STRUCTURED LEARNING OF BEHAVIORALLY RELEVANT NEURAL DYNAMICS IN MULTIMODAL NEURAL TIME-SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal neural data can enable a more complete understanding of brain dynamics underlying behavior. Modalities such as neuronal spiking activity, local field potentials (LFPs), and behavioral signals capture diverse spatiotemporal aspects of brain processes. By leveraging these complementary strengths, multimodal neural fusion can provide a unified, rich representation of brain-behavior processes and address the limitations of single-modality analyses, such as incomplete or noisy data. While recent works have jointly modeled behavior and neural data to separate sources of variability, they largely rely on latent variable models that use a single modality of neural data. Here we develop BREM-NET, a multimodal dynamical model that integrates behavioral signals and multiple neural modalities—such as LFPs and spike counts—with distinct statistical characteristics and temporal resolutions. BREM-NET performs multimodal neural fusion during inference while also separating behaviorally relevant and residual neural dynamics in multimodal neural time-series via a structured learning framework. In two independent public multimodal neural datasets spanning multiple behaviors and neural modalities, our method learns a low-dimensional behaviorally relevant latent subspace that is sufficient for behavior decoding while separately modeling residual neural dynamics that improve neural prediction. Furthermore, our method achieves these capabilities even when different neural time-series modalities are asynchronous and have distinct temporal resolutions, which is a major challenge in real-world neural recordings. This framework provides a new tool for studying behaviorally relevant neural computations across different spatiotemporal scales of brain activity.

1 INTRODUCTION

Understanding neural dynamics across different spatiotemporal scales and their relationship to behavior is a fundamental challenge in neuroscience. These different scales can be captured via different modalities. Neuronal spike trains, local field potentials (LFPs), and behavioral signals each offer distinct yet complementary insights into brain function (Pesaran et al., 2002; Penttonen & Buzsáki, 2003; Belitski et al., 2010; Einevoll et al., 2013; Stavisky et al., 2015; Pesaran et al., 2018; Lu et al., 2021; Ramezani et al., 2025). Spikes capture the spatial scale of neurons, reflect rapid transient neural events at a millisecond timescale, and are often modeled with Poisson distributions due to their discrete nature. LFPs measure a larger spatial scale and slower oscillatory processes that evolve over longer timescales than spikes, and are typically modeled with Gaussian distributions (Buzsáki et al., 2012; Hsieh et al., 2019; Ahmadi et al., 2021). Behavioral signals represent high-level outputs of the nervous system, introducing an additional difficulty due to their complex relationships with neural activity (Gallego et al., 2020; Urai et al., 2022).

Given the above complementary aspects, integrating these modalities in a unified dynamical model can help uncover the multimodal neural mechanisms driving behavior. It can also improve neural-behavioral predictions, for example for brain-computer interfaces (BCIs), where spike signals often degrade over time while LFPs remain comparatively stable (Stavisky et al., 2015). However, developing such models remains challenging for several reasons. First, neural modalities contain both *behaviorally relevant* dynamics that are shared with behavior, and other neural dynamics that these behaviorally relevant dynamics leave unexplained, which we refer to as *residual neural dynamics*.

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

Separating these two types of neural dynamics enables new scientific analyses – for example, characterizing which aspects of multimodal neural activity are sufficient to describe behavior, apart from the remaining neural variability. Such analyses require functionally separated latent subspaces rather than a single unified latent representation. Second, different modalities such as discrete spikes and continuous LFPs have distinct statistical and temporal characteristics. Third, in real-world recordings, neural modalities are often asynchronous—that is, they are sampled at different temporal resolutions—or suffer from dropouts due to noise or electrode failures, complicating multimodal dynamical modeling. Finally, the dynamics in these modalities can be complex and nonlinear. Despite recent advances in multimodal frameworks, existing approaches do not simultaneously incorporate behavior and multiple asynchronous neural modalities, such as spikes and LFPs, into a unified dynamical model (Ramezani et al., 2025).

Indeed, learning behaviorally relevant neural dynamics first, before modeling other neural dynamics, is an established objective in neural-behavioral modeling (Sani et al., 2021; Hurwitz et al., 2021; Sani et al., 2024; Vahidi et al., 2025): because these dynamics can be low-variance and masked by other neural dynamics, models learned from neural activity alone can miss them or mix them with unrelated variability (Sani et al., 2021; 2024). Learning a compact behaviorally relevant latent state first, and modeling the residual neural dynamics separately, thus enables interpretable analyses of how neural activity relates to behavior. However, these and related joint neural-behavioral models (Schneider et al., 2023; Gondur et al., 2024; Zhang et al., 2025; Schulz et al., 2025) are constrained to a single neural modality and do not perform multimodal neural fusion. Thus, a major unaddressed challenge is to learn behaviorally relevant neural dynamics jointly from multiple modalities that reflect different spatiotemporal scales, while modeling residual neural dynamics separately.

Another line of work has developed multimodal models that fuse information across multiple neural modalities, but without separating their behaviorally relevant and residual neural dynamics (Coleman et al., 2011; Abbaspourazad et al., 2021; Singh Alvarado et al., 2021; Kramer et al., 2022; Rezaei et al., 2023; Ahmadipour et al., 2024). Without this separation, behaviorally relevant neural dynamics can be conflated with residual neural dynamics. Addressing this challenge requires a new multimodal modeling framework.

Contributions To address this gap, we propose **B**ehaviorally **Re**levant modeling of **M**ultimodal **Ne**ural **T**ime-series (**BREM-NET**), a nonlinear multimodal dynamical model that simultaneously achieves the following capabilities: 1) It jointly incorporates multiple neural modalities with behavioral signals into a unified model. 2) It learns behaviorally relevant neural dynamics from multimodal neural time-series first and then models the residual neural dynamics in a separate set of latents via a structured learning framework. 3) It supports modeling and fusion of modalities with heterogeneous statistical distributions, different temporal resolutions, or dropouts. We validate BREM-NET on two independent multimodal non-human primate (NHP) datasets from different brain regions and during different tasks (Flint et al., 2012; O'Doherty et al., 2020). We find that BREM-NET successfully fuses information across neural modalities—even when they are sampled asynchronously—while also separating behaviorally relevant and residual neural dynamics. Doing so, the method learns more accurate low-dimensional behaviorally relevant neural latents as reflected in behavior decoding accuracy, while its behaviorally relevant trajectories are strongly aligned with behavior. Further, via its residual neural latents, it achieves accurate neural prediction by capturing the neural variability that the behaviorally relevant latents leave unexplained. By modeling different spatiotemporal neural scales with behavior, our approach provides a new tool for investigating multiscale neural computations that underlie behavior in neuroscience and has potential implications for multimodal BCIs under realistic, asynchronous conditions.

2 Related Work

Prior related works fall into two broad categories: (i) models that separate behaviorally relevant and residual neural dynamics but just for a single neural modality without enabling multimodal neural fusion, and (ii) models that fuse information across multiple neural modalities but do not separate behaviorally relevant and residual neural dynamics. BREM-NET achieves both these capabilities: it not only fuses information across multiple neural modalities even in asynchronous scenarios, but also separates the behaviorally relevant and residual neural dynamics in multimodal neural time-series.

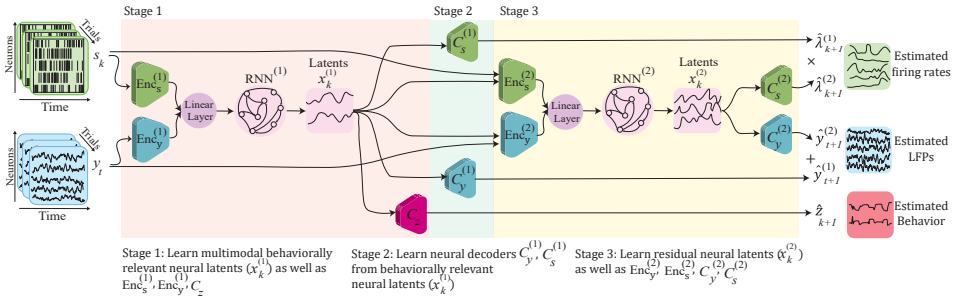

Figure 1: **BREM-NET model architecture.** At each time step, multimodal neural observations are used to infer latent states that predict behavior and neural activity. **Stage 1** trains an RNN with modality-specific encoders and a behavior decoder to learn behaviorally relevant latents $\mathbf{x}_k^{(1)}$. **Stage 2** fits modality-specific neural decoders given $\mathbf{x}_k^{(1)}$. **Stage 3** trains a second RNN, conditioned on $\mathbf{x}_k^{(1)}$, to learn residual neural latents $\mathbf{x}_k^{(2)}$ that capture the multimodal neural dynamics not predicted by $\mathbf{x}_k^{(1)}$.

The first category includes both linear models (Kobak et al., 2016; Sani et al., 2021) and nonlinear deep learning models (Zhou & Wei, 2020; Hurwitz et al., 2021; Whiteway et al., 2021; Schneider et al., 2023; Gondur et al., 2024; Sani et al., 2024; Wang et al., 2024; Wu et al., 2025; Vahidi et al., 2025; Yi et al., 2025) for joint neural-behavioral modeling. Among these works, some recent methods include TNDM (Hurwitz et al., 2021), DPAD (Sani et al., 2024), CEBRA (Schneider et al., 2023), and MMGPVAE (Gondur et al., 2024). These approaches learn latent representations using a variety of modeling strategies, including recurrent dynamical models (Hurwitz et al., 2021; Sani et al., 2024), contrastive objectives (Schneider et al., 2023), and variational inference (Gondur et al., 2024). However, none of these methods is designed for handling behavior and multiple neural time-series simultaneously, which our method enables even when neural modalities exhibit distinct temporal resolutions.

The second category spans both linear and nonlinear models that fuse information across multiple neural modalities. These approaches include linear multiscale dynamical systems (Coleman et al., 2011; Abbaspourazad et al., 2021; Rezaei et al., 2023), a linear subspace identification method named MSID (Ahmadipour et al., 2024), and nonlinear methods (Kramer et al., 2022; Brenner et al., 2024; Krishna et al., 2025). However, these methods do not separate behaviorally relevant and residual neural dynamics. Further, current nonlinear multimodal dynamical models are designed for fusing modalities with the same (rather than distinct) temporal resolutions.

3 METHODS

The architecture of BREM-NET, depicted in Fig. 1, is designed for joint modeling of behavior and multiple neural modalities observed simultaneously. It addresses two key challenges: 1) separating behaviorally relevant from residual neural dynamics, which we achieve with two RNNs trained through a structured three-stage learning process, and 2) handling asynchronous modalities and dropouts, which we achieve with modality-specific encoders that process each modality at its native temporal resolution.

We model multimodal neural and behavior time-series as observations of a nonlinear dynamical system (Eq.1), whose evolution is captured with latent states $\mathbf{X} = \{\mathbf{x}_k : \mathbf{x}_k \in \mathbb{R}^{n_x}, k \in \mathcal{K}\}$, where k is the time index and $\mathcal{K} = \{1, \ldots, \mathbf{K}\}$. Specifically, we infer the latent states by fusing information across two neural time-series which can have distinct temporal resolutions: $\mathbf{S} = \{\mathbf{s}_k : \mathbf{s}_k \in \mathbb{R}^{n_s}, k = 1, \ldots, K\}$, $\mathbf{Y} = \{\mathbf{y}_t : \mathbf{y}_t \in \mathbb{R}^{n_y}, t \in \mathcal{T}\}$ with $\mathcal{T} = \{1, \ldots, \mathbf{T}\}$. These distinct index sets (k, t) emphasize that neural modalities can be sampled at different rates, with k indexing the faster sampling rate. Our goal, however, is to perform latent state inference on the timescale of the faster modality, thereby enabling more flexible behavior decoding. In addition to neural observations, during learning, BREM-NET incorporates $\mathbf{Z} = \{\mathbf{z}_k : \mathbf{z}_k \in \mathbb{R}^{n_z}, k = 1, \ldots, K\}$, a behavioral time-series that can be

decoded from the latent states. We separate the latents $\mathbf{X}$ into two sets: behaviorally relevant neural latents $\mathbf{X}^{(1)} = \{\mathbf{x}_k^{(1)} : \mathbf{x}_k^{(1)} \in \mathbb{R}^{n_1}\}$ and residual neural latents $\mathbf{X}^{(2)} = \{\mathbf{x}_k^{(2)} : \mathbf{x}_k^{(2)} \in \mathbb{R}^{n_x - n_1}\}$.

3.1 Model Formulation

We assume that the multimodal neural and behavioral time-series are generated from a nonlinear latent dynamical system described by:

$$\begin{aligned} \mathbf{x}_{k+1} &= F(\mathbf{x}_k) + \mathbf{w}_k \\ \mathbf{y}_t \mid \mathbf{x}_t &\sim p_{\mathbf{y}}\big(\mathbf{y}_t \mid \bar{\boldsymbol{C}}_{\mathbf{y}}(\mathbf{x}_t)\big) \\ \mathbf{s}_k \mid \mathbf{x}_k &\sim p_{\mathbf{s}}\big(\mathbf{s}_k \mid \bar{\boldsymbol{C}}_{\mathbf{s}}(\mathbf{x}_k)\big) \\ \mathbf{z}_k \mid \mathbf{x}_k &\sim p_{\mathbf{z}}\big(\mathbf{z}_k \mid \bar{\boldsymbol{C}}_{\mathbf{z}}(\mathbf{x}_k)\big) \end{aligned} \tag{1}$$

In this system, the latent state evolves in time through a recursion function $F(\cdot)$ with additive noise $\mathbf{w}_k$. The latent state is connected to three types of time-series through different distributions $p_y\big(\mathbf{y}_t \mid \bar{\boldsymbol{C}}_{\mathbf{y}}(\mathbf{x}_t)\big)$, $p_s\big(\mathbf{s}_k \mid \bar{\boldsymbol{C}}_{\mathbf{s}}(\mathbf{x}_k)\big)$, and $p_z\big(\mathbf{z}_k \mid \bar{\boldsymbol{C}}_{\mathbf{z}}(\mathbf{x}_k)\big)$ that describe the likelihood distributions of distinct neural time-series modalities and behavior time-series respectively, with $\bar{\boldsymbol{C}}_{\mathbf{y}}(\cdot), \bar{\boldsymbol{C}}_{\mathbf{s}}(\cdot), \bar{\boldsymbol{C}}_{\mathbf{z}}(\cdot)$ being nonlinear decoder networks. These distributions can take forms such as Poisson for spike counts, Gaussian for continuous signals, and Bernoulli for binary outcomes.

