are generated using session- and area-specific GLMs with sparse, random weights, allowing partially overlapping RNN units to contribute differently across sessions. (B) Prediction performance for synthetic neurons in unrecorded areas across 10 sessions. The bar plot displays the mean DFE and one standard error across neurons for each method, while the flow diagram shows the computation of DFE. (C) Spike data and ground truth firing rates (GT) for two example synthetic neurons, compared with firing rate predictions from NeuroPaint (trained with reconstruction and consistency losses) and the two baselines (GLM and LFADS). (D) Prediction performance for synthetic neurons from area 1 (unrecorded) in session 1. Each dot represents a neuron, showing DFE achieved by NeuroPaint (trained with reconstruction and consistency loss), the two baselines, and the upper bound. Dots are color-coded by each neuron's mean firing rate.

In each session, 3 ~ 4 out of 5 areas are designated as “recorded,” and the rest as unrecorded. Each area contains 20 ~ 60 simulated neurons per session.

IBL dataset. We use the International Brain Laboratory (IBL) brain-wide map dataset [28] (Fig. 3A, B). This dataset consists of Neuropixels recordings collected from 12 labs which utilize a standardized experimental pipeline. The recordings target 279 brain areas across 139 adult mice performing the same visual decision-making task. The probe was localized after the experiments using reconstructed histology and the brain areas were annotated. We utilize trial-aligned, spike-sorted data from 20 mice (1 session per mice), and select 8 brain areas for analysis: PO, LP, DG, CA1, VISa, VPM, APN, and MRN (see Appendix A.1 for full area names). From these recordings, we have a total of 21568 neurons for training and evaluation. We bin the spike trains using 10 ms windows and we fix the trial-length to 2 seconds (200 time bins). Trials are categorized into two types: left choice and right choice. Each of the 20 selected sessions has 3 to 7 brain areas simultaneously recorded. We randomly hold out one recorded brain area per session for evaluation.

MAP dataset. The Mesoscale Activity Project (MAP) dataset records neural activity underlying memory-guided movement in mice using Neuropixels probes targeting motor cortex, thalamus, midbrain, and hindbrain regions, spanning 293 cortical and subcortical structures [10] (Fig. 3E, F). It includes 173 sessions from 28 mice. We use trial-aligned, spike-sorted data from 40 sessions across 16 mice, binned in 10 ms windows over fixed 4-second trials (400 time bins). Analyses are restricted to two trial types, *hit left* and *hit right*, which are also used to compute the consistency loss. For our analysis, we focus on 8 brain areas, 6 highly relevant to the task [19, 29, 18, 49, 9] and 2 orbital areas: ALM, lOrb, vOrb, Pallidum, Striatum, VAL-VM, MRN, and SC (see Appendix A.1 for full area names). Each of the selected sessions has 4 to 6 brain areas simultaneously recorded. We randomly hold out one recorded brain area per session for evaluation.

4.2 Baselines

We compare NeuroPaint against linear and non-linear baselines. We use a standard generalized linear model (GLM) for our linear baseline. For our non-linear baseline, we compare to Latent Factor Analysis via Dynamical Systems (LFADS) [33, 46], a sequential variational autoencoder that has demonstrated strong performance in capturing latent neural dynamics. We utilize a re-implemented version of LFADS [38] for all our analyses, as discussed below.

GLM. We implement GLMs that map from instantaneous spike activity in all the recorded areas to instantaneous spike activity in an unrecorded (in the synthetic dataset) or held-out (in the Neuropixels datasets) area. The GLM consists of a linear layer followed by an exponential nonlinearity, and is trained using the Poisson negative log-likelihood loss. We fit a separate GLM for each session and each held-out area.

LFADS. We re-implement multi-session LFADS [46, 33] to model shared neural dynamics across multiple animals. The objective of LFADS is to maximize the likelihood of observed neural activity by reconstructing spike trains from a low-dimensional latent dynamical system. For each session's neural population, multi-session LFADS learns a linear projection layer (read-in stitcher) to embed the neural activity into a shared latent space. It also learns a corresponding set of read-out stitchers to map the latent representations back to neural activity space. Unlike NeuroPaint, LFADS embeds activity from all recorded areas into a shared latent space but cannot infer latent dynamics for unrecorded areas, as it is only trained to reconstruct observed data and does not use masked modeling to predict missing areas. Most importantly, LFADS lacks area-specific latent factors, which limits its interpretability in multi-area regimes. We apply principal component regression to select the latent dimensionality and to pre-condition both the read-in and read-out stitchers [33] (see Appendix A.5). Further details on architecture, weight initialization, and implementation can be found in Appendix A.10 and A.11.

In summary, we evaluate the ability of each approach to predict neural activity in unrecorded areas, by using a supervised GLM (trained during the test phase) to predict neural activity in held-out areas. Specifically, we compare the predictive power of three types of inputs to the GLM: (1) spike activity from recorded areas (corresponding to the GLM baseline), (2) LFADS latent factors shared across recorded areas, and (3) NeuroPaint's area-specific latent factors for the unrecorded area. Although we are comparing NeuroPaint's performance with these other baselines, they are not comparable in the important sense that NeuroPaint is the only approach that explicitly learns separate latent factors for each brain area, including those unrecorded ones, whereas the two baselines cannot infer area-specific latent dynamics.

5 Results

5.1 Synthetic dataset

We evaluate NeuroPaint on a synthetic dataset with known ground truth firing rates, enabling computation of an upper-bound DFE using spike data in the unrecorded areas and true rates. The model is trained on recorded areas from 10 synthetic sessions and evaluated on its ability to infer single-neuron firing rates in unrecorded areas. The architecture follows Section 3.1, excluding hemisphere embeddings due to the absence of hemispheric structure in the simulation. We compare two model variants, one trained with reconstruction plus regularization loss, the other with reconstruction plus consistency loss; combining all three losses (not shown) results in over-regularization and performance degradation below baseline. This is likely due to the fast, high-firing-rate dynamics in the synthetic data; as suggested by an additional experiment on a separate low-firing-rate synthetic dataset, where the full model (all three losses) outperforms the ablated variant lacking one loss term (see Appendix A.6 and Supplementary Fig. 2). We compare each model against GLM and LFADS baselines and the upper bound. The consistency-loss variant outperforms the two baselines and the regularized model, and is the closest to the upper bound (Fig. 2B, D). As shown in Fig. 2C, NeuroPaint more accurately captures temporal structure than GLM, which tends to overfit to Poisson noise, and slightly outperforms LFADS in recovering fine-grained features of the ground truth firing rates. These results show that under the idealized conditions of the synthetic data, *i.e.* low-dimensional activity and consistent cross-area mapping, NeuroPaint can accurately infer neural dynamics in unrecorded areas.

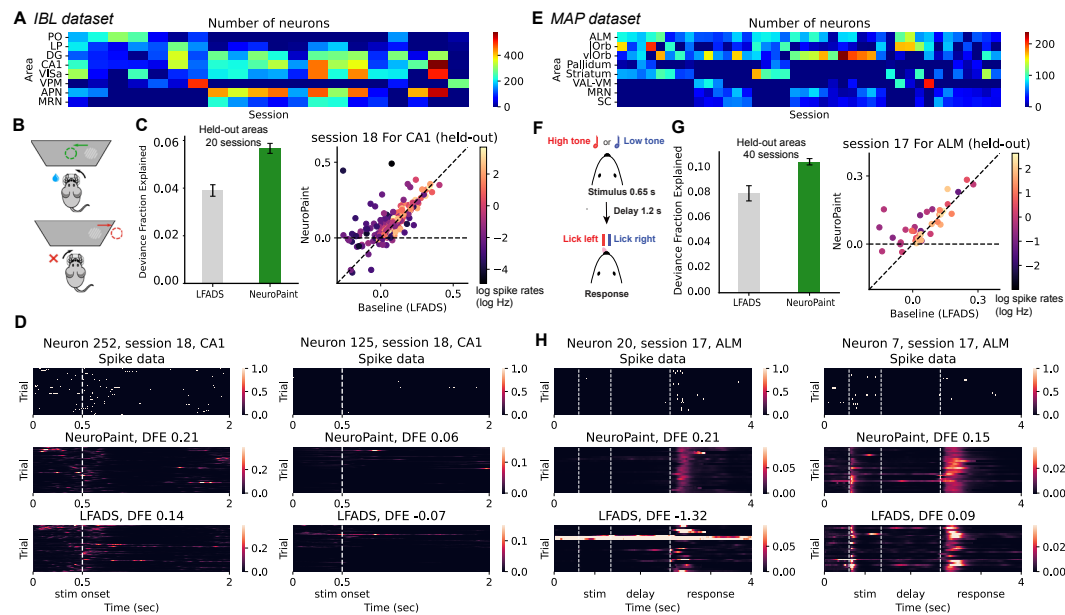


Figure 3: Quantitative and qualitative comparison of NeuroPaint with baseline models on IBL and MAP datasets. (A) Number of neurons recorded per area in 20 IBL sessions. (B) In the IBL experimental setup, mice perform a visual decision-making task by turning a wheel to the left or right to indicate whether the stimulus is presented on the left or right screen [28]. (C) Left: The mean and standard error of DFE for LFADS and NeuroPaint, calculated for neurons from held-out areas pooled across 20 IBL sessions, excluding the worst 1% of neurons for visualization; see Fig. S1A for results without exclusion. Right: Per-neuron DFE for the held-out area (CA1) of session 18. (D) Spike trains and predicted firing rates for two example neurons, compared between NeuroPaint and LFADS predictions. (E) Number of neurons recorded per area in 40 sessions from MAP dataset. (F) The MAP dataset records neural activity from mice performing a memory-guided directional licking task, in which they are instructed to lick left or right according to auditory tones presented before a fixed delay period. [9]. (G, H) show similar content to (C, D), but with neurons from held-out areas pooled across 40 MAP sessions (see Fig. S1B for results with the worst 1% exclusion).

