arXiv:2611.05167v1 [cs.AI] 3 Nov 2026

Learning Behaviorally Relevant Dynamics in Multimodal Neural Recordings

Parastoo Azizeddin[1] **Eray Erturk**[1] **Maryam M. Shanechi**[*1–4]
University of Southern California, Los Angeles, CA
{azizeddi,eerturk,shanechi}@usc.edu

Abstract

Multimodal neural data can enable a more complete understanding of brain dynamics underlying behavior. Modalities such as spiking activity, local field potentials (LFPs), and behavior capture diverse spatiotemporal aspects of brain processes, yet prior neural-behavioral models that separate sources of variability rely on a single neural modality. 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. Here we develop BREM-NET, a nonlinear multimodal dynamical model that integrates behavior with multiple neural modalities having distinct statistical characteristics and temporal resolutions. BREM-NET performs multimodal neural fusion during inference while separating behaviorally relevant and residual neural dynamics via a structured learning framework. In a public multimodal non-human primate dataset with simultaneous spiking and local field potential recordings during a 2D reaching task, BREM-NET separates these two types of dynamics and learns them more accurately, as reflected in better behavior decoding, better neural prediction, and stronger latent alignment with behavior, and it retains these capabilities when neural modalities are asynchronous or suffer dropouts, which are major challenges in real-world recordings. This framework provides a new tool for studying behaviorally relevant neural computations across spatiotemporal scales of brain activity.

1 Introduction

Understanding neural dynamics across spatiotemporal scales and their relationship to behavior is a fundamental challenge in neuroscience, and different scales are captured by different modalities: spikes reflect rapid transient events of individual neurons and are often modeled as Poisson, LFPs capture slower oscillatory processes at a larger spatial scale and are typically modeled as Gaussian, and behavior is a high-level output of the nervous system with complex relationships to neural activity [1–9]. Integrating them can provide a unified dynamical framework for uncovering the multimodal neural mechanisms driving behavior and improve neural-behavioral prediction, e.g., for brain-computer interfaces (BCIs), where spike signals often degrade over time while LFPs remain comparatively stable [10]. However, developing such models remains challenging. First, neural modalities have both *behaviorally relevant* dynamics shared with behavior and other neural dynamics that these leave unexplained, which we refer to as *residual neural dynamics*; separating them enables new scientific analyses – such as characterizing which aspects of multimodal neural activity are sufficient to describe behavior, apart from the remaining neural variability – and requires functionally

[*]Corresponding author: shanechi@usc.edu.
[1]Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California
[2]Thomas Lord Department of Computer Science, University of Southern California
[3]Alfred E. Mann Department of Biomedical Engineering, University of Southern California
[4]Neuroscience Graduate Program, University of Southern California

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Workshop: Foundation Models for the Brain and Body.

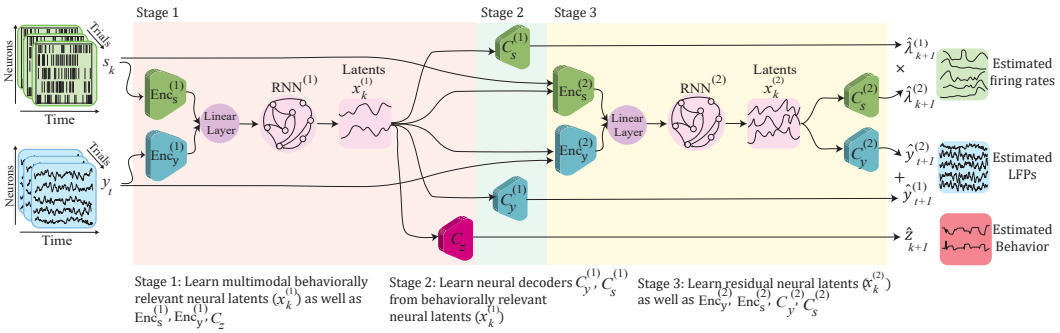


Figure 1: **BREM-NET model architecture.** At each time step, multimodal neural observations are used to infer latent states that predict both behavior and neural activity. **Stage 1** learns behaviorally relevant latent states $\mathbf{x}_k^{(1)}$ from multimodal neural data by jointly training an RNN with modality-specific encoders and a behavior decoder to minimize the negative log-likelihood (NLL) of behavior. **Stage 2** fits two modality-specific neural decoders to minimize the NLL of the neural time-series given $\mathbf{x}_k^{(1)}$. **Stage 3** learns residual neural latent states $\mathbf{x}_k^{(2)}$ by jointly fitting a second RNN with modality-specific encoders and decoders, minimizing the multimodal NLL of neural observations given both latent sets, so that it learns residual neural dynamics not predicted by $\mathbf{x}_k^{(1)}$.

separated latent subspaces rather than a single unified representation. Second, discrete spikes and continuous LFPs have distinct statistical characteristics. Third, in real recordings modalities are often asynchronous, i.e., sampled at different temporal resolutions, or suffer dropouts. Finally, their dynamics can be nonlinear.

Prior work falls into two categories, neither of which addresses all of the above challenges. The first jointly models a *single* neural modality with behavior to learn behaviorally relevant dynamics, using linear and nonlinear models [11–22]; these separate the two types of dynamics but do not fuse multiple neural modalities in modeling or inference. The second fuses multiple neural time-series [23, 18, 24–30] – but does not separate behaviorally relevant from residual neural dynamics, so the two can be conflated. Further, many of these methods are designed for fusing modalities with the same temporal resolution.

