Exploiting correlations across trials and behavioral sessions to improve neural decoding

Yizi Zhang[1,⋆], Hanrui Lyu[2,⋆], Cole Hurwitz[2], Shuqi Wang[3], Charles Findling[6], Felix Hubert[6], Alexandre Pouget[6], International Brain Laboratory[5], Erdem Varol[4], Liam Paninski[1,2,†]

1 Department of Statistics, Columbia University, New York, New York, United States of America
2 Center for Theoretical Neuroscience, Columbia University, New York, New York, United States of America
3 Department of Computer Science, École Polytechnique Fédérale de Lausanne, Écublens, Vaud, Switzerland
4 Department of Computer Science and Engineering, New York University, New York, New York, United States of America
5 The International Brain Laboratory
6 Department of Basic Neurosciences, University of Geneva, Geneva, Switzerland

⋆ These authors contributed equally to this work
† Correspondence: liam@stat.columbia.edu

Abstract

Traditional neural decoders model the relationship between neural activity and behavior within individual trials of a single experimental session, neglecting correlations across trials and sessions. However, animals exhibit similar neural activities when performing the same behavioral task, and their behaviors are influenced by past experiences from previous trials. To exploit these informative correlations in large datasets, we introduce two complementary models: a multi-session reduced-rank model that shares similar behaviorally-relevant statistical structure in neural activity across sessions to improve decoding, and a multi-session state-space model that shares similar behavioral statistical structure across trials and sessions. Applied across 433 sessions spanning 270 brain regions in the International Brain Laboratory public mouse Neuropixels dataset, our decoders demonstrate improved decoding accuracy for four distinct behaviors compared to traditional approaches. Unlike existing deep learning approaches, our models are interpretable and efficient, uncovering latent behavioral dynamics that govern animal decision-making, quantifying single-neuron contributions to decoding behaviors, and identifying different activation timescales of neural activity across the brain. Code: https://github.com/yzhang511/neural_decoding.

1 Introduction

Neural decoding is a critical tool for understanding the relationship between behavior and brain activity. Traditional neural decoders operate within a single-trial, single-session context [1, 2], modeling the relationship between neural activity and behavior within individual trials of each experimental session. However, these decoders overlook informative correlations across trials and sessions in both the neural and behavioral data, missing opportunities to leverage information from large datasets collected across numerous experiments.

Similar neural activities emerge across experimental sessions when animals engage in the same behavioral task [3, 4, 5]. Incorporating such inter-session neural similarities offers an opportunity to improve single-session decoding accuracy. However, directly sharing this information across sessions is challenging, since typically different populations of neurons are recorded in each session. An alternative approach is to focus on the important neural population variations relevant to the behavior, utilizing their correlation structures across sessions. Previous unsupervised studies have adopted this strategy to improve neural dynamics estimation by sharing activities across sessions [6, 7, 8]. However, the learned neural latents may not be behaviorally relevant, and have to be fine-tuned for supervised decoding tasks. While supervised pre-training can learn shared neural representations by training models on multiple sessions before fine-tuning them to decode specific behaviors, existing methods [9] require substantial computing resources and result in complex black-box models that lack interpretability. For a more lightweight and interpretable solution, a simple yet effective model is needed for sharing behaviorally relevant neural variations across many sessions.

Similarly, animal behavior is shaped not only by the current task, but also by the animal's experiences from previous trials. For example, [10] found that mouse decision-making evince internal states persisting across tens to hundreds of trials, effectively modeled by hidden Markov models (HMMs). These latent states are reproducible across animals and experiment sessions. Many neuroscience experiments exhibit trial-to-trial behavioral correlations arising from such reproducible latent states. Explicitly accounting for these behavioral correlations across sequential trials, in addition to modeling inter-session neural similarities, can potentially improve neural decoding performance.

In this work we develop two complementary methods to leverage these neural and behavioral correlations for improved neural decoding. For neural data, we employ a multi-session reduced-rank model that shares similar temporal patterns in the neural activity across sessions while retaining session-specific differences to accommodate individual variations. For behavioral data, we use multi-session state-space models to learn latent behavioral states from trial-to-trial correlations in animal behaviors across multiple sessions. These learned neural and behavioral representations are then used to improve single-trial, single-session decoders. Unlike existing deep learning methods that share data across sessions through complex black-box models, our models are simple, highly interpretable, and easy to fit. We evaluate our neural and behavioral data-sharing models using mouse Neuropixels recordings from the International Brain Lab [11, 12], which include 433 sessions and 270 brain regions. The results show improved decoding accuracy across different behavioral tasks. Our approach is computationally efficient and enables us to create a brain-wide map of behaviorally-relevant timescales and identify key neurons associated with each behavioral task.

2 Formulation of the neural data-sharing model

All analyses here are based on spike-sorted and temporally-binned spike count data. We split the recording into equal-length trials of 2 seconds. We further divide each trial into 20-millisecond time bins, yielding 90 timesteps per trial. For each trial from a session, the spike counts of N neurons are used to construct the input $X \in \mathbb{R}^{N \times T}$, where T denotes the number of timesteps per trial, to obtain an decoder $d \in \mathbb{R}^P$ of the true behavior $y \in \mathbb{R}^P$. When $P = 1$, the value of y remains constant throughout a trial (*per-trial decoded behavior*). When $P = T$, the value of y varies over time within a trial (*per-timestep decoded behavior*). To simplify our notation, we initially present the following model specification assuming y is a scalar (i.e., $P = 1$). However, when $P = T$, the decoding problem remains the same across all dimensions of P. In this case, we can apply the same solution independently to each element of y.

Traditional single-session decoders use *full-rank* models, where a full-rank (unconstrained) $N \times T$ weight matrix is fit to X; this basic full-rank approach is prone to overfitting when the number of neurons and timesteps is large. See Table 1 for notation of model parameters and variables. To reduce overfitting, we impose a low-rank constraint on the single-session decoder by factorizing the high-dimensional parameters into neural and temporal low-rank basis sets:

$$d = f(X^\top (UV) + b), \tag{1}$$

where U and V are the neural and temporal basis sets used to constrain the dimensionality of the weight matrix applied to X, and $b \in \mathbb{R}$ is the intercept term. The function f can be either linear or nonlinear, depending on the specific application. $U \in \mathbb{R}^{N \times R}$ projects N neurons' activity to a low-dimensional space of size R, while $V \in \mathbb{R}^{R \times T}$ weights each timestep differently. For $y \in \mathbb{R}$, the full-rank model has $N \times T$ parameters, while the **reduced-rank model** (Eq 1) has $R \times (N + T)$ parameters, where $R < \min(N, T)$.

The solutions of U, V and b can be obtained through either automatic differentiation or closed-form expressions. When f is an identity function, closed-form solutions are attainable. The closed-form solution of U reveals that it can be interpreted as a subspace that maximizes the correlation between neural activity X and behavior y while capturing the major variations in y. Thus, this reduced-rank model (RRM) can be viewed as a latent variable model, where the rank R determines the number of latent variables required to capture the behaviorally relevant variations in neural activity. See "Closed-form solution for theoretical interpretation" in Methods for details.

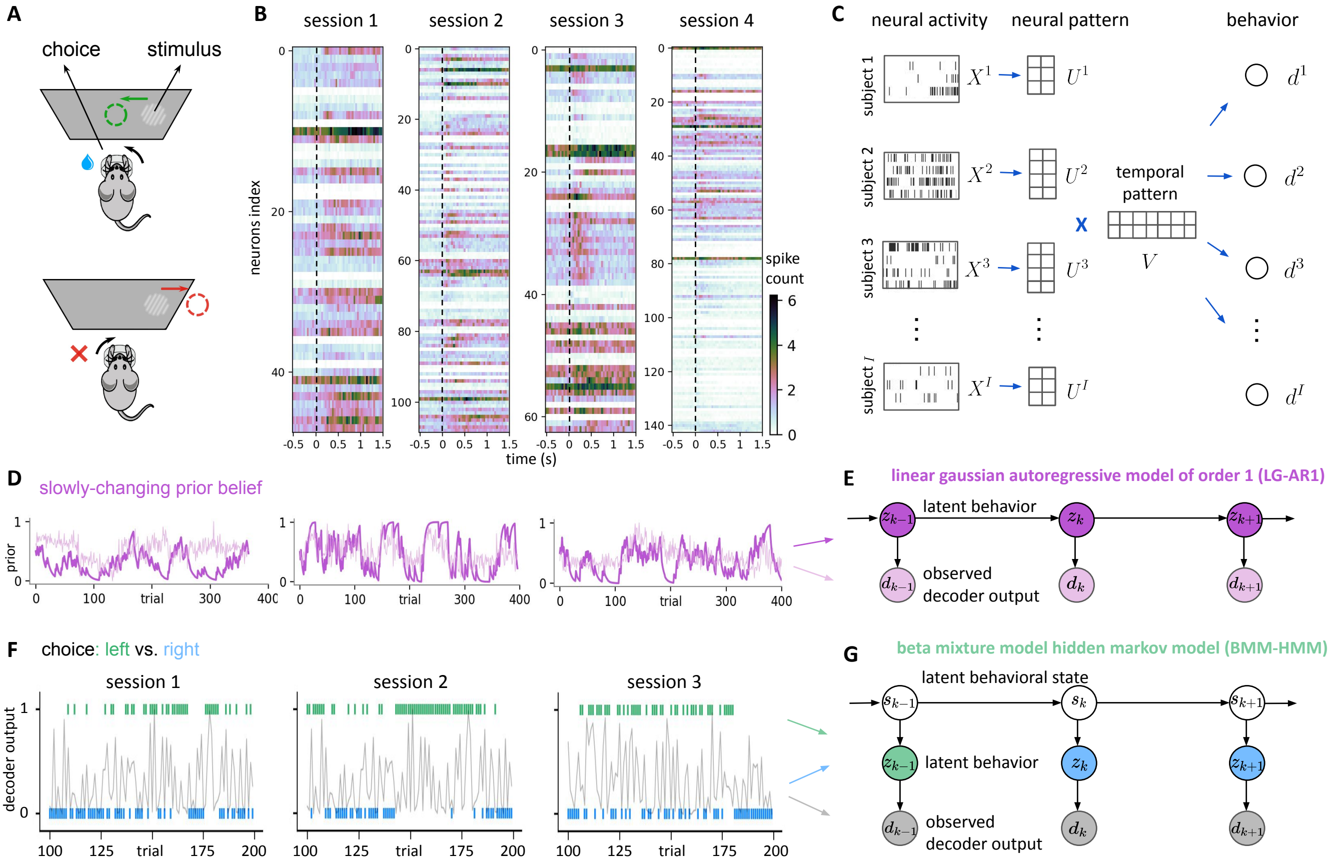

Figure 1: **Schematic illustration of the neural and behavioral data-sharing models. (A)** Schematic of the experiment where mice indicate the location of a visual stimulus by rotating a wheel. **(B)** Neural activity shows consistent activation following stimulus onset (dashed line at time t = 0s) across 6 selected sessions. Each spike train raster plot depicts the average spike count across all trials in a session. Each row in the plot represents the Peri-Stimulus Time Histogram (PSTH) of a single neuron. **(C)** Multi-session reduced-rank model with session-specific neural patterns U^i and shared temporal patterns V. **(D)** For slowly-changing prior belief y_k (dark purple), trial-to-trial correlations exist which single-trial decoders (light purple) neglect. Behavioral patterns are similar across sessions. **(E)** The LG-AR1 graphical model features latent behaviors z_k and observed single-trial decoder outputs d_k, with colors corresponding to the examples in panel D. **(F)** For binary choice y_k (blue and green), trial-to-trial correlations exist, which single-trial decoders d_k (grey) fail to capture, leading to suboptimal performance. Similar behavioral patterns also occur across sessions. **(G)** The BMM-HMM graphical model features latent behavioral states s_k, latent behaviors z_k, and observed single-trial decoder outputs d_k, with colors corresponding to the examples in panel F.