5.2 IBL and MAP datasets

In the Neuropixels datasets, although we have ground truth spiking data, the absence of ground truth firing rates precludes computation of an upper-bound DFE. While absolute DFE values may appear lower than those observed in the synthetic data, they reflect the intrinsic performance ceiling set by the low and sparsely modulated firing rates typical of real neuronal activity. Rather than indicating suboptimal performance, these values underscore the challenge of the task and the robustness of the model's performance under realistic biological constraints.

We train two separate NeuroPaint models, one for the 20 IBL sessions and one for the 40 MAP sessions. In both datasets, we compare NeuroPaint to LFADS, which is the highest performing baseline for single-neuron prediction accuracy in held-out areas. The GLM baseline performs poorly across both datasets due to its inability to model non-linear, cross-area interactions in real neural data [56, 54], and is therefore omitted from the comparison (results are provided in Appendix A.9).

Across held-out areas, NeuroPaint outperforms LFADS by a large margin in both the IBL and MAP datasets (Fig. 3C, G; Fig. S1). Example raster plots of individual neurons further show that NeuroPaint captures structured, single-trial dynamics that LFADS fails to recover (Fig. 3D, H). In particular, LFADS exhibits large outlier errors for some neurons (e.g., neuron 20 in Fig. 3H), whereas NeuroPaint delivers more stable and accurate firing rate predictions. In the neural data, which has relatively low firing rates compared to the synthetic data of Fig. 2, the combination of all three loss terms performs best: ablation experiments show that removing either the regularization or the consistency loss degrades performance relative to the full NeuroPaint model (see Appendix A.13, Supplementary Fig. 1). Moreover, increasing the number of parameters in the LFADS model does not close the performance gap to NeuroPaint (see Appendix A.15), suggesting that NeuroPaint's advantage is not explained by model size alone. These results demonstrate that NeuroPaint not

only infers interpretable, area-specific latent factors for unrecorded brain regions, but also achieves state-of-the-art predictive performance.

5.3 Interpretable area-specific latent dynamics revealed by NeuroPaint

We use the MAP dataset to illustrate that NeuroPaint generates consistent and interpretable area-specific latent dynamics across trials and sessions. See similar results for IBL dataset in Appendix A.14.

Preprocessing. To highlight dynamic structure over static offsets, we subtract the temporal mean of each trial from each latent factor before computing correlations or visualizing dynamics in Fig. 4. This preprocessing step helps mitigate spurious correlations driven by differences in the latent factor's non-zero temporal means.

Consistent and context-dependent latent dynamics. We evaluate (1) the consistency of latent factors across trials for both recorded and unrecorded areas, and (2) the context-dependent variability of latent factors that reflects distinct behavioral conditions, such as *hit left* vs. *hit right* trials where mice respond by licking in different directions. We find that the inferred area-specific latent factors exhibit consistent temporal structure across trials within the same behavioral context (Fig. 4A), even in sessions where the corresponding area is unrecorded (Fig. 4B). Additionally, the latent factors exhibit clear context-dependent variability that captures task-relevant differences; for instance, the latent factors for area SC show distinct patterns between hit left and hit right trials (Fig. 4A). To quantify the consistency and context-dependent variability, we compute the Pearson correlation between latent factors across all trial pairs, averaged over brain areas (see Appendix Table 5). First, latent factors from the same brain areas exhibit positive correlations across trials, regardless of whether those areas were recorded or not in a given session. Second, trial pairs of the same type have higher correlations than those of different types, suggesting that the latent factors capture context-dependent variability. (see Appendix A.16 for more details)

To directly evaluate whether the latent factors capture stimulus- or behavior-dependent variability, we performed decoding analysis from the latent factors. We found that the latent factors in most areas reliably support decoding for both stimulus and behavioral choice, demonstrating that they encode meaningful, task-relevant information. (see Appendix A.17 for more details)

Inferring area-to-area interactions from inpainted neural data. We demonstrate that NeuroPaint's inferred latent factors enable novel analyses of area-to-area interactions during behavior. As shown in Fig. 4C, the latent factors from both recorded and unrecorded areas in a single trial reveal rich and distinct temporally structured activity across the 8 selected brain areas. With a fully inpainted neural picture at a single-trial resolution across 40 sessions, NeuroPaint enables analyses on area-to-area interactions that were previously infeasible with Neuropixels recordings, which lack simultaneous coverage of all areas. As one such analysis, we quantify representational similarity across all pairs of brain areas. For each area and each trial period (stimulus, delay, and response), we first compute a time-by-time representational dissimilarity matrix (RDM), defined as one minus the pairwise correlation between flattened latent factors. We then measure area-to-area similarity by correlating the upper-triangular elements of these RDMs, averaged across trials over multiple sessions (Fig. 4D). This analysis reveals that inter-area relationships evolve across behavioral epochs: similarity is relatively low during stimulus and delay periods but increases during the response period, which is consistent with previous findings of increased inter-area coordination during motor output [9]. While Chen et al. [9] were limited to analyzing interactions between ALM and a few simultaneously recorded areas, NeuroPaint overcomes this limitation by enabling comparisons across all area pairs, even including unrecorded areas, via latent inpainting. Furthermore, we observe that the two orbital areas (lOrb and vOrb) exhibit high mutual similarity, reflecting their shared anatomical and functional roles [32]. In contrast, these areas show weak similarity to the remaining brain areas, even during the response period, which aligns with prior knowledge about the involvement of the selected non-orbital areas in task-relevant computations [19, 29, 18, 49, 9].

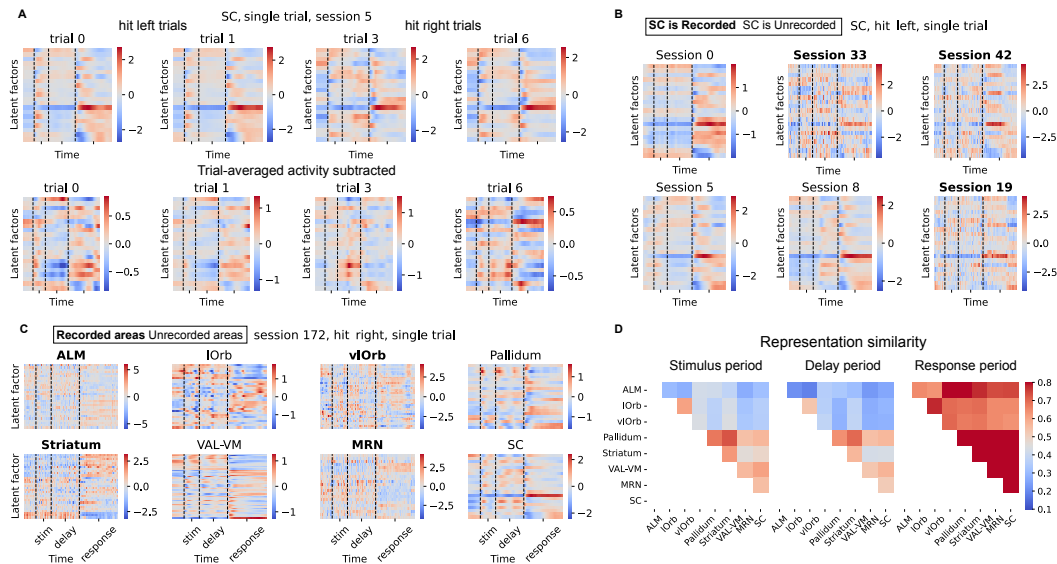


Figure 4: NeuroPaint's area-specific latent factors are consistent, capture context-dependent variability, and enable inference of area-to-area interactions. (A) Inferred latent factors for the Superior Colliculus (SC), an unrecorded area, during example trials across the *hit left* and *hit right* conditions in a single session. The bottom row shows the same latent factors with the trial-averaged activity subtracted. (B) Inferred single-trial latent factors for SC (both recorded and unrecorded) across six sessions. (C) Latent factors for all eight areas (both recorded and unrecorded) in a *hit right* trial from a single session in the MAP dataset. (D) Representation similarity analysis (RSA) of latent factors across eight brain areas (recorded and unrecorded) for the stimulus, delay, and response periods in the MAP dataset, with values averaged across trials and pooled over sessions.

6 Discussion

In this work, we introduce NeuroPaint, a masked transformer-based model that infers the dynamics of unrecorded brain areas by leveraging shared activity structure across animals. Our experiments on synthetic and large-scale Neuropixels datasets show that NeuroPaint outperforms both linear and non-linear baselines, including LFADS and GLMs, in predicting single-neuron activity in held-out areas. Beyond predictive accuracy, NeuroPaint produces interpretable latent dynamics that are consistent across sessions and sensitive to behavioral context, enabling new forms of cross-area and cross-session analysis that were inaccessible with existing tools.

Despite these strengths, several limitations remain. First, training NeuroPaint is computationally expensive, as the self-attention mechanism scales quadratically with the number of brain areas and time steps. Future work could address this by incorporating sparse or low-rank attention mechanisms [8]. Second, the latent dimensionality for each brain area is not manually selected, it is estimated via a principled procedure (see Appendix A.5). Nonetheless, the procedure involved design choices that introduce some arbitrariness, and we find that the selected number of latent factors often exceeds the intrinsic dimensionality of the predicted firing rates in each brain area (see Appendix A.18). More automated hyperparameter selection methods, such as Population Based Training [26], could reduce this arbitrariness. Finally, while we demonstrate proof-of-concept results on two datasets, our experiments involve a limited subset of brain areas and sessions relative to the full datasets. Scaling to hundreds of sessions is straightforward, but extending to hundreds of brain areas will require architectural innovations and improved training strategies. Nonetheless, this represents a critical step toward building truly brain-wide models.