We propose **B**ehaviorally **Re**levant modeling of **M**ultimodal **Ne**ural **T**ime-series (**BREM-NET**), a nonlinear multimodal dynamical model that simultaneously: 1) incorporates multiple neural modalities and behavior in a unified model; 2) 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; and 3) supports fusion of modalities with heterogeneous distributions, different temporal resolutions, or dropouts. On a public multimodal non-human primate (NHP) dataset with simultaneous spiking and LFP recordings during a 2D reaching task [31], and in simulations, BREM-NET fuses information across neural modalities – even when sampled asynchronously – while separating the two types of dynamics. Doing so, it learns more accurate behaviorally relevant latents, as reflected in behavior decoding and alignment with behavior, and achieves accurate neural prediction via its residual neural latents, which capture the neural variability that the behaviorally relevant latents leave unexplained. This provides a new tool for investigating multiscale neural computations underlying behavior, with implications for multimodal BCIs under realistic, asynchronous conditions.

2 Methods

The architecture of BREM-NET (Fig. 1) is designed for joint modeling of behavior and multiple neural modalities observed simultaneously, and addresses two key challenges: separating the behaviorally relevant from the residual neural dynamics, and handling asynchronous modalities. For the separation, BREM-NET employs two RNNs trained through a structured three-stage process: the first RNN learns behaviorally relevant neural latents in Stage 1, whose mapping to the neural modalities is learned in Stage 2, while the second RNN learns residual neural latents in Stage 3 that capture neural dynamics unexplained by the first. To handle asynchronous modalities, we introduce separate modality-specific encoders that process each modality at its native temporal resolution.

2.1 Model formulation

We model the multimodal neural and behavior time-series as observations of a nonlinear latent dynamical system. The system state $\mathbf{x}_k \in \mathbb{R}^{n_x}$ evolves in time through a nonlinear recursion with additive noise, $\mathbf{x}_{k+1} = F(\mathbf{x}_k) + \mathbf{w}_k$, and each observed time-series is generated from the state at the corresponding time step, conditionally independently of the others. Concretely, the state is passed through a modality-specific nonlinear decoder network, $\bar{\boldsymbol{C}}_{\mathbf{y}}(\cdot)$, $\bar{\boldsymbol{C}}_{\mathbf{s}}(\cdot)$ or $\bar{\boldsymbol{C}}_{\mathbf{z}}(\cdot)$, whose output parameterizes a modality-specific likelihood: $\mathbf{y}_t \sim p_{\mathbf{y}}(\cdot \mid \bar{\boldsymbol{C}}_{\mathbf{y}}(\mathbf{x}_t))$ for the LFP, $\mathbf{s}_k \sim p_{\mathbf{s}}(\cdot \mid \bar{\boldsymbol{C}}_{\mathbf{s}}(\mathbf{x}_k))$ for the spiking activity, and $\mathbf{z}_k \sim p_{\mathbf{z}}(\cdot \mid \bar{\boldsymbol{C}}_{\mathbf{z}}(\mathbf{x}_k))$ for behavior (Appendix A.1.1). Because each modality keeps its own likelihood, sources with heterogeneous statistics are modeled jointly: we use Poisson for spike counts and Gaussian for LFP and for the continuous behavior signals, and the same formulation accommodates other choices such as Bernoulli for binary outcomes.

Here $\mathbf{s}_k$ is indexed by k and $\mathbf{y}_t$ by t. The distinct index sets emphasize that the two neural modalities can be sampled at different rates, with k indexing the faster one; we perform latent state inference on that faster timescale, which enables more flexible behavior decoding. The behavior time-series $\mathbf{z}_k \in \mathbb{R}^{n_z}$ is used only during learning and is decoded from the latent states at inference time.

We jointly and nonlinearly fuse the two neural time-series to infer the latent states using an RNN, $\mathbf{x}_{k+1} = A(\mathbf{x}_k) + \mathbf{Enc}_{\mathbf{y}}(\mathbf{y}_t) + \mathbf{Enc}_{\mathbf{s}}(\mathbf{s}_k)$, where $A(\cdot)$ captures the temporal evolution of the latents and $\mathbf{Enc}_{\mathbf{y}}(\cdot)$, $\mathbf{Enc}_{\mathbf{s}}(\cdot)$ are modality-specific nonlinear encoders. This lets BREM-NET leverage complementary information across the neural modalities in a shared latent space, while the separate modality-specific encoders address the different temporal resolutions and modality-specific dropouts common in real-world recordings: if one modality is unavailable at a given k, the corresponding encoder output is set to zero, so the state update relies only on the available modality (Appendix A.1).