Instead of manually aligning neurons from different populations based on their firing or physical properties [13, 14], we aim to automatically learn a common neural representational space crucial for decoding from multiple neural populations. To this end, we introduce a **multi-session reduced-rank model** to learn such common neural representations and improve neural decoding. Since neural populations within a given region may exhibit similar activation patterns [3] (Fig 1B), we can share the low-rank temporal basis set V across sessions and retain session-specific differences via the neural basis set U^i:

$$d^i = f(X^{i\top}(U^i V) + b^i), \qquad (2)$$

where $X^i \in \mathbb{R}^{N^i \times T}$ and $d^i \in \mathbb{R}$ are the neural activity and predicted behavior from a single trial in session i with N^i neurons, corresponding to the terms X and d in Eq 1. As V is shared across sessions, a more robust estimation can be obtained since fewer parameters need to be learned from the same amount of data. The model schematic is summarized in Fig 1C.

The multi-session reduced-rank model, by sharing temporal basis across sessions covering diverse brain regions, assumes uniform spiking activation patterns across regions. However, different brain regions may activate at varying time steps within a trial due to functional differences [15, 16, 17]. For instance, sensory-related areas might activate earlier than cognition-related areas. To capture potential differences in temporal activation across brain regions while still enjoying the benefits of a low-rank model that combines information across multiple sessions, we propose a **multi-region reduced rank model**, decomposing the across-session temporal basis V into two low-rank matrices, allowing flexible temporal bases for different regions indexed by j:

$$d^{ij} = f({X^{ij}}^\top (U^i V^j) + b^i), \quad V^j = {A^j}^\top B, \tag{3}$$

where $X^{ij} \in \mathbb{R}^{N^{ij} \times T}$ represents the neural activity from region j in session i, and $d^{ij} \in \mathbb{R}$ is the behavior decoded from X^{ij}. Intuitively, $A^j \in \mathbb{R}^{L \times R}$ captures regional differences, allowing varying timescales across brain regions. $B \in \mathbb{R}^{L \times T}$ represents shared similarities across regions, capturing major temporal variations associated with the behavior. In this context, L represents the rank of both the region-specific temporal basis set A^j and the global temporal basis set B. For $y^{ij} \in \mathbb{R}$, fitting a multi-session reduced-rank model (Eq 2) on J brain regions from I sessions learns $R \times T$ parameters for the temporal basis set V. In contrast, fitting a multi-region reduced-rank model (Eq 3) on the same data slightly increases the temporal basis set parameters to $L \times (J \times R + T)$. We typically select $L, R < 10$ based on empirical studies. This approach allows for unique temporal basis sets to flexibly accommodate each brain region.

3 Formulation of the behavioral data-sharing model

In neuroscience experiments, animal behaviors often display trial-to-trial correlations. We can leverage these correlations to improve upon traditional single-trial decoders. For example, when neural signals are insufficient to obtain adequate decoding performance in a given trial or session, the decoder can potentially improve by incorporating information from adjacent trials or other sessions.

For traditional decoders, we use neural activity X_k in trial k to make predictions about the true behavior y_k, and obtain a decoder estimate d_k. The index k emphasizes that X_k, y_k and d_k are single-trial quantities, with X_k corresponding to X in Eq 1 and X^i in Eq 2, and d_k corresponding to d in Eq 1 and d^i in Eq 2. (We focus on per-trial decoded scalar quantities $d_k \in \mathbb{R}$ in this section, but this can be generalized.) Our goal is to improve the quality of d_k produced by the baseline decoder, which generates each d_k independently without information from other trials. We propose an approach to improve d_k by exploiting trial-to-trial correlations in d_k across all trials, and the statistical structure present in multiple sessions. Our method assumes that observations d_k are generated from latent variables z_k representing the unknown behavior, which follow a latent dynamic process. For continuous-valued behavior (e.g., an animal's prior belief about stimulus side probability [18]), we model the transitions of z_k between trials using a first-order autoregressive process. Here, z_k in the current trial depends on z_{k-1} from the previous trial, while the continuous-valued $d_k \in \mathbb{R}$ linearly depends on the latent z_k in the same trial. This is a **linear Gaussian autoregressive model of order 1 (LG-AR1)**. Given the sequence of decoder estimates $\vec{d} = (d_1, d_2, \ldots, d_k)$, we can infer the latent variable z_k via standard Kalman smoothing forward-backward inference [19]. This inferred z_k serves as an improved decoder estimate, potentially closer to the true behavior y_k than the single-trial estimate d_k, by incorporating information from neighboring trials and other sessions. For the data generating mechanism, see Fig 1 D-E and "LG-AR1: Model details" in Methods.

While the LG-AR1 / Kalman smoother can provide improved estimates of continuous-valued y_k from noisy single-trial decoder estimates d_k, this model is not applicable to binary-valued $y_k \in \{0, 1\}$, such as an animal's choice in IBL's experimental setup [11, 12]. In the IBL experiments, mice indicate the location of a visual stimulus by rotating a wheel. The stimulus appears randomly on either side with equal probability for the first 90 trials, then predominantly on one side (left or right) over blocks of subsequent trials. This setup creates a three-level data generating mechanism: (1) The animal forms an internal belief about the stimulus-generating behavioral state (s_k); (2) Different choices (z_k) are made based on the animal's perceived state; (3) The decoder estimate d_k is generated depending on z_k. This hierarchical structure requires a different modeling approach than LG-AR1.

For binary $y_k \in \{0, 1\}$, the output from single-trial decoder $d_k \in [0, 1]$ represents the probability of $y_k = 1$. Our method assumes that d_k is generated from a mixture of beta distributions, with the mixture assignment dependent on the latent variable z_k. When the single-trial decoder accurately predicts the behavior from neural signals, we expect well-separated beta mixture components. Specifically, d_k should be distributed close to 1 when $z_k = 1$ correctly predicts the true $y_k = 1$, and close to 0 when $z_k = 0$ correctly predicts the true $y_k = 0$. Conversely, if the decoder struggles due to insufficient neural information, the two beta distributions in the mixture become less distinguishable. We further assume that the latent variable z_k depends on latent behavioral states s_k, whose transitions are governed by a hidden Markov model with H discrete hidden states. For instance, in the binary choice task, at least three hidden states exist:

random switching (stimulus appears randomly), left-biased, and right-biased (stimulus predominantly appears on one side). The likelihood of z_k being 0 or 1 varies with the latent state, defined by emission probabilities. We term this model the "**beta mixture model hidden Markov model (BMM-HMM)**". Given the sequence of decoder estimates $\vec{d} = (d_1, d_2, \ldots, d_k)$, we infer both s_k and z_k. The inferred z_k serves as an improved decoder estimate, potentially closer to the true behavior y_k than the original d_k, by incorporating information from neighboring trials and other sessions. For the data generating mechanism, see Fig 1 F-G and "BMM-HMM: Model details" in Methods.

Single-session LG-AR1 and BMM-HMM models may yield inaccurate parameter estimates when neural signals in the target session are insufficient, leading to unreliable single-trial decoder estimates $\vec{d}$. To address this, we develop multi-session versions of these models that leverage shared statistical structure across sessions to improve parameter estimation. Our multi-session approach learns empirical prior distributions of model parameters using observable behaviors from training sessions, and applies these priors to constrain model parameter updates during inference on the target test session. This method, grounded in empirical Bayes techniques, [20, 21, 22], pools data more effectively to constrain model parameters and improve characterization of underlying dynamics [23, 24]. For details on prior distribution selection and implementation, refer to "BMM-HMM: Model details" and "LG-AR1: Model details" in Methods.

4 Results

We apply the new decoders described above to 433 sessions in the IBL datasets [11], covering 270 brain regions and 5 behavioral signals: choice, prior, wheel speed, motion energy, and pupil diameter, which we describe in detail below. While our experiments use IBL data, the proposed approaches should be applicable to all settings where neural activity exhibits similar temporal patterns during the same behavioral task, and behaviors show trial-to-trial correlations across sessions.

In the IBL experiments, mice rotate a wheel to indicate the location of a visual stimulus, which is considered their *choice* (Fig 1A). For the first 90 trials, the stimulus appears randomly on either the left or right side of the screen with equal probability. In the subsequent trials, the stimulus appears predominantly on one side (either left or right) over blocks of trials [11, 12]. The mice are learning and adapting their behavior based on the changing probabilities in the experiment. This adaptive behavior allows us to estimate each mouse's "prior belief" (*prior*) about the probability of where the stimulus appears per trial. The prior we consider is not the actual probability of stimulus occurrence. Instead, it represents an estimate of this probability for the current trial, based on the mouse's behavior; see [18] for details. *Wheel speed*, *motion energy* near the whisker pad, and *pupil diameter* are also recorded. Motion energy is quantified by computing the mean absolute difference between adjacent video frames in the whisker pad area [25], defined using a bounding box anchored between the nose tip and the eye, identified using DeepLabCut (DLC) [26]. Pupil diameter is extracted from the videos using Lightning Pose [27]. Choice and prior are static within a trial, while wheel speed, motion energy, and pupil diameter are time-varying signals sampled at 60 Hz. Details about data processing, baseline decoders, and evaluation procedures are described in "Data details" and "Hyperparameter selection" in the Methods section.

4.1 Learning behaviorally relevant neural variations across sessions

The reduced-rank model improves decoding performance by capturing behaviorally relevant neural variations in a low-rank subspace. In binary decoding tasks, it projects neural activity onto this subspace, effectively separating variations based on the behavior of interest. Unlike principal component analysis (PCA) [30], which may capture both task-relevant and -irrelevant variations [31, 32], the reduced-rank model focuses on variations that are most informative for decoding the target behavior [33]. (See the Methods section "Differences between RRM, PCA, CCA, and demixed PCA" for a discussion comparing PCA and the reduced-rank model.) Fig 2A shows how neural projections related to different behavior classes are separated in the low-rank subspace identified by the multi-session reduced-rank model but remain intertwined in the PCA subspace. We quantify the distinction between projections in left and right trials using K-means clustering. The resulting cluster assignments are then evaluated using the adjusted Rand index (ARI) [28, 29]. A higher score on this index indicates greater separation between