Acknowledgments and Disclosure of Funding

This project was supported by the Simons foundation (543017), National Institutes of Health (1RF1DA056397, 1K99NS144599), the Gatsby Foundation. This work used NCSA Delta GPU at National Center for Supercomputing Applications through allocation OTH250001 from the Ad-

vanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by U.S. National Science Foundation grants (2138259, 2138286, 2138307, 2137603, and 2138296). We are grateful to Susu Chen, Thinh Nguyen, Nuo Li for discussions about accessing the MAP dataset. We thank Mehdi Azabou for discussion about the design of cross-attention stitcher. We thank Manuel Beiran and David Clark for discussion about neural dynamics in random recurrent neural networks used in the synthetic dataset.

References

- [1] Misha B Ahrens, Jennifer M Li, Michael B Orger, Drew N Robson, Alexander F Schier, Florian Engert, and Ruben Portugues. Brain-wide neuronal dynamics during motor adaptation in zebrafish. *Nature*, 485 (7399):471–477, 2012.
- [2] 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.
- [3] Mehdi Azabou, Krystal Xuejing Pan, Vinam Arora, Ian Jarratt Knight, Eva L Dyer, and Blake Aaron Richards. Multi-session, multi-task neural decoding from distinct cell-types and brain regions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=IuU0wc00mo>.
- [4] Kush Banga, Julius Benson, Jai Bhagat, Dan Biderman, Daniel Birman, Niccolò Bonacchi, Sebastian A Bruijns, Kelly Buchanan, Robert AA Campbell, Matteo Carandini, et al. Reproducibility of in vivo electrophysiological measurements in mice. *eLife*, 13:RP100840, 2025.
- [5] William E Bishop and Byron M Yu. Deterministic symmetric positive semidefinite matrix completion. *Advances in Neural Information Processing Systems*, 27, 2014.
- [6] Andersen Chang, Lili Zheng, and Genevera I Allen. Low-rank covariance completion for graph quilting with applications to functional connectivity. *arXiv preprint arXiv:2209.08273*, 2022.
- [7] Andersen Chang, Lili Zheng, Gautam Dasarathy, and Genevera I Allen. Nonparanormal graph quilting with applications to calcium imaging. *Stat*, 12(1):e623, 2023.
- [8] Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. *Advances in Neural Information Processing Systems*, 34:17413–17426, 2021.
- [9] Susu Chen, Yi Liu, Ziyue Aiden Wang, Jennifer Colonell, Liu D. Liu, Han Hou, Nai-Wen Tien, Tim Wang, Timothy Harris, Shaul Druckmann, Nuo Li, and Karel Svoboda. Brain-wide neural activity underlying memory-guided movement. *Cell*, 187(3):676–691.e16, February 2024. ISSN 00928674. doi: 10.1016/j.cell.2023.12.035. URL <https://linkinghub.elsevier.com/retrieve/pii/S0092867423014459>.
- [10] Susu Chen, Yi Liu, Ziyue Aiden Wang, Jennifer Colonell, Liu D Liu, Han Hou, Nai-Wen Tien, Tim Wang, Timothy Harris, Shaul Druckmann, et al. Brain-wide neural activity underlying memory-guided movement. *Cell*, 187(3):676–691, 2024.
- [11] David G Clark, LF Abbott, and Ashok Litwin-Kumar. Dimension of activity in random neural networks. *Physical Review Letters*, 131(11):118401, 2023.
- [12] Marlene R Cohen and Adam Kohn. Measuring and interpreting neuronal correlations. *Nature neuroscience*, 14(7):811–819, 2011.
- [13] Peiran Gao and Surya Ganguli. On simplicity and complexity in the brave new world of large-scale neuroscience. *Current opinion in neurobiology*, 32:148–155, 2015.
- [14] Joshua Glaser, Matthew Whiteway, John P Cunningham, Liam Paninski, and Scott Linderman. Recurrent switching dynamical systems models for multiple interacting neural populations. *Advances in neural information processing systems*, 33:14867–14878, 2020.
- [15] Evren Gokcen, Anna I Jasper, João D Semedo, Amin Zandvakili, Adam Kohn, Christian K Machens, and Byron M Yu. Disentangling the flow of signals between populations of neurons. *Nature Computational Science*, 2(8):512–525, 2022.

- [16] Evren Gokcen, Anna Jasper, Alison Xu, Adam Kohn, Christian K Machens, and Byron M Yu. Uncovering motifs of concurrent signaling across multiple neuronal populations. *Advances in Neural Information Processing Systems*, 36, 2024.
- [17] Joshua I Gold and Michael N Shadlen. The neural basis of decision making. *Annu. Rev. Neurosci.*, 30(1): 535–574, 2007.
- [18] Zengcai V. Guo, Hidehiko K. Inagaki, Kayvon Daie, Shaul Druckmann, Charles R. Gerfen, and Karel Svoboda. Maintenance of persistent activity in a frontal thalamocortical loop. *Nature*, 545(7653):181–186, May 2017. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature22324. URL <https://www.nature.com/articles/nature22324>.
- [19] Zengcai V. Guo, Nuo Li, Daniel Huber, Eran Ophir, Diego Gutnisky, Jonathan T. Ting, Guoping Feng, and Karel Svoboda. Flow of Cortical Activity Underlying a Tactile Decision in Mice. *Neuron*, 81(1):179–194, January 2014. ISSN 08966273. doi: 10.1016/j.neuron.2013.10.020. URL <https://linkinghub.elsevier.com/retrieve/pii/S0896627313009240>.
- [20] Diksha Gupta, Charles D Kopec, Adrian G Bondy, Thomas Z Luo, Verity A Elliott, and Carlos D Brody. A multi-region recurrent circuit for evidence accumulation in rats. *BioRxiv*, 2024.
- [21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
- [22] Rainbo Hultman, Kyle Ulrich, Benjamin D Sachs, Cameron Blount, David E Carlson, Nkemdilim Ndubuizu, Rosemary C Bagot, Eric M Parise, Mai-Anh T Vu, Neil M Gallagher, et al. Brain-wide electrical spatiotemporal dynamics encode depression vulnerability. *Cell*, 173(1):166–180, 2018.
- [23] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.
- [24] Xiaoxuan Jia, Joshua H Siegle, Séverine Durand, Gregory Heller, Tamina K Ramirez, Christof Koch, and Shawn R Olsen. Multi-regional module-based signal transmission in mouse visual cortex. *Neuron*, 110(9): 1585–1598, 2022.
- [25] James J Jun, Nicholas A Steinmetz, Joshua H Siegle, Daniel J Denman, Marius Bauza, Brian Barbarits, Albert K Lee, Costas A Anastassiou, Alexandru Andrei, Çağatay Aydın, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017.
- [26] Mohammad Reza Keshtkaran, Andrew R Sedler, Raeed H Chowdhury, Raghav Tandon, Diya Basrai, Sarah L Nguyen, Hansem Sohn, Mehrdad Jazayeri, Lee E Miller, and Chethan Pandarinath. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nature Methods*, 19(12):1572–1577, 2022.
- [27] Andrei Khilkevich, Michael Lohse, Ryan Low, Ivana Orsolic, Tadej Bozic, Paige Windmill, and Thomas D Mrsic-Flogel. Brain-wide dynamics linking sensation to action during decision-making. *Nature*, 634 (8035):890–900, 2024.
- [28] International Brain Laboratory, Brandon Benson, Julius Benson, Daniel Birman, Niccolò Bonacchi, Kcénia Bougrova, Sebastian A Bruijns, Matteo Carandini, Joana A Catarino, Gaelle A Chapuis, et al. A brain-wide map of neural activity during complex behaviour. *bioRxiv*, pages 2023–07, 2023.
- [29] Nuo Li, Tsai-Wen Chen, Zengcai V. Guo, Charles R. Gerfen, and Karel Svoboda. A motor cortex circuit for motor planning and movement. *Nature*, 519(7541):51–56, March 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14178. URL <https://www.nature.com/articles/nature14178>.
- [30] Camden J MacDowell, Alexandra Libby, Caroline I Jahn, Sina Tafazoli, Adel Ardalan, and Timothy J Buschman. Multiplexed subspaces route neural activity across brain-wide networks. *Nature communications*, 16(1):3359, 2025.
- [31] Marcel Nonnenmacher, Srinivas C Turaga, and Jakob H Macke. Extracting low-dimensional dynamics from multiple large-scale neural population recordings by learning to predict correlations. *Advances in neural information processing systems*, 30, 2017.
- [32] Seung Wook Oh, Julie A Harris, Lydia Ng, Brent Winslow, Nicholas Cain, Stefan Mihalas, Quanxin Wang, Chris Lau, Leonard Kuan, Alex M Henry, et al. A mesoscale connectome of the mouse brain. *Nature*, 508 (7495):207–214, 2014.