We jointly and nonlinearly model multimodal neural time-series $\mathbf{S}$ and $\mathbf{Y}$ with distinct temporal resolutions to infer the latent states using an RNN:

$$\mathbf{x}_{k+1} = A(\mathbf{x}_k) + \mathbf{Enc}_{\mathbf{y}}(\mathbf{y}_t) + \mathbf{Enc}_{\mathbf{s}}(\mathbf{s}_k) \tag{2}$$

where $A(\cdot)$ captures the temporal evolution of latent states, and $\mathbf{Enc}_{\mathbf{y}}(\cdot)$ and $\mathbf{Enc}_{\mathbf{s}}(\cdot)$ are modality-specific nonlinear encoder networks. This nonlinear multimodal dynamical modeling framework enables BREM-NET to leverage complementary information across distinct neural modalities to infer a shared latent space with more expressive dynamics. In addition, BREM-NET explicitly addresses the challenge of different temporal resolutions and modality-specific dropouts, which are common in real-world neural recordings; this is achieved via the separate modality-specific encoders $\mathbf{Enc}_{\mathbf{y}}(\cdot)$ and $\mathbf{Enc}_{\mathbf{s}}(\cdot)$. If one modality is unavailable at a given k, the corresponding encoder output is set to zero in Eq. 2, so the state update relies only on any available modality (See Appendix A.1 for details).

In addition to multimodal neural dynamical modeling, another major goal of BREM-NET is to functionally separate the behaviorally relevant and residual neural dynamics within the multimodal neural time-series. We define two sets of latent states (Sani et al., 2021; Hurwitz et al., 2021; Sani et al., 2024): (1) behaviorally relevant neural latents $\mathbf{x}_k^{(1)} \in \mathbb{R}^{n_1}$, which capture the shared dynamics between behavior and multimodal neural time-series, and (2) residual neural latents $\mathbf{x}_k^{(2)} \in \mathbb{R}^{n_x - n_1}$, which encode the residual neural dynamics that $\mathbf{x}_k^{(1)}$ leaves unexplained (see Appendix A.1.2 for details). Following these frameworks, the two sets of latents are defined by what they are learned to predict, not by statistical independence: $\mathbf{x}_k^{(1)}$ is learned first to predict behavior, and $\mathbf{x}_k^{(2)}$ is then learned, given $\mathbf{x}_k^{(1)}$, to predict the neural variability that $\mathbf{x}_k^{(1)}$ leaves unexplained. We do not constrain the two to be orthogonal or independent, so $\mathbf{x}_k^{(2)}$ may carry behavior-correlated variability; we instead test whether it adds behavior information beyond $\mathbf{x}_k^{(1)}$ (Section 4.2.1). Likewise, n_1 is the smallest dimensionality at which behavior decoding saturates for this model class (Appendix A.5.7), a model-selection choice rather than an estimate of intrinsic dimensionality. We then expand our formulation in a structured form as follows,

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_{k+1}^{(1)} \\ \mathbf{x}_{k+1}^{(2)} \end{bmatrix} &= \begin{bmatrix} f^{(1)}\big(\mathbf{x}_k^{(1)}, \mathbf{y}_t, \mathbf{s}_k\big) \\ f^{(2)}\big(\mathbf{x}_k^{(2)}, \mathbf{y}_t, \mathbf{s}_k, \mathbf{x}_{k+1}^{(1)}\big) \end{bmatrix} \\ \hat{\mathbf{y}}_t &= \boldsymbol{C}_{\mathbf{y}}^{(1)}(\mathbf{x}_t^{(1)}) + \boldsymbol{C}_{\mathbf{y}}^{(2)}(\mathbf{x}_t^{(2)}) \\ \hat{\lambda}_k &= \boldsymbol{C}_{\mathbf{s}}^{(1)}(\mathbf{x}_k^{(1)}) \times \boldsymbol{C}_{\mathbf{s}}^{(2)}(\mathbf{x}_k^{(2)}) \\ \hat{\mathbf{z}}_k &= \boldsymbol{C}_{\mathbf{z}}(\mathbf{x}_k^{(1)}) \end{aligned} \tag{3}$$

where $f^{(1)}(\cdot)$ and $f^{(2)}(\cdot)$ denote the RNNs that describe the dynamics of the behaviorally relevant neural latents and residual neural latents, respectively. Note that as the latent state at time $k+1$ is

estimated purely using past neural data (that is, $y_1, ..., y_t, s_1, ...s_k$), $\mathbf{x}_{k+1}$ supports causal predictions and is generated by the two RNNs as follows (using Eq. 2),

$$\begin{aligned}\mathbf{x}_{k+1}^{(1)} &= f^{(1)}\big(\mathbf{x}_k^{(1)}, \mathbf{y}_t, \mathbf{s}_k\big) = A^{(1)}(\mathbf{x}_k^{(1)}) + \mathbf{Enc}_{\mathbf{y}}^{(1)}(\mathbf{y}_t) + \mathbf{Enc}_{\mathbf{s}}^{(1)}(\mathbf{s}_k), \\ \mathbf{x}_{k+1}^{(2)} &= f^{(2)}\big(\mathbf{x}_k^{(2)}, \mathbf{y}_t, \mathbf{s}_k, \mathbf{x}_{k+1}^{(1)}\big) = A^{(2)}(\mathbf{x}_k^{(2)}) + \mathbf{Enc}_{\mathbf{y}}^{(2)}(\mathbf{y}_t, \mathbf{x}_{k+1}^{(1)}) + \mathbf{Enc}_{\mathbf{s}}^{(2)}(\mathbf{s}_k, \mathbf{x}_{k+1}^{(1)})\end{aligned} \tag{4}$$

In this structured form, each RNN, i.e., $f^{(1)}(\cdot)$ and $f^{(2)}(\cdot)$, is parameterized by a recursion function, $A(\cdot)$, and by modality-specific encoders, $\mathbf{Enc}_{\mathbf{s}}(\cdot)$ and $\mathbf{Enc}_{\mathbf{y}}(\cdot)$, for behaviorally relevant neural latents and residual neural latents, which are denoted by the appropriate superscripts. Further, $\boldsymbol{C}_{\mathbf{y}}(\cdot)$, $\boldsymbol{C}_{\mathbf{s}}(\cdot)$, and $\boldsymbol{C}_{\mathbf{z}}(\cdot)$ are modality-specific decoders that predict the observed multimodal neural and behavior time-series from the two sets of latents. In Eq. 3, $\hat{\lambda}_k$ is the predicted firing rate for Poisson neural modality $\mathbf{s}_k$, $\hat{\mathbf{y}}_t$ is the predicted Gaussian neural modality, and $\hat{\mathbf{z}}_k$ represents the predicted behavior. Note that behavior is predicted from the behaviorally relevant neural latents while the neural modalities are predicted using both behaviorally relevant neural latents and residual neural latents; their contributions are fused by adding the Gaussian components and multiplying the Poisson firing-rates (equivalently summing log-rates; see Appendix A.1.1). Also, in our model, we pass $\mathbf{x}_{k+1}^{(1)}$ to the second RNN when computing $\mathbf{x}_{k+1}^{(2)}$ such that it learns the residual neural dynamics not already learned by the first RNN as detailed below and in Appendix A.1.2.

3.2 Structured learning of behaviorally relevant and residual neural dynamics

To functionally separate behaviorally relevant multimodal neural dynamics from residual neural dynamics in multiple neural time-series, we learn our model in Eq. 3 using a structured three-stage learning procedure.

Stage 1 (Supervised training of behaviorally relevant neural latents): We first train the RNN, $f^{(1)}(\cdot)$ – composed of networks $A^{(1)}(\cdot)$, $\mathbf{Enc}_{\mathbf{y}}{}^{(1)}(\cdot)$, and $\mathbf{Enc}_{\mathbf{s}}{}^{(1)}(\cdot)$ – jointly with the behavior decoder $\boldsymbol{C}_{\mathbf{z}}(\cdot)$. These components are parameterized as multi-layer perceptrons (MLPs) and optimized to predict the behavior time-series from neural inputs. Specifically, Stage 1 minimizes the negative log-likelihood (NLL) of behavior given the inferred behaviorally relevant neural latents $\mathbf{x}_k^{(1)}$:

$$\mathbf{L}_{\mathbf{z}} = \sum_k \mathbf{NLL}\big(\mathbf{z}_k; \boldsymbol{C}_{\mathbf{z}}(\mathbf{x}_k^{(1)})\big) \tag{5}$$

This optimization ensures that the learned latents in this stage capture behaviorally-relevant information in the multimodal neural data.

Stage 2 (Training neural decoders for behaviorally-relevant neural latents): Next, we train the neural decoders, $\boldsymbol{C}_{\mathbf{y}}^{(1)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{s}}^{(1)}(\cdot)$, which are again formed as MLPs and predict the neural modalities from the behaviorally relevant neural latents $\mathbf{x}_k^{(1)}$. In this stage, $f^{(1)}(\cdot)$ (and thus $\mathbf{x}_k^{(1)}$) is fixed, and we optimize $\boldsymbol{C}_{\mathbf{y}}^{(1)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{s}}^{(1)}(\cdot)$ by minimizing the combined NLL of both modalities using their different likelihood distributions, with a scaling factor α to balance their scale differences (see Appendix A.1.3).

$$\mathbf{L}_{\mathbf{y},\mathbf{s}}^{(1)} = \sum_t \mathbf{NLL}_{\mathbf{y}}\big(\mathbf{y}_t; \boldsymbol{C}_{\mathbf{y}}^{(1)}(\mathbf{x}_t^{(1)})\big) + \alpha \sum_k \mathbf{NLL}_{\mathbf{s}}\big(\mathbf{s}_k; \boldsymbol{C}_{\mathbf{s}}^{(1)}(\mathbf{x}_k^{(1)})\big) \tag{6}$$

Stage 3 (Unsupervised training of residual neural latents): We now train the second RNN, $f^{(2)}(\cdot)$ – composed of $A^{(2)}(\cdot)$, $\mathbf{Enc}_{\mathbf{y}}{}^{(2)}(\cdot)$ and $\mathbf{Enc}_{\mathbf{s}}{}^{(2)}(\cdot)$ – jointly with the decoders $\boldsymbol{C}_{\mathbf{y}}^{(2)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{s}}^{(2)}(\cdot)$ by minimizing the combined NLL of both neural modalities given both behaviorally relevant neural latents ($\mathbf{x}_k^{(1)}$) and residual neural latents ($\mathbf{x}_k^{(2)}$). Since $\boldsymbol{C}_{\mathbf{y}}^{(1)}(\cdot)$, and $\boldsymbol{C}_{\mathbf{s}}^{(1)}(\cdot)$ and the value of $\mathbf{x}_k^{(1)}$ are already learned in Stages 1 and 2 and account for the part of $\mathbf{y}_t$ and $\mathbf{s}_k$ that are predictable with behaviorally relevant neural latents, this stage is trained to explain any residual dynamics in the neural data. Thus, to learn these residual dynamics, we form the loss as:

$$\mathbf{L}_{\mathbf{y},\mathbf{s}}^{(2)} = \sum_t \mathbf{NLL}_{\mathbf{y}}\big(\mathbf{y}_t; \boldsymbol{C}_{\mathbf{y}}^{(1)}(\mathbf{x}_t^{(1)}), \boldsymbol{C}_{\mathbf{y}}^{(2)}(\mathbf{x}_t^{(2)})\big) + \alpha \sum_k \mathbf{NLL}_{\mathbf{s}}\big(\mathbf{s}_k; \boldsymbol{C}_{\mathbf{s}}^{(1)}(\mathbf{x}_k^{(1)}), \boldsymbol{C}_{\mathbf{s}}^{(2)}(\mathbf{x}_k^{(2)})\big) \tag{7}$$

Overall, this process learns the evolution of spikes, LFPs, and behavior with a unified nonlinear dynamical model, first fusing neural modalities to learn behaviorally relevant latents and then modeling the residual multimodal neural dynamics that these latents leave unexplained.

4 EXPERIMENTAL RESULTS

4.1 DISTINCT DATASETS

We validated BREM-NET on two distinct NHP datasets from different labs and involving different behavioral tasks, neural signal modalities, and experimental paradigms, as well as on simulated data (Appendix A.5.2). The dataset in (O'Doherty et al., 2020) consists of simultaneous recordings of discrete spiking activity and raw LFP waveforms from the primary motor cortex (M1) while an NHP performed sequential 2D reaching movements to random targets in virtual reality (Fig. 3a, Appendix A.3.1). The dataset in (Flint et al., 2012) includes spiking activity and LFP power features recorded from motor and premotor cortices during a center-out reaching task, whose results are provided in Appendix A.5.1.

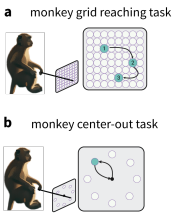

Figure 2: Datasets.

4.2 MULTIMODAL MODELING IN THE NHP GRID REACHING DATASET

We first applied BREM-NET to the grid-reaching dataset, treating spike counts and raw LFPs as neural time-series and the 2D cursor velocity as behavior. For all sessions, we fixed the total latent dimensionality at 64 ($n_x = 64$), allocating 16 dimensions for behaviorally relevant neural latents ($n_1 = 16$). These fixed choices were made based on initial exploration of the model on a single session of the dataset (Fig. A.7a-d, Appendix A.1.4).

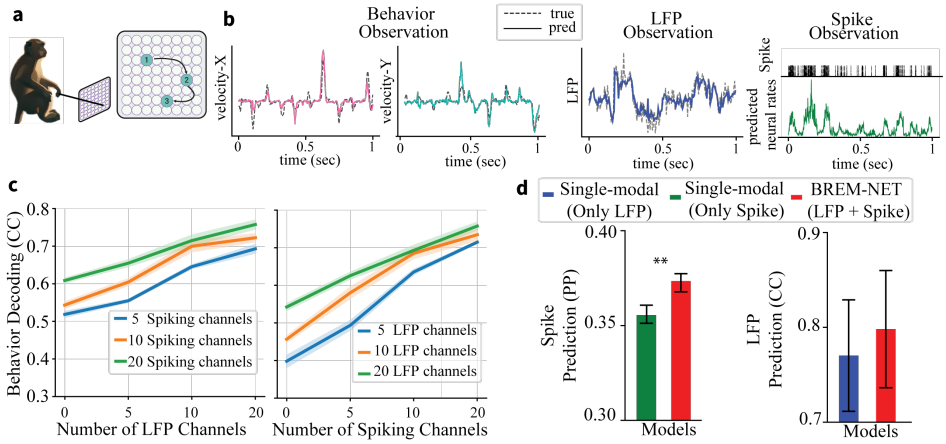

Figure 3: **BREM-NET successfully fuses multimodal neural time-series, thus improving neural-behavioral predictions in the NHP grid reaching dataset.** **(a)** Dataset and task. **(b)** True (dashed) vs. predicted (solid) behavior, LFP, and spikes for a representative session. **(c)** *left*: Cross-validated behavior decoding CC when 5, 10, or 20 spiking channels were the primary modality and increasing numbers of LFP channels were fused; *right*: same with LFP as the primary modality. **(d)** Spike prediction (PP, *left*) and LFP prediction (CC, *right*) of BREM-NET vs. single-modal models. Mean $\pm$ SEM (N=20). Asterisks indicate significance of comparison; **: $p < 0.001$, one-sided Wilcoxon signed-rank test.