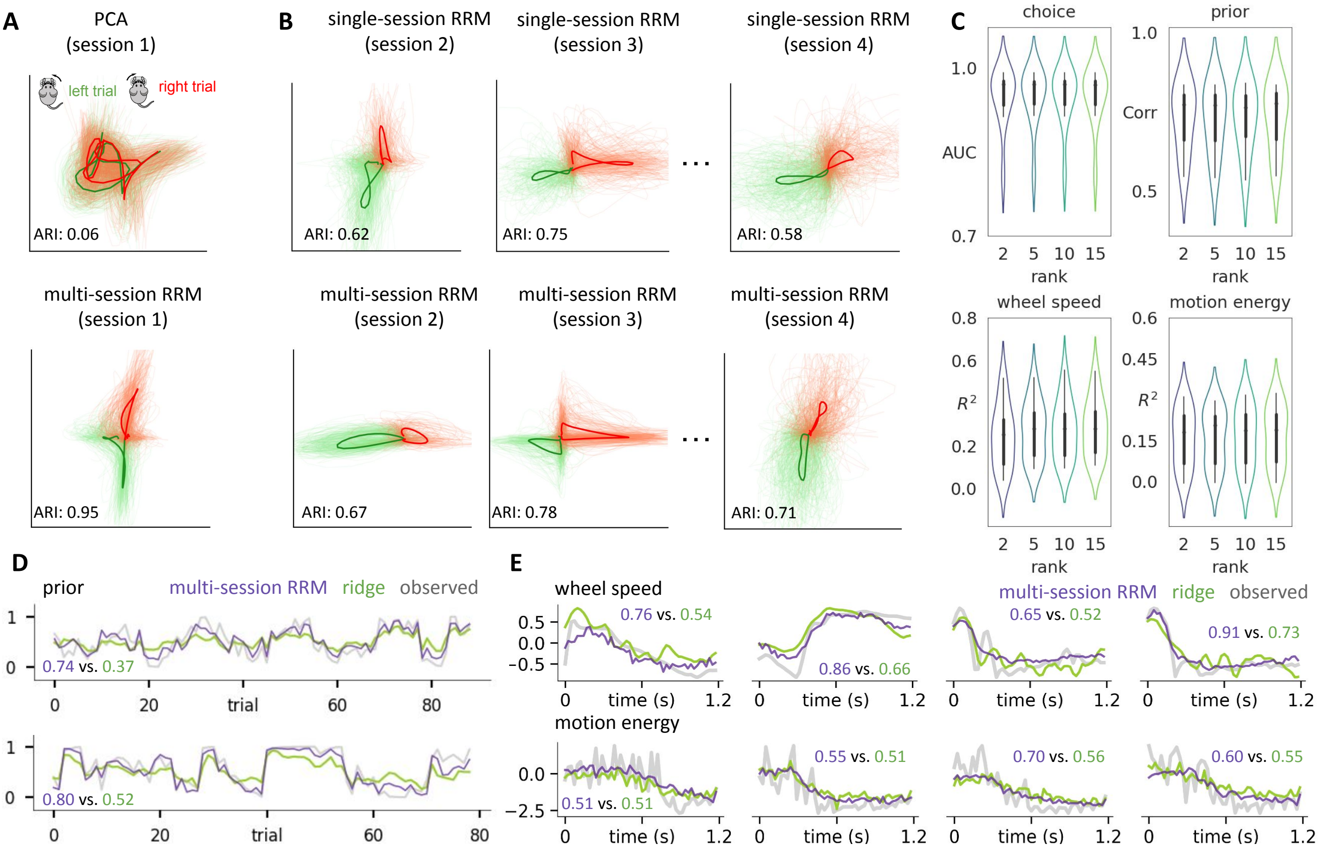

Figure 2: **The reduced-rank model achieves strong decoding performance by learning behaviorally relevant neural variations through multi-session learning.** **(A)** Projections of neural activity on the PCA subspace and the low-rank subspace identified by U^i from the multi-session reduced-rank model (RRM) are color-coded based on the binary behavioral variable. Light curves show single-trial projections from a single session, while dark curves represent trial-averaged projections. K-means clustering (2 clusters) is applied to the projections to separate left and right trials. Cluster similarity is assessed using the adjusted Rand index (ARI) [28, 29], where a higher score indicates better separation. Visualizations of the temporal basis V are depicted in Figure 9A. **(B)** Neural activity projections onto the low-rank subspace identified by the single-session and multi-session reduced-rank models, following the same color-coding convention as in panel A. K-means clustering is used to cluster the projections into left and right trials, and ARI measures cluster separation. **(C)** All behaviors are well-predicted when using a low-rank reduced rank model; however, wheel speed shows improvement with higher rank. AUC (Area Under the Curve), Pearson's correlation, and R^2 are used to evaluate decoding performance for choice, prior, and dynamic behaviors, respectively. AUC is a metric for binary classifiers with values ranging from 0 to 1, where 1 indicates a perfect classifier and 0.5 represents random guessing. **(D)** Decoded prior from the multi-session reduced-rank model (purple) vs. ridge regression decoder (green), with Pearson's correlation between the decoded and true prior shown as a numeric value for each example. The true prior (observed) is shown in grey. **(E)** Decoded motion energy and wheel speed traces from the reduced-rank model vs. ridge regression, with R^2 values shown in purple (reduced-rank) and green (ridge) for each example. The true behavior traces (observed) are shown in grey.

the clusters. Moreover, multi-session training allows the model to learn fewer parameters with more data and draw upon information from other sessions when the neural signals from a particular session lack information about the behaviors, thereby improving decoding performance. This results in less noisy parameter estimates and learned neural representations that better capture behavioral variations. Fig 2B shows that multi-session reduced-rank model leads to more separated neural representations compared to single-session reduced-rank model.

Fig 2C shows a sensitivity analysis examining the effect of the reduced-rank model's rank on decoding quality. Both static behaviors (choice and prior) and dynamic behaviors (wheel speed and motion energy) achieve good performance with a small rank, after which performance plateaus. In addition, Fig 2D and E demonstrate that our model's decoded behavior traces align more closely with the observed behavior traces than the baseline ridge regression decoder. We evaluate the reduced-rank model's performance in decoding continuous behaviors using two criteria: (1) predicting behavior averaged across trials under various stimulus conditions, and (2) capturing individual trial behavioral differences after subtracting the trial-averaged behavior. We also examine residual behavior (the difference between observed and predicted behavior) to identify any systematic errors. Fig 3, S2 and S3 illustrate the model's decoding

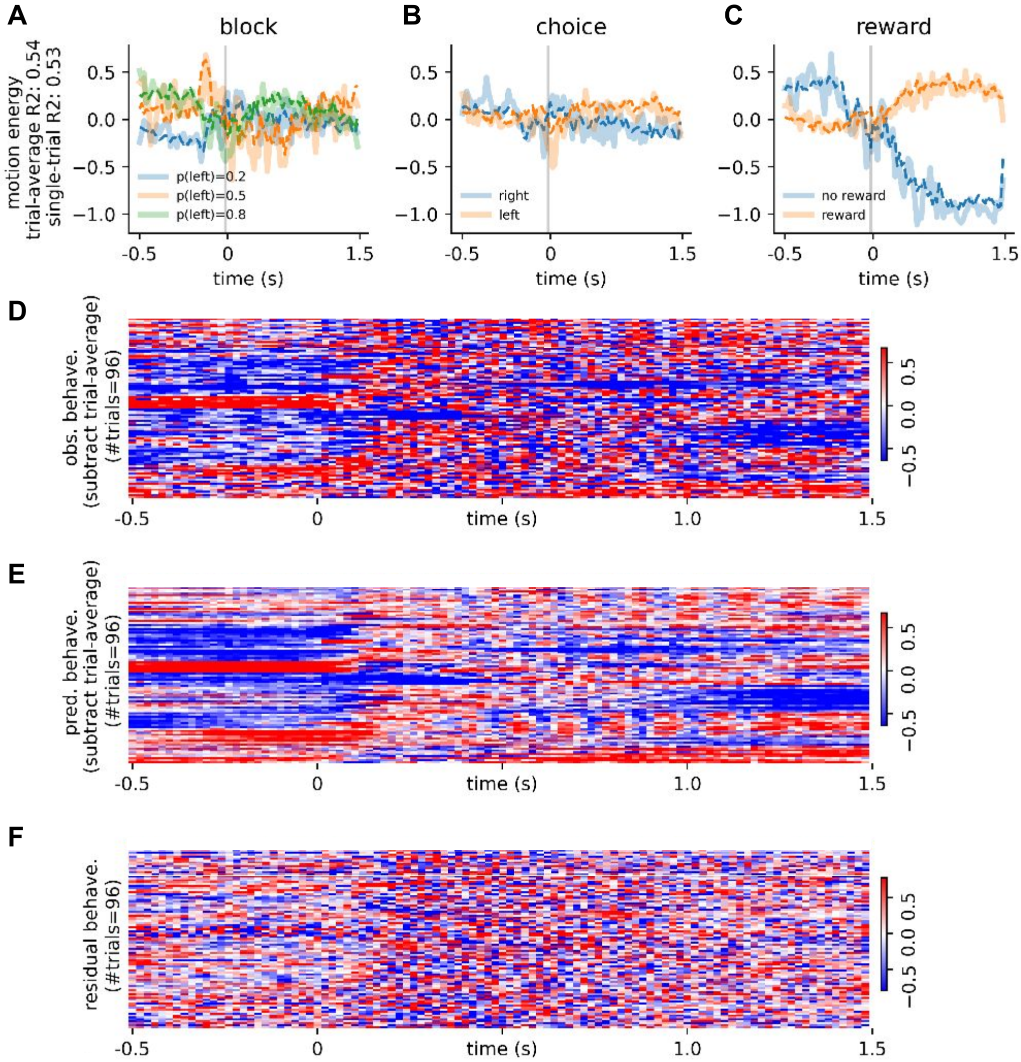

Figure 3: **Evaluating motion energy decoding quality using spiking activity from 1313 neurons in a RE dataset session.** **(A)** Comparison between the reduced-rank model's predicted motion energy (dotted curves) and observed ground truth behavior (solid curves) across different block conditions. For example, blue curves represent average predicted (dotted) and observed (solid) behavior for trials with a block value of 0.2. The grey vertical line denotes stimulus onset. **(B)** The predicted and observed whisker motion energy averaged across trials based on choice conditions (right and left). **(C)** Similar comparison based on reward outcomes (reward and no reward). **(D-F)** illustrate data from individual experimental trials in this session. Panel D displays observed behavior, panel E shows predicted behavior from the reduced-rank model, and panel F shows residual behavior (the difference between observed and predicted behavior). In each panel, the raster plot's rows depict behavior over time for individual trials, while columns represent timesteps within a trial. To emphasize trial-to-trial variations, we center both observed and predicted behaviors by subtracting their respective trial averages. For visualization purposes, we standardize the observed behaviors, predicted behaviors, and residuals. We also apply spectral clustering to the observed behavior, which groups trials exhibiting similar behavioral patterns, allowing for easier interpretation of the results.

performance for motion energy, wheel speed, and pupil diameter respectively. Panels A, B, and C compare the model's predictions to the observed behavior averaged across trials under different stimulus conditions (e.g., left vs. right choice). The decoder accurately predicts trial-averaged wheel speed and motion energy under different conditions, but is less accurate for pupil diameter. Panels D and E evaluate the decoder's ability to capture individual trial differences after subtracting the trial-average of predicted or observed behavior. Again, performance is better for wheel speed and motion energy than for pupil diameter. Panel F shows the residual behavior. In an ideal scenario, this should display small, random fluctuations without noticeable patterns, indicating accurate prediction of observed behaviors. Our analysis reveals that the decoder performs best in predicting motion energy, while systematic residual errors remain for wheel speed and pupil diameter.

4.2 Learning latent behavioral dynamics across trials

Next we turn to the behavioral data-sharing model. This model learns latent behavioral states $\vec{s}$ that infer the unknown behavior $\vec{z}$ (Eq 12) given the neural activity X, leveraging the correlation between trials in the same state to improve single-session and single-trial decoder outputs $\vec{d}$. Fig 4A shows the latent state inference of a multi-session BMM-HMM applied to the IBL binary decision behavior [11]. Recall that the stimulus probability switches between three discrete states: 1) a right (R) state (stimulus predominantly on the right), 2) a left (L) state (stimulus mostly on the left), and 3) a "middle" (M) state (stimulus randomly switching sides). Note that the three stimulus-generating states discussed here are different from the three decision-making states ("engaged", "disengaged" and "biased") in [10]. The model accurately infers the occurrence of the three discrete states using only single-trial decoder outputs, without prior knowledge of the true choices or the timing of the stimulus probability block state changes. (Fig 4B). Note that we only use neural data from the decoded session to learn the model (the behavior in that session is unobserved). However, we do use observed behavior from other sessions to learn the multi-session model.

Ideally, when the single-trial decoder accurately predicts behavior, the model can more precisely infer the states. Conversely, when the single-trial decoder makes errors, the model can compensate by borrowing decoder outputs from other trials (trial-to-trial correlation) and behavioral patterns from other sessions to refine its state estimation. Fig 4C visually compares the improved decoder outputs (Eq 12), from the multi-trial and multi-session BMM-HMM to the baseline single-trial and single-session decoder outputs. The single-trial and single-session decoder outputs exhibit considerable noise and frequent errors, while the multi-trial and multi-session outputs better follow the smooth "block" structure due to their knowledge of the latent states in the data. Quantitatively, the proposed model achieves a higher AUC (area under the ROC curve) than the baseline, highlighting the effectiveness of using trial-to-trial correlations and latent states to improve decoding. The decoder performance segregated by block type is also shown in Fig 4C. The decoding AUC of the baseline single-trial decoder is shown in black, while that of the BMM-HMM is shown in purple.

Next we apply similar ideas to improve the decoding of the continuous-valued *prior*, using the multi-session LG-AR1 model. Recall that this prior signal represents a running estimate of the stimulus side probability [18]. Similar to the BMM-HMM model, the LG-AR1 model infers the latent behavior (the true yet unobserved prior) by exploiting trial-to-trial correlations in the single-trial decoder outputs, borrowing behavioral information from other sessions to correct estimates when decoding errors occur. Fig 4D visually compares the improved decoder outputs, $\tilde{d}_k$ (Eq 42), from the multi-trial and multi-session LG-AR1 to the baseline single-trial and single-session decoder outputs, d_k. The single-session baseline decoder struggles to accurately predict the prior, as it doesn't incorporate information from previous trials. In contrast, the LG-AR1 model, by considering trial-to-trial correlations, produces outputs that more closely align with the true prior, resulting in a higher Pearson's correlation. This improved performance reflects the model's ability to capture the mice's prior beliefs, which are based on past experiences [18].