- [33] 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.
- [34] Matthew G Perich and Kanaka Rajan. Rethinking brain-wide interactions through multi-region 'network of networks' models. *Current opinion in neurobiology*, 65:146–151, 2020.
- [35] Matthew G Perich, Charlotte Arlt, Sofia Soares, Megan E Young, Clayton P Mosher, Juri Minxha, Eugene Carter, Ueli Rutishauser, Peter H Rudebeck, Christopher D Harvey, et al. Inferring brain-wide interactions using data-constrained recurrent neural network models. *BioRxiv*, pages 2020–12, 2020.
- [36] Lorenzo Posani, Shuqi Wang, Samuel P Muscinelli, Liam Paninski, and Stefano Fusi. Rarely categorical, always high-dimensional: how the neural code changes along the cortical hierarchy. *bioRxiv*, pages 2024–11, 2025.
- [37] Mostafa Safaie, Joanna C Chang, Junchol Park, Lee E Miller, Joshua T Dudman, Matthew G Perich, and Juan A Gallego. Preserved neural dynamics across animals performing similar behaviour. *Nature*, 623(7988):765–771, 2023.
- [38] Andrew R Sedler and Chethan Pandarinath. Ifads-torch: A modular and extensible implementation of latent factor analysis via dynamical systems. *arXiv preprint arXiv:2309.01230*, 2023.
- [39] João D Semedo, Amin Zandvakili, Christian K Machens, Byron M Yu, and Adam Kohn. Cortical areas interact through a communication subspace. *Neuron*, 102(1):249–259, 2019.
- [40] Haim Sompolinsky, Andrea Crisanti, and Hans-Jurgen Sommers. Chaos in random neural networks. *Physical review letters*, 61(3):259, 1988.
- [41] Oleksandr Sorochynskyi, Stéphane Deny, Olivier Marre, and Ulisse Ferrari. Predicting synchronous firing of large neural populations from sequential recordings. *PLoS computational biology*, 17(1):e1008501, 2021.
- [42] Daniel Soudry, Suraj Keshri, Patrick Stinson, Min-hwan Oh, Garud Iyengar, and Liam Paninski. Efficient "shotgun" inference of neural connectivity from highly sub-sampled activity data. *PLoS computational biology*, 11(10):e1004464, 2015.
- [43] Nicholas A. Steinmetz, Peter Zatzka-Haas, Matteo Carandini, and Kenneth D. Harris. Distributed coding of choice, action and engagement across the mouse brain. *Nature*, 576(7786):266–273, December 2019. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-019-1787-x. URL <https://www.nature.com/articles/s41586-019-1787-x>.
- [44] Nicholas A Steinmetz, Cagatay Aydin, Anna Lebedeva, Michael Okun, Marius Pachitariu, Marius Bauza, Maxime Beau, Jai Bhagat, Claudia Böhm, Martijn Broux, et al. Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588, 2021.
- [45] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [46] David Sussillo, Rafal Jozefowicz, LF Abbott, and Chethan Pandarinath. Lfads-latent factor analysis via dynamical systems. *arXiv preprint arXiv:1608.06315*, 2016.
- [47] Karel Svoboda, susu chen, and Nuo Li. Single cell reconstruction data from MouseLight database and related Matlab analysis scripts. 9 2023. doi: 10.25378/janelia.24088431.v1. URL https://janelia.figshare.com/articles/dataset/Single_cell_reconstruction_data_from_MouseLight_database_and_related_Matlab_analysis_scripts/24088431.
- [48] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.
- [49] Alyse Thomas, Weiguo Yang, Catherine Wang, Sri Laasya Tipparaju, Guang Chen, Brennan Sullivan, Kylie Swiekatowski, Mahima Tatam, Charles Gerfen, and Nuo Li. Superior colliculus bidirectionally modulates choice activity in frontal cortex. *Nature Communications*, 14(1):7358, November 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-43252-9. URL <https://www.nature.com/articles/s41467-023-43252-9>.
- [50] Eric M Trautmann, Janis K Hesse, Gabriel M Stine, Ruobing Xia, Shude Zhu, Daniel J O'Shea, Bill Karsh, Jennifer Colonell, Frank F Lanfranchi, Saurabh Vyas, et al. Large-scale high-density brain-wide neural recording in nonhuman primates. *bioRxiv*, pages 2023–02, 2023.

- [51] Sridhar Turaga, Lars Buesing, Adam M Packer, Henry Dalglish, Noah Pettit, Michael Hausser, and Jakob H Macke. Inferring neural population dynamics from multiple partial recordings of the same neural circuit. *Advances in Neural Information Processing Systems*, 26, 2013.
- [52] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- [53] Giuseppe Vinci, Gautam Dasarathy, and Genevera I Allen. Graph quilting: graphical model selection from partially observed covariances. *arXiv preprint arXiv:1912.05573*, 2019.
- [54] Martin Vinck, Cem Uran, Georgios Spyropoulos, Irene Onorato, Ana Clara Broggin, Marius Schneider, and Andres Canales-Johnson. Principles of large-scale neural interactions. *Neuron*, 111(7):987–1002, 2023.
- [55] Adrien Wöhrer, Ranulfo Romo, and Christian K Machens. Linear readout from a neural population with partial correlation data. *Advances in Neural Information Processing Systems*, 23, 2010.
- [56] Ji Xia, Anna Jasper, Adam Kohn, and Kenneth D Miller. Circuit-motivated generalized affine models characterize stimulus-dependent visual cortical shared variability. *Science*, 27(8), 2024.
- [57] Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.
- [58] 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, 2024.
- [59] Zhiwen Ye, Andrew M Shelton, Jordan R Shaker, Julien Boussard, Jennifer Colonell, Sahar Manavi, Susu Chen, Charlie Windolf, Cole Hurwitz, Tomoyuki Namima, et al. Ultra-high density electrodes improve detection, yield, and cell type specificity of brain recordings. *bioRxiv*, 2023.
- [60] Yizi Zhang, Hanrui Lyu, Cole Hurwitz, Shuqi Wang, Charles Findling, Felix Hubert, Alexandre Pouget, International Brain Laboratory, Erdem Varol, and Liam Paninski. Exploiting correlations across trials and behavioral sessions to improve neural decoding. *bioRxiv*, 2024.
- [61] Yizi Zhang, Yanchen Wang, Donato Jiménez-Beneto, Zixuan Wang, Mehdi Azabou, Blake Richards, Renee Tung, Olivier Winter, Eva Dyer, Liam Paninski, et al. Towards a "universal translator" for neural dynamics at single-cell, single-spike resolution. *Advances in Neural Information Processing Systems*, 37: 80495–80521, 2024.
- [62] 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 preprint arXiv:2504.08201*, 2025.
- [63] Lili Zheng, Zachary T Rewolinski, and Genevera I Allen. A low-rank tensor completion approach for imputing functional neuronal data from multiple recordings. In *2022 IEEE Data Science and Learning Workshop (DSLW)*, pages 1–6. IEEE, 2022.

A Technical Appendices and Supplementary Material

A.1 Dataset details

This section provides the full anatomical names and corresponding acronyms of the brain areas analyzed throughout the paper.

IBL dataset. The posterior thalamus (PO), lateral posterior nucleus (LP), dentate gyrus (DG), hippocampal CA1 (CA1), anterior/anteromedial visual area (VISa), ventral posteromedial nucleus (VPM), anterior pretectal nucleus (APN), and midbrain reticular nucleus (MRN).

MAP dataset. The anterior lateral motor cortex (ALM), orbital area, lateral part (lOrb), orbital area, ventrolateral part (vlOrb), pallidum combined with globus pallidus, external segment (Pallidum), striatum combined with caudoputamen (Striatum), ventral anterior-lateral complex of the thalamus combined with ventral medial nucleus of the thalamus (VAL-VM), midbrain reticular nucleus (MRN), and superior colliculus, motor related (SC). ALM is identified following methods described in [47].

A.2 Generative process

For the N_r neurons in area r , we construct neural data tokens $Z_{X_r} \in \mathbb{R}^{N_r \times T^{*'}} by concatenating neural activity $X_r \in \mathbb{R}^{N_r \times T}$ with area, hemisphere, and unit embeddings: $Z_{\text{area}} \in \mathbb{R}^{D_a}$, $Z_{\text{hemi}} \in \mathbb{R}^{D_h}$, $Z_{\text{unit}} \in \mathbb{R}^{D_u}$, where $T^{*'} = T + D_a + D_h + D_u$. We then compute keys (K_r) and values (V_r) for the cross-attention module using linear projections $W_K, W_V \in \mathbb{R}^{T^{*'} \times T^{*'}}$. A shared set of learnable latent tokens $Z_0 \in \mathbb{R}^{P \times T^*}$ acts as the query Q . The resulting cross-attention output is projected via a multi-layer perceptron to obtain embedding factors $Z_{\text{embed},r} \in \mathbb{R}^{P \times T}$.$

For each trial, a masking proportion p is chosen from a uniform distribution from 0 to 0.6. If $p \leq 0.05$, then no areas are masked, otherwise, if the trial has recordings from R areas (where held-out areas are considered unrecorded), then $\text{Ceiling}[p * R]$ areas are randomly chosen to be masked. A variable M_r is set to 1 for masked areas and 0 for unmasked areas. The embedding factors $Z_{\text{embed},r}$ are tokenized by treating each timestep as a token, resulting in T tokens $Z_r^{\text{token}} \in \mathbb{R}^{T \times H}$, where an MLP projects each token from $\mathbb{R}^P$ to $\mathbb{R}^H$. For masked or unrecorded areas, these neural data tokens are replaced by a learned mask token Z_{mask} . We define a binary indicator $U_r \in \{0, 1\}$ to denote whether area r is unrecorded. Each token is then concatenated with a new learnable area embedding $Z'_{\text{area}} \in \mathbb{R}^{D_{a'}}$, producing the input tokens $Z_r^{\text{input}} \in \mathbb{R}^{T \times H'}$, $H' = H + D_{a'}$. These token sequences are passed through a transformer (with RoPE applied to queries and keys), and the outputs (also $\in \mathbb{R}^{T \times H'}$) are projected via $W_r^{\text{latent}} \in \mathbb{R}^{H' \times D_r}$ to obtain latent factors $Z_{\text{latent},r} \in \mathbb{R}^{T \times D_r}$. Finally, firing rate predictions $\hat{X}_r$ are computed via a linear readout $W_r^{\text{rate}} \in \mathbb{R}^{D_r \times N_r}$ followed by exponentiation.