4.2.1 MULTIMODAL FUSION EXTRACTS MORE ACCURATE BEHAVIORALLY RELEVANT NEURAL LATENTS

To assess multimodal neural fusion for learning behaviorally relevant and residual neural latents, we conducted a series of experiments where either spikes or LFPs served as the primary modality, and increasing numbers of channels from the other modality were fused (Fig. 3c). When spiking activity served as the primary modality, the addition of LFP channels consistently enhanced behavior decoding across varying spike channel counts (Fig. 3c, left, $p < 10^{-5}$, $n = 20$, one-sided Wilcoxon signed-rank test). A similar trend was observed when LFPs were used as the primary modality (Fig. 3c, right).

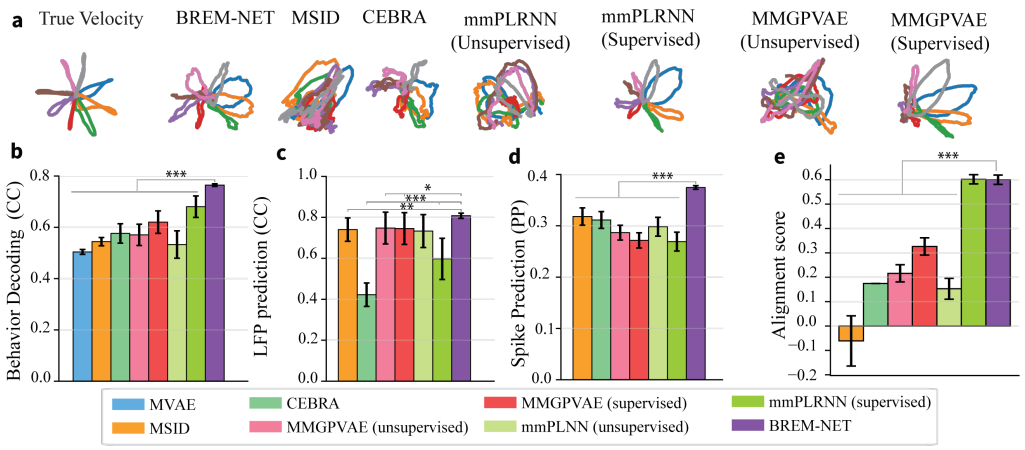

Figure 4: **Latents inferred by BREM-NET are more aligned with true behavior trajectories and achieve accurate neural-behavioral prediction. (a)** True velocity trajectories vs. condition-averaged latent trajectories learned by BREM-NET and baselines for the NHP grid-reaching task. **(b)** Behavior decoding CC. **(c)** LFP prediction CC. **(d)** Spike prediction PP. **(e)** Alignment score. Bars represent mean $\pm$ SEM across all sessions, 5 folds (N=20) and asterisks indicate statistical significance (*: $p < 0.05$, **: $p < 0.001$, ***: $p < 10^{-7}$; one-sided Wilcoxon signed-rank test).

Notably, the improvements were more pronounced when LFP channels were the primary modality, suggesting that spiking activity encoded more behavior-related information compared to LFPs in this dataset. More generally, fusion yields its largest gains when the primary modality is information-limited (e.g., fewer primary channels in Fig. 3c), the regime characteristic of chronic BCIs in which spike quality degrades over time while LFPs remain comparatively stable (Stavisky et al., 2015). In addition to using 20 spiking and 20 LFP channels as done for the main analyses, we also performed an extended analysis using all available channels and again observed that BREM-NET resulted in improvements over single-modal variants; consistent with the trend above, this improvement was smaller in this high-information regime (Appendix A.5.3).

To evaluate the separation of learned behaviorally relevant latents ($\mathbf{x}^{(1)}$) and residual neural latents ($\mathbf{x}^{(2)}$), we assessed two properties: sufficiency of $\mathbf{x}^{(1)}$ for behavior, and specialization of the two sets of latents. First, for sufficiency, we evaluated the informativeness of each of the 64 learned latents for neural-behavioral prediction (Fig. A.8). Behavior decoding saturates within the 16 latents learned in Stage 1, and adding the residual neural latents provides no further improvement, indicating that $\mathbf{x}^{(2)}$ adds no decodable behavior information beyond $\mathbf{x}^{(1)}$. In contrast, neural prediction improves gradually across all 64 latents learned across Stages 1 and 3. Second, for specialization, we used $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ separately (Fig. A.6b) and found that $\mathbf{x}^{(1)}$ accurately decodes behavior, while $\mathbf{x}^{(2)}$ excels in predicting neural activity but decodes behavior substantially less accurately. Finally, as a complementary geometric characterization, true velocity trajectories are much more aligned with the trajectories of $\mathbf{x}^{(1)}$ than with those of $\mathbf{x}^{(2)}$ (Fig. A.6c). Together, these results show that behavior information is concentrated in $\mathbf{x}^{(1)}$, learned in Stage 1, while $\mathbf{x}^{(2)}$, learned in Stage 3, primarily captures the remaining neural variability. We further evaluated whether the learned behaviorally relevant latent structure is consistent across sessions by extending our model to a multi-session setting. Our results in Fig. A.5 show that a shared model recovers similar latent representations and maintains comparable decoding performance across sessions (see Appendix A.5.6).

Finally, we compared BREM-NET to several multimodal baselines: MVAE (Wu & Goodman, 2018) (generic variational autoencoder without temporal modeling or behavioral supervision), MSID (Ahmadipour et al., 2024), multi-session CEBRA (Schneider et al., 2023), mmPLRNN (Kramer et al., 2022), and MMGPVAE (Gondur et al., 2024) (Fig. 4; see Appendix A.2 for details). For MMGPVAE and mmPLRNN, we used both neural modalities as inputs and trained their original unsupervised versions as well as supervised versions with an added behavior reconstruction loss (Appendix A.2.9, A.2.10). First, we found that the behaviorally relevant neural latents learned by BREM-NET are more aligned to true behavior signals (velocity) (Fig. 4a,e), as quantified by

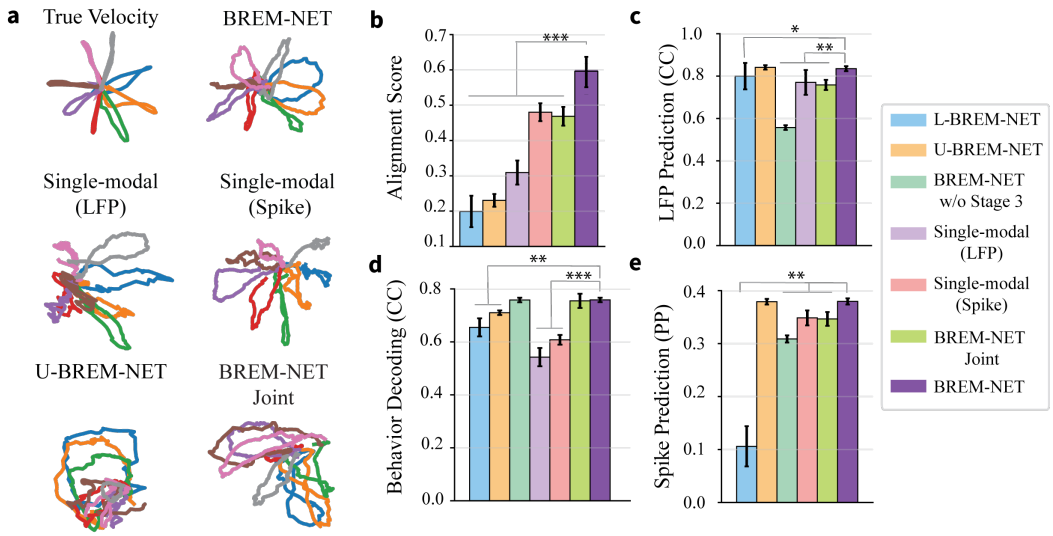

Figure 5: **Ablations show that multimodal structured modeling improves the accuracy and behavior alignment of behaviorally relevant latents while capturing residual neural variability.** **(a)** Condition-averaged behaviorally relevant latent trajectories of BREM-NET and its variants. **(b)** Alignment score. **(c)** LFP prediction (CC). **(d)** Behavior decoding (CC). **(e)** Spike prediction (PP). Mean $\pm$ SEM across all sessions and 5 folds (N=20); asterisks indicate statistical significance.

an alignment score defined as the Pearson correlation between the Euclidean distance matrices of behavior signals and the latent trajectories (Jiang et al., 2025). Second, we found that BREM-NET achieves higher behavior decoding and neural prediction compared to baselines (Fig. 4b-d), showing that it extracts accurate behaviorally relevant and residual neural latents.

Next, we conducted a comprehensive ablation study to evaluate the contribution of components in BREM-NET (Fig. 5). To examine the effect of multimodal neural fusion on the quality of latents, we compared BREM-NET with Single-modal variants. As shown in Fig. 5a, the behaviorally relevant latent trajectories $\mathbf{x}^{(1)}$ inferred by BREM-NET are more aligned with behavior trajectories than those inferred from either of the Single-modal variants (LFP or Spike alone). Consistently, as shown in Fig. 5b, BREM-NET has a significantly higher alignment score than Single-modal variants. Also, BREM-NET surpasses Single-modal variants in neural-behavioral prediction (Fig. 5c-e), confirming the benefit of multimodal neural fusion while separating behaviorally relevant and residual neural dynamics.

4.2.2 Multimodal modeling and structured learning in BREM-NET are critical for uncovering accurate and aligned behaviorally relevant neural dynamics

To assess the impact of the architectural design, we compare BREM-NET to four variants: (1) L-BREM-NET, a fully linear counterpart, (2) U-BREM-NET, an unsupervised variant where Stages 1 and 2 are omitted, (3) BREM-NET without Stage 3, and (4) BREM-NET Joint, a variant without structured learning in which all latents are trained jointly using a combined loss for behavior decoding and neural prediction (Appendix A.2.4). First, as shown in Fig. 5b-e, BREM-NET outperforms L-BREM-NET in neural-behavioral predictions and alignment score. Second, compared to U-BREM-NET, true behavior trajectories are much more aligned with BREM-NET's latents than with U-BREM-NET's latents as shown in Fig. 5a-b. Further, while U-BREM-NET matches BREM-NET on neural predictions, BREM-NET significantly outperformed U-BREM-NET in behavioral decoding (Fig. 5c-e). Together, these results show the importance of behavior supervision in training the multimodal dynamical model. Third, while BREM-NET w/o Stage 3 matches BREM-NET's behavior prediction accuracy, it has a substantially lower neural prediction accuracy, showing the importance of learning residual neural dynamics in addition to behaviorally relevant neural dynamics

for accurate prediction of both neural-behavioral data (Fig. 5). Finally, true behavior trajectories are much more aligned with BREM-NET's latents than with BREM-NET Joint's latents (Fig. 5a-b).

4.2.3 BREM-NET FUSES NEURAL MODALITIES WITH ASYNCHRONOUS SAMPLING RATES AND DROPOUTS

We evaluated the robustness of BREM-NET when neural modalities are partially dropped out or sampled at different rates (different temporal resolutions). First, in the NHP grid-reaching dataset, we downsampled LFPs by a factor of 5. As shown in Tab. A.7, BREM-NET maintained its behavior decoding performance in this asynchronous setting ($p > 0.5$, $n = 20$, two-sided Wilcoxon signed-rank test), confirming that it does not require the modalities to be aligned in time. Appendix A.5.4 further compares masking with interpolation- and zero-imputation-based handling of the missing samples, and with MMGPVAE under the same setting. Next, to simulate noise, artifacts, or recording failures, we randomly dropped spike or LFP samples at each timestep with probability $p \in \{0.2, 0.4, 0.6, 0.8\}$. As demonstrated in Tab. A.8, BREM-NET showed strong robustness. Even for spikes that have a primary role in behavior decoding (Fig. 5a), performance dropped by only about 5% even with 40% spike dropout, and further our method was more robust to LFP dropouts. Overall, results show that BREM-NET can robustly fuse even asynchronous and incomplete neural modalities.

4.3 GENERALIZATION ON A DISTINCT MULTIMODAL NHP DATASET WITH A DIFFERENT CENTER-OUT TASK

To evaluate the generalizability of BREM-NET, we applied it to another distinct NHP dataset (Flint et al., 2012), in which an NHP performed center-out reaches with a manipulandum while spiking activity and LFP power bands were recorded from M1 and PMd; we used the 2D manipulandum velocity as behavior (Fig. A.1a, Appendix A.3.2). We trained BREM-NET with $n_1 = 16$ and $n_x = 32$ (Fig. A.7e-h). Similar to the NHP grid reaching dataset, behavior decoding performance improved through multimodal neural fusion (Fig. A.1c). Further, again our method separated behaviorally relevant neural dynamics from residual neural dynamics as shown in Fig. A.3b. Finally, BREM-NET's behavior decoding outperformed the baselines (Fig. A.2). While some unsupervised or contrastive baselines outperformed BREM-NET in neural prediction in this dataset, they achieved much lower behavior decoding performance than BREM-NET.

5 DISCUSSION

We introduced BREM-NET, a nonlinear dynamical model that integrates multiple neural modalities alongside behavior, using modality-specific encoders and decoders and separate behaviorally relevant and residual neural latents. Across two independent multimodal NHP datasets, our method functionally separated behaviorally relevant and residual neural dynamics and learned them more accurately, leading to better neural and behavior inferences, even when modalities had distinct temporal resolutions or modality-specific dropouts, which are natural challenges in real-world applications such as BCIs. Our ablations showed that these benefits arise from multimodal modeling and supervised structured learning.

By first learning a compact latent state that is predictive of behavior from multiple neural modalities, and then modeling the remaining neural dynamics in a separate set of latents, BREM-NET provides a tool to study how neural activity at different spatiotemporal scales jointly relates to behavior. For example, it can quantify how each modality contributes to behaviorally relevant dynamics and how fusion changes this contribution, as in our analyses of spikes and LFPs (Fig. 3c), while its compact behaviorally relevant state supports accurate behavior decoding for BCIs.

Limitations and future directions. We applied our method to spikes, raw LFPs, and LFP power features given their importance in neuroscience and BCIs. Important future directions include extending it to more modalities, such as EEG, intracranial EEG, or physiological signals, and to other brain regions and behaviors. Also, future work can extend BREM-NET to study shared latent dynamics across multiple brains, such as in social interaction paradigms. In such settings, a structured multimodal model could help isolate task-relevant dynamics shared across brains from those private to an individual brain.

AI USE STATEMENT

In this work, we used generative AI tools only to polish the writing of the manuscript, such as improving grammar, clarity, and wording. We did not use generative AI tools for research ideation, method development, implementation, experiments, data analysis, or the generation of results or figures. All AI-assisted edits were reviewed by the authors, and we take full responsibility for the final content of this work, including all text and claims.

ETHICS STATEMENT

This work analyzes publicly available, previously collected non-human primate electrophysiology datasets (O'Doherty et al., 2020; Flint et al., 2012); no new animal experiments or human-subject data were involved.

REPRODUCIBILITY STATEMENT

The model formulation and the three-stage learning procedure are described in Section 3 and Appendix A.1. Hyperparameters, architectures, and training settings are listed in Appendix A.1.4; baseline configurations are described in Appendix A.2; and dataset preprocessing is described in Appendix A.3. An anonymized link to implementation is provided as supplementary material.

REFERENCES

Hamidreza Abbaspourazad, Mahdi Choudhury, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Multiscale low-dimensional motor cortical state dynamics predict naturalistic reach-and-grasp behavior. *Nature communications*, 12(1):607, 2021.