Beyond multimodal fusion, another major goal of BREM-NET is to functionally separate the behaviorally relevant and the residual neural dynamics within the multimodal neural time-series. Following [12, 14, 15], we define two sets of latent states: behaviorally relevant neural latents $\mathbf{x}_k^{(1)} \in \mathbb{R}^{n_1}$, which capture the dynamics shared between behavior and the multimodal neural time-series, and residual neural latents $\mathbf{x}_k^{(2)} \in \mathbb{R}^{n_x - n_1}$, which encode the residual neural dynamics. We then expand the formulation into the following structured form:

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

where $f^{(1)}(\cdot)$ and $f^{(2)}(\cdot)$ denote the RNNs describing the behaviorally relevant and the residual neural dynamics, each parameterized by a recursion network $A(\cdot)$ and modality-specific encoders with the corresponding superscript, and $\boldsymbol{C}_{\mathbf{y}}(\cdot)$, $\boldsymbol{C}_{\mathbf{s}}(\cdot)$, $\boldsymbol{C}_{\mathbf{z}}(\cdot)$ are modality-specific decoders: $\hat{\lambda}_k$ is the predicted Poisson firing rate, $\hat{\mathbf{y}}_t$ the predicted Gaussian modality, and $\hat{\mathbf{z}}_k$ the predicted behavior. Behavior is predicted from the behaviorally relevant latents alone, whereas the neural modalities are predicted from both latent sets, whose contributions are fused by adding the Gaussian components and multiplying the Poisson firing rates, equivalently summing log-rates (Appendix A.1.1). Since $\mathbf{x}_{k+1}$ is estimated purely from past neural data, the model supports causal prediction, and passing $\mathbf{x}_{k+1}^{(1)}$ to the second RNN makes it learn the residual dynamics not already captured by the first.

2.2 Structured learning of behaviorally relevant and residual neural dynamics

To functionally separate the two types of dynamics, we learn the model in Eq. 1 with a structured three-stage procedure, in which each stage optimizes a subset of the parameters while holding those learned earlier fixed (Appendix A.1.2).

Stage 1. We train the RNN $f^{(1)}(\cdot)$ jointly with the behavior decoder $\boldsymbol{C}_{\mathbf{z}}(\cdot)$, minimizing the negative log-likelihood (NLL) of behavior given the inferred latents $\mathbf{x}_k^{(1)}$:

$$
\mathbf{L}_{\mathbf{z}} = \sum_k \mathbf{NLL}\big(\mathbf{z}_k; \boldsymbol{C}_{\mathbf{z}}(\mathbf{x}_k^{(1)})\big). \tag{2}
$$

This ensures these latents capture the behaviorally relevant information in the multimodal neural data.

Stage 2. We train the neural decoders $\boldsymbol{C}_{\mathbf{y}}^{(1)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{s}}^{(1)}(\cdot)$, which predict the neural modalities from $\mathbf{x}_k^{(1)}$, by minimizing the combined NLL of both modalities under their respective likelihoods, with a scaling factor α that balances their scale differences (Appendix A.1.3):

$$\mathbf{L}_{\mathbf{y},\mathbf{s}}^{(1)} = \sum_t \mathbf{NLL}_{\mathbf{y}}\big(\mathbf{y}_t; \boldsymbol{C}_{\mathbf{y}}^{(1)}(\mathbf{x}_t^{(1)})\big) + \alpha \sum_k \mathbf{NLL}_{\mathbf{s}}\big(\mathbf{s}_k; \boldsymbol{C}_{\mathbf{s}}^{(1)}(\mathbf{x}_k^{(1)})\big). \tag{3}$$

Stage 3. Finally, we train the second RNN $f^{(2)}(\cdot)$ jointly with the decoders $\boldsymbol{C}_{\mathbf{y}}^{(2)}(\cdot)$ and $\boldsymbol{C}_{\mathbf{s}}^{(2)}(\cdot)$, minimizing the combined NLL of both neural modalities given *both* $\mathbf{x}_k^{(1)}$ and $\mathbf{x}_k^{(2)}$. Because Stages 1–2 already account for the part of $\mathbf{y}_t$ and $\mathbf{s}_k$ predictable from the behaviorally relevant latents, this stage explains the residual neural dynamics:

$$\mathbf{L}_{\mathbf{y},\mathbf{s}}^{(2)} = \sum_t \mathbf{NLL}_{\mathbf{y}}\big(\mathbf{y}_t; \boldsymbol{C}_{\mathbf{y}}^{(1)}(\mathbf{x}_t^{(1)}), \boldsymbol{C}_{\mathbf{y}}^{(2)}(\mathbf{x}_t^{(2)})\big) + \alpha \sum_k \mathbf{NLL}_{\mathbf{s}}\big(\mathbf{s}_k; \boldsymbol{C}_{\mathbf{s}}^{(1)}(\mathbf{x}_k^{(1)}), \boldsymbol{C}_{\mathbf{s}}^{(2)}(\mathbf{x}_k^{(2)})\big). \tag{4}$$