The training procedure is as follows:

Cross-attention stitcher	Remaining modules of NeuroPaint
$Z_{X_r} = [X_r, Z_{\text{area}}, Z_{\text{hemi}}, Z_{\text{unit}}]$	$Z_r^{\text{token}} = \text{Tokenizer}(Z_{\text{embed},r})$
$K_r, V_r = W_K(Z_{X_r}^T), W_V(Z_{X_r}^T)$	$\delta_r = \begin{cases} 1 & \text{if } M_r = 0 \wedge U_r = 0 \\ 0 & \text{otherwise} \end{cases}$
$Q = Z_0$	$Z_r^{\text{input}}(t) = [(1 - \delta_r)Z_{\text{mask}} + \delta_r Z_r^{\text{token}}(t), Z'_{\text{area}}]$
$Z_{\text{cross-attn},r} = \text{Cross-Attention}(Q, K_r, V_r)$	$Z_{\text{latent},r} = (\text{Transformer}(\text{RoPE}(Z_r^{\text{input}})))W_r^{\text{latent}}$
$Z_{\text{embed},r} = \text{MLP}(Z_{\text{cross-attn},r})$	$\hat{X}_r^T = \text{Poisson}(\exp(Z_{\text{latent},r} W_r^{\text{rate}}))$

(1)

The model assumes a Poisson emission process with time-varying rates. For notational simplicity, we describe the generative model using data from one trial in one session. Symbols with the subscript r denote area-specific parameters, while those without r are shared across areas. Most parameters are shared across sessions, except for Z_{unit} and W_r^{rate} .

A.3 Loss function details

The consistency and regularization losses are formally defined below.

To ensure the embedding factors maintain a stable correlation structure across sessions and penalize deviations from a session-averaged target, we introduce the following consistency loss.

Definition A.1 (Consistency loss). Let $Z_{\text{embed},r}^{(b)}(t) \in \mathbb{R}^P$ denote the embedding factors for area $r \in [R]$ at time t under trial type $b \in [B]$. Define $K_{\text{model}}^{(b)}(r, r') \in \mathbb{R}^{P \times P}$ as the Pearson correlation matrix between $Z_{\text{embed},r}^{(b)}(t)$ and $Z_{\text{embed},r'}^{(b)}(t)$ computed across time after aggregating time steps over trials of type b within a batch. The corresponding target correlation matrix $K_{\text{target}}^{(b)}(r, r') \in \mathbb{R}^{P \times P}$ is obtained by averaging the correlation between $Z_{\text{embed},r}^{(b)}(t)$ and $Z_{\text{embed},r'}^{(b)}(t)$ across time steps in trials of type b , aggregated over sessions. The consistency loss is then defined as follows.

$$\mathcal{L}_{\text{consist.}} = \sum_b \sum_{r'=1}^R \sum_{r=1}^R 1 - \cos \left(\text{vec} \left(K_{\text{target}}^{(b)}(r, r') \right), \text{vec} \left(K_{\text{model}}^{(b)}(r, r') \right) \right), \quad (2)$$

where $\cos(\cdot, \cdot)$ denotes the cosine similarity, and $\text{vec}(\cdot)$ is a vectorization operator that flattens the correlation matrix, using only the upper triangular elements when $r = r'$.

The consistency loss encourages each area's embedding factors to encode information predictive of all areas observed across sessions, in addition to those recorded in the current session, thereby improving generalization to unrecorded areas. In practice, we approximate the session-averaged target $K_{\text{target}}^{(b)}(r, r')$ using a buffer of past batch correlations, computed with an exponential moving average (EMA) of the cross-attention stitcher to improve stability (Appendix A.4) [48].

Under the assumption that neural dynamics evolve smoothly over time, we define the following regularization loss.

Definition A.2 (Regularization loss). Let $Z_{\text{latent},r}(t) \in \mathbb{R}^{D_r}$ denote the latent factors at time t for area $r \in [R]$ and let $Z_{\text{latent},r}(t, i)$ denote its i -th component. The regularization loss is defined as follows.

$$\mathcal{L}_{\text{reg.}} = \sum_{t=1}^{T-1} \sum_{r=1}^R \sum_{i=1}^{D_r} |Z_{\text{latent},r}(t+1, i) - Z_{\text{latent},r}(t, i)|. \quad (3)$$

The regularization loss penalizes abrupt changes across consecutive time steps in the latent space, thereby improving temporal smoothness of the learned latent factors.

The relative weighting of the loss terms was selected to ensure that the reconstruction loss remains the primary driver during training, while the consistency and regularization terms serve as auxiliary constraints. Specifically:

- The reconstruction loss is scaled by a weight of 1 and normalized by the number of timesteps and neurons.
- The consistency loss is scaled by a weight of 1 and normalized by the number of area pairs.
- The regularization loss is scaled by a weight of 0.1 and normalized by the number of timesteps and latent factors.

A.4 Exponential moving average (EMA) of the cross-attention stitcher

To stabilize the correlation target calculated from the area-specific embedding factors, we maintain an EMA version of the cross-attention stitcher during training. At each iteration, the parameters in the EMA version are updated as a weighted sum of their previous values and the current weight of the cross-attention stitcher in the main model, controlled by a decay rate α . Specifically, the EMA parameter θ is updated as follows:

$$\theta^{EMA} \leftarrow \alpha \theta^{EMA} + (1 - \alpha) \theta \quad (4)$$

To reduce bias in early training when parameter estimates are still unstable, we adapt the decay rate α based on the number of iteration steps n_{step} :

$$\alpha = \min \left(1 - \frac{1}{n_{\text{step}} + 1}, \alpha_{\text{max}} \right) \quad (5)$$

which increase α as training progresses. Here $\alpha_{\max} = 0.999$. This ensures faster adaptation in the early iterations and more stable averaging later on. We use the EMA cross-attention stitcher to compute the area-specific embedding factors and then compute the correlation between embedding factors for each batch, and store them in a running buffer. These stored correlation matrices are then averaged to generate the target correlation matrix for each area pair and trial type for the current iteration.

A.5 Determining the area-specific latent factor dimension in NeuroPaint and LFADS

NeuroPaint. To determine the number of latent factors for each brain area, we estimated the dimensionality of smoothed neural activity across sessions. For each area and session, we first smoothed the spike trains with a Gaussian window (standard deviation = 50 ms), concatenated all trials, and subtracted the mean to center the data. We then computed the participation ratio of the resulting activity matrix, which provides a measure of its linear dimensionality [13]. This yielded one dimensionality estimate per area per session. To ensure sufficient capacity, we set the number of latent factors for each brain area to the maximum participation ratio observed across sessions for that area, plus a margin of 10 dimensions. We observed that for most brain areas, the estimated dimensionality saturated as the number of recorded neurons increased, suggesting that the low-dimensional assumption we mentioned at the end of the introduction is approximately satisfied by the neural data. This procedure ensures that NeuroPaint is equipped with enough latent factors to model the full range of dynamics expressed in each area.

Area	Dim
PO	61
LP	84
DG	72
CA1	39
VISa	35
VPM	43
APN	58
MRN	24
ALM	29
lOrb	31
vlOrb	37
Pallidum	23
Striatum	31
VAL-VM	49
MRN	22
SC	21

Table 1: NeuroPaint latent factor dimensions picked for areas from IBL and MAP datasets.

LFADS. The multi-session LFADS [33] uses per-session linear read-in and read-out layers to align neural activity from different sessions into a shared latent space. These session-specific “stitching” layers map observed spike counts to input factors (read-in) and latent factors to firing rates (read-out). The shape of these matrices can vary to match the number of neurons recorded in each session. A shared encoder, generator, and factor matrix are shared across sessions and learned jointly from all sessions. The per-session read-in and read-out matrices are learned using data from only the corresponding session. To promote alignment across sessions, we follow the standard procedure to initialize the weights of the read-in and read-out matrices using principal component regression (PCR) [38]. PCR maps the trial-averaged firing rates from each individual session to the shared principal components across all sessions. This provides the model with an input that is already in a shared subspace and is critical for ensuring a shared set of dynamics is learned. We first reshape the trial-averaged firing rates from all sessions into a matrix of size $(n_{\text{timepoints}} \times n_{\text{conditions}}) \times (n_{\text{sessions}} \times n_{\text{neurons}})$, and PCA is applied to identify a set of global principal components (PCs). We find the number of PCs needed to explain 90% of the variance, which defines the dimensionality of the latent factors used in LFADS. Each session’s data is then projected onto this global PC space via Ridge regression, and the

resulting weights and biases from the trained regression model are saved to be used to initialize the read-in weights and biases in LFADS.

A.6 Details of the simulated recurrent neural network in the synthetic dataset

To generate synthetic spike data with multi-area dynamics, we simulated a continuous-time recurrent neural network (RNN). This RNN is composed of 5 areas, each area contains 200 units. Each unit has a tanh nonlinearity and with a time constant $\tau = 25$ ms. We simulate the network using a timestep of $\Delta t = 10$ ms. Unit activity evolves according to the following update rule:

$$h_{t+1} = (1 - \beta)h_t + \beta \tanh(W h_t) \quad (6)$$

Here $\beta = \frac{\Delta t}{\tau}$. W is the recurrent connectivity between units. h_t denotes unit activities at time step t .