Nur Ahmadi, Timothy G Constandinou, and Christos-Savvas Bouganis. Inferring entire spiking activity from local field potentials. *Scientific Reports*, 11(1):19045, 2021.

Parima Ahmadipour, Omid G Sani, Bijan Pesaran, and Maryam M Shanechi. Multimodal subspace identification for modeling discrete-continuous spiking and field potential population activity. *Journal of Neural Engineering*, 21(2):026001, 2024.

Andrei Belitski, Stefano Panzeri, Cesare Magri, Nikos K Logothetis, and Christoph Kayser. Sensory information in local field potentials and spikes from visual and auditory cortices: time scales and frequency bands. *Journal of computational neuroscience*, 29:533–545, 2010.

Manuel Brenner, Florian Hess, Georgia Koppe, and Daniel Durstewitz. Integrating multimodal data for joint generative modeling of complex dynamics. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4482–4516, 2024.

Eli Brookner. *Tracking and Kalman Filtering Made Easy*. John Wiley & Sons, 1998.

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

Todd P Coleman, Marianna Yanike, Wendy A Suzuki, and Emery N Brown. A mixed-filter algorithm for dynamically tracking learning from multiple behavioral and neurophysiological measures. In Mingzhou Ding and Dennis L. Glanzman (eds.), *The Dynamic Brain: An Exploration of Neuronal Variability and Its Functional Significance*, pp. 3–28. Oxford University Press, 2011.

Daniel Durstewitz. A state space approach for piecewise-linear recurrent neural networks for identifying computational dynamics from neural measurements. *PLoS computational biology*, 13 (6):e1005542, 2017.

Gaute T Einevoll, Christoph Kayser, Nikos K Logothetis, and Stefano Panzeri. Modelling and analysis of local field potentials for studying the function of cortical circuits. *Nature Reviews Neuroscience*, 14(11):770–785, 2013.

Robert D Flint, Eric W Lindberg, Luke R Jordan, Lee E Miller, and Marc W Slutzky. Accurate decoding of reaching movements from field potentials in the absence of spikes. *Journal of neural engineering*, 9(4):046006, 2012.

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

Rong Ge, Sham M Kakade, Rahul Kidambi, and Praneeth Netrapalli. The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares. *Advances in neural information processing systems*, 32, 2019.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Rabia Gondur, Usama Bin Sikandar, Evan Schaffer, Mikio Christian Aoi, and Stephen L Keeley. Multi-modal Gaussian process variational autoencoders for neural and behavioral data. In *The Twelfth International Conference on Learning Representations*, 2024.

Han-Lin Hsieh, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Multiscale modeling and decoding algorithms for spike-field activity. *Journal of Neural Engineering*, 16(1):016018, 2019.

Cole Hurwitz, Akash Srivastava, Kai Xu, Justin Jude, Matthew Perich, Lee Miller, and Matthias Hennig. Targeted neural dynamical modeling. *Advances in Neural Information Processing Systems*, 34:29379–29392, 2021.

Yiqi Jiang, Kaiwen Sheng, Yujia Gao, E. Kelly Buchanan, Yu Shikano, Seung Je Woo, Yixiu Zhao, Tony Hyun Kim, Fatih Dinc, Scott Linderman, and Mark Schnitzer. Extracting task-relevant preserved dynamics from contrastive aligned neural recordings. In *Advances in Neural Information Processing Systems*, 2025.

Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Xue-Lian Qi, Ranulfo Romo, Naoshige Uchida, and Christian K Machens. Demixed principal component analysis of neural population data. *elife*, 5:e10989, 2016.

Daniel Kramer, Philine Lou Bommer, Carlo Tombolini, Georgia Koppe, and Daniel Durstewitz. Reconstructing nonlinear dynamical systems from multi-modal time series. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11613–11633, 2022.

Nanda H. Krishna, Mathys Loiselle, Avery Hee-Woon Ryoo, Matthew G. Perich, and Guillaume Lajoie. Towards a generalizable, unified framework for multimodal neural decoding. In *NeurIPS 2025 Workshop on Foundation Models for the Brain and Body*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Hung-Yun Lu, Elizabeth S Lorenc, Hanlin Zhu, Justin Kilmarx, James Sulzer, Chong Xie, Philippe N Tobler, Andrew J Watrous, Amy L Orsborn, Jarrod Lewis-Peacock, and Samantha R Santacruz. Multi-scale neural decoding and analysis. *Journal of Neural Engineering*, 18(4):045013, 2021.

Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. *Advances in neural information processing systems*, 24, 2011.

Joseph E. O'Doherty, Mariana M. B. Cardoso, Joseph G. Makin, and Philip N. Sabes. Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology. Zenodo dataset, 2020.

Markku Penttonen and György Buzsáki. Natural logarithmic relationship between brain oscillators. *Thalamus & Related Systems*, 2(2):145–152, 2003.

Bijan Pesaran, John S Pezaris, Maneesh Sahani, Partha P Mitra, and Richard A Andersen. Temporal structure in neuronal activity during working memory in macaque parietal cortex. *Nature neuroscience*, 5(8):805–811, 2002.

Bijan Pesaran, Martin Vinck, Gaute T Einevoll, Anton Sirota, Pascal Fries, Markus Siegel, Wilson Truccolo, Charles E Schroeder, and Ramesh Srinivasan. Investigating large-scale brain dynamics using field potential recordings: analysis and interpretation. *Nature neuroscience*, 21(7):903–919, 2018.

Mehrdad Ramezani, Yundong Ren, Ertugrul Cubukcu, and Duygu Kuzum. Innovating beyond electrophysiology through multimodal neural interfaces. *Nature Reviews Electrical Engineering*, 2 (1):42–57, 2025.

Mohammad R Rezaei, Haseul Jeoung, Ayda Gharamani, Utpal Saha, Venkat Bhat, Milos R Popovic, Ali Yousefi, Robert Chen, and Milad Lankarany. Inferring cognitive state underlying conflict choices in verbal Stroop task using heterogeneous input discriminative-generative decoder model. *Journal of Neural Engineering*, 20(5):056016, 2023.

Omid G Sani, Hamidreza Abbaspourazad, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1):140–149, 2021.

Omid G Sani, Bijan Pesaran, and Maryam M Shanechi. Dissociative and prioritized modeling of behaviorally relevant neural dynamics using recurrent neural networks. *Nature neuroscience*, 27 (10):2033–2045, 2024.

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

Auguste Schulz, Julius Vetter, Richard Gao, Daniel Morales, Victor Lobato-Rios, Pavan Ramdya, Pedro J Gonçalves, and Jakob H Macke. Modeling conditional distributions of neural and behavioral data with masked variational autoencoders. *Cell Reports*, 44(3):115338, 2025.

Jonnathan Singh Alvarado, Jack Goffinet, Valerie Michael, William Liberti III, Jordan Hatfield, Timothy Gardner, John Pearson, and Richard Mooney. Neural dynamics underlying birdsong practice and performance. *Nature*, 599(7886):635–639, 2021.

Sergey D Stavisky, Jonathan C Kao, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. A high performing brain–machine interface driven by low-frequency local field potentials alone and together with spikes. *Journal of neural engineering*, 12(3):036009, 2015.

Wilson Truccolo, Leigh R Hochberg, and John P Donoghue. Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes. *Nature neuroscience*, 13(1):105–111, 2010.

Anne E Urai, Brent Doiron, Andrew M Leifer, and Anne K Churchland. Large-scale neural recordings call for new insights to link brain and behavior. *Nature neuroscience*, 25(1):11–19, 2022.

Parsa Vahidi, Omid G. Sani, and Maryam M. Shanechi. BRAID: Input-driven nonlinear dynamical modeling of neural-behavioral data. In *International Conference on Learning Representations*, 2025.

Yule Wang, Chengrui Li, Weihan Li, and Anqi Wu. Exploring behavior-relevant and disentangled neural dynamics with generative diffusion models. *Advances in Neural Information Processing Systems*, 37:34712–34736, 2024.

Matthew R Whiteway, Dan Biderman, Yoni Friedman, Mario Dipoppa, E Kelly Buchanan, Anqi Wu, John Zhou, Niccolò Bonacchi, Nathaniel J Miska, Jean-Paul Noel, et al. Partitioning variability in animal behavioral videos using semi-supervised variational autoencoders. *PLoS computational biology*, 17(9):e1009439, 2021.

Joshua Huang Wu, Hari Koneru, James Russell Ravenel, Anshuman Sabath, James Michael Roach, Shaun Sze-Xian Lim, Michael R Tadross, Alex H Williams, and Timothy W Dunn. Disentangling 3D animal pose dynamics with scrubbed conditional latent variables. In *The Thirteenth International Conference on Learning Representations*, 2025.

Mike Wu and Noah Goodman. Multimodal generative models for scalable weakly-supervised learning. *Advances in neural information processing systems*, 31, 2018.

Daiyao Yi, Hao Dong, Michael James Higley, Anne Churchland, and Shreya Saxena. Shared-AE: Automatic identification of shared subspaces in high-dimensional neural and behavioral activity. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yizi Zhang, Yanchen Wang, Mehdi Azabou, Alexandre Andre, Zixuan Wang, Hanrui Lyu, The International Brain Laboratory, Eva Dyer, Liam Paninski, and Cole Hurwitz. Neural encoding and decoding at scale. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76175–76192, 2025.

Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-VAE. *Advances in Neural Information Processing Systems*, 33:7234–7247, 2020.

A APPENDIX

A.1 METHOD DETAILS

A.1.1 METHOD FORMULATION

In Equation 3, we provided a compact formulation of our RNN-based architecture. Here, we present a more detailed formulation that explicitly describes the latent dynamics and the associated model components used in BREM-NET. The full model is defined as:

$$\begin{cases} \begin{bmatrix} \mathbf{x}_{k+1}^{(1)} \\ \mathbf{x}_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} A^{(1)}(\mathbf{x}_k^{(1)}) \\ A^{(2)}(\mathbf{x}_k^{(2)}) \end{bmatrix} + \begin{bmatrix} \mathbf{Enc}_{\mathbf{y}}^{(1)}(\mathbf{y}_t) + \mathbf{Enc}_{\mathbf{s}}^{(1)}(\mathbf{s}_k) \\ \mathbf{Enc}_{\mathbf{y}}^{(2)}(\mathbf{y}_t, \mathbf{x}_{t+1}^{(1)}) + \mathbf{Enc}_{\mathbf{s}}^{(2)}(\mathbf{s}_k, \mathbf{x}_{k+1}^{(1)}) \end{bmatrix} \\ \hat{\mathbf{y}}_t = \boldsymbol{C}_{\mathbf{y}}^{(1)}(\mathbf{x}_t^{(1)}) + \boldsymbol{C}_{\mathbf{y}}^{(2)}(\mathbf{x}_t^{(2)}) \\ \hat{\lambda}_k = \boldsymbol{C}_{\mathbf{s}}^{(1)}(\mathbf{x}_k^{(1)}) \times \boldsymbol{C}_{\mathbf{s}}^{(2)}(\mathbf{x}_k^{(2)}) \\ \hat{\mathbf{s}}_k \sim \mathcal{P}ois(\hat{\lambda}_k) \\ \hat{\mathbf{z}}_k = \boldsymbol{C}_{\boldsymbol{z}}(\mathbf{x}_k^{(1)}) \end{cases} \tag{8}$$

where $A^{(1)}(\cdot)$, and $A^{(2)}(\cdot)$ represents the state transition function, capturing the temporal evolution of the latent state, and $\mathbf{Enc}_{\mathbf{y}}^{(1)}(\cdot)$, $\mathbf{Enc}_{\mathbf{s}}^{(1)}(\cdot)$, $\mathbf{Enc}_{\mathbf{y}}^{(2)}(\cdot)$, and $\mathbf{Enc}_{\mathbf{s}}^{(2)}(\cdot)$ are modality-specific encoders for $\mathbf{y}_t$ and $\mathbf{s}_k$, respectively. We use distinct time indices t and k for y and s respectively to indicate that neural modalities can be sampled at different temporal resolutions with k indexing the faster sampling rate. At each time step k, the latent state is updated by combining the predicted state evolution with contributions from any available modalities through their corresponding encoders. When a modality is not observed at a given time step (e.g., due to asynchronous sampling or dropout), its encoder contribution is set to zero, such that no update is introduced from that modality. Consequently, the latent state is inferred using the prior prediction together with any available modalities at that time step. This mechanism is analogous to Kalman filtering (Brookner, 1998) with missing observations, where the Kalman gain for unobserved measurement dimensions becomes zero (equivalently, assuming infinite observation noise), resulting in no correction from those dimensions and relying solely on the prediction and available measurements.

Here, the overall latent state $\mathbf{x}_k \in \mathbb{R}^{n_x}$, which captures the underlying dynamics of the neural-behavioral data, is designed to separate behaviorally relevant neural dynamics, represented by $\mathbf{x}_k^{(1)} \in \mathbb{R}^{n_1}$, from multimodal residual neural dynamics, represented by $\mathbf{x}_k^{(2)} \in \mathbb{R}^{n_x - n_1}$. We also note that each $\mathbf{x}_k$ is estimated purely using past neural data (that is, $y_1, ..., y_{t-1}, s_1, ...s_{k-1}$), so $\mathbf{x}_{k+1}$ has the information about neural modalities up to time k. As such, our model supports causal predictions.

Lines 2-4 of Equation 8 describe how the model predicts observations:

2: $\hat{\mathbf{y}}_t$ is the predicted Gaussian neural signal, combining contributions from both latent states $\mathbf{x}_k^{(1)}$ and $\mathbf{x}_k^{(2)}$ through the nonlinear decoders $\boldsymbol{C}_{\mathbf{y}}^{(1)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{y}}^{(2)}(\cdot)$. This prediction is performed in Stage 2 and Stage 3 of the architecture (Figure 1).

3: $\hat{\lambda}_k$ represents the predicted firing rate of spike counts, which also combines $\mathbf{x}_k^{(1)}$ and $\mathbf{x}_k^{(2)}$ through the nonlinear decoders $\boldsymbol{C}_{\mathbf{s}}^{(1)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{s}}^{(2)}(\cdot)$. Notably, unlike the Gaussian decoders $\boldsymbol{C}_{\mathbf{y}}^{(1)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{y}}^{(2)}(\cdot)$, our Poisson decoders $\boldsymbol{C}_{\mathbf{s}}^{(1)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{s}}^{(2)}(\cdot)$, employ an $\exp(.)$ nonlinearity output layer activation function to force non-negativity of Poisson rates. Therefore, Poisson decoders can be expressed as:

$$\begin{aligned} \boldsymbol{C}_{\boldsymbol{s}}^{(1)}(\cdot) &= \exp\left(\boldsymbol{C}_{\boldsymbol{s}}''^{(1)}(\cdot)\right) \\ \boldsymbol{C}_{\boldsymbol{s}}^{(2)}(\cdot) &= \exp\left(\boldsymbol{C}_{\boldsymbol{s}}''^{(2)}(\cdot)\right) \end{aligned} \tag{9}$$

where, $\boldsymbol{C}_{\boldsymbol{s}}''(\cdot)$ has no output layer activation function. As a result, predicted firing rate can be formulated as:

$$\hat{\lambda}_k = \exp\left(\boldsymbol{C}_{\boldsymbol{s}}''^{(1)}(\mathbf{x}_k^{(1)}) + \boldsymbol{C}_{\boldsymbol{s}}''^{(2)}(\mathbf{x}_k^{(2)})\right) \tag{10}$$

This step is also done in Stage 2 and Stage 3 of the architecture (Figure 1).