Next we evaluate the impact of incorporating behavioral information from other sessions on the performance of the BMM-HMM and LG-AR1 models. We explore three model variants: a *single-session model*, a *multi-session model*, and an *oracle model* that uses true behaviors to learn parameters and improve decoder estimates (see Methods for details). The oracle models assume that the true values of the latent behavioral variable z_k are known a priori. In this scenario, rather than inferring the latent behaviors, we directly substitute the ground truth observed behaviors y_k for z_k, effectively treating z_k as a known quantity. However, the oracle models cannot simply use the observed y_k as the final improved decoder output, as this would result in a trivial decoding problem. Instead, these models must still generate a distinct output given the known z_k values and the learned model parameters. Thus the oracle model serves as an upper bound to assess the performance of single-session versus multi-session models. Fig 4E and F compare the estimated parameters of BMM-HMM and LG-AR1 from the three variants, showing that parameters estimated by the multi-session model align more closely with the oracle estimates than those from the single-session model. In addition, Fig 4G and H compare the outputs of the model variants, suggesting that predictions from the multi-session model are closer to the oracle model predictions than those from the single-session model. These findings underscore the importance of multi-session learning in improving both parameter estimation and decoding performance.

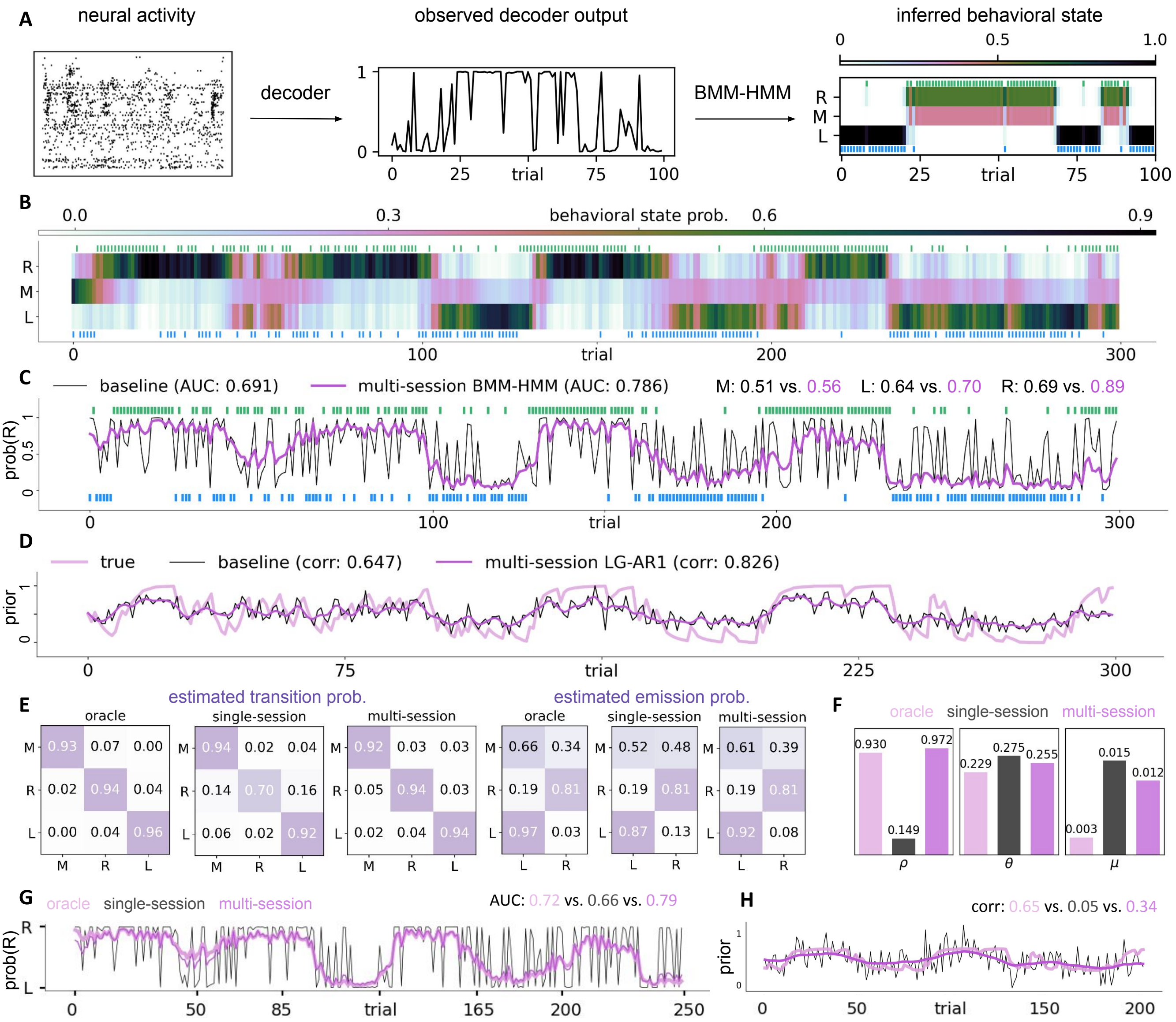

Figure 4: **The behavioral data-sharing model improves single-trial decoding by inferring latent behavioral states from trial-to-trial correlations within individual sesssions, and sharing behavioral information across sessions. (A)** A schematic showing the BMM-HMM's latent state inference from neural activity. A decoder is fitted to single-session, single-trial activity X_k, yielding decoder output d_k. The BMM-HMM is fitted to d_k to infer latent states s_k, which alternate between left (L), right (R), and a random "middle" switching state (M), producing an improved decoder output. **(B)** The latent states s_k estimated from neural activity exhibit "block" structures, switching between states L, R, and M; these blocks mirror the true block probabilities in the IBL task but note that these states are learned, not pre-specified, and the state names in the plot are assigned post hoc. Color bar indicates state probabilities. Observed mouse choices are shown in green (right trials) and blue (left trials). **(C)** Improved decoder outputs $P(z_k = 1 \mid \vec{d})$ from the multi-trial and multi-session BMM-HMM (purple) overlaid on baseline single-trial and single-session decoder traces d_k (black), exploiting trial-to-trial correlations and achieving higher AUC. "Multi-session" refers to borrowing behavioral information from multiple training sessions to improve neural state estimates in the test session. d_k is observed and z_k is latent. We additionally show the decoder performance for each block type: random switching (M), left-biased (L), and right-biased (R). The decoding AUC of the baseline single-trial decoder is shown in black, while that of the BMM-HMM is shown in purple. **(D)** Improved decoder outputs $\tilde{d}_k$ from the multi-trial and multi-session LG-AR1 (purple) superimposed on baseline single-trial and single-session decoder outputs d_k, aligning more closely with the true prior (pink) and achieving higher Pearson's correlation. **(E)** Estimated transition and emission probabilities from the oracle (pink), single-session (black), and multi-session (purple) BMM-HMM models. **(F)** Parameter estimates from the oracle, single-session, and multi-session LG-AR1 models. **(G)** Decoded probabilities of choosing the right side from the oracle (pink), single-session (black), and multi-session (purple) BMM-HMM models. **(H)** Decoded priors from the oracle (pink), single-session (black), and multi-session (purple) LG-AR1 models.

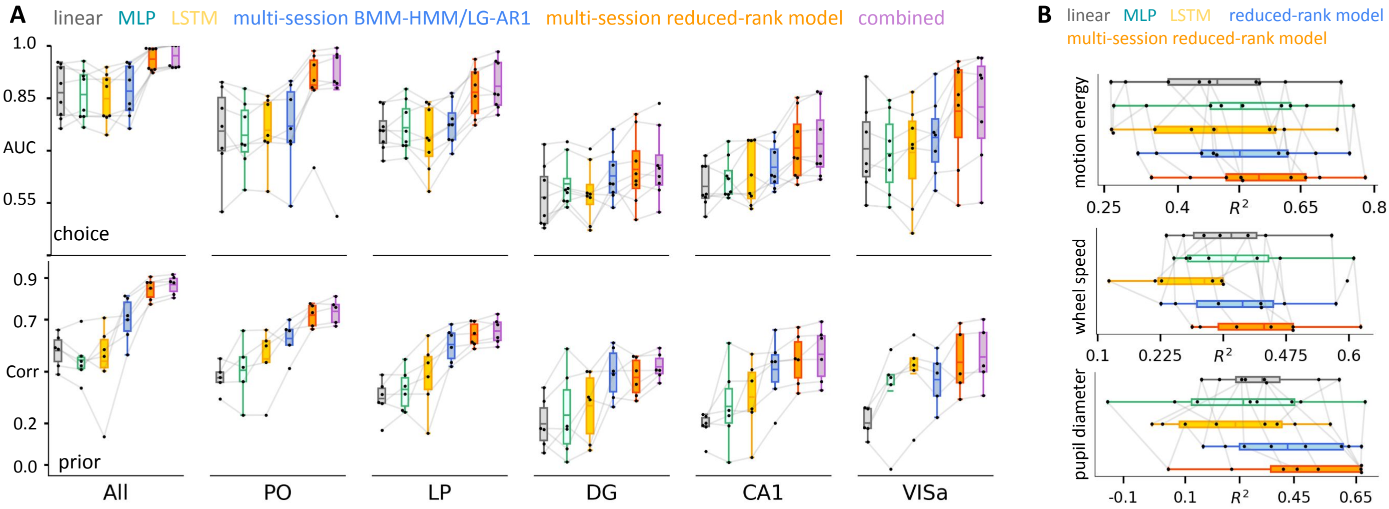

Figure 5: **Quantitative improvement in decoding accuracy achieved by the neural and behavioral data-sharing models compared to the baseline decoder. (A)** Cross-validated decoding AUC (Pearson's correlation) for decoding choice (prior) using spikes from all brain areas across 10 sessions, focusing on 5 selected regions. Box plots show the min, max, first and third quartiles, and mean of the metrics. Each point is one session, with colors differentiating decoders. The multi-session reduced-rank model is defined in Eq 2. The "combined" decoder involves a two-step process: first, initial decoder estimates are derived from the multi-session reduced-rank model; these estimates are then refined using the multi-session BMM-HMM or LG-AR1 model. **(B)** Cross-validated decoding R^2 for decoding dynamic behaviors using spikes from all brain areas across 10 sessions. Box plots show the min, max, first and third quartiles, and mean R^2. Each point represents one session, and colors differentiate the decoders. The "combined" decoder is not implemented in this case, as the multi-session BMM-HMM and LG-AR1 model do not currently apply to vector-valued dynamic behaviors.

4.3 Increasing information decoded from various brain regions

To evaluate our proposed multi-session decoders, we compare them to baseline single-session decoders: L2-regularized linear decoders, nonlinear neural networks (MLPs), and long short-term memory (LSTM) decoders [2, 34]. Hyperparameter selection and model architecture details are in the Methods section. A common approach to reduce the number of model parameters is using a temporal convolutional model, which fits one temporal filter and slides it against the input neural activity for each of the P timesteps. This contrasts with the reduced-rank model (Eq 1), which fits a separate $R \times T$ dimensional temporal basis for each timestep. We implemented the temporal convolutional model as a baseline in a pilot study. However, this model did not outperform ridge regression, and therefore we only used ridge regression as the linear baseline for decoding continuous behaviors in our remaining analyses.

We benchmark all the methods in decoding choice, prior, wheel speed, motion energy and pupil diameter using spikes from all brain regions in the brain-wide map (BWM) dataset [12] and also 5 selected areas in the reproducible electrophysiology (RE) datasets [11]: the posterior thalamic nucleus (*PO*), the lateral posterior nucleus (*LP*), the dentate gyrus (*DG*), the cornu ammonis (*CA1*), and the anterior visual area of the visual cortex (*VISa*). We focus on RE regions due to their large number of recorded cells and use a per-region evaluation scheme to avoid the ceiling effect that may occur when using all regions for decoding (e.g., all decoders achieving an AUC near the "ceiling" AUC = 1), which can hinder decoder performance comparison. The selected areas, distributed across the brain, likely contain less information per area than all regions combined, resulting in lower expected decoding accuracy compared to using all regions. The multi-region reduced-rank model (Eq 3) improves region-wise decoding in some areas (Fig 7 and 9), but requires the input matrix X^{ij} to contain spiking activity from neurons in region j from session i, allowing a region-specific temporal basis V^j. When decoding from all brain regions (X^i), V^j becomes shared across sessions regardless of region, reducing to V in the multi-session reduced-rank model (Eq 2). Therefore, we exclude the multi-region model as a baseline here, discussing it only in the subsections "The benefit of training with more data" and "Mapping behaviorally-relevant timescales across the brain."