The network receives no external input, its activity is driven entirely by autonomous recurrent dynamics. Each recurrent connection has a weight sampled from a normal distribution $W_{ij} \sim \mathcal{N}(0, g^2/N)$, here W_{ij} is the weight between unit i and unit j , $N = 1000$ is the total number of units in the network, we pick $g = 3$ so that we will have chaotic circuit dynamics [40]. Connectivity between areas is sparse: each pair of units are connected with 1% probability. Units within the same area have all-to-all connectivity.

For each trial, we randomly initialize the unit activity in the network. Due to chaotic dynamics, each trial will have qualitatively different dynamics. We simulate trials from the same RNN across synthetic sessions.

For each synthetic session, we simulate neuronal spike trains from the unit activity using GLMs. Each session has its own set of synthetic neurons, with the number of neurons in each area sampled uniformly from the range [20, 60]. The number of trials in each synthetic sessions are sampled uniformly from the range [200, 300].

Each area-specific and session-specific GLM first projects the unit activity onto neuronal instantaneous log firing rates using a sparse linear layer (2% sparsity). The log firing rates for each synthetic neuron are then scaled to a fixed range ([0,2]). Next, we pass the log rates to an exponential nonlinearity and a Poisson emission process. For each session, we held-out spike activities generated from 1-2 RNN areas as unrecorded.

In summary, partially overlapped RNN units contribute to the simulated spike trains across synthetic sessions.

A.7 Deviance fraction explained (DFE)

Deviance fraction explained is a metric that ranges from 0 to 1, where a larger value indicates a better model fit. It is defined as:

$$1 - \frac{D_{\text{model}}}{D_{\text{null}}}$$

where the deviance terms are given by:

$$\begin{aligned} D_{\text{model}} &= \log p(\mathbf{x}_{1:T} | M_{\text{sat}}) - \log p(\mathbf{x}_{1:T} | M) \\ D_{\text{null}} &= \log p(\mathbf{x}_{1:T} | M_{\text{sat}}) - \log p(\mathbf{x}_{1:T} | M_{\text{null}}) \end{aligned}$$

Here, M_{sat} represents the **saturated model**, while M_{null} represents the **null model**. Both the saturated and null models assume that the instantaneous spike data is generated from an i.i.d. Poisson distribution. The **saturated model** (M_{sat}) assumes that the mean of the Poisson distribution is given by the instantaneous spike count, while the **null model** (M_{null}) assumes that the mean is given by the time-averaged and trial-averaged spike count.

A.8 Evaluation on Synthetic Data with Lower Firing Rates (Matching Real Data Statistics)

To better align our synthetic data with the firing rate statistics observed in the real neural recordings, we generated an additional synthetic dataset, where the log firing rates of each neuron were scaled to a fixed range of $[-3, 3]$, in contrast to the range $[0, 2]$ used in the previous synthetic dataset shown in Fig. 2. This adjustment results in lower and more realistic firing rates.

We evaluated the performance of several models for predicting neural activity in the unrecorded brain areas on this synthetic dataset: two baselines (GLM and LFADS) and multiple variants of NeuroPaint (including the linear version described in A.12 and other variants trained with different combinations of loss terms). Overall, we found that NeuroPaint trained with all three loss terms (reconstruction, consistency and regularization) achieved the best performance for this low-rate synthetic dataset (see Fig. S2), unlike for the higher-rate synthetic dataset discussed in the main text (Fig. 2). Empirically, we found that including the consistency loss was particularly important for improving NeuroPaint’s performance, enabling it to outperform the baselines.

It is important to note that for the synthetic dataset, the GLM baseline also performs very well, suggesting that a linear model is sufficient to predict neural activity in one area based on data from other areas, i.e. to capture inter-areal communication in this simplified setting. In future work, we plan to construct more complex and biologically realistic synthetic datasets that include strong nonlinear inter-areal dependencies, to better evaluate the advantage of nonlinear models.

A.9 GLM baseline performance on the IBL and MAP datasets

We evaluated GLM baselines on both the IBL and MAP datasets. For the IBL dataset, the mean deviance fraction explained (DFE) across neurons in the held-out areas was extremely negative (-3.8×10^{32}), indicating severe overfitting or model mismatch. Notably, 90.7% of neurons had negative DFE values. For the MAP dataset, the mean DFE was -5.1×10^{23} . 63.3% of neurons exhibited negative DFE values.

A.10 Model and hyperparameter details

NeuroPaint. To avoid overfitting, we add dropout (40%) to all the attention layers and a dropout (20%) at the very beginning. Other hyperparameters of the NeuroPaint model is listed in Table 2. Notations for the hyperparameters are introduced in A.2. For synthetic dataset, we use 24 latent factors for each area. For the IBL and MAP dataset, we set the dimensionality of latent factors according to A.5.

Hyperparameter	Value
D_a	20
D_h	3
D_u	50
T	400 (MAP), 200 (IBL)
T^*	512
P	48
H	236
D'_a	20
H^a	256
number of transformer layers	5

Table 2: Hyperparameters used for NeuroPaint.

LFADS. We use a version of LFADS (*lfads-torch*) re-implemented in PyTorch by [38]. The hyperparameters of our multi-session LFADS are provided in Table 3. The shared encoder, generator, and factor matrix comprise about 0.3 million parameters. When including session-specific read-in and read-out matrices, the total number of model parameters increases to 0.4 million, 0.5 million, and 0.6 million for the synthetic, IBL and MAP datasets. The latent factor dimension of multi-session LFADS is chosen via PCR (Appendix A.5). We set the latent factor dimension to 28, 11, and 27 for the synthetic, IBL and MAP datasets. The LFADS model computes KL divergence between posteriors and priors for both initial condition and inferred input distributions, which are added to the reconstruction cost in the variational ELBO. The priors are multivariate normal for the initial conditions and autoregressive multivariate normal for the inferred inputs.

NeuroPaint-Linear. The linear, session-specific, and area-specific stitchers take as input the flattened population activity from 3 neighboring bins centered around the target time point. Prior to this, the binned spike counts are smoothed using a 1D Gaussian kernel with a standard deviation of 5

Hyperparameter	Value
Initial Condition Encoder Dimension	100
Controller Input Encoder Dimension	100
Controller Input Lag	1
Controller Dimension	100
Controller Output Dimension	6
Initial Condition Dimension	100
Generator Dimension	100
Controller Output Prior Temporal Decay Constant	10
Controller Output Prior Innovation Variance	0.1
Initial Condition Prior Mean	0
Initial Condition Prior Variance	0.1
Dropout Rate	0.02
Coordinated Dropout Rate	0.3
Weight Decay	0.0
Learning Rate	0.0003
Batch Size	1024

Table 3: Hyperparameters used for multi-session LFADS.

time bins. The stitchers output area-specific embedding factors, which, along with the latent factors, are set to have the same dimensionality as in the original NeuroPaint model. The session-shared, reduced-rank linear transformation applied to the concatenated embeddings has a fixed rank of 20. The population activity in the masked areas, which the model is trained to predict, is also smoothed using a Gaussian kernel with the same standard deviation of 5 time bins.

A.11 Training details

NeuroPaint. We trained our model on 2-12 Nvidia A40 or A100 GPU using AdamW optimizer for 1000 epochs with a learning rate of $(1e^{-3}/256 \times \text{global batch size})$ using a OneCycleLR scheduler. We put a weight decay 0.01 to avoid overfitting. We utilized a batch size of 16 on each GPU during the training. Global batch size is $16 \times$ the number of GPU nodes. We split our dataset based on the session to training, validation, and test set with a proportion of 60%, 20%, and 20%. We saved the model checkpoint based on the validation loss. The 10-session model for the synthetic dataset is trained with 2 GPU nodes in around 6 hours. The 20-session model for the IBL dataset is trained with 4 GPU nodes in around 35 hours. The 40-session model for the MAP dataset is trained with 12 GPU nodes in around 20 hours.

LFADS. We train the multi-session LFADS using the AdamW optimizer without a learning rate scheduler, which is the default setting in the *lfads-torch* package. All multi-session LFADS models are trained on a single Nvidia A40 or A100 GPU for 1000 epochs. The best model checkpoint is selected based on the reconstruction performance on the validation set. The 10-session model for the synthetic dataset is trained in under 4 hours. The 20-session LFADS for the IBL dataset takes approximately 10 hours, while the 40-session LFADS for the MAP dataset is trained in about 16 hours.

NeuroPaint-Linear. Training details are provided in Appendix A.12. All models are trained on a single NVIDIA RTX 2080 Ti GPU and complete within one hour.

A.12 NeuroPaint-Linear

We introduce NeuroPaint-Linear, a *linear* variant of the original NeuroPaint model. Designed for simplicity and interpretability.

Architecture NeuroPaint-Linear retains the core architectural components of NeuroPaint, comprising three main components:

1. **Linear, session-specific, and area-specific** stitchers, which map population activity to a set of embedding factors.
2. A **low-rank, session-shared** weight matrix, which transforms the concatenated embedding factors (across all areas) into area-specific latent factors, capturing a mapping between the latent dynamics of different brain areas that remains consistent across sessions and animals.
3. **Linear, session-specific, and area-specific** readout layers, which map latent factors to predicted population activity (for all the recorded and unrecorded areas).

While the overall structure closely mirrors that of NeuroPaint, NeuroPaint-Linear introduces three simplifications:

1. The cross-attention readin-stitcher is replaced by a simple linear mapping.
2. The transformer encoder is replaced with a reduced-rank linear transformation, restricting inter-area mappings to linear operations.
3. In contrast to the nonlinear NeuroPaint, where embeddings for masked or unrecorded areas are replaced with a learned mask token after tokenization, we set these embedding factors to zero directly, bypassing the tokenizer.