4: $\hat{\mathbf{z}}_k$ is the predicted behavior, decoded from the low-dimensional latents of Stage 1 using the nonlinear decoder $\boldsymbol{C}_{\mathbf{z}}(\cdot)$ learned during Stage 1 (Figure 1).

A.1.2 Learning Stages

This section details the structured three-stage optimization procedure used to train BREM-NET as described in Equation 8. Each stage is designed to optimize distinct components of the model in order to accurately predict observations while separating behaviorally relevant neural dynamics from residual neural dynamics. Across all stages, the negative log-likelihood (NLL) of the observed data is used as the training objective.

Stage 1:

First, we fit an RNN, composed of $A^{(1)}(\cdot)$ and modality-specific encoders, $\mathbf{Enc}_{\mathbf{y}}^{(1)}(\cdot)$ and $\mathbf{Enc}_{\mathbf{s}}^{(1)}(\cdot)$, jointly with the behavior decoder network $\boldsymbol{C}_{\mathbf{z}}(\cdot)$, using n_1 states, to minimize behavior prediction NLL given past neural activities $\mathbf{y}_k$, and $\mathbf{s}_k$ (Equation 11). This ensures that the neural dynamics that are predictive of behavior are learned. The resulting latent states, $\mathbf{x}_k^{(1)}$, form the first set of latents in BREM-NET which are behaviorally relevant latents. The optimization can be expressed as:

$$\begin{cases} \mathbf{x}_{k+1}^{(1)} = A^{(1)}(\mathbf{x}_k) + \mathbf{Enc}_{\mathbf{y}}^{(1)}(\mathbf{y}_t) + \mathbf{Enc}_{\mathbf{s}}^{(1)}(\mathbf{s}_k) \\ \hat{\mathbf{z}}_k = \boldsymbol{C}_{\mathbf{z}}(\mathbf{x}_{\mathbf{k}}^{(\mathbf{1})}) \\ \mathbf{L}_z^{(1)} = -\sum_k \log\left(p_z(\mathbf{z}_k \mid \mathbf{x}_k^{(1)})\right) \end{cases} \tag{11}$$

where p_z represent the conditional likelihood of behavior $\mathbf{z}_k$ given behaviorally relevant latents in Stage 1.

Stage 2:

Next, we fit two modality-specific decoders to the latents $\mathbf{x}_k^{(1)}$ to predict the two neural modalities. The objective is to simultaneously minimize NLL of both neural modalities, ensuring accurate predictions across these diverse data types. Since BREM-NET accommodates neural modalities from different statistical distributions, the loss function must reflect these differences. To address this, we optimize a weighted combination of the NLLs for the two modalities, incorporating a scaling factor α to balance the disparity in scales between them (Eq. 12). This α factor ensures that both modalities contribute meaningfully to the optimization process, regardless of their inherent scale.

$$\begin{cases} \mathbf{x}_{k+1}^{(1)} = A^{(1)}(\mathbf{x}_k) + \mathbf{Enc}_{\mathbf{y}}^{(1)}(\mathbf{y}_t) + \mathbf{Enc}_{\mathbf{s}}^{(1)}(\mathbf{s}_k) \\ \hat{\tilde{\mathbf{y}}}_t = \boldsymbol{C}_{\mathbf{y}}^{(1)}(\mathbf{x}_t^{(1)}) \\ \hat{\tilde{\lambda}}_k = \boldsymbol{C}_{\mathbf{s}}^{(1)}(\mathbf{x}_k^{(1)}) \\ \mathbf{L}_{\mathbf{y},\mathbf{s}}^{(1)} = -\big(\sum_t \log\left(p_y(\mathbf{y}_t \mid \mathbf{x}_t^{(1)})\right) + \alpha \sum_k \log\left(p_s(\mathbf{s}_k \mid \mathbf{x}_k^{(1)})\right)\big) \end{cases} \tag{12}$$

Here, p_y and p_s represent the conditional likelihoods of the two neural modalities, $\mathbf{y}_k$ and $\mathbf{s}_k$, respectively. This stage ensures that $\mathbf{x}_k^{(1)}$ adequately explains the behaviorally relevant components of the neural data. Therefore, $\hat{\tilde{\mathbf{y}}}_k$, and $\hat{\tilde{\lambda}}_k$ correspond to the part of neural modalities $\mathbf{y}_k$, and $\mathbf{s}_k$, that are predictable with behaviorally relevant latents.

Stage 3:

In the final stage, we learn the second RNN, composed of $A^{(2)}(\cdot)$ and modality-specific encoders, $\mathbf{Enc}_{\mathbf{y}}^{(2)}(\cdot)$ and $\mathbf{Enc}_{\mathbf{s}}^{(2)}(\cdot)$ jointly with the modality-specific neural decoders $\boldsymbol{C}_{\mathbf{y}}^{(2)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{s}}^{(2)}(\cdot)$ to predict the residual neural dynamics of the two neural modalities that are not already explained by the behaviorally relevant neural latents $\mathbf{x}_k^{(1)}$. We represent the unpredicted part of multimodal neural activities with $\tilde{\mathbf{y}}_k$, $\tilde{\mathbf{s}}_k$, such that the full neural predictions are given by:

$$\begin{aligned} \hat{\mathbf{y}}_t &= \hat{\tilde{\mathbf{y}}}_t + \hat{\tilde{\mathbf{y}}}_t \\ \hat{\mathbf{s}}_k &\sim \mathcal{P}ois(\hat{\tilde{\lambda}}_k \times \hat{\tilde{\lambda}}_k) \end{aligned} \tag{13}$$

This final stage ensures that the residual neural dynamics (not explained by $\mathbf{x}_k^{(1)}$) are also captured using a separate set of latents. The optimization for this stage is defined as:

$$\begin{cases} \mathbf{x}_{k+1}^{(2)} = A^{(2)}(\mathbf{x}_k) + \mathbf{Enc}_{\mathbf{y}}^{(2)}(\mathbf{y}_t) + \mathbf{Enc}_{\mathbf{s}}^{(2)}(\mathbf{s}_k) \\ \hat{\hat{\mathbf{y}}}_t = \boldsymbol{C}_{\mathbf{y}}^{(2)}(\mathbf{x}_t^{(2)}) \\ \hat{\hat{\lambda}}_k = \boldsymbol{C}_{\mathbf{s}}^{(2)}(\mathbf{x}_k^{(2)}) \\ \mathbf{L}_{\mathbf{y},\mathbf{s}}^{(2)} = -\big(\sum_t \log\big(p_y(\mathbf{y}_t \mid \mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)})\big) + \alpha \sum_k \log\big(p_s(\mathbf{s}_k \mid \mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)})\big)\big) \end{cases} \tag{14}$$

A.1.3 Scaling factor α

To address the differences in scale between the likelihoods of two neural modalities, we use a scaling factor, α, in the loss function to ensure that both neural modalities contribute proportionally to the optimization process. To determine the value of α, we first compute the average value across all available time steps for each neural modality. This average serves as a baseline, representing the overall scale or magnitude of the observed data for that modality. Then, assuming the mean value as the predicted value for each modality, we calculate the log-likelihood of the observed data. This step quantifies the "scale" of each modality in terms of their likelihood under their respective distributions. Finally, the scaling factor α is determined as the ratio of the likelihoods of the modalities. Specifically, α is set such that it normalizes the contributions of the modalities, effectively equalizing their influence during the optimization process.

Sensitivity to α. Because α appears only in the Stage 2 and Stage 3 objectives, behavior decoding, which is performed from $\mathbf{x}^{(1)}$ learned exclusively in Stage 1, is invariant to α. We therefore assessed the sensitivity of neural prediction by sweeping α around the closed-form estimate in the NHP grid-reaching dataset (Table A.1). LFP prediction varied only marginally across the entire range, while spike prediction improved with increasing α and saturated near the value given by the mean-predictor estimate. Thus, the closed-form estimate provides a favorable balance between the two likelihood terms. To assess whether this estimate, computed before training, remains appropriate at convergence, we recomputed the likelihood-scale ratio from the trained model's Gaussian and Poisson negative log-likelihoods at every training epoch. The ratio remained within 0.01 of its initial closed-form estimate throughout training ($\alpha_{\text{conv}}/\alpha_{\text{init}} \approx 1$), indicating that the relative scales of the two likelihood terms remain essentially unchanged during optimization.

Table A.1: **Sensitivity of BREM-NET to the likelihood scaling factor** α in the NHP grid-reaching dataset. Behavior decoding is unaffected because α enters only the Stage 2 and Stage 3 objectives.

α	**Behavior decoding (CC)**	**LFP prediction (CC)**	**Spike prediction (PP)**
0.1	0.7645	0.80462	0.2009
1	0.7645	0.80458	0.2694
Mean-predictor estimate (ours)	0.7645	0.80432	0.3560
10	0.7645	0.80304	0.3584

A.1.4 Hyperparameters and Implementation

For BREM-NET, we employed consistent neural network architectures across both simulated and real-world evaluations. The specific hyperparameters employed for training on the simulation dataset, as well as analyses of both real-world datasets, are detailed in Table A.2. For encoders and decoders in our model i.e., $\mathbf{Enc}_{\mathbf{y}}^{(1)}(\cdot)$, $\mathbf{Enc}_{\mathbf{s}}^{(1)}(\cdot)$, $\mathbf{Enc}_{\mathbf{y}}^{(2)}(\cdot)$, $\mathbf{Enc}_{\mathbf{s}}^{(2)}(\cdot)$, $\boldsymbol{C}_{\mathbf{y}}^{(1)}(\cdot)$, $\boldsymbol{C}_{\mathbf{s}}^{(1)}(\cdot)$, $\boldsymbol{C}_{\mathbf{y}}^{(2)}(\cdot)$, $\boldsymbol{C}_{\mathbf{s}}^{(2)}(\cdot)$ we used multi-layer perceptrons (MLPs) with each having specific number of hidden layers and neurons (Table A.2). Otherwise, for our L-BREM-NET analysis, we remove the hidden layers and nonlinearity in feedforward neural networks. Notably, unlike the Gaussian decoders $\boldsymbol{C}_{\mathbf{y}}^{(1)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{y}}^{(2)}(\cdot)$, our Poisson decoders $\boldsymbol{C}_{\mathbf{s}}^{(1)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{s}}^{(2)}(\cdot)$ employ an $\exp(.)$ nonlinearity output layer activation function to force non-negativity of Poisson rates. We adopted a Step Decay learning rate scheduler (Ge

et al., 2019), which is known for its effectiveness in training deep neural networks. It was initialized with a learning rate of $1e-3$, which was halved ($\gamma = 0.5$) every 400 steps. To ensure stable training and convergence, we initialized all model parameters using the Xavier-normal initialization method (Glorot & Bengio, 2010). For optimization, we utilized the AdamW optimizer (Loshchilov & Hutter, 2019). We trained model for a maximum of 1000 epochs to ensure thorough learning, while employing early stopping to avoid overfitting. All models were trained on CPU servers equipped with AMD EPYC 7513 and 7542 processors (2.90 GHz, 32 cores) with parallelization. Our implementation is available at `https://anonymous.4open.science/r/BREM-NET-D9C9/`.

In simulation experiments, the state dimensions were aligned with those of the true underlying model. For real-world data analyses, we explored the impact of varying the state dimensions n_1 and n_x in one session of data. Specifically, we tested dimensions $n_1, n_x \in [2, 4, 8, 16, 32, 64, 128]$ and report the corresponding results in Figure A.7 for one dataset. After choosing the best n_1 and n_x based on one session of data, we kept it consistent for all models and baselines.

Table A.2: **Hyperparameter settings of BREM-NET.** Encoders and decoders architectures, as well as training configurations, used across all experiments and models.

Hyperparameter	Value
Encoders ($\mathbf{Enc_y}, \mathbf{Enc_s}$)	[64, 128]
Neural decoders ($\boldsymbol{C}_\mathbf{y}, \boldsymbol{C}_\mathbf{s}$)	[128, 128, 128]
Behavior decoder ($\boldsymbol{C}_z$)	[64]
Initial learning rate	$1e-3$
LR scheduler	StepLR
Batch size	32
Number of epochs	1000
Weight decay	$1e-3$

A.1.5 Model Complexity Analysis

In this section, we report the number of trainable parameters for each model. These counts are provided to document the complexity of BREM-NET compared to our baselines (Table A.3).

Table A.3: **Parameter counts for models.** Number of trainable parameters for each model.

Model	Number of parameters
MMGPVAE	103,578
mmPLRNN	353,328
CEBRA	697,344
BREM-NET	218,946

A.1.6 Evaluation Metrics

After learning BREM-NET, we utilize the learned nonlinear modality-specific encoders $\mathbf{Enc}_\mathbf{y}^{(1)}(\cdot)$, $\mathbf{Enc}_\mathbf{s}^{(1)}(\cdot)$, $\mathbf{Enc}_\mathbf{y}^{(2)}(\cdot)$, $\mathbf{Enc}_\mathbf{s}^{(2)}(\cdot)$, and recursion networks $A^{(1)}(\cdot)$, $A^{(2)}(\cdot)$ to infer 1-step-ahead predicted latent states using Equation 3 for the held-out test data in cross-validation. Using these inferred latent states, we compute the predicted neural activity and behavior by applying the corresponding decoders $\boldsymbol{C}_\mathbf{y}(\cdot)$, $\boldsymbol{C}_\mathbf{s}(\cdot)$, and $\boldsymbol{C}_\mathbf{z}(\cdot)$. For behavior predictions, we refer to this process as *decoding*, as the model exclusively utilizes neural modalities,$\mathbf{Y}$ and $\mathbf{S}$, and does not incorporate behavior itself as an input during inference.

To evaluate model performance, we use 5-fold cross-validation for both simulation and real data analyses. For LFP neural activity and continuous behavioral signals, which are modeled using a

Gaussian distribution, we report the Pearson Correlation Coefficient (CC) in all of our analysis (Figures 3, A.2, 5, A.1) between predicted and actual observations.

For spike count data, modeled with a Poisson distribution, we quantify one-step-ahead prediction accuracy using the Prediction Power (PP) metric, defined as $\mathbf{PP} = \mathbf{2\,AUC} - \mathbf{1}$, where $\mathbf{AUC}$ is the area under the curve of the receiver operating characteristic (ROC) curve (Macke et al., 2011). The ROC curve is constructed by using the one-step-ahead predicted firing rates ($\hat{\lambda}_{k+1}$) as classification scores to determine whether a time step contains a spike (Truccolo et al., 2010).

A.2 BASELINES

To evaluate the performance of BREM-NET and highlight its contributions, we compare it against a diverse set multimodal baseline models.