3 Experimental Results

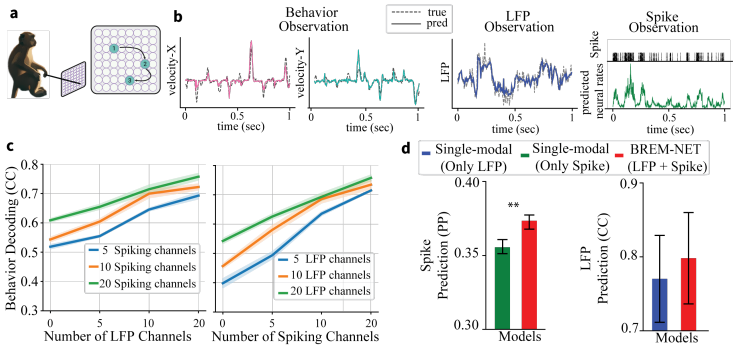


Figure 2: **BREM-NET successfully fuses multimodal neural time-series, thus improving neural-behavioral predictions in the NHP grid reaching dataset. (a)** Dataset and task. **(b)** True behavior, LFP and spike time-series (dashed) versus their BREM-NET predictions (solid) for a representative session. **(c)** Cross-validated behavior decoding CC when 5, 10 or 20 spiking (*left*) or LFP (*right*) channels were the primary modality and channels of the other modality were fused; lines show the mean, shaded areas the SEM. **(d)** Cross-validated spike prediction PP (*left*) and LFP prediction CC (*right*) with BREM-NET vs. single-modal models. N=20; **: $p < 0.001$, one-sided Wilcoxon signed-rank test.

Dataset and metrics. We validated BREM-NET on a public NHP dataset [31] with simultaneous spiking activity and raw LFP from primary motor cortex during sequential 2D reaches to random targets in virtual reality (Fig. 2a, Appendix A.3), and on simulated Gaussian and Poisson systems (Appendix A.5.1). All results use 5-fold cross-validation across all sessions. For Gaussian signals we report the Pearson correlation coefficient (CC); for spike counts, prediction power ($\mathrm{PP} = 2\,\mathrm{AUC} - 1$) [32, 33]; both averaged across dimensions (Appendix A.1.5).

3.1 Multimodal fusion extracts more accurate behaviorally relevant neural latents

We treated spike counts and raw LFPs as the neural time-series and 2D cursor velocity as behavior, with $n_x = 64$ and $n_1 = 16$ selected by sweeping both dimensionalities on a single session (Appendix A.1.4). To assess fusion, either spikes or LFPs served as the primary modality while increasing numbers of channels of the other were fused (Fig. 2c). Adding LFP channels to spiking activity consistently improved behavior decoding ($p < 10^{-5}$, $n = 20$, one-sided Wilcoxon signed-rank test), and a similar trend held when LFPs were primary, with larger gains in that case, suggesting spikes encoded more behavior-related information here. More generally, fusion yields its largest gains when the primary modality is information-limited (e.g., fewer primary channels in Fig. 2c), the regime characteristic of chronic BCIs in which spike quality degrades over time while LFPs

remain comparatively stable [10]. Fusion also improved neural prediction of each modality over single-modal models (Fig. 2d).

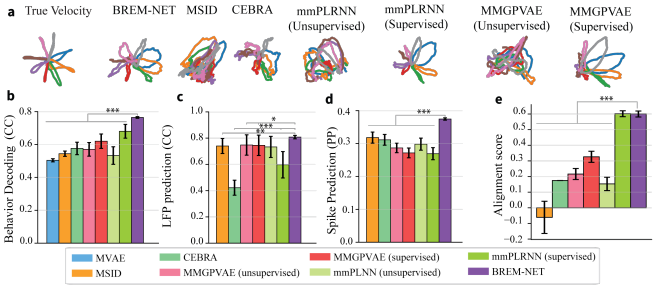


Figure 3: **Latents inferred by BREM-NET are more aligned with true behavior trajectories and achieve accurate neural-behavioral prediction.** **(a)** True velocity trajectories vs. condition-averaged latent trajectories for BREM-NET and baselines. **(b)** Behavior decoding CC. **(c)** LFP prediction CC. **(d)** Spike prediction PP. **(e)** Alignment score. Bars show mean $\pm$ SEM (N=20); *: $p < 0.05$, **: $p < 0.001$, ***: $p < 10^{-7}$, one-sided Wilcoxon signed-rank test.

To assess the functional separation of the two sets of latents, we evaluated their specialization by using $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ separately (Fig. A.2b, Appendix A.5.3): $\mathbf{x}^{(1)}$ accurately decoded behavior, while $\mathbf{x}^{(2)}$ excelled at predicting neural activity but decoded behavior substantially less accurately. As a complementary geometric characterization, true velocity trajectories were much more aligned with $\mathbf{x}^{(1)}$ than with $\mathbf{x}^{(2)}$ trajectories (Fig. A.2c), as quantified by an alignment score, defined as the Pearson correlation between the Euclidean distance matrices of the behavior signals and the neural latent trajectories [34]. Together, these results indicate that Stage 1 captures the behaviorally relevant dynamics while Stage 3 captures the residual neural variability.

Comparison to multimodal baselines. We compared BREM-NET to MVAE [35], MSID [28], multi-session CEBRA [16], mmPLRNN [26], and MMGPVAE [17] (Fig. 3; Appendix A.2). For fairness, mmPLRNN and MMGPVAE received the two neural modalities as inputs and were trained both unsupervised (their original form) and supervised, with an added behavior reconstruction loss. As shown in Fig. 3, BREM-NET's behaviorally relevant latents were more aligned with true velocity than those of all baselines (Fig. 3a,e), and it decoded behavior and predicted neural activity more accurately (Fig. 3b–d).