For static behaviors, Fig 5A shows that the multi-session reduced-rank model consistently outperforms the baseline decoders in decoding choice and prior, while the multi-session state-space model outperforms baselines in most cases. The proposed models consistently outperform the single-session linear decoder and frequently outperform single-session MLP and LSTM decoders. Despite hyperparameter tuning, the MLP and LSTM may not have reached optimal performance, highlighting the advantage of our models

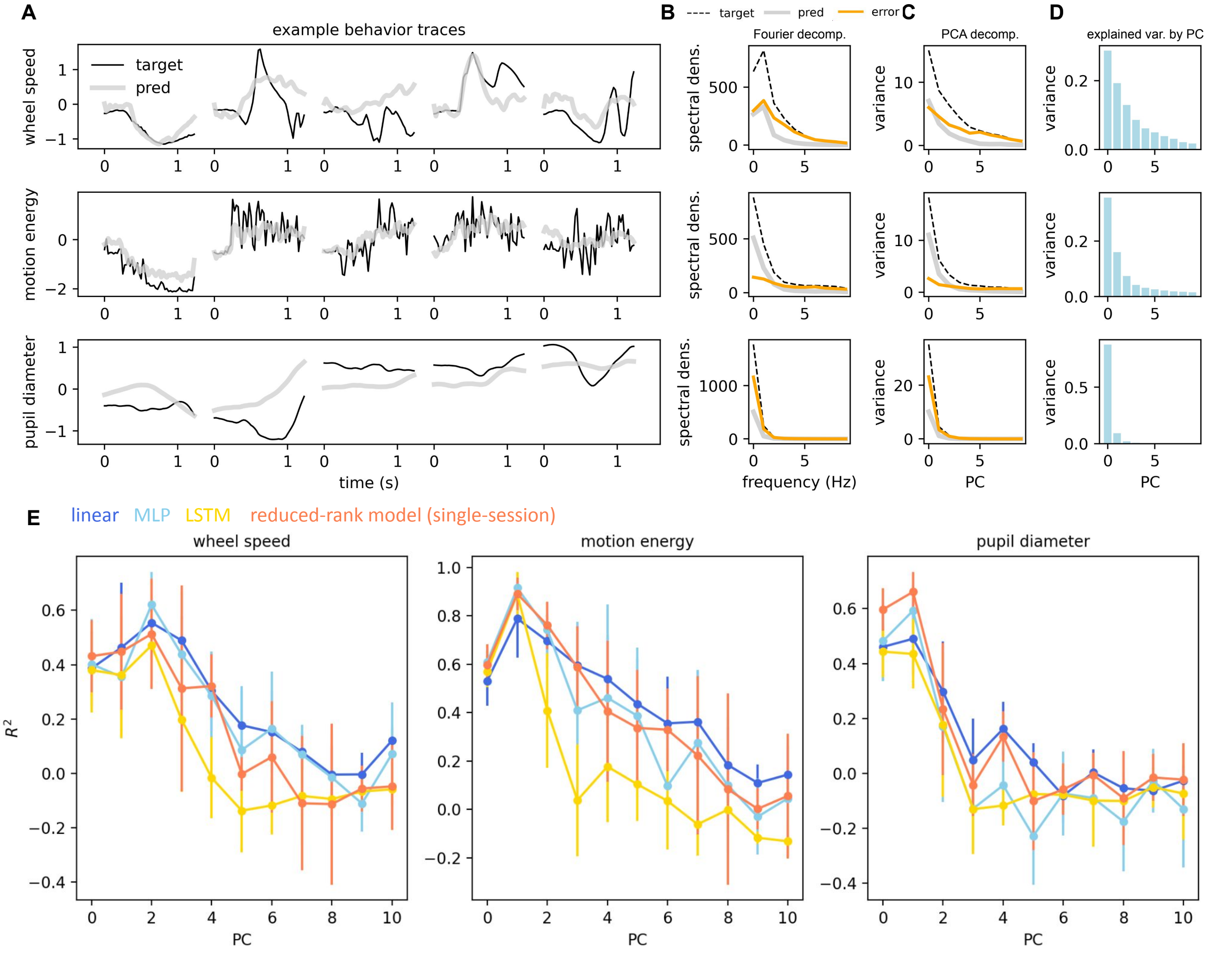

Figure 6: **Evaluating the reduced-rank model against baseline decoders in capturing the primary components of the target behaviors.** **(A)** Examples of real ("target") and predicted ("pred") behaviors from the reduced-rank model in 5 selected trials. Motion energy has higher frequency than other behaviors, while pupil diameter has lower frequency. **(B)** Power spectral density vs. frequency for real behaviors ("target"), predicted behaviors ("pred") from the reduced-rank model, and decoding error ("error" = real − predicted). Results are averaged across 10 sessions. **(C)** Variance of real behaviors ("target"), predicted behaviors ("pred"), and decoding error ("error") vs. principal component (PC). The initial PCs, corresponding roughly to low-frequency Fourier components, capture the majority of the behavioral variations. Results are averaged across 10 sessions. **(D)** Explained variance ratio of the real behaviors by the first 10 PCs (again averaged across 10 sessions). Explained variance ratio is the percentage of the total variance in the original behavior explained by each PC. **(E)** Decoding quality (R^2) of behaviors reconstructed from each PC of the real behavior for all baseline decoders. Decoders generally perform better at decoding the initial PCs linked to low-frequency Fourier components. Mean and standard deviation of decoding R^2 across 10 sessions are shown for the first 10 PCs.

which have fewer parameters, making it easier to thoroughly explore the model space. Note that multi-session BMM-HMM/LG-AR1 performs worse than multi-session reduced-rank model, because multi-session BMM-HMM/LG-AR1 improves the outputs from the single-session and single-trial decoder. Although the multi-session reduced-rank and BMM-HMM/LG-AR1 models in Fig 5A are fitted independently, they can be combined for decoding. The multi-session reduced-rank model provides initial decoder estimates, which are then refined using the multi-session state-space model's smoothing. The performance of this "combined" decoder is shown in Fig 5A. However, combining both models only leads to marginal improvement over the best-performing multi-session reduced-rank model.

For dynamic behaviors, we compare the reduced-rank model to baselines in decoding wheel speed, motion energy, and pupil diameter. Fig 5B shows that the single-session reduced-rank decoder outperforms the linear decoder, with similar performance to the MLP and LSTM decoders. However, the multi-session reduced-rank model outperforms all single-session models. These results highlight the importance of prioritizing behaviorally relevant neural variations and training with more data for improving decoding performance.

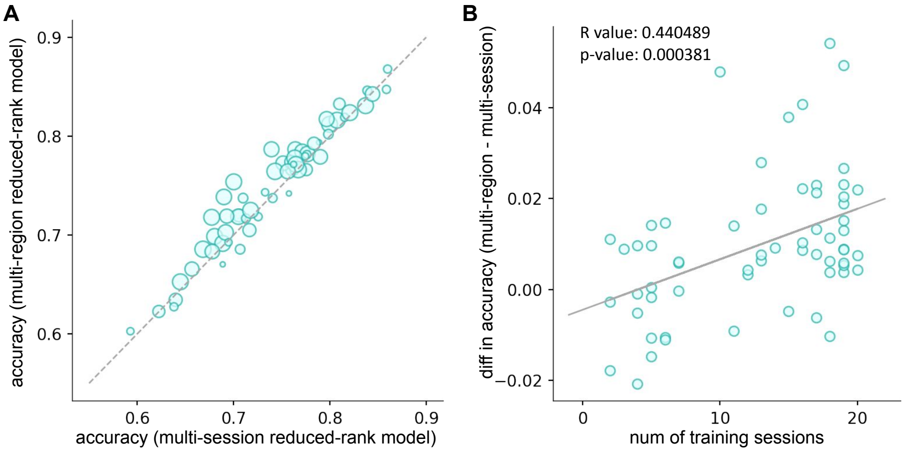

Figure 7: **Comparison of the multi-region vs. multi-session reduced-rank models and a scaling curve showing the improvement in decoding accuracy vs. training data size.** **(A)** A scatter plot comparing the multi-region (Eq 3) and multi-session reduced-rank models (Eq 2) in decoding choice using neural activity from a specific brain region. Each point shows the 5-fold cross-validated accuracy per region, averaged across sessions. Point size is proportional to the number of training sessions used in the multi-region model. **(B)** A scatter plot showing the difference in decoding accuracy between the models vs. the number of training sessions. Each point shows the accuracy difference per region, averaged across sessions. A linear regression fitted to the data demonstrates a positive relationship between training data size and model performance improvement. The correlation coefficient (R value) and its p-value are shown.

4.4 Decoding frequency components of behavior

Fig 6A illustrates that motion energy has higher frequency components than the smoother wheel speed and pupil diameter. Although a decoder may not accurately decode the entire behavior, it could still effectively capture slower variations in the behavior. We analyze which frequency bands of each behavior are captured by our decoders, and compare the performance of different decoders in capturing different behavioral components.

To quantify the fraction of behavior reconstructed at each frequency, we calculate the power spectral density of the real behavior, predicted behavior, and prediction error, following the approach in [35]. Fig 6B shows that the spectral density for both real and predicted behaviors, as well as the prediction error, diminishes sharply at higher frequencies. Beyond 5 Hz, the decoder extracts no information about the behavior, suggesting that lower frequency components capture the major variations and the decoder primarily extracts information from these frequencies.

We also perform PCA on the real behavior and project the real behavior, predicted behavior, and prediction error onto the obtained principal components (PCs). Fig 6C shows the variance of these projections across the first 10 PCs, while Fig 6D shows the variance in the real behavior explained by each PC. The results indicate that the first few PCs capture the major variations in the real behavior, with the decoder predominantly extracting information from these PCs. These PCs likely represent low-frequency components that capture slow behavioral variations.

To determine if the baseline decoders capture both slow and fast behavioral variations, we extract the first 10 PCs of the real behavior, and reconstruct the behavior using each of the 10 PCs. We then train each decoder to decode the reconstructed behavior from each PC. Fig 6E shows the decoding R^2 per PC for all baseline decoders. In decoding low-frequency components, most decoders, except LSTM, show comparable performance, and the reduced-rank model slightly outperforms other baselines in decoding pupil diameter. Effective decoding is mainly achieved at lower frequencies.