For each baseline, we performed hyperparameter sweeps including latent dimensionality. We observed that increasing latent dimensionality did not improve performance for some baselines and in some cases led to overfitting. Therefore, the reported results reflect tuned baseline configurations, aiming to provide a fair comparison. Also, all the model variants and baselines are evaluated using the same cross-validation protocol and number of channels per modality.

First, to assess the role of nonlinear modeling in BREM-NET, we compare it against two fully linear dynamical methods: MSID (Ahmadipour et al., 2024) and a fully linear version of BREM-NET, denoted L-BREM-NET, in which all nonlinear components are removed. Second, to highlight the significance of using multimodal neural modalities on the neural-behavioral dynamics, we include single-modal variants of our model trained with either LFP or spike data alone. These baselines help highlight the benefit of combining neural modalities when modeling behaviorally relevant dynamics. Third, to evaluate the importance of behavior supervision and structured learning, we include an unsupervised variant of our model, denoted U-BREM-NET, in which behavior is not used during training. In addition, we include a variant, denoted BREM-NET w/o Stage 3, in which Stage 3 is omitted, and a variant, denoted BREM-NET Joint, in which all latents are trained jointly without structured learning. We further compare BREM-NET against several recent works developed for neural-behavioral modeling, including: mmPLRNN (Kramer et al., 2022), MMGPVAE (Gondur et al., 2024), and CEBRA (Schneider et al., 2023). Finally, we include MVAE (Wu & Goodman, 2018), a generic multimodal variational autoencoder baseline that aligns latent spaces across modalities without modeling dynamics and not developed for neural-behavioral modeling.

A.2.1 L-BREM-NET

L-BREM-NET serves as a simplified linear baseline. It preserves the same multi-stage architecture and overall structure of BREM-NET but replaces all encoders and decoders with linear transformations. This baseline uses identical loss functions and optimization procedures to ensure fair comparison. Nonlinear activations and hidden layers are removed, making the model purely linear while maintaining the core structure of our approach.

A.2.2 U-BREM-NET

U-BREM-NET is an unsupervised baseline that performs only the third stage of the BREM-NET learning procedure (see Appendix A.1.2). Unlike BREM-NET, U-BREM-NET learns the neural dynamics without any supervision or consideration of their relevance to behavior. Behavior information is not utilized during the learning process, and the extracted latent states are subsequently mapped to behavior data through a downstream decoder. Effectively, U-BREM-NET represents a special case of BREM-NET where $n_1 = 0$, meaning that no behaviorally relevant latent states are separated during training.

A.2.3 BREM-NET W/O STAGE 3

This ablated variant of BREM-NET includes the first two stages of BREM-NET where Stage 3 is omitted. Specifically, the model learns the behavior-relevant latent states $x^{(1)}$ with $n_1 = 16$ and behavior decoder $\boldsymbol{C}_z$ from multimodal neural input in stage 1, and fits neural decoders ($\boldsymbol{C}_y$ and $\boldsymbol{C}_s$) in stage 2 which is used for prediction of multimodal neural activities. The residual neural

latent states $x^{(2)}$, introduced in Stage 3 of the full model, are removed in this variant. Consequently, the model lacks a dedicated mechanism for modeling the residual neural dynamics that $x^{(1)}$ leaves unexplained.

A.2.4 BREM-NET JOINT

BREM-NET Joint is an ablated variant that removes the structured learning procedure of BREM-NET (see Appendix A.1.2). Instead of learning the behaviorally relevant and residual neural latents in separate stages, it trains all model components jointly in a single stage by minimizing a combined loss for behavior decoding and neural prediction. The latent space is therefore not partitioned into behaviorally relevant and residual neural latents: all n_x latent dimensions are learned jointly, and the full latent state $\mathbf{x}_k$ is used both to decode behavior and to predict the multimodal neural activity. The combined loss is

$$\mathbf{L}_{\text{joint}} = \lambda \sum_k \text{NLL}_{\mathbf{z}}\big(\mathbf{z}_k; \boldsymbol{C}_{\mathbf{z}}(\mathbf{x}_k)\big) + \sum_t \text{NLL}_{\mathbf{y}}\big(\mathbf{y}_t; \boldsymbol{C}_{\mathbf{y}}(\mathbf{x}_t)\big) + \alpha \sum_k \text{NLL}_{\mathbf{s}}\big(\mathbf{s}_k; \boldsymbol{C}_{\mathbf{s}}(\mathbf{x}_k)\big), \quad (15)$$

where λ weights the behavior loss relative to the neural losses and α is the scaling factor between the two neural modalities (Appendix A.1.3). We use $\lambda = 1$ for the results reported in Fig. 5. To assess whether the comparison with BREM-NET depends on this choice, we additionally swept λ over a 1000-fold range; across this range, BREM-NET Joint did not outperform BREM-NET in behavior decoding or neural prediction, so the conclusions of the ablation study are not sensitive to the choice of λ. Accordingly, the behavior decoding reported for BREM-NET Joint (Fig. 5d) is obtained from its full latent space. Comparing BREM-NET with BREM-NET Joint thus assesses the effect of the structured learning procedure on how behaviorally relevant information is organized in the latent space.

A.2.5 SINGLE-MODAL MODELS

For single-modal models, we adopt a simplified version of the architecture shown in Figure 1, where the encoder and decoder components corresponding to the additional neural modality are removed. In this configuration, the model focuses solely on one modality of neural data, such as LFPs or spike counts and learns behaviorally relevant dynamics from that single modality and thus does not fuse multimodal neural during inference. The first stage of the model is dedicated to modeling the latent structure of the chosen modality, capturing the most informative features that are behaviorally relevant. Subsequently, these learned representations are used to predict the associated behavioral signals, under the assumption that the chosen modality alone contains sufficient information to account for the observed behavior. This single-modality setup serves as a baseline comparison to our multimodal model, enabling an evaluation of the individual contribution of each neural modality. By isolating the dynamics of a single modality, we are able to assess its relative effectiveness in capturing behavior-related information and gain insight into the strengths and limitations of using single neural modality for behavior decoding.

A.2.6 MSID

Multiscale Subspace Identification (MSID) is a linear model designed for analyzing multiscale dynamical systems in neural activity, operating under the assumption of linear dynamics (Ahmadipour et al., 2024). In our analysis, we compared the performance of BREM-NET against MSID and demonstrated that, unlike the linear approach of MSID, the nonlinear information aggregation and behavior supervision property of BREM-NET significantly enhances both behavior decoding and neural prediction. We used the official implementation of MSID provided by the authors for training [1]. For training MSID, we set the horizon hyperparameters as specified in the original manuscript (Ahmadipour et al., 2024), specifically $h_y = h_z = 10$.

A.2.7 MVAE

Multimodal Variational Autoencoder (MVAE) (Wu & Goodman, 2018) is a VAE-based architecture designed to model multimodal data using a mixture-of-experts formulation over the posterior distributions. MVAE is trained to align latent representations across modalities. To adapt MVAE for our use

[1] We use the implementation provided in https://github.com/ShanechiLab/MultiscaleSID

case, we treated each time-step as an independent data point and trained the model in a non-sequential manner, effectively bypassing the lack of a dynamical modeling backbone. Behavior decoding was performed by fitting linear regression models on the inferred latent representations. Since MVAE does not model temporal dependencies, it is not capable of performing neural time-series prediction, which requires explicit dynamical modeling.

A.2.8 CEBRA

CEBRA (Schneider et al., 2023) is a recent method designed to extract latent embeddings from neural recordings, optionally guided by behavioral information. It employs a 1D convolutional neural network to extract embeddings from small windows of neural data, and optimized through a contrastive objective. In the supervised variant (CEBRA-Behavior), behavior labels are used to define positive and negative sample pairs, encouraging the model to learn embeddings that are discriminative with respect to behavior and thus behaviorally relevant. In the unsupervised version, positive samples are determined based on time-difference to the anchor sequence.

Unlike RNN-based methods, CEBRA models the temporal dynamics of neural activity through 1D convolutional layers. Further, CEBRA architecture do not include decoder networks but it only consists of encoder networks to extract latent states from input signals. To enable performance comparisons on behavior decoding and neural prediction tasks, we therefore fit separate linear regression models on top of the CEBRA embeddings to decode neural and behavior signals. For a fair comparison, we use the multisession version of CEBRA that jointly processes both continuous LFP and discrete spike data, which generates consistent embeddings across the two recording modalities. This allows CEBRA to operate as a multimodal model in our experiments. We follow the original implementation and adopt the default hyperparameters provided in the original work for CEBRA[2].

A.2.9 MMPLRNN

Multi-modal piecewise-linear RNN (mmPLRNN) is a variational method previously introduced for multimodal dynamical modeling with piecewise-linear RNNs (Kramer et al., 2022). mmPLRNN builds on a prior work, PLRNN (Durstewitz, 2017), by fusing information from two modalities. By design, mmPLRNN utilizes two modalities during inference. To enable a fair comparison with BREM-NET, which does not incorporate behavior during inference, we trained two versions of mmPLRNN. In the first version, mmPLRNN was trained in an unsupervised manner, using only neural modalities (e.g., LFP and spiking signals) as input during training and inference. For the second version, we extended mmPLRNN to a supervised framework. Specifically, we introduced an additional behavior prediction objective during training. This was achieved by learning a feedforward neural network that maps the inferred latent factors from mmPLRNN to the behavior signals. The behavior reconstruction loss was added to the mmPLRNN's training objective to encourage the latent space to capture behaviorally relevant dynamics. Importantly, the behavior signal was not treated as an input to the model. Instead, the goal was to infer latent factors from neural modalities (LFP and spikes) that could accurately reconstruct the behavior. However, unlike BREM-NET, this extended mmPLRNN approach does not separate behaviorally relevant and residual neural latents in latent space. Instead, it only encourages the learned latent space to be more behaviorally informative overall, without explicit separation.

Furthermore, mmPLRNN does not support predicted inference of latent factors. Thus, to provide a fair comparison between all baseline methods, we modified mmPLRNN's inference algorithm to infer one-step-ahead predicted latent factors. For each timestep $\mathbf{k}$, we passed the neural modalities up to timestep $\mathbf{k-1}$ to the inference network, and forward propagated the latent states at $\mathbf{k-1}$ through PLRNN dynamics to obtain one-step-ahead predicted latent states for $\mathbf{k}$. After this procedure is applied for all time horizon, we then obtained one-step-ahead predicted neural modalities by passing the predicted latent factors through decoder networks. Further, the original mmPLRNN implementation supports only Gaussian and categorical observation models. To accommodate our experimental setup, we implemented a Poisson observation model following the methodology outlined in Appendix C of (Kramer et al., 2022). Furthermore, we trained mmPLRNN models with nonlinear readouts comparable to BREM-NET and evaluated their neural-behavioral prediction performance.

[2]We use the implementation provided in https://github.com/AdaptiveMotorControlLab/CEBRA

We used the official implementation and recommended hyperparameters of mmPLRNN for the aforementioned extensions and data analyses[3].

A.2.10 MMGPVAE

The Multimodal Gaussian Process Variational Autoencoder (MMGPVAE) is a recent multimodal framework that leverages Gaussian processes to model the latent distribution underlying multimodal observations (Gondur et al., 2024). MMGPVAE's inference network first extracts the frequency content of the latent factors and subsequently converts them into time-domain representations, rather than directly estimating them in the time domain. This approach allows MMGPVAE to filter out high-frequency components from the latent factors, resulting in smoother representations. As noted in the authors' manuscript, the choice of a Gaussian Process (GP) prior is particularly advantageous in experimental settings where the latent dynamics are assumed to be smooth (Gondur et al., 2024).

Similar to mmPLRNN, MMGPVAE utilizes two modalities during inference. Thus, for the comparisons shown in Figs. 4 and A.2, we followed the same procedure as for mmPLRNN-unsupervised and supervised variations. To ensure consistency with BREM-NET, we trained MMGPVAE with 64-dimensional latent factors per modality, where 32 out of the 64 dimensions were shared across modalities. All MMGPVAE variations were trained for 100 epochs. We also modified the encoder/decoder architecture, as the default configuration led to suboptimal performance on our dataset. The new configuration is the same as what we use in BREM-NET (refer to Table A.2). Similar to mmPLRNN, MMGPVAE also does not support predicted inference of latent states. To infer one-step-ahead predicted latent states, for each timestep $\mathbf{k}$, we replaced the neural modalities after timestep $\mathbf{k-1}$ with zeros such that the latent states at $\mathbf{k}$ is inferred only by neural observations up to timestep $\mathbf{k-1}$ (i.e., one-step-ahead predicted). Then, we passed these neural signals to MMGPVAE's inference network, and we obtained one-step-ahead predicted latent states. After this procedure is applied for all time horizon, we then obtained one-step-ahead predicted neural modalities by passing the predicted latent states through decoder networks. Furthermore, we extended MMGPVAE to a supervised setting similar to the supervised extension of mmPLRNN, see Appendix A.2.9 for details. We used the implementation of MMGPVAE provided by the authors for our analysis [4].

A.3 Additional information on Real Datasets

A.3.1 Nonhuman Primate (NHP) Grid Reaching Dataset

We analyzed a publicly available dataset (O'Doherty et al., 2020) in which a macaque monkey (Monkey I), performed a grid reaching task. Neural activity was recorded from the primary motor cortex (M1) using a 96-channel electrode array as the subject controlled a 2D cursor to reach randomly appearing targets on a grid within a virtual reality environment. Targets were presented sequentially, with no intervening time gaps between their appearances. In our analysis, we took the subject's 2D fingertip velocity recorded at a 10 ms timescale as the behavior time-series to decode.

For the spiking data, we utilized multi-unit spiking activity recorded at a 10 ms timescale. From the four recording sessions, we selected the top 20 spiking channels based on their behavior prediction accuracies. For the local field potential (LFP) data, signals were extracted from the raw neural recordings using a low-pass filter with a 300 Hz cut-off frequency. The filtered signals were downsampled to 100 Hz (10 ms timescale). Similar to the spiking data, we identified the top 20 LFP channels across the four sessions based on their behavior prediction accuracies. We report the mean and SEM computed across four sessions and five cross-validated folds (N=20).

A.3.2 NHP Center-Out Reaching Dataset

In this publicly available dataset (Flint et al., 2012), a macaque monkey (Monkey C) performed a 2D center-out reaching task while grasping a two-link manipulandum. The task involved reaching from a central position to one of eight outer targets arranged in a circle, followed by returning to the center to begin the next trial. In our analysis, we used the 2D manipulandum velocity recorded at a 10 ms timescale as the behavior variable to decode. Neural recordings were obtained from a 96-channel

[3]We use the implementation provided in https://github.com/DurstewitzLab/mmPLRNN

[4]We use the implementation provided in https://github.com/RabiaGondur/MM-GPVAE

silicon microelectrode array (Blackrock Microsystems) chronically implanted in the primary motor (M1) and premotor (PMd) cortices of the monkey's arm.