3.2 Multimodal fusion and structured learning are both critical

We ablated BREM-NET's components (Fig. A.3, Appendix A.5.4). Multimodal fusion, behavior supervision and the staged procedure each contribute: latents from the single-modal variants, from the unsupervised U-BREM-NET and from a single-stage joint-loss variant are all markedly less aligned with behavior, and BREM-NET also beats its linear counterpart L-BREM-NET.

3.3 Robustness to asynchronous sampling and modality 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.2, 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 strictly aligned in time. Furthermore, under stochastic modality-specific dropouts ($q \in \{0.2, 0.4, 0.6, 0.8\}$ per timestep), we show that behavior decoding declined by less than 2% even with 80% of LFP samples removed and by about 5% at 40% spike dropout (Appendix A.5.2). Overall, these results show that BREM-NET can robustly fuse even asynchronous and incomplete neural modalities.

4 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. In a multimodal NHP dataset and in simulations, 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. 2c), while its compact behaviorally relevant state supports accurate behavior decoding for BCIs.

Limitations and future directions. We applied our method to spikes and raw LFPs given their importance in neuroscience and BCIs. Important future directions include extending it to more modalities, such as EEG [36], intracranial EEG [37], or physiological signals such as heart rate, and to other brain regions and behaviors [38].

Acknowledgments and Disclosure of Funding

This work was supported by National Institutes of Health (NIH) grants R61MH135407, R01MH123770, and RF1DA056402, the One Mind Rising Star Award, and the Foundation for OCD Research (FFOR).

References

[1] 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.

[2] 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.

[3] 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.

[4] 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, 2018.

[5] 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.

[6] Mehrdad Ramezani, Yundong Ren, Ertugrul Cubukcu, and Duygu Kuzum. Innovating beyond electrophysiology through multimodal neural interfaces. *Nature Reviews Electrical Engineering*, pages 1–16, 2024.

[7] Omid Sharafi, Timothy Silliman, Hadi Mokhtari Dowlatabad, Gengle Niu, Steven T. Walston, Jean-Marie C. Bouteiller, Kimberly K. Gokoffski, and Gianluca Lazzi. Population-scale analysis of frequency-dependent calcium dynamics in retinal ganglion cells under electric field stimulation. *Scientific Reports*, 16:18948, 2026.

[8] Nicholas Householder, Omid Sharafi, Anahit Simonyan, Steven T Walston, Gianluca Lazzi, Michael Bienkowski, and Kimberly K Gokoffski. Extraocular electrical stimulation activates retinal ganglion cells in vivo. *bioRxiv*, 2026.

[9] Yasaman Torabi, Parsa Razmara, Hamed Ajorlou, and Bardia Baraeinejad. NeuroMambaLLM: Dynamic graph learning of fMRI functional connectivity in autistic brains using Mamba and language model reasoning. *arXiv preprint arXiv:2602.13770*, 2026.

[10] 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.

[11] 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.

[12] 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.

[13] Parsa Vahidi, Omid G Sani, and Maryam M Shanechi. Modeling and dissociation of intrinsic and input-driven neural population dynamics underlying behavior. *Proceedings of the National Academy of Sciences*, 121(7):e2212887121, 2024.

[14] 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.

[15] 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.

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

[17] 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. *arXiv preprint arXiv:2310.03111*, 2023.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] Sayed Mohammad Hosseini and Maryam M. Shanechi. Dynamical modeling of behaviorally relevant spatiotemporal patterns in neural imaging data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 23846–23872, 2025.

[23] 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. *The dynamic brain: An exploration of neuronal variability and its functional significance*, pages 3–28, 2011.

[24] 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.

[25] 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.

[26] Daniel Kramer, Philine Lou Bommer, Carlo Tombolini, Georgia Koppe, and Daniel Durstewitz. Reconstructing nonlinear dynamical systems from multi-modal time series. *arXiv preprint arXiv:2111.02922*, 2021.

[27] 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.

[28] 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.

[29] DongKyu Kim, Christian Y Song, Han-Lin Hsieh, and Maryam M Shanechi. Unsupervised learning of multiscale switching dynamical system models from multimodal neural data. *Journal of Neural Engineering*, 23(2):026039, 2026.

[30] Eray Erturk and Maryam M. Shanechi. Dynamical modeling of nonlinear latent factors in multiscale neural activity with real-time inference. In *Advances in Neural Information Processing Systems*, 2025.

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

[32] 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.

[33] 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.

[34] Yiqi Jiang, Kaiwen Sheng, Yujia Gao, E Kelly Buchanan, Yu Shikano, Tony Hyun Kim, Yixiu Zhao, Seung Je Woo, Fatih Dinc, Scott Linderman, et al. Extracting task-relevant preserved dynamics from contrastive aligned neural recordings. *bioRxiv*, 2025.

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

[36] Parastoo Azizeddin, Omid Sharafi, and Maryam M Shanechi. Multimodal LLMs can learn to read brain signals: A vision–language model for unified multi-task EEG decoding. *arXiv preprint arXiv:2610.09355*, 2026.

[37] Kiarash Vaziri, Lucine L. Oganesian, HyeongChan Jo, Roberto M. C. Vera, Charles Y. Liu, Brian Lee, and Maryam M. Shanechi. Nonlinear dynamical modeling of human intracranial brain activity with flexible inference. *arXiv preprint arXiv:2512.22785*, 2025.