In addition, NeuroPaint-Linear operates in a local and time-independent fashion: latent factors and the corresponding predicted activity are computed independently at each time step, using only the recorded activity from a short temporal window (3 time bins, *i.e.* ± 1 bin) around that time—without leveraging long temporal context.

Training details Training follows a similar inter-area masking strategy used in NeuroPaint. For each epoch, one recorded area per session is randomly selected and masked from the input. The model is then trained to predict the activity of the masked area.

Optimization is performed using the L-BFGS algorithm in full-batch mode, minimizing the mean squared error computed across all time steps and all neurons in the masked areas over sessions. For each training epoch, L-BFGS is reinitialized from scratch and run until convergence. We train models for each of the MAP and IBL datasets for 50 epochs each (we only show results for the MAP dataset), using the default L-BFGS hyperparameters provided by PyTorch. We chose L-BFGS and mean squared error due to their stability during training and their effectiveness.

Comparison with standard reduced-rank regression methods NeuroPaint-Linear differs fundamentally from standard reduced-rank regression approaches used in encoding models [4, 36] and inter-areal correlational analyses (e.g., communication subspace) [39], both in its objectives and methodology. The key innovations lie in the introduction of a session-shared mapping from the full set of area-specific embedding factors to area-specific latent factors, and the use of an inter-area masking strategy during training. Together, these components enable NeuroPaint-Linear to predict the dynamics of unrecorded brain areas based solely on the activity of recorded areas within the same session.

A.13 Impact of loss terms on NeuroPaint performance (MAP dataset)

To evaluate the contribution of each designed loss term on real data, we compared the performance of different NeuroPaint variants on the MAP dataset. In line with the synthetic data results in Fig. 2 and Fig. S2, the inclusion of the consistency loss was essential for enabling NeuroPaint to outperform the LFADS baseline. In addition, NeuroPaint trained with all three loss terms performs the best. Surprisingly, NeuroPaint-Linear, a purely linear variant of the model, also performed competitively, comparably with the LFADS baseline, highlighting the strength of the overall NeuroPaint framework and its potential to reveal interpretable inter-areal communication among all areas of interest, including both recorded and unrecorded ones.

Interestingly, we found that the variant of NeuroPaint trained with reconstruction and regularization loss (but without consistency loss) exhibited unstable performance across datasets. For example, while it performed reasonably well on the original synthetic dataset (Fig. 2), its performance degraded substantially on the lower firing rate synthetic data (Fig. S2) and on the MAP dataset (Fig. S3). This suggests that, in the absence of a consistency constraint, regularization can sometimes hinder model

performance—possibly by over-constraining the latent space or penalizing useful variability. These observations underscore the stabilizing role of the consistency loss in guiding the latent representation, especially under more realistic data distributions.

A.14 Interpretable area-specific latent dynamics in the IBL dataset revealed by NeuroPaint

Similar to results shown in Fig. 4, here we examine the inferred latent factors for the IBL dataset. Unlike the latent factors for SC in the MAP dataset, the latent factors for MRN in the IBL dataset did not show strong trial-type modulations (Fig. S4A). However, in line with results for the MAP dataset, the inferred latent factors in the IBL dataset remain consistent across trials over sessions, regardless of whether a given area was recorded or unrecorded (Fig. S4A, B). We evaluated area-to-area representation similarity based on the inferred latent factors, across three trial periods, defined as three consecutive 0.5-second intervals after stimulus onset (Fig. S4C,D). Consistent with anatomical expectations, DG and CA1, both belonging to the hippocampus, exhibited high similarity with each other but not with other areas. In addition, two thalamic nuclei, PO and VPM, also exhibit high similarity across the three periods. Notably, the overall inter-areal similarity structure changed over time: although coherent sub-networks of areas remained present throughout the trial, they became more fragmented in the later periods, exhibiting both reduced similarity and a narrower set of similar areas. This dynamic change likely reflects a shift in task phase—from visual processing to decision-making and motor execution.

A.15 LFADS with larger architectures does not outperform NeuroPaint

To address the concerns regarding potential model capacity limitations in LFADS, we trained LFADS models on the MAP dataset with larger architectures (256 units in encoder, controller and generator). We tested the number of latent factors of 243 (matching the number of pooled NeuroPaint latent factors across areas) and 49 (matching the largest single-area NeuroPaint latent factors). To maintain training stability and prevent loss divergence, we reduced the learning rate from $3e^{-4}$ to $2e^{-4}$ here. Training converged successfully under this setting. The DFE for neurons in held-out areas, pooled over 40 sessions, is reported in Table 4. We found that NeuroPaint continued to outperform LFADS with larger architectures in terms of reconstruction accuracy. Interestingly, the higher-capacity LFADS models performed worse than the smaller model used in our main experiments, potentially due to overfitting given the increased number of parameters. These results suggest that the performance gap is not solely attributable to architectural capacity, and that the paradigm introduced in NeuroPaint is the key to its improved performance.

	LFADS variants			NeuroPaint (used in Fig. 3G,H)
	27 factors, 100 units (used in Fig. 3G,H)	49 factors, 256 units	243 factors, 256 units	
DFE	0.08 ± 0.006	-2.93×10^{20}	-9.12×10^{20}	0.10 ± 0.003
DFE (Worst 5% excluded)	0.10 ± 0.002	0.08 ± 0.002	-9.31 ± 1.31	0.12 ± 0.002

Table 4: Performance comparison across LFADS models of different capacities and NeuroPaint, evaluated by DFE (mean $\pm$ s.t.e. over neurons in the held-out areas).

A.16 Calculating correlations between trial pairs of latent factors (MAP dataset)

To assess the consistency and trial-type-dependent modulation of inferred latent factors, we computed Pearson correlations between trial pairs of area-specific latent factors. All analyses were performed separately for each brain area.

Preprocessing. For each trial, the latent factor matrix was first mean-subtracted across time for each factor to remove the static offsets. The resulting matrix was then flattened into a vector by concatenating all factors.

Consistency results in the left block of Table 5. For each area, we computed Pearson correlations between flattened latent-factor vectors for every pair of trials, and grouped trial pairs into three categories: 1) Recorded - Recorded: both trials from sessions where the area is recorded 2) Unrecorded -

Table 5: Mean Pearson correlation between latent factors computed from trial pairs pooled across all sessions and averaged over brain areas (mean $\pm$ 1 standard deviation). See Appendix A.16 for calculation details.

From all sessions		Trial-average subtracted, within a session	
Trial pairs	Pearson's correlation	Trial pairs	Pearson's correlation
Recorded - Recorded	0.25 $\pm$ 0.05	Hit Left	0.59 $\pm$ 0.28
Unrecorded - Unrecorded	0.57 $\pm$ 0.10	Hit Right	0.59 $\pm$ 0.28
Recorded - Unrecorded	0.35 $\pm$ 0.06	Across Trial Type	0.53 $\pm$ 0.31

Unrecorded: both trials from sessions where the area is unrecorded 3) Recorded & Unrecorded: one trial from a session where the area is recorded, one from an unrecorded session.

For each area, we computed the averaged correlation within each group. The reported mean was obtained by averaging these per-area values across all areas, and the standard deviation reflects variability across areas.

Trial-type-dependent modulation results in the right block of Table 5. To control for the session-to-session variability, we perform the following analysis on trial pairs within each session, treating each session separately, and again also separately for each area.

For each session, we first computed the trial-averaged flattened latent factor by averaging over all *hit* trials. We then subtracted this average from each trial's flattened latent factor.

We calculated Pearson correlations between all pairs of trials within a session and grouped them into three categories, based on the trial types of the trial pair 1) Hit Left 2) Hit Right 3) Across Trial Type: one hit left and one hit right trial.

For each area and session, we averaged correlations within each group. The reported means and standard deviations were computed by averaging over sessions and areas.

A.17 Decoding choice and stimulus from NeuroPaint latent factors for held-out areas

To evaluate whether the latent factors capture the stimulus and behavioral variability, we performed decoding analysis on the MAP dataset. Specifically, we used logistic regression with cross-validation to decode either stimulus (high/low auditory tone) or choice (lick left/lick right/no lick) from the latent factors for each held-out area, and we compared the balanced accuracy to that decoding from held-out spike data (see Table 6). Stimulus was decoded using summed activity over the last 100 ms of the stimulus period; choice was decoded using summed activity over the last 100 ms of the delay period. Note that due to randomness, no data from VM-VAL was held out in the MAP dataset. As a result, we are unable to report results for this area.

We found that decoding accuracy for inferred latent factors is typically slightly higher than that from held-out spike data, suggesting that the inferred latent factors successfully capture the stimulus/behavioral variability in the brain area. This decoding improvement also suggests that the inferred latent factors provide a denoised and more complete representation than the held-out population spike data subsampled from the area of interest.