4.5 The benefit of training with more data

Are our models sufficiently flexible to demonstrate improved performance as the training set size increases? To analyze this question, we compare the multi-region reduced-rank model in Eq 3, which uses 433 sessions across 270 brain regions to predict choice per region, with the multi-session reduced-rank model in Eq 2, trained for each region with around 20 sessions. Fig 7A shows that the multi-region model outperforms the multi-session model in choice decoding across many regions. Although the multi-region model's global temporal basis B (Eq 3) is learned using all 433 sessions, the region-specific basis $V^{(j)}$ (Eq 3) is learned

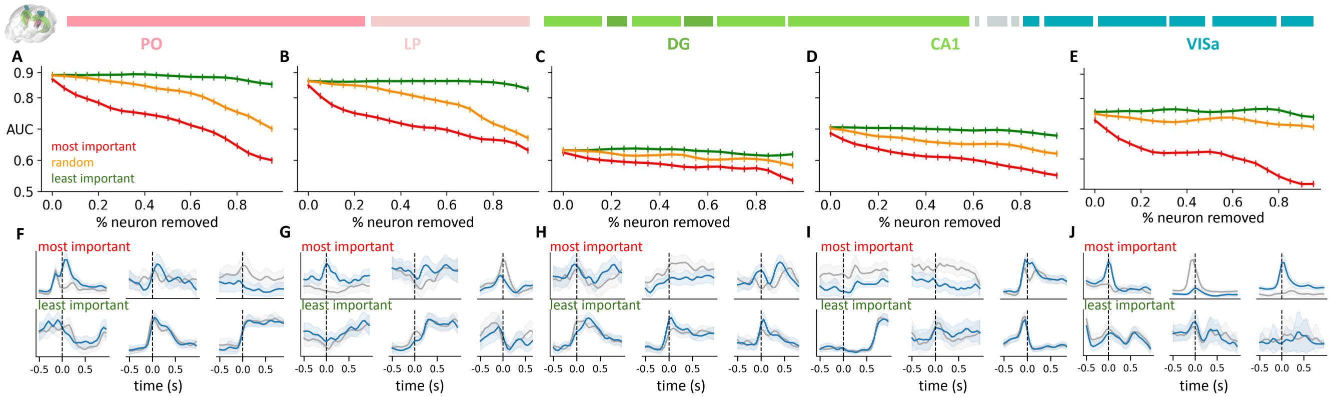

Figure 8: **Reduced-rank models identify important neurons for decoding choice in brain regions including PO, LP, DG, CA1, and VISa.** **(A-E)** Region-specific performance degradation from the "neuron pruning" experiment using three neuron removal strategies. Decoding accuracy is quantified by AUC and averaged across 10 sessions from each region. **(F-J)** Trial-averaged neural activities conditioned on choice for the most and least important choice-decoding neurons from example sessions in each brain region. Blue and black solid curves show the mean spiking patterns for left and right trials, respectively, with light-colored ribbons indicating one standard deviation. Stimulus onset is indicated by a dashed line.

using sessions from the given region. Fig 7B visualizes the relationship between decoding accuracy and training data size, comparing the difference in accuracy between models against the number of training sessions per region. Additionally, a linear regression fitted to the data illustrates a positive correlation between training data size and model performance improvement.

4.6 Identifying important neurons for decoding

The reduced-rank model not only improves decoding outcomes but also offers intrinsic interpretability. In this section, we show that the neural basis set U quantifies individual neurons' contribution to behavior decoding (see Eq 10 for theoretical justification). We validate this claim through a "neuron pruning" experiment, where the magnitude of U's first rank indicates neuron importance, with larger values signifying higher importance. Starting with all neurons, we iteratively remove 5% of neurons from each session. After each removal, we fit a L2-regularized logistic regression to the remaining neurons' activities and track the decrease in decoding accuracy measured by AUC. We compare three removal strategies: removing the least important neurons first, removing the most important neurons first, and removing randomly selected neurons. Fig 8 A-E show that removing the least important neurons first minimally impacts decoding performance, while removing the most important ones leads to a faster decline in choice decoding accuracy than random removal. Moreover, accurate decoding can be achieved with only a small proportion of the important neurons (green curves in Fig 8 A-E). Fig 8 F-J show the choice-conditioned, trial-averaged activity of the most and least important neurons identified based on the reduced-rank model's U values from example sessions in each region. The most important neurons exhibit choice-selective firing patterns, while the least important neurons show similar activity in left and right trials, indicating limited task responsiveness.

4.7 Mapping behaviorally-relevant timescales across the brain

Prior studies show that functionally distinct brain regions have different intrinsic timescales [15, 16, 17], with motor and sensory areas exhibiting faster timescales than cognitive areas. However, a comprehensive investigation of temporal dynamics linked to specific behaviors is lacking. We fit the multi-region reduced-rank model on 433 sessions across 270 brain areas to perform choice and prior decoding tasks, using the first rank of the region-specific temporal basis V^j to represent each brain region's timescale. Fig 9A reveals distinct activation timescales for different brain regions in decoding choice, including the Gigantocellular Reticular Nucleus (GRN), motor cortex (MOp), nucleus accumbens (ACB), amygdala complex (CEA), CA1 region in the hippocampus, basomedial amygdala (BMA), and visual cortex (VISa). The peak activation time ("peak") corresponds to the highest point of a curve. The activation duration ("width") is defined as the interval spanning points on either side of the peak where the curve covers 90% of the peak height. While activation patterns peak around similar times after stimulus onset, ACB and BMA show longer durations than other regions.

We use the peak activation time and duration of each area (Fig 9A) to compare behaviorally relevant timescales across brain regions. Figure 9A shows that for the choice decoding task, most brain regions exhibit peak activation within 1.5 seconds of stimulus onset. This timing aligns closely with the "reaction time", defined as the interval between stimulus onset and the initial movement (Figure 1c of [12]). For the choice decoding task (visual decision-making), Fig 9B (first row) shows most regions have similar peak activation times, except the olfactory bulb and cerebellum, which may have delayed activation upon receiving the water reward. Fig 9C (first row) shows that activation durations vary, with hindbrain areas having shorter durations than forebrain and midbrain regions. For the prior decoding task (learning from past experiences), Fig 9B (bottom row) shows the cerebral cortex has earlier activation, while regions in the cerebellum have delayed activation. Fig 9C (bottom row) shows the cerebral cortex and thalamus have longer activation durations than other areas. White areas denote brain regions not decoded due to the absence of corresponding behavioral data (choice or prior) in sessions containing these regions.

In addition to showing the behaviorally-relevant timescales in each brain region to explain their responsiveness to the task, we analyze the amount of decodable behavior information from the neural activity in each region. While [12] creates a brain-wide map of decoding accuracy for selected behavior tasks, they only use L2-regularized linear decoders. In Fig 9D, we show that the multi-region reduced-rank model, a more constrained and interpretable linear decoder trained with more data, improves choice and prior decoding across most brain regions compared to the linear decoder baseline used in [12]. This suggests that regularized linear decoders may not fully capture all decision-making task information in each region, potentially influencing the interpretation of results derived from these decoders.

Finally, in Section 6.7 "Assessing statistical significance," we verify that multi-region reduced-rank model improves information decoded from each region compared to the baseline linear decoder, while controlling for spurious correlations [36] through null distributions generated from "imposter sessions" as per [12]. Analysis of representative brain regions (PO, LP, DG, CA1, and VISa) in Figure S1 reveals that while absolute decoding improvement varies slightly between original and adjusted scores, the relative ranking of regional improvements remains largely consistent.

5 Discussion

We propose a reduced-rank and multi-session state-space models to share neural and behavioral data across sessions, improving decoding performance. Applied to a large collection of sessions from various brain regions, our decoders improve multiple behavioral decoding tasks. Our interpretable approach identifies important neurons for decoding, behaviorally relevant timescales per brain area, and infers latent behavioral states from neural activity.

Several existing methods relate to our neural data-sharing model [37, 38, 39, 40, 41]. [3] uses canonical correlation analysis (CCA) to align latent dynamics across sessions, while our model substitutes the unsupervised CCA with reduced-rank regression using a supervised decoding loss. CCA maximizes neural-behavioral correlation, but reduced-rank regression minimizes the normalized mean squared error between the real and predicted behavior. Demixed PCA [33] isolates neural activity variations related to different conditions, maximizing neural-behavioral correlations and prioritizing neural variability for reconstruction. In contrast, our reduced-rank regression emphasizes behavioral variation for accurate decoding. The preferential subspace identification (PSID) [32] and targeted neural dynamical modeling (TNDM) [42] also extract low-dimensional, behaviorally relevant neural dynamics but rely on more complex state-space models. Our reduced-rank model is a latent variable model without constraints on neural dynamics. See "Differences between RRM, PCA, CCA, and demixed PCA" in Methods for a detailed comparison.

Previous studies like [43, 44] relate to our behavioral data-sharing model. [10] models mouse decision-making using HMM with generalized linear model (GLM) observations, allowing behavioral states to persist across trials and depend on the stimulus and other covariates. Unlike these methods that infer HMM states only from the behaviors, we also use neural data. While [45, 46, 47, 48, 49] apply HMMs to understand how different neural states generate the observed neural activities, we learn HMM states that generate the observed decoder estimates, which rely on both neural activity and behavior. Another related approach is that of [50], which uses a Bayesian decoder to decode continuous and discrete states of the

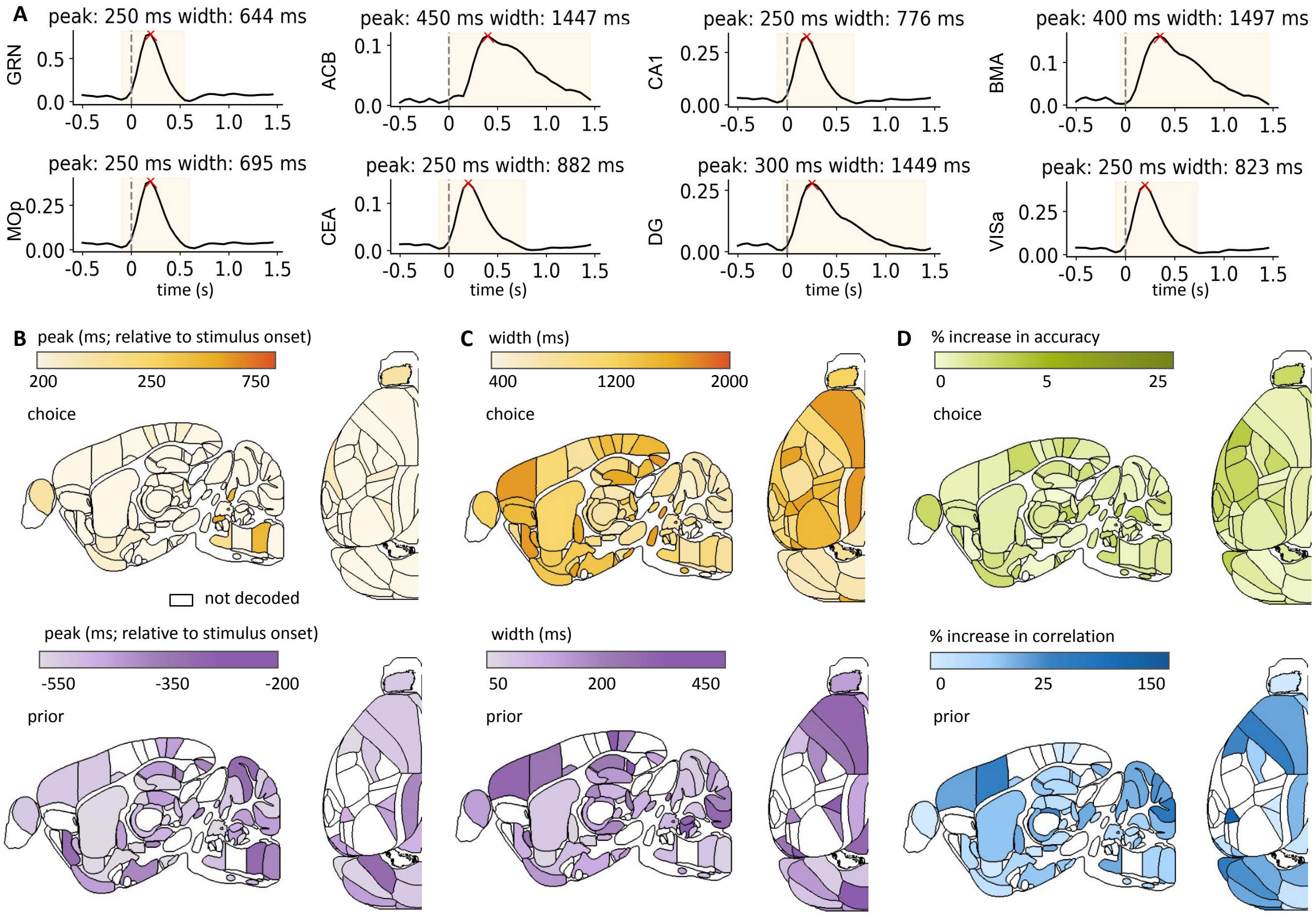

Figure 9: **Mapping behaviorally relevant timescales and decoding quality improvement across the brain. (A)** The first rank of each brain region's temporal basis V^j in the multi-region reduced-rank model (Eq 3) is shown. Stimulus onset is indicated by a dashed line, peak activation time ("peak") by a red cross, and activation duration ("width") by a yellow segment. "Peak" corresponds to the highest point of a curve. "Width" is defined as the interval spanning points on either side of the peak where the curve covers 90% of the peak height. **(B)** Brain-wide map of relative peak activation time w.r.t. stimulus onset. **(C)** Brain-wide map of activation duration (width). Colors distinguish choice (yellow) from prior (purple); intensity represents peak time and duration magnitude. White regions indicate non-decoded areas. **(D)** Region-specific improvement in choice decoding accuracy and the correlation between the real and predicted prior. The multi-region reduced-rank model's improvement is compared to the baseline L2-regularized linear decoder. Color intensity represents the magnitude of improvement.

behavioral video data, and then combine those with a behavior-based autoregressive HMM to smooth the original neural predictions.