[38] Trisha Jha, Dong Song, and Maryam M Shanechi. Prioritized nonlinear modeling of shared dynamics across neural populations. *Journal of Neural Engineering*, 23(5):056054, 2026.

[39] Eli Brookner. *Tracking and Kalman filtering made easy*. Wiley, New York, 1998.

[40] 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.

[41] 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*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

A Appendix

A.1 Method Details

A.1.1 Method Formulation

In Equation 1, 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:

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

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 [39] 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 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 A.1 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:

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

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

$$\hat{\lambda}_k = \exp\left(\boldsymbol{C}_{\boldsymbol{s}}''^{(1)}(\mathbf{x}_k^{(1)}) + \boldsymbol{C}_{\boldsymbol{s}}''^{(2)}(\mathbf{x}_k^{(2)})\right) \tag{A.3}$$

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 A.1. Each stage is designed to optimize distinct components of the model in order to accurately predict observations while separating behaviorally relevant neural dynamics and 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 $\mathbf{y}_t$ and $\mathbf{s}_k$ (Equation A.4). 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:

$$\begin{cases} \mathbf{x}_{k+1}^{(1)} = A^{(1)}(\mathbf{x}_k) + \mathbf{Enc}_{\mathbf{y}}^{(1)}(\mathbf{y}_t) + \mathbf{Enc}_{\mathbf{s}}^{(1)}(\mathbf{s}_k) \\ \hat{\mathbf{z}}_k = \boldsymbol{C}_{\mathbf{z}}(\mathbf{x}_{\mathbf{k}}^{(\mathbf{1})}) \\ \mathbf{L}_z^{(1)} = -\sum_k \log\left(p_z(\mathbf{z}_k \mid \mathbf{x}_k^{(1)})\right) \end{cases} \tag{A.4}$$

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. A.5). This α factor ensures that both modalities contribute meaningfully to the optimization process, regardless of their inherent scale.

$$\begin{cases} \mathbf{x}_{k+1}^{(1)} = A^{(1)}(\mathbf{x}_k) + \mathbf{Enc}_{\mathbf{y}}^{(1)}(\mathbf{y}_t) + \mathbf{Enc}_{\mathbf{s}}^{(1)}(\mathbf{s}_k) \\ \hat{\tilde{\mathbf{y}}}_t = \boldsymbol{C}_{\mathbf{y}}^{(1)}(\mathbf{x}_t^{(1)}) \\ \hat{\tilde{\lambda}}_k = \boldsymbol{C}_{\mathbf{s}}^{(1)}(\mathbf{x}_k^{(1)}) \\ \mathbf{L}_{\mathbf{y},\mathbf{s}}^{(1)} = -\big(\sum_t \log\left(p_y(\mathbf{y}_t \mid \mathbf{x}_t^{(1)})\right) + \alpha \sum_k \log\left(p_s(\mathbf{s}_k \mid \mathbf{x}_k^{(1)})\right)\big) \end{cases} \tag{A.5}$$

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

$$\begin{aligned} \hat{\mathbf{y}}_t &= \hat{\tilde{\mathbf{y}}}_t + \hat{\tilde{\mathbf{y}}}_t \\ \hat{\mathbf{s}}_k &\sim \mathcal{P}ois(\hat{\tilde{\lambda}}_k \times \hat{\tilde{\lambda}}_k) \end{aligned} \tag{A.6}$$

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:

$$\begin{cases} \mathbf{x}_{k+1}^{(2)} = A^{(2)}(\mathbf{x}_k) + \mathbf{Enc}_{\mathbf{y}}^{(2)}(\mathbf{y}_t) + \mathbf{Enc}_{\mathbf{s}}^{(2)}(\mathbf{s}_k) \\ \hat{\hat{\mathbf{y}}}_t = \boldsymbol{C}_{\mathbf{y}}^{(2)}(\mathbf{x}_t^{(2)}) \\ \hat{\hat{\lambda}}_k = \boldsymbol{C}_{\mathbf{s}}^{(2)}(\mathbf{x}_k^{(2)}) \\ \mathbf{L}_{\mathbf{y},\mathbf{s}}^{(2)} = -\big(\sum_t \log\left(p_y(\mathbf{y}_t \mid \mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)})\right) + \alpha \sum_k \log\left(p_s(\mathbf{s}_k \mid \mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)})\right)\big) \end{cases} \tag{A.7}$$

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.

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 the real-world dataset, are detailed in Table A.1. 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.1). 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 [40], which is known for its effectiveness in training deep neural networks. It was initialized with a learning rate of 10^{-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 [41]. For optimization, we utilized the AdamW optimizer [42]. We trained the 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.

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 selected $n_1 = 16$ and $n_x = 64$ based on behavior decoding and neural prediction accuracy on a held-out session. This choice is a model-selection criterion rather than an estimate of the intrinsic dimensionality of behaviorally relevant neural activity.

A.1.5 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 1 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 the neural modalities $\mathbf{y}_t$ and $\mathbf{s}_k$, and does not incorporate behavior itself as an input during inference.

Table A.1: **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 ($\boldsymbol{C_y}, \boldsymbol{C_s}$)	[128, 128, 128]
Behavior decoder ($\boldsymbol{C}_z$)	[64]
Initial learning rate	10^{-3}
LR scheduler	StepLR
Batch size	32
Number of epochs	1000
Weight decay	10^{-3}

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 (Figs. 2, 3, A.3) 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 [32]. 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 [33].