For the spiking data, we utilized multi-unit spiking activity recorded at a 10 ms timescale. From the three recording sessions, we selected the top 20 spiking channels based on their behavior prediction accuracies. For the local field potential (LFP) data, raw signals were band-pass filtered between 0.5 and 500 Hz and sampled at 2 kHz. LFP power was computed across five frequency bands: 0-4 Hz, 7-20 Hz, 70-115 Hz, 130-200 Hz, and 200-300 Hz, using a 256 ms window with 10 ms resolution, yielding signals at 100 Hz. Similar to the spiking data, we identified the top 20 LFP channels across the three sessions based on their behavior prediction accuracies. We report the mean and SEM computed across three sessions and five cross-validated folds (N=15).

A.4 Simulation Details

For our simulation data in section A.5.2, we generated Gaussian and Poisson observations from four randomly initialized systems as defined by the dynamics in

$$\begin{aligned} \mathbf{x}_{k+1} &= F(\mathbf{x}_k) + \mathbf{w}_k \\ \mathbf{y}_k \mid \mathbf{x}_k &\sim \mathcal{N}\big(\mathbf{y}_k \mid \boldsymbol{C}_{\mathbf{y}}(\mathbf{x}_k)\big) \\ \mathbf{s}_k \mid \mathbf{x}_k &\sim \mathcal{P}ois\big(\mathbf{s}_k \mid \boldsymbol{C}_{\mathbf{s}}(\mathbf{x}_k)\big) \\ \mathbf{z}_k \mid \mathbf{x}_k &\sim \mathcal{N}\big(\mathbf{z}_k \mid \boldsymbol{C}_{\mathbf{z}}(\mathbf{x}_k)\big) \end{aligned} \tag{16}$$

In this setup, the latent state evolves in time according to a recursion function $F(\cdot)$ with additive noise $\mathbf{w}_k$. $\mathbf{y}_k$ represents Gaussian neural activity, $\mathbf{s}_k$ corresponds to spiking neural activity, and $\mathbf{z}_k$ denotes behavior. A total of eight latent variables ($n_x = 8$) were drawn from the system, where $n_1 = 4$ latent dimensions were allocated to the behaviorally relevant subspace and the remaining $n_2 = n_x - n_1 = 4$ to the behaviorally independent subspace. These latent variables were then mapped to their corresponding observations using nonlinear functions $\boldsymbol{C}_{\mathbf{y}}(\cdot)$, $\boldsymbol{C}_{\mathbf{s}}(\cdot)$, and $\boldsymbol{C}_{\mathbf{z}}(\cdot)$. For each randomly initialized system, we generated a total of $2e6$ samples. These simulation settings demonstrate the benefits of fusing multimodal information from two neural modalities while separating the behaviorally relevant neural latents and residual neural latents, leading to improved predictions of both neural and behavior time-series.

A.5 Supplementary Results

A.5.1 NHP Center-out Reaching Task Results

A.5.2 Our method fuses multimodal data, thus better predicting all modalities in simulations

We first validated the ability of BREM-NET to effectively fuse information across multiple modalities through a series of simulation experiments. These simulations involved three modalities with shared and independent dynamics. The underlying dynamical system used to generate the synthetic time-series for these modalities is defined in Equation 16. We set the total simulated neural latent dimensionality to $n_x = 8$, with four dimensions allocated to behaviorally relevant latents ($n_1 = 4$) and the remaining four to residual neural latents ($n_2 = 4$), which in this simulation are by construction independent of behavior.

We generated observations from both Gaussian and Poisson distributions using 100 channels each and compared BREM-NET to single-modal models that only used either Gaussian or Poisson observations alongside behavior (Appendix A.2.5). Specifically, in Figure A.4a, we chose a primary neural modality, Poisson in the left panel and Gaussian on the right panel. We then fused this primary modality with gradually increasing numbers of channels of the other secondary modality, as shown on the x axis. As such, the points corresponding to 0 on the x axis indicate the single-modal performance and all other points show BREM-NET that fuses different numbers of channels of the two different modalities.

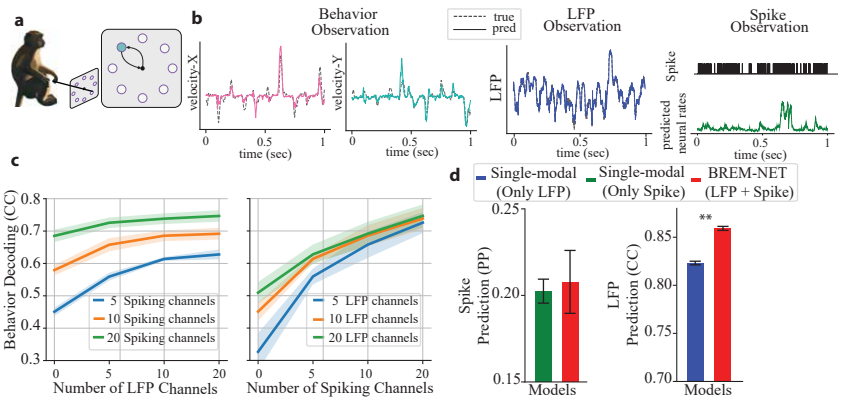

Figure A.1: **BREM-NET again improves neural-behavioral predictions due to multimodal neural fusion in another NHP dataset with a center-out reaching task. (a)** Dataset and task visualization. **(b)** True behavior, LFP, and spike time-series (dashed) versus their predictions with BREM-NET (solid) for a representative session. **(c)** *left*: Cross-validated behavior decoding CC when 5, 10, or 20 spiking channels were the primary modality and an increasing number of LFP channels were fused with them (N=15). Lines represent the mean and shaded areas show the SEM. *right*: Similar to *left* when LFP channels were the primary modality. **(d)** *left*: Cross-validated Predicted Power (PP) of firing rates using BREM-NET vs. when training a single-modal model on the spiking modality alone (N=15). Bars represent the mean and error bars represent the SEM. *right*: Cross-validated Prediction CC of LFP modality with BREM-NET vs. when training a single-modal model with the LFP modality alone. Asterisks indicate significance of comparison (**: $p < 0.001$, one-sided Wilcoxon signed-rank test)

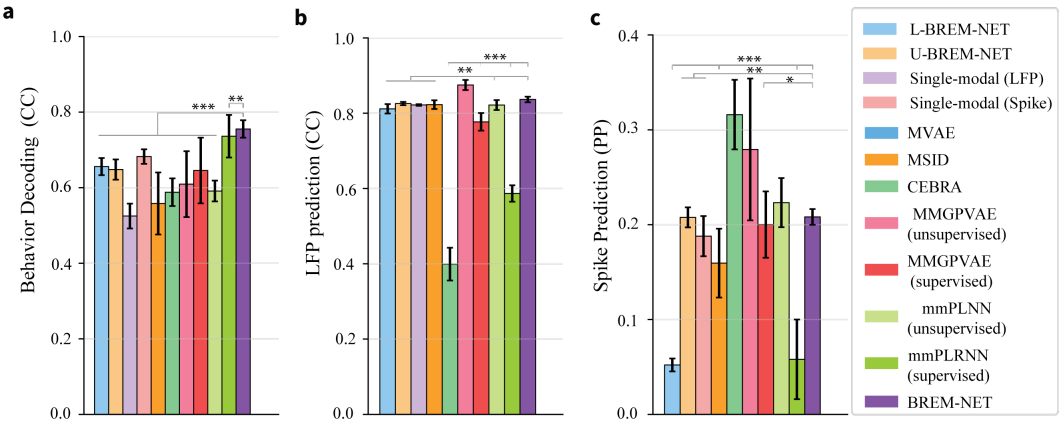

Figure A.2: **Results on a different center-out reaching dataset.** Prediction performance of BREM-NET compared to baselines in NHP center-out reaching dataset. For all multimodal models, 20 spiking channels as well as 20 LFP channels were used. For single-modal baselines, 20 channels of the chosen modality were used. Mean ± SEM is across all sessions and 5 cross-validation folds (N = 15).

We found that BREM-NET demonstrated superior performance in behavior decoding compared to single-modal models, as shown in Figure A.4a. In scenarios where the Poisson modality served as the primary modality, incorporating Gaussian channels consistently improved behavior decoding accuracy (Figure A.4a, left). Similar trend was observed when the Gaussian modality was the primary modality and Poisson channels were fused (Figure A.4a, right). Further, incorporating information

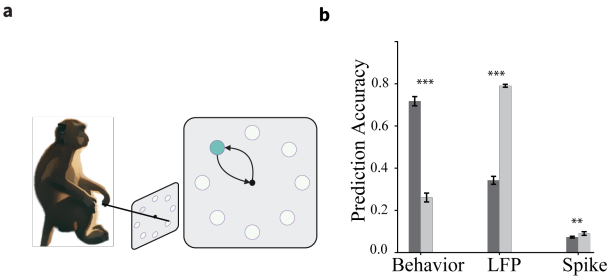

Figure A.3: **Separation of behaviorally relevant and residual neural dynamics through structured multi-stage learning.** Evaluation of the separation of the learned latent states by using $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ separately to predict behavior and neural modalities. **(a)** NHP center-out reaching dataset. **(b)** Behavior decoding is substantially more accurate when using the behaviorally relevant latents $\mathbf{x}^{(1)}$ compared to the residual neural latents $\mathbf{x}^{(2)}$. Conversely, neural prediction accuracy improves when using $\mathbf{x}^{(2)}$, demonstrating its specialization in capturing residual neural variability. Together, these results indicate a functional specialization: $\mathbf{x}^{(1)}$ carries the behaviorally relevant dynamics learned in Stage 1, while $\mathbf{x}^{(2)}$ captures the residual neural dynamics learned in Stage 3 that improve neural prediction.

from one neural modality enhanced the prediction accuracy of the other modality, showing the power of cross-modality information fusion (Figure A.4b).

Consistently, BREM-NET successfully predicted both neural time-series modalities and behavioral time-series from the learned latent states (Figure A.4c).

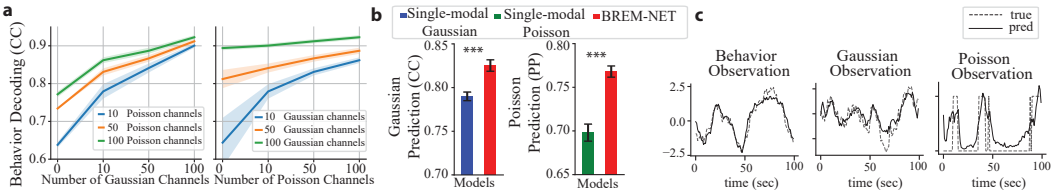

Figure A.4: **BREM-NET improves behavior decoding and multimodal neural predictions in a simulated dataset.** Cross-validated accuracy of behavior decoding and Gaussian and Poisson neural prediction in four simulated systems (N=20). **(a)** *left*: behavior decoding correlation coefficient (CC) when 10, 50, or 100 Poisson channels were the primary modality and an increasing number of Gaussian channels were fused with them. Lines represent the mean and shaded areas show the standard error of the mean (SEM). *right*: Similar to *left* when Gaussian channels were the primary modality. **(b)** *left*: Prediction CC of Gaussian modality using BREM-NET vs. when training a single-modal model with Gaussian modality alone. Bars represent the mean and error bars show the SEM. *right*: Predicted Power (PP) of the Poisson modality using BREM-NET vs. when training a single-modal model on the Poisson modality alone. Asterisks indicate significance of comparison (***: $p < 10^{-7}$ one-sided Wilcoxon signed-rank test)**(c)** True versus predicted time-series using BREM-NET, showing the model's ability to accurately predict all observation modalities.

A.5.3 Effect of using all recording channels

We performed an additional analysis using all 96 spiking and 96 LFP channels and obtained a behavior decoding (CC) of 0.7711 ± 0.0179, slightly higher than 0.7645 ± 0.0052 with the top 20 channels (Fig. 3c). The small improvement indicates that including low-SNR or noisy channels does not significantly enhance decoding performance, as these channels contain limited behaviorally relevant

information. This justifies our choice to focus on the top predictive channels for improved efficiency with minimal performance trade-off. Nevertheless, BREM-NET still has benefits over single-modal variants without any preselection and using all channels as shown below:

Table A.4: **BREM-NET still has benefits over single-modal variants without any preselection and using all channels.** Prediction performance of BREM-NET compared to single-modal variants in NHP grid reaching dataset. For BREM-NET, all 96 spiking channels as well as 96 LFP channels were used. For single-modal variants, 96 channels of the chosen modality were used. Mean ± SEM is across all sessions and 5 cross-validation folds (N = 20).

Models	**Behavior Decoding (CC)**
Single-modal (LFP)	0.5752 ± 0.0325
Single-modal (Spike)	0.7575 ± 0.0189
BREM-NET (multimodal)	$\mathbf{0.7711 \pm 0.0179}$

A.5.4 Robustness to missing data, temporal resolution mismatch, and dropout

In practical neural recording scenarios, different modalities—such as LFPs and spikes—are often sampled at different temporal resolutions due to hardware limitations or preprocessing constraints. Additionally, missing data are common in real-world datasets due to noise, sensor dropout, or transmission errors. Although the datasets used in our experiments are fully observed, BREM-NET is explicitly designed to handle missing data and asynchronous sampling as explained in Methods section 3. This is achieved through modality-specific encoder networks, which flexibly incorporate available modalities at each time step. If a modality is missing, its encoder output can be set to zero, allowing the latent dynamics to update using only the available inputs.

To empirically test robustness to mismatched sampling rates, we downsampled the LFP signals by a factor of 5 in the NHP grid-reaching dataset, while leaving the spike and behavior data unchanged. This setup mimics real-world conditions where continuous signals are collected at lower frequencies than discrete spike trains, or where LFP observations may be intermittently unavailable. As shown in Table A.7, behavior decoding and spike prediction performance remained largely stable, whereas LFP prediction degraded due to masked predictions on downsampled timesteps. This confirms that BREM-NET is resilient to both temporal resolution mismatches and partial observability, effectively integrating modalities sampled at different rates without requiring strict alignment.

To isolate the benefit of performing masked latent-state inference directly from asynchronous observations, we kept the BREM-NET architecture fixed and changed only how the missing samples were handled (Table A.5). For the continuous LFP modality under $5\times$ downsampling, we replaced masking with interpolation of the missing LFP samples. For the discrete spiking modality under $5\times$ downsampling, interpolation is not naturally defined for count data, so we replaced masking with zero-imputation. In both cases, masking yielded higher behavior decoding. We further compared against MMGPVAE (supervised; Appendix A.2.10) under the same $5\times$ LFP downsampling (Table A.6). Because MMGPVAE requires fully sampled inputs, the missing LFP samples were first interpolated. Under this setting, MMGPVAE's behavior decoding decreased from 0.6201 to 0.5303, whereas BREM-NET decreased only from 0.7645 to 0.7571. These results indicate that the robustness of BREM-NET is not explained by redundancy in the data alone. Moreover, unlike interpolation, which applies only to continuous modalities, masking handles both continuous and discrete modalities, as well as irregular sampling and modality-specific dropouts, without imputing or modifying the observations.