Technological advancements now enable the simultaneous collection of multiple data modalities, like local field potentials and calcium imaging, during neuroscience experiments. Moreover, the reduced-rank model has applications beyond neural decoding, including neural encoding (predicting neural activity from behavior) and inter-region activity prediction (reconstructing activity in one brain region using data from another). Therefore, important future directions include incorporating more data modalities into the model and adapting the model to perform additional tasks. The interpretability of this approach helps understand connections between changes in neural activities, behaviors, and information flow among brain regions. For multi-session state-space models, exploring nonlinear time series models and high-order dynamical systems [34, 51, 52] can facilitate modeling more complex latent behavioral dynamics. Finally, all of our models are compatible with the density-based decoding approach from [53], allowing decoding from unsorted spike features rather than spike-sorted data; we expect that combining these approaches would lead to further accuracy improvements.

Acknowledgments

We thank Matt Whiteway for many useful discussions, Peter Latham and Jonathan Pillow for helpful comments on the manuscript, and Chris Langfield for code review. This work is supported by grants from the Wellcome Trust (209558 and 216324), National Institutes of Health (1U19NS123716 and K99MH128772), the Simons Foundation, the National Science Foundation, and by DoD OUSD (R&E) under Cooperative Agreement PHY-2229929 (The NSF AI Institute for Artificial and Natural Intelligence).

Declaration of interests

The authors declare no competing interests.

References

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

[2] Joshua I Glaser, Ari S Benjamin, Raeed H Chowdhury, Matthew G Perich, Lee E Miller, and Konrad P Kording. Machine learning for neural decoding. *Eneuro*, 7(4), 2020.

[3] Juan A Gallego, Matthew G Perich, Stephanie N Naufel, Christian Ethier, Sara A Solla, and Lee E Miller. Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature communications*, 9(1):4233, 2018.

[4] Svenja Melbaum, Eleonora Russo, David Eriksson, Artur Schneider, Daniel Durstewitz, Thomas Brox, and Ilka Diester. Conserved structures of neural activity in sensorimotor cortex of freely moving rats allow cross-subject decoding. *Nature Communications*, 13(1):7420, 2022.

[5] Arthur Pellegrino, N Alex Cayco-Gajic, and Angus Chadwick. Low tensor rank learning of neural dynamics. *arXiv preprint arXiv:2308.11567*, 2023.

[6] Srini Turaga, Lars Buesing, Adam M Packer, Henry Dalgleish, Noah Pettit, Michael Hausser, and Jakob H Macke. Inferring neural population dynamics from multiple partial recordings of the same neural circuit. *Advances in Neural Information Processing Systems*, 26, 2013.

[7] Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

[8] Joel Ye, Jennifer Collinger, Leila Wehbe, and Robert Gaunt. Neural data transformer 2: multi-context pretraining for neural spiking activity. *bioRxiv*, pages 2023–09, 2023.

[9] Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael J Mendelson, Blake Richards, Matthew G Perich, Guillaume Lajoie, and Eva L Dyer. A unified, scalable framework for neural population decoding. *arXiv preprint arXiv:2310.16046*, 2023.

[10] Zoe C Ashwood, Nicholas A Roy, Iris R Stone, International Brain Laboratory, Anne E Urai, Anne K Churchland, Alexandre Pouget, and Jonathan W Pillow. Mice alternate between discrete strategies during perceptual decision-making. *Nature Neuroscience*, 25(2):201–212, 2022.

[11] Kush IBL, Banga, Julius Benson, Niccolò Bonacchi, Sebastian A Bruijns, Rob Campbell, Gaëlle A Chapuis, Anne K Churchland, M Felicia Davatolhagh, Hyun Dong Lee, et al. Reproducibility of in-vivo electrophysiological measurements in mice. *bioRxiv*, pages 2022–05, 2022.

[12] Brandon IBL, Benson, Julius Benson, Daniel Birman, Niccolo Bonacchi, Matteo Carandini, Joana A Catarino, Gaelle A Chapuis, Anne K Churchland, Yang Dan, et al. A brain-wide map of neural activity during complex behaviour. *bioRxiv*, pages 2023–07, 2023.

[13] James V Haxby, J Swaroop Guntupalli, Samuel A Nastase, and Ma Feilong. Hyperalignment: Modeling shared information encoded in idiosyncratic cortical topographies. *elife*, 9:e56601, 2020.

[14] Erica L Busch, Lukas Slipski, Ma Feilong, J Swaroop Guntupalli, Matteo Visconti di Oleggio Castello, Jeremy F Huckins, Samuel A Nastase, M Ida Gobbini, Tor D Wager, and James V Haxby. Hybrid hyperalignment: A single high-dimensional model of shared information embedded in cortical patterns of response and functional connectivity. *NeuroImage*, 233:117975, 2021.

[15] Stefan J Kiebel, Jean Daunizeau, and Karl J Friston. A hierarchy of time-scales and the brain. *PLoS computational biology*, 4(11):e1000209, 2008.

[16] Benjamin B Scott, Christine M Constantinople, Athena Akrami, Timothy D Hanks, Carlos D Brody, and David W Tank. Fronto-parietal cortical circuits encode accumulated evidence with a diversity of timescales. *Neuron*, 95(2):385–398, 2017.

[17] Roxana Zeraati, Yan-Liang Shi, Nicholas A Steinmetz, Marc A Gieselmann, Alexander Thiele, Tirin Moore, Anna Levina, and Tatiana A Engel. Intrinsic timescales in the visual cortex change with selective attention and reflect spatial connectivity. *Nature communications*, 14(1):1858, 2023.

[18] Charles Findling, Felix Hubert, International Brain Laboratory, Luigi Acerbi, Brandon Benson, Julius Benson, Daniel Birman, Niccolò Bonacchi, Matteo Carandini, Joana A Catarino, et al. Brain-wide representations of prior information in mouse decision-making. *BioRxiv*, pages 2023–07, 2023.

[19] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[20] JJ Deely and DV Lindley. Bayes empirical bayes. *Journal of the American Statistical Association*, 76(376):833–841, 1981.

[21] Herbert E Robbins. An empirical bayes approach to statistics. In *Breakthroughs in Statistics: Foundations and basic theory*, pages 388–394. Springer, 1992.

[22] Bradley Efron. Empirical bayes methods for combining likelihoods. *Journal of the American Statistical Association*, 91(434):538–550, 1996.

[23] Greg M Allenby and Peter E Rossi. Hierarchical bayes models. *The handbook of marketing research: Uses, misuses, and future advances*, pages 418–440, 2006.

[24] John K Kruschke and Wolf Vanpaemel. Bayesian estimation in hierarchical models. *The Oxford handbook of computational and mathematical psychology*, pages 279–299, 2015.

[25] International Brain Laboratory, D Birman, N Bonacchi, K Buchanan, G Chapuis, J Huntenburg, G Meijer, L Paninski, M Schartner, K Svoboda, et al. Video hardware and software for the international brain laboratory. *figshare*, 2022.

[26] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289, 2018.

[27] Dan Biderman, Matthew R Whiteway, Cole Hurwitz, Nicholas Greenspan, Robert S Lee, Ankit Vishnubhotla, Richard Warren, Federico Pedraja, Dillon Noone, Michael M Schartner, et al. Lightning pose: improved animal pose estimation via semi-supervised learning, bayesian ensembling and cloud-native open-source tools. *Nature Methods*, pages 1–13, 2024.

[28] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

[29] Douglas Steinley. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386, 2004.

[30] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

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

[32] 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.

[33] 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.

[34] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[35] David K Warland, Pamela Reinagel, and Markus Meister. Decoding visual information from a population of retinal ganglion cells. *Journal of neurophysiology*, 78(5):2336–2350, 1997.

[36] Kenneth D Harris. Nonsense correlations in neuroscience. *Biorxiv*, pages 2020–11, 2020.

[37] Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 503(7474):78–84, 2013.

[38] Mikio Aoi and Jonathan W Pillow. Model-based targeted dimensionality reduction for neuronal population data. *Advances in neural information processing systems*, 31, 2018.

[39] João D Semedo, Amin Zandvakili, Christian K Machens, M Yu Byron, and Adam Kohn. Cortical areas interact through a communication subspace. *Neuron*, 102(1):249–259, 2019.

[40] João D Semedo, Evren Gokcen, Christian K Machens, Adam Kohn, and M Yu Byron. Statistical methods for dissecting interactions between brain areas. *Current opinion in neurobiology*, 65:59–69, 2020.

[41] Atika Syeda, Lin Zhong, Renee Tung, Will Long, Marius Pachitariu, and Carsen Stringer. Facemap: a framework for modeling neural activity based on orofacial tracking. *Nature Neuroscience*, pages 1–9, 2023.

[42] 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.

[43] Kim Whoriskey, Marie Auger-Méthé, Christoffer M Albertsen, Frederick G Whoriskey, Thomas R Binder, Charles C Krueger, and Joanna Mills Flemming. A hidden markov movement model for rapidly identifying behavioral states from animal tracks. *Ecology and evolution*, 7(7):2112–2121, 2017.

[44] Guiming Wang. Machine learning for inferring animal behavior from location and movement data. *Ecological informatics*, 49:69–76, 2019.

[45] Moshe Abeles, Hagai Bergman, Itay Gat, Isaac Meilijson, Eyal Seidemann, Naftali Tishby, and Eilon Vaadia. Cortical activity flips among quasi-stationary states. *Proceedings of the National Academy of Sciences*, 92(19):8616–8620, 1995.

[46] Caleb Kemere, Gopal Santhanam, Byron M Yu, Afsheen Afshar, Stephen I Ryu, Teresa H Meng, and Krishna V Shenoy. Detecting neural-state transitions using hidden markov models for motor cortical prostheses. *Journal of neurophysiology*, 100(4):2441–2452, 2008.

[47] Márton G Danóczy and Richard Hahnloser. Efficient estimation of hidden state dynamics from spike trains. *Advances in neural information processing systems*, 18, 2005.

[48] Gregor Rainer and Earl K Miller. Neural ensemble states in prefrontal cortex identified using a hidden markov model with a modified em algorithm. *Neurocomputing*, 32:961–966, 2000.

[49] Günter Radons, JD Becker, B Dülfer, and J Krüger. Analysis, classification, and coding of multielectrode spike trains with hidden markov models. *Biological cybernetics*, 71(4):359–373, 1994.

[50] Eleanor Batty, Matthew Whiteway, Shreya Saxena, Dan Biderman, Taiga Abe, Simon Musall, Winthrop Gillis, Jeffrey Markowitz, Anne Churchland, John P Cunningham, et al. Behavenet: nonlinear embedding and bayesian neural decoding of behavioral videos. *Advances in Neural Information Processing Systems*, 32, 2019.

[51] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[52] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

[53] Yizi Zhang, Tianxiao He, Julien Boussard, Charles Windolf, Olivier Winter, Eric Trautmann, Noam Roth, Hailey Barrell, Mark Churchland, Nicholas A Steinmetz, et al. Bypassing spike sorting: Density-based decoding using spike localization from dense multielectrode probes. *Advances in Neural Information Processing Systems*, 36, 2024.

[54] Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of multivariate analysis*, 5(2):248–264, 1975.

[55] Shuai Zheng, Xiao Cai, Chris Ding, Feiping Nie, and Heng Huang. A closed form solution to multi-view low-rank regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[56] S Joe Qin. An overview of subspace identification. *Computers & chemical engineering*, 30(10-12):1502–1513, 2006.