A.2 Baselines and model variants

We compare BREM-NET against multimodal baselines and against ablated variants of our own model. All models use the same cross-validation protocol and the same number of channels per modality, and we performed hyperparameter sweeps (including latent dimensionality) for each baseline while keeping the total number of trainable parameters comparable across methods, so that gains are not confounded by model capacity.

Model variants. L-BREM-NET keeps the three-stage architecture but replaces every encoder and decoder with linear maps. U-BREM-NET runs Stage 3 only, learning neural dynamics without behavior supervision (equivalently $n_1 = 0$) and fitting a behavior decoder downstream on the resulting latents. BREM-NET w/o Stage 3 keeps Stages 1–2 only and therefore has no residual neural latents. BREM-NET Joint trains all latents in a single stage with a combined behavior-decoding and neural-prediction loss. Single-modal variants remove the encoder and decoder of one neural modality, so no fusion occurs during inference.

External baselines. MSID [28] is a linear multiscale subspace-identification method; we used the authors' implementation with horizons $h_y = h_z = 10$ as specified in the original work. MVAE [35] is a generic multimodal variational autoencoder without temporal modeling or behavior supervision; we treated each time step as an independent sample and fit linear regressions on the inferred latents, and it cannot perform time-series prediction. CEBRA [16] is a convolutional encoder trained with a contrastive objective; we used its multi-session mode to process LFP and spikes jointly and fit linear readouts on the embeddings, since CEBRA has no decoder. mmPLRNN [26] uses piecewise-linear RNN dynamics with variational inference; we added a Poisson observation model following the authors' Appendix C. MMGPVAE [17] places Gaussian-process priors over the latents and optimizes the ELBO; we used 64-dimensional latents per modality with 32 shared and matched its encoder/decoder architecture to ours. For all external baselines we used the authors' public implementations and recommended hyperparameters.

Neither mmPLRNN nor MMGPVAE supports predicted (one-step-ahead) inference, so for a fair comparison we modified their inference to use only neural observations up to time $k-1$ when inferring the latent state at k, then decoded the predicted observations from those latents. Both were additionally trained in a supervised variant in which a feedforward map from the inferred latents to behavior, and its reconstruction loss, were added to the original objective; behavior is never an input

at inference. Unlike BREM-NET, this makes their latent space more behaviorally informative overall but does not separate behaviorally relevant from residual neural dynamics.

A.3 NHP grid reaching dataset

We analyzed a publicly available dataset [31] 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 cursor 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.4 Simulation Details

For our simulation data in section A.5.1, we generated Gaussian and Poisson observations from four randomly initialized systems as defined by the dynamics in

$$\begin{aligned} \mathbf{x}_{k+1} &= F(\mathbf{x}_k) + \mathbf{w}_k \\ \mathbf{y}_k \mid \mathbf{x}_k &\sim \mathcal{N}\big(\mathbf{y}_k \mid \boldsymbol{C}_{\mathbf{y}}(\mathbf{x}_k)\big) \\ \mathbf{s}_k \mid \mathbf{x}_k &\sim \mathcal{P}ois\big(\mathbf{s}_k \mid \boldsymbol{C}_{\mathbf{s}}(\mathbf{x}_k)\big) \\ \mathbf{z}_k \mid \mathbf{x}_k &\sim \mathcal{N}\big(\mathbf{z}_k \mid \boldsymbol{C}_{\mathbf{z}}(\mathbf{x}_k)\big) \end{aligned} \tag{A.8}$$

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 2×10^6 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 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 A.8. 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 ($n_2 = 4$).

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). Specifically, in Figure A.1a, 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.

We found that BREM-NET demonstrated superior performance in behavior decoding compared to single-modal models, as shown in Figure A.1a. In scenarios where the Poisson modality served as the primary modality, incorporating Gaussian channels consistently improved behavior decoding accuracy (Figure A.1a, left). Similar trend was observed when the Gaussian modality was the primary modality and Poisson channels were fused (Figure A.1a, right). Further, incorporating information from one neural modality enhanced the prediction accuracy of the other modality, showing the power of cross-modality information fusion (Figure A.1b). Consistently, BREM-NET successfully predicted both neural time-series modalities and behavioral time-series from the learned latent states (Figure A.1c).

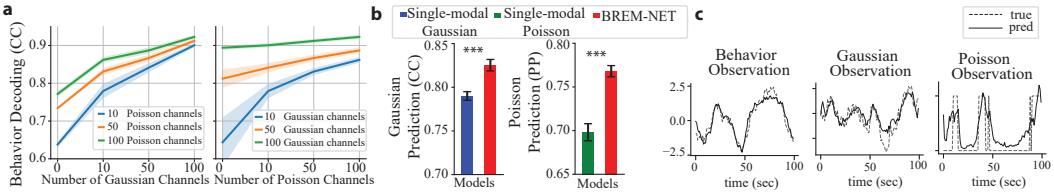


Figure A.1: **BREM-NET improves behavior decoding and multimodal neural predictions in a simulated dataset.** Cross-validated accuracy of behavior decoding and Gaussian and Poisson neural prediction in four simulated systems (N=20). **(a)** *left*: behavior decoding correlation coefficient (CC) when 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*: Prediction 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.5.2 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 dataset used in our experiments is fully observed, BREM-NET is explicitly designed to handle missing data and asynchronous sampling as explained in Methods section 2. 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.2, 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.