In addition to resolution mismatches, we also evaluated robustness under irregular and stochastic missing data. At each timestep, either spikes or LFPs were randomly dropped with probability $p \in \{0.2, 0.4, 0.6, 0.8\}$, creating asynchronous gaps where only one modality was available. Unlike the resolution mismatch experiment, where missing samples occur regularly, this dropout analysis simulates challenges such as channel noise, transient artifacts, or partial recording failures that occur irregularly. As shown in Table A.8, decoding performance remained remarkably stable across dropout probabilities, particularly for LFPs. Even when 80% of LFP samples were removed, behavior decoding declined by less than 2%. Spike dropouts produced a larger impact, consistent with

Table A.5: **Masking outperforms imputation-based handling of asynchronous observations.** Behavior decoding (CC) of BREM-NET with the architecture fixed and only the missing-data handling changed, when LFPs or spikes are downsampled by a factor of 5 in the NHP grid-reaching dataset. Interpolation is not naturally defined for spike counts, so zero-imputation is used as the alternative for the spiking modality.

Missing-data handling	**5× LFP downsampling**	**5× spike downsampling**
Masking (BREM-NET)	$\mathbf{0.7571 \pm 0.0051}$	$\mathbf{0.6475 \pm 0.0052}$
Interpolation	0.7342 ± 0.0021	–
Zero-imputation	–	0.6110 ± 0.0024

Table A.6: **BREM-NET is more robust to asynchronous sampling than an imputation-based multimodal baseline.** Behavior decoding (CC) in the NHP grid-reaching dataset with same-resolution inputs and with LFPs downsampled by a factor of 5. MMGPVAE requires fully sampled inputs, so the missing LFP samples were interpolated; BREM-NET masks them.

Model	**Same resolution**	**5× LFP downsampling**
MMGPVAE (supervised, interpolation)	0.6201	0.5303
BREM-NET (masking)	**0.7645**	**0.7571**

spikes carrying richer behavior-related information in this dataset (Fig. 3c). Still, the performance degradation remained modest, with about a 5% drop at 40% spike dropout. Together, these analyses demonstrate that BREM-NET flexibly integrates multimodal inputs under asynchronous sampling and partial observability, maintaining robust decoding accuracy under realistic and adverse recording conditions.

Table A.7: **BREM-NET is robust to temporal resolution mismatch and missing observations.** We compare BREM-NET trained under two conditions: (1) **same-resolution**, where both LFP and spike data are aligned and sampled at the same rate, and (2) **different-resolution**, where LFP signals are downsampled by a factor of 5 to simulate asynchronous sampling. These results demonstrate that BREM-NET can effectively integrate multimodal signals under realistic constraints such as sampling mismatch and missing observations without compromising decoding performance.

Models	**Behavior Decoding**	**Neural prediction**	
	(CC)	**LFP (CC)**	**Spike (PP)**
BREM-NET w/ different-resolution	0.7571 ± 0.0051	0.6272 ± 0.0234	0.3701 ± 0.0031
BREM-NET w/ same-resolution	0.7645 ± 0.0052	0.8078 ± 0.0125	0.3746 ± 0.0036

A.5.5 Latent Trajectory Plotting Details

To visualize the learned latent states, we plotted the condition-averaged latent trajectories in grid reaching dataset (O'Doherty et al., 2020). Trials were grouped into 8 discrete categories based on the direction of the reach movement, corresponding to the target directions in the task. For each condition, we computed the average latent trajectory across all trials within that condition. To enable 2D visualization, we applied Principal Component Analysis (PCA) on the full set of inferred $\mathbf{x}_k^{(1)}$ latents across all trials and retained the first two principal components. These top two components were then used to project the condition-averaged latent trajectories into a 2D plane. Each trajectory was plotted using a unique color corresponding to the reach direction condition. For comparison,

Table A.8: **BREM-NET is robust to dropouts.** We evaluated decoding performance when either spikes or LFPs were randomly dropped at each timestep with probability $p \in \{0.2, 0.4, 0.6, 0.8\}$. BREM-NET remained highly resilient, particularly to LFP dropouts. Spike dropouts produced a larger effect, consistent with their primary role in behavior decoding. These results demonstrate that BREM-NET flexibly integrates asynchronous and incomplete modalities while maintaining robust behavior decoding performance.

Dropout probability	**Behavior Decoding (CC)**	
	LFP dropout	**Spike dropout**
0.0	0.7645 ± 0.0052	0.7645 ± 0.0052
0.2	0.7613 ± 0.0061	0.7543 ± 0.0042
0.4	0.7601 ± 0.0071	0.7245 ± 0.0063
0.6	0.7573 ± 0.0067	0.6993 ± 0.0072
0.8	0.7516 ± 0.007	0.6292 ± 0.0065

we also plotted the condition-averaged true behavior trajectories (e.g., 2D hand velocity) using the same procedure, aligning and averaging behavioral traces over trials per direction. This provides an intuitive comparison of whether the latent dynamics learned by the model align with true task structure.

A.5.6 Consistency of Behaviorally Relevant Latent Representations Across Sessions

To assess the consistency of learned behaviorally-relevant neural latent representations across sessions, we also extended BREM-NET to a multi-session setting. Specifically, we introduced session-specific linear projection heads to account for inter-session variability, while keeping the core model architecture shared across sessions. Our hypothesis was that if a unified model—with the same latent dimensionality as the single-session variant—could accurately decode behavior across sessions, it would indicate that a consistent latent structure is recoverable across sessions. To test this, we pooled data from multiple sessions and trained a model with the same number of latent states used in single-session experiments. Despite increased data heterogeneity, the multi-session model achieved comparable behavior decoding accuracy (0.7648 vs. 0.7645 CC), and its latent trajectories were congruent with the true behavior and smoother than those from the single-session model (Fig. A.5). This result suggests the consistency of behaviorally-relevant neural latent representations across sessions and the ability of our model to recover these consistent latents.

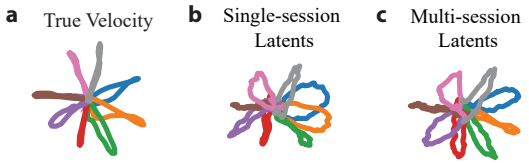

Figure A.5: **Multi-session fitting extracts consistent latents across sessions.** Visualization of behaviorally relevant latent trajectories under single-session and multi-session fitting. Latents from the multi-session model are smoother and closely track true velocity.

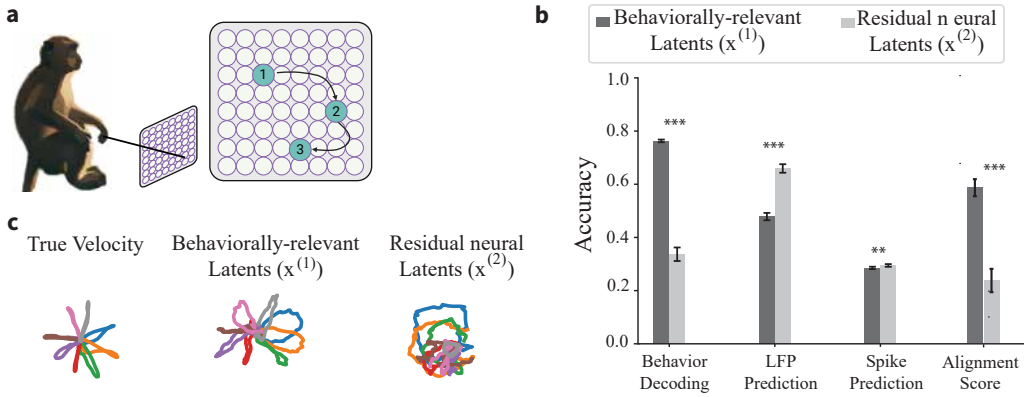

Figure A.6: **Separation of behaviorally relevant and residual neural dynamics through structured multi-stage learning.** Evaluation of the separation of the learned latent states by using $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ separately to predict behavior and neural modalities. **(a)** NHP grid-reaching dataset. **(b)** Behavior decoding and alignment score are substantially higher when using the behaviorally relevant latents $\mathbf{x}^{(1)}$ compared to the residual neural latents $\mathbf{x}^{(2)}$. Conversely, neural prediction accuracy improves when using $\mathbf{x}^{(2)}$, demonstrating its specialization in capturing residual neural variability. Together, these results indicate a functional specialization: $\mathbf{x}^{(1)}$ carries the behaviorally relevant dynamics learned in Stage 1, while $\mathbf{x}^{(2)}$ captures the residual neural dynamics learned in Stage 3 that improve neural prediction. **(c)** True velocity trajectories are much more aligned with behaviorally relevant neural latent trajectories than with residual neural latent trajectories.

A.5.7 LATENT DIMENSIONALITY SELECTION

To select the latent dimensionalities for each real-world dataset, we evaluated model performance on one session from each dataset across a range of dimensionalities. Specifically, we varied the number of behaviorally relevant latent dimensions n_1 and the total latent dimensions n_x in $[2, 4, 8, 16, 32, 64, 128]$, and assessed their impact on behavior decoding and neural prediction accuracy. In BREM-NET, Stage 1 is designed to extract behaviorally relevant latent states, which are used exclusively for behavior decoding. Therefore, we selected n_1 based on the behavior decoding performance. As shown in Figure A.7b,f, performance peaks at $n_1 = 16$ for both datasets, which we selected as the behaviorally relevant latent dimensionality. This choice identifies the smallest n_1 at which behavior decoding saturates for this architecture and training objective; as in related work, it is a model-selection criterion rather than an estimate of the intrinsic dimensionality of behaviorally relevant neural activity.

Stage 3 is responsible for modeling residual neural dynamics to improve neural predictions beyond what is captured by behaviorally relevant latents. Thus, we selected n_x based on neural prediction performance. As shown in Figure A.7c,d,g,h, neural prediction performance peaks around $n_x = 64$ for the first dataset and around $n_x = 32$ for the second dataset, which we used as the total latent dimensionalities for subsequent analyses.

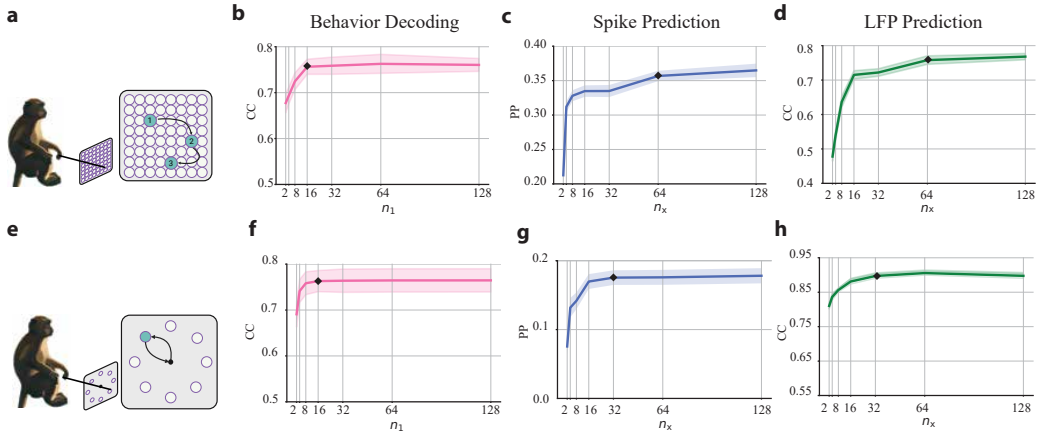

Figure A.7: **Selecting latent dimensions based on behavior decoding and neural prediction performance in one session of datasets.** This figure illustrates how behavior decoding and neural prediction accuracy vary with the number of latent dimensions. **(a)** NHP grid reaching dataset. **(b)** Behavior decoding performance (CC) peaks around $n_1 = 16$, which is therefore selected as the number of behaviorally relevant latent dimensions. **(c, d)** Neural prediction accuracy—measured via Gaussian prediction CC and spike prediction PP—peaks around $n_x = 64$, which is selected as the total latent dimensionality. **(e)** NHP center-out reaching dataset. **(f)** Behavior decoding performance (CC) peaks around $n_1 = 16$, which is therefore selected as the number of behaviorally relevant latent dimensions. **(g, h)** Neural prediction accuracy—measured via Gaussian prediction CC and spike prediction PP—peaks around $n_x = 32$, which is selected as the total latent dimensionality.

A.5.8 LATENT INFORMATIVENESS FOR BEHAVIOR AND MULTIMODAL NEURAL TIMESERIES

As described in Appendix A.5.7, for the first dataset we fixed the total latent dimensionality to 64, allocating 16 dimensions to behaviorally relevant latents learned in Stage 1. To quantify how much information each of the 64 latent dimensions carries about behavior and neural activity, we conducted a latent–informativeness analysis. We constructed cumulative sets of latents by adding latents one by one to the sets; for example, set i includes latents $\{1, \ldots, i\}$, with latents $1, \ldots, 16$ being the behaviorally relevant neural latents and latents $17, \ldots, 64$ being the residual neural latents. We then trained nonlinear MLP decoders on each set of latents and evaluated the behavior decoding performance as well as spike and LFP prediction performances using those latents. Plotting each of these performances versus the number of included latents yielded the cumulative informativeness plots shown in Figure A.8. This figure shows the following:

1. **Behavior information is highly concentrated in latents learned in Stage 1.** The cumulative behavior informativeness curve rises sharply within the first $\sim$10–16 latent dimensions and then saturates (Figure A.8 (a)). Adding the remaining latents (up to 64 total) provides no additional improvement in behavior decoding, indicating that the behaviorally relevant information in neural data is captured by the behaviorally relevant neural latents learned in Stage 1; that is, $\mathbf{x}^{(1)}$ is sufficient for behavior decoding and the residual neural latents add no further behavior information.
2. **Neural informativeness accumulates gradually across all 64 latents (Stages 1 and 3).** For neural activity, informativeness increases steadily as more residual neural latents are included (Figure A.8 (b,c)). This demonstrates that the residual neural latents learned in Stage 3 capture the neural variability left unexplained by the behaviorally relevant latents.

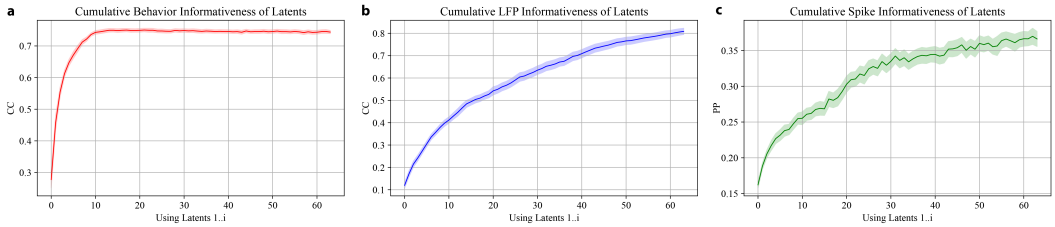

Figure A.8: **Cumulative latent informativeness for behavior, LFP, and spike activity. (a)** Behavior decoding (CC) saturates within the first $\sim$ 10–16 latent dimensions, indicating that behaviorally relevant information in multimodal neural data is captured in Stage 1 with 16 latent dimensions. **(b,c)** Informativeness for LFP (CC) and spikes (PP) increases gradually across the full 64-dimensional latent space, reflecting that residual neural latents in Stage 3 provide additional information about multimodal neural dynamics.