		ALM	lOrb	vlOrb	Pallidum	Striatum	MRN	SC
Stimulus (chance level= 0.5)	spike data	0.60 $\pm$ 0.12	0.58 $\pm$ 0.08	0.53 $\pm$ 0.08	0.65 $\pm$ 0.07	0.53 $\pm$ 0.04	0.62 $\pm$ 0.08	0.66 $\pm$ 0.08
	latent factors	0.71 $\pm$ 0.08	0.50 $\pm$ 0.13	0.61 $\pm$ 0.13	0.80 $\pm$ 0.13	0.73 $\pm$ 0.06	0.68 $\pm$ 0.12	0.76 $\pm$ 0.09
Choice (chance level= 0.33)	spike data	0.40 $\pm$ 0.15	0.41 $\pm$ 0.13	0.37 $\pm$ 0.12	0.46 $\pm$ 0.17	0.58 $\pm$ 0.13	0.45 $\pm$ 0.09	0.52 $\pm$ 0.21
	latent factors	0.50 $\pm$ 0.14	0.51 $\pm$ 0.13	0.57 $\pm$ 0.14	0.68 $\pm$ 0.12	0.63 $\pm$ 0.11	0.63 $\pm$ 0.16	0.70 $\pm$ 0.16

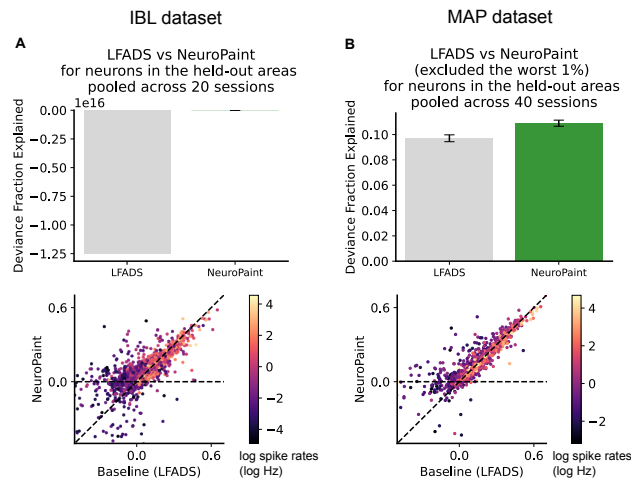
Table 6: Decoding accuracy from spike data or latent factors for the held-out areas with logistic regression, evaluated by balanced accuracy (mean $\pm$ s.t.d. over sessions).

A.18 Intrinsic dimensionality of predicted firing rates for each brain area from NeuroPaint latent factors

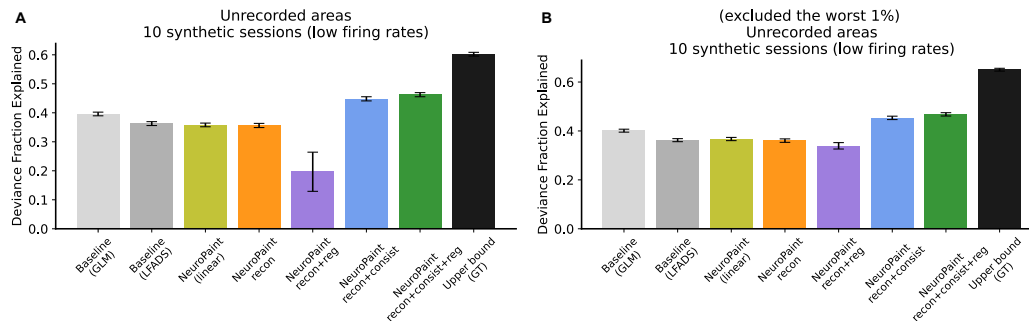
In Table 7, we report the intrinsic dimension of the reconstructed firing rates based on the NeuroPaint latent factors. Specifically, for each recorded area in the MAP dataset, we calculated the participation ratio of the predicted firing rates, which quantifies the effective number of linear dimensions contributing to the variance. Notably, the number of latent factors we used is much larger than the intrinsic dimensionality of predicted firing rates, suggesting that similar performance may be achievable with fewer latent factors. We will explore this in future work.

	ALM	lOrb	vlOrb	Pallidum	Striatum	VAL-VM	MRN	SC
Number of latent factors	29	31	37	23	31	49	22	21
Intrinsic dimensionality (mean $\pm$ s.t.d.)	4.71 ± 1.45	4.82 ± 1.40	4.21 ± 1.43	3.51 ± 1.36	4.19 ± 1.53	5.63 ± 1.40	3.90 ± 0.98	4.50 ± 1.05

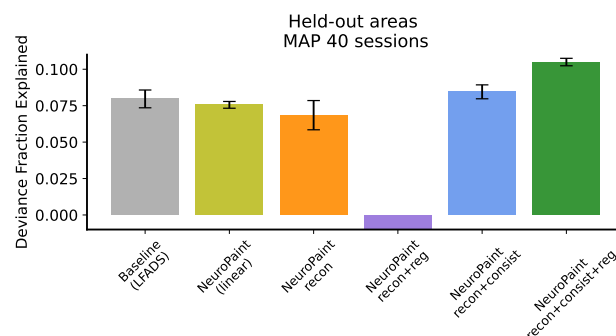
Table 7: Intrinsic dimensionality of predicted firing rates reported against the number of latent factors from NeuroPaint for the MAP dataset.



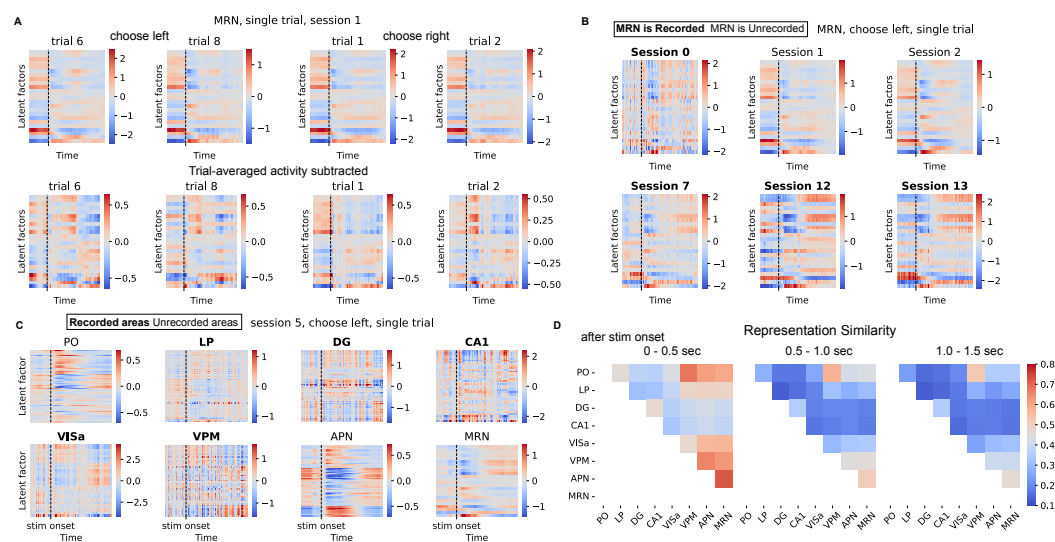
Supplemental Figure 1: Prediction performance for neurons from held-out brain areas across the two datasets. **(A)** Top: mean and standard error of DFE on the IBL dataset across all neurons, here LFADS yields a mean DFE on the order of $-1e16$, primarily due to a subset of neurons with extremely poor predictions; compare Fig. 3C where the 1% of neurons with the poorest predictions were excluded. Bottom: per-neuron DFE for all the held-out areas in 20 IBL sessions. **(B)** Top: mean and standard error of DFE on the MAP dataset, excluding the 1% of neurons with the poorest predictions; compare Fig. 3G, which included all neurons. Bottom: per-neuron DFE for all the held-out areas in 40 MAP sessions.



Supplemental Figure 2: Prediction performance from baselines, variants of NeuroPaint and ground truth firing rates for neurons from unrecorded areas in the synthetic dataset with lower firing rates. **(A)** Mean and standard error of DFE on the synthetic dataset with lower firing rates across all neurons. **(B)** Mean and standard error of DFE on the synthetic dataset with lower firing rates, excluding the 1% of neurons with the poorest predictions.



Supplemental Figure 3: Prediction performance from LFADS and variants of NeuroPaint for neurons from held-out brain areas in the MAP dataset. Bar plot shows mean and standard error of DFE on the MAP dataset across all neurons. For visualization purpose, we truncate the y axis below -0.01 . For NeuroPaint trained with reconstruction and regularization loss, its mean of DFE is -3.83 , standard error of DFE is 0.11 .



Supplemental Figure 4: *NeuroPaint's area-specific latent factors for the IBL dataset.* **(A)** Inferred latent factors for the Midbrain Reticular Nucleus (MRN), an unrecorded area in session 1 from the IBL dataset, during example trials across the *choose left* and *choose right* conditions. The bottom row shows the same latent factors with the trial-averaged activity subtracted. **(B)** Inferred single-trial latent factors for MRN (both recorded and unrecorded) across six sessions. **(C)** Latent factors for all eight areas (both recorded and unrecorded) in a *choose left* trial from a single session in the IBL dataset. **(D)** Representation similarity analysis (RSA) of latent factors across eight brain areas (recorded and unrecorded) for choose-left trials for three periods after stimulus onset in the IBL dataset, with values averaged across trials and pooled over sessions.

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: We clearly declare and introduce our main contributions as inferring unrecorded brain area dynamics.

Guidelines:

- The answer NA 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 NA 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: Please refer to the discussion section.

Guidelines:

- The answer NA 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: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA 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: Implementation details are included in the Appendix and the github repository to be shared.

Guidelines:

- The answer NA 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: We use public datasets for our experiments. Our code is attached for reproducibility of our results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify all the training and hyperparameters details in Appendix.

Guidelines:

- The answer NA 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: We report error bars whenever necessary in the plots.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. Experiments compute resources

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

Answer: [Yes]

Justification: We document the computing resources used in Appendix.

Guidelines:

- The answer NA 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 carefully read and follow the NeurIPS Code of Ethics. All of our code and main text are anonymous.

Guidelines:

- The answer NA 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: [NA]

Justification: Our work is not specifically linked to any social applications, except for its relevance to fundamental neuroscience research.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not include any high risk data and all results and models are free to share.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: We clearly indicate our cited existing assets by citing the work.

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA 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: [NA]

Justification: Our paper does not involve any human subjects.

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.