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 $q \in \{0.2, 0.4, 0.6, 0.8\}$, 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.3, 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 spikes carrying richer behavior-related information in this dataset (Fig. 2c). 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.2: **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

Table A.3: **BREM-NET is robust to dropouts.** We evaluated decoding performance when either spikes or LFPs were randomly dropped at each timestep with probability $q \in \{0.2, 0.4, 0.6, 0.8\}$. 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

A.5.3 Functional Separation of behaviorally relevant and residual neural latents

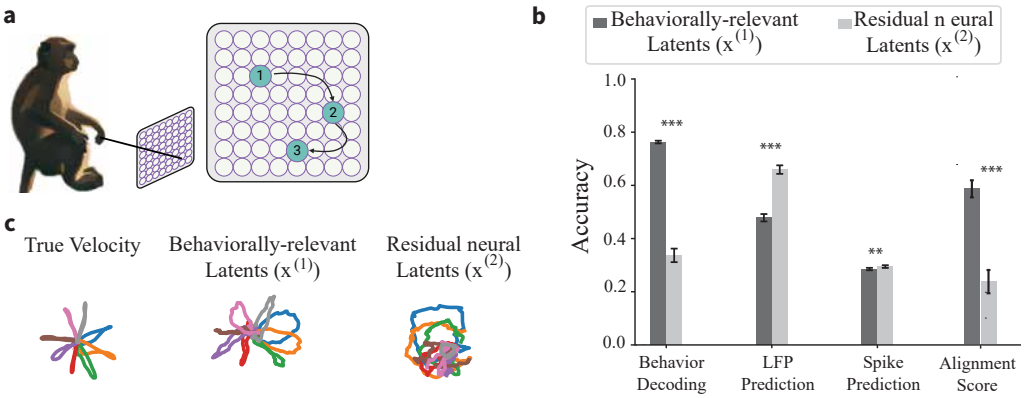


Figure A.2: **Separation of behaviorally relevant and residual neural dynamics through structured multi-stage learning.** Evaluation of the separation of the learned latent states by using $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ separately to predict behavior and neural modalities. **(a)** NHP grid-reaching 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.5.4 Ablation studies

We ablated the components of BREM-NET on the NHP grid reaching dataset (Fig. A.3; see Appendix A.2 for variant definitions). Multimodal fusion, behavior supervision and the staged learning procedure each contribute: the behaviorally relevant latents of the full model are more aligned with behavior than those of the single-modal variants, of U-BREM-NET, and of the joint-loss variant, and BREM-NET matches or exceeds every variant on behavior decoding and neural prediction. Removing Stage 3 leaves behavior decoding intact but substantially degrades neural prediction, confirming that the residual neural latents carry residual neural variability.

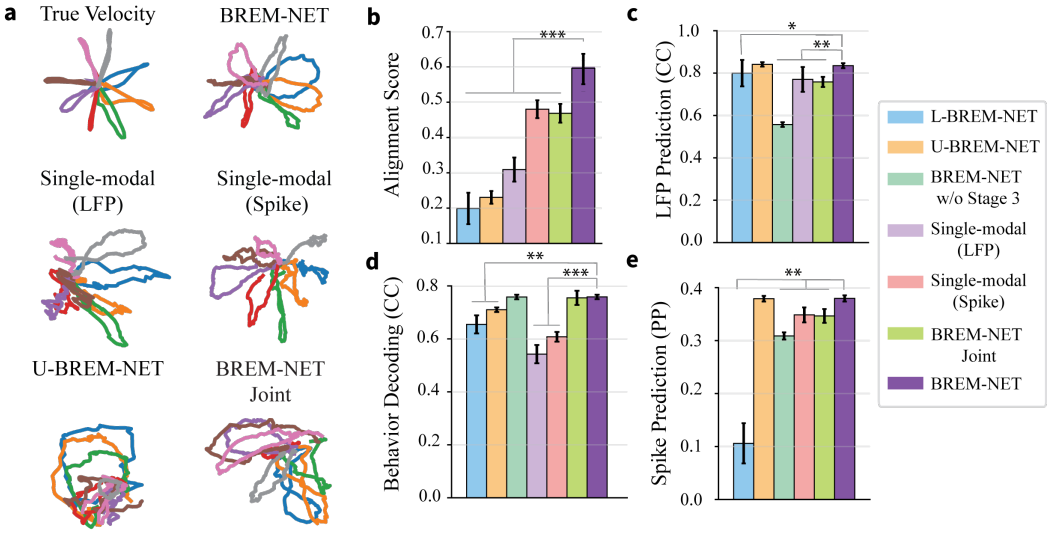


Figure A.3: **Ablations show that multimodal structured modeling improves the accuracy and behavior alignment of behaviorally relevant neural latents while also capturing residual neural variability.** **(a)** Condition-averaged behaviorally relevant latent trajectories. **(b)** Alignment score. **(c)** LFP prediction CC. **(d)** Behavior decoding CC. **(e)** Spike prediction PP. Bars show mean ± SEM across all sessions and 5 folds (N=20); asterisks indicate significance.