[57] Christian Gruhl and Bernhard Sick. Variational bayesian inference for hidden markov models with multivariate gaussian output distributions. *arXiv preprint arXiv:1605.08618*, 2016.

[58] Robert Bassett and Julio Deride. Maximum a posteriori estimators as a limit of bayes estimators. *Mathematical Programming*, 174:129–144, 2019.

[59] Stephen Brooks. Markov chain monte carlo method and its application. *Journal of the royal statistical society: series D (the Statistician)*, 47(1):69–100, 1998.

[60] Marius Pachitariu, Nicholas Steinmetz, Shabnam Kadir, Matteo Carandini, and Harris Kenneth D. Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. *BioRxiv*, page 061481, 2016.

[61] Fernando De la Torre. A least-squares framework for component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1041–1055, 2012.

	Notation	Definition
RRM	X	single-trial neural activity
	y	single-trial ground truth behavior
	d	single-trial predicted behavior (decoder estimate)
	U	reduced-rank model's neural basis set
	V	reduced-rank model's temporal basis set
	b	reduced-rank model's intercept term
	A	multi-region reduced-rank model's temporal basis set for each brain region
	B	multi-region reduced-rank model's temporal basis set shared across all regions
	N	number of neurons in a session
	T	number of time bins in each trial
	K	number of trials in a session
	P	dimension of the behavior of interest
	R	rank of the (multi-session) reduced-rank model's U and V basis sets
	L	rank of the multi-region reduced-rank model's A and B basis sets
BMM-HMM	y_k	true behavior in trial k
	d_k	single-trial, single-session decoder estimate in trial k
	z_k	latent mixture assignment for trial k in the beta-mixture model
	s_k	hidden Markov model's latent state in trial k
	$\alpha_k(h)$	probability of past observations $\{d_1, d_2, \ldots, d_k\}$ at state h in trial k
	$\beta_k(h)$	probability of future observations $\{d_{k+1}, d_{k+1}, \ldots, d_K\}$ at state h in trial k
	$\gamma_k(h, y)$	probability of y at state h in trial k given observations $\{d_1, \ldots, d_K\}$
	$\xi_k(h, m)$	transition probability from state h in trial k to state m in trial $k+1$ given $\{d_1, \ldots, d_K\}$
	π	HMM's initial state distribution
	η	HMM's transition probability matrix
	ϕ	HMM's emission probability matrix
	H	number of latent states in the HMM
LG-AR1	y_k	true behavior in trial k
	d_k	single-trial, single-session decoder estimate in trial k
	z_k	LG-AR1's latent state in trial k
	$\tilde{d}_k$	improved decoder estimate in trial k given observations $\{d_1, \ldots, d_K\}$
	Λ	LG-AR1's model parameters including $\theta, \rho, \mu, \sigma_\epsilon^2, \sigma_\tau^2$
	θ	LG-AR1's observation model parameter controlling the generation of d_k from the latent state
	ρ	LG-AR1's dynamic model parameter governing the latent state transition from trial $k-1$ to k
	μ	the intercept term of LG-AR1's observation model
	$\sigma_\epsilon^2, \sigma_\tau^2$	LG-AR1 noise term variance

Table 1: Table of notation.

6 Methods

6.1 Reduced-rank model: Model details

6.1.1 Closed-form solution for theoretical interpretation

In practice, the reduced-rank model parameters can be learned using automatic differentiation. However, in this section, we derive a closed-form solution for computational efficiency and theoretical interpretation. For notational simplicity, we omit the session index i and denote the neural activity and behavior from all trials as X and D. We use the centered neural activity and behavior matrices $X^c = X - \bar{X}$ and $D^c = D - \bar{D}$ to avoid dealing with the intercept term b from Eq 1.

Our proposed reduced-rank model in Eq 1 solves the following optimization problem:

$$\mathcal{L}_{\text{RRM}} = ||D^c - X^{c\top}(UV)||^2 + \lambda||UV||^2, \tag{4}$$

where $||\cdot||^2$ is the Frobenius norm and λ is the regularization strength. While reduced-rank regression has a standard closed-form solution [54], it cannot be directly applied to our problem when decoding vector-valued behavior ($P = T$), as its objective is to solve the following optimization problem:

$$\mathcal{L}_{\text{standard-RRM}} = ||D^c - X^{c\top}(FE)||^2 + \lambda||FE||^2, \tag{5}$$

where $F \in \mathbb{R}^{NT\times R}$ serves as the basis set for the entire neural activity X^c, while $E \in \mathbb{R}^{R\times P}$ serves as the basis set for the entire behavior D^c, respectively. In contrast, our decoding model explicitly disentangles the parameters into a neural basis set $U \in \mathbb{R}^{N\times R}$ and a temporal basis set $V \in \mathbb{R}^{R\times T}$ for each of the P timesteps, separating the effects of neurons and time. A separate temporal basis set V also allows for multi-session training by sharing V across sessions.

The intercept solution is $\tilde{b} = \bar{D} - \bar{X}^\top(UV)$. Taking the derivative of Eq 4 w.r.t. V, we have

$$\frac{\partial \mathcal{L}_{\text{RRM}}}{\partial V} = -2U^\top X^c D^c + 2U^\top X^c X^{c\top} UV + 2\lambda U^\top UV. \tag{6}$$

Setting Eq 6 to 0, we have the optimal solution

$$\tilde{V} = G^{-1}H, \quad G = U^\top(X^c X^{c\top} + \lambda I)U, \quad H = U^\top X^c D^c. \tag{7}$$

Substituting $\tilde{V}$ into Eq 4, the objective becomes minimizing $-Tr\{G^{-1}HH^\top\}$ according to [55]. Then, U is given by the optimal solution of the following problem:

$$\tilde{U} = -\underset{U}{\text{argmin}} Tr\{G^{-1}HH^\top\} = \underset{U}{\text{argmax}}\, Tr\{(U^\top S_t U)^{-1} U^\top S_b U\}, \tag{8}$$

where

$$S_b = X^c D^c D^{c\top} X^{c\top}, \quad S_t = \text{diag}\{X^c X^{c\top} + \lambda I\}. \tag{9}$$

We obtain $\tilde{U}$ from the left singular vectors of $X^c D^c (S_t)^{-1/2} \in \mathbb{R}^{N\times T}$ corresponding to the R largest singular values. In practice, the regularization strength λ is selected via cross-validation and grid-search.

Without regularization ($\lambda = 0$), $\tilde{U}$ being the left singular vectors of $X^c D^c (S_t)^{-1/2}$ implies that $\tilde{U}$ maximizes the correlation between neural activity X and behavior D, and captures major variations in D:

$$\mathbb{E}[X^c D^c (S_t)^{-1/2}] = \frac{\mathbb{E}[(X - \bar{X})(D - \bar{D})]}{\sqrt{\mathbb{E}[(X - \bar{X})(X - \bar{X})^\top]}} = \frac{\text{Cov}(X, D)}{\sqrt{\text{Var}(X)}} = \text{Corr}(X, D)\sqrt{\text{Var}(D)}. \tag{10}$$

Therefore, $\tilde{U}$ quantifies each neuron's contribution to behavior decoding, and therefore identifies the most relevant neurons for the decoding task. After learning the optimal U and V, we project the neural activity X onto the learned low-rank subspace U to obtain the low-dimensional data representation $W = (X^\top U)$ capturing behaviorally-relevant neural variations [56, 33, 32].

The closed-form solution provided is restricted to linear models. For greater flexibility with nonlinear decoders or more complex data structures than the present case, we recommend using automatic differentiation.

6.1.2 Multi-trial reduced-rank model for prior decoding

To improve prior decoding, we employ a multi-trial reduced-rank model that exploits trial-to-trial correlations. The decoding results are shown in Figure 2 and 5. The main idea is to use neural activity from neighboring trials, denoted as $\vec{X}_k := [X_k - l, X_k, X_k + l] \in \mathbb{R}^{N\times T\times L}$, to decode scalar-valued behavior in trial k, where $L = 2l + 1$ denotes the trial window length. Due to the large number of parameters that need to be learned, a reduced-rank model is a natural choice to prevent overfitting: $d_k = f(\vec{X}_k^\top(UV) + b)$, where $U \in \mathbb{R}^{N\times R}$, $V \in \mathbb{R}^{R\times T\times L}$ and $b \in \mathbb{R}$.

6.2 BMM-HMM: Model details

This section presents algorithms and implementation details for various BMM-HMM model variants. The BMM-HMM model consists of a dynamic process governing transitions among discrete latent states $\vec{s}$ and an observation process describing the generation of decoder estimates $\vec{d}$ given the latent state. The dynamic model, $P(s_k \mid s_{k-1})$, describes the state transition from trial $k-1$ to k, parameterized by a state transition matrix. The observation model, $p(d_k \mid s_k) = p(d_k \mid z_k)p(z_k \mid s_k)$, is characterized by a beta mixture model, where $p(z_k \mid s_k)$ is the emission probability at each state, $p(d_k \mid z_k)$ is the observation probability, and z_k controls the assignment of beta distributions in the mixture.

Specifically, we assume the single-session, single-trial decoder output $d_k = P(y_k = 1 \mid X_k) \in [0,1]$ follows a mixture of beta distributions, with mixture assignment z_k depending on a latent state s_k, governed by an H-state HMM. The data generation process for d_k is formulated as

$$p(d_k \mid s_k) = \sum_{z_k=0}^{1} \phi_{s_k z_k} \mathrm{Beta}(d_k; a_{z_k}, b_{z_k}), \quad \phi_{s_k z_k} := P(z_k = 1 \mid s_k), \tag{11}$$

where a_{z_k} and b_{z_k} are parameters of a beta distribution. In each trial, the latent state s_k generates z_k with emission probability $\phi_{s_k z_k}$, and d_k is drawn from a beta mixture with observation probability $p(d_k \mid z_k)$, where d_k values cluster around 1 when $z_k = 1$ and around 0 when $z_k = 0$.

The main idea is to substitute the single-session and single-trial decoder output d_k, which only considers information from the neural activity X_k, with the inferred z_k. The inferred z_k contains information about the underlying behavioral states deduced from the trial-to-trial correlations in $\vec{d}$. Specifically, the improved decoder output is

$$\begin{aligned} P(z_k = 1 \mid \vec{d}) &= \sum_{s_k=1}^{H} P(z_k, s_k \mid \vec{d}) = \sum_{s_k=1}^{H} P(s_k \mid \vec{d}) P(z_k \mid s_k, d_k) \\ &= \sum_{s_k=1}^{H} \frac{p(s_k, \vec{d})}{p(\vec{d})} \frac{p(d_k, z_k \mid s_k)}{p(d_k \mid s_k)} = \sum_{s_k=1}^{H} \frac{\alpha_k(s_k)\beta_k(s_k)}{\sum_{s'_k=1}^{H} \alpha_k(s'_k)\beta_k(s'_k)} \frac{f(d_k, z_k \mid s_k)}{f(d_k \mid s_k)}, \end{aligned} \tag{12}$$

where $f(d_k \mid s_k) = \sum_{z_k=0}^{1} p(d_k, z_k \mid s_k)$, as defined in Eq 11. $\alpha_k(s_k)$ and $\beta_k(s_k)$ are outputs from the forward and backward passes in an Expectation-Maximization (EM) algorithm, described in more depth below.

6.2.1 EM algorithm for BMM-HMM

The EM (Baum–Welch) algorithm is used for iterative HMM parameter estimation. Each iteration consists of the following Expectation and Maximization steps:

- **(E step)** Let k index trial, $z \in \{0,1\}$ index the beta mixture component and $h, m \in \{1, \ldots, H\}$ index the state. For all component and state pairs, we recursively compute the forward and backward probabilities $\alpha_k(h,z)$ and $\beta_k(h,z)$, defined below. We then compute the component and state occupation probabilities $\gamma_k(h,z)$ and $\xi_k(h,m)$.
- **(M step)** Using the estimated $\gamma_k(h)$ and $\xi_k(h)$, we then update the model parameters, including the transition probabilities η_{hm} and the emission probabilities ϕ_{hz} of the HMM, and the parameters of the beta mixture a_z, b_z.

Forward pass. We define the probability of observing the sequence of decoder outputs $\vec{d}$ being in state h in trial k as

$$\alpha_k(h) := p(d_1, d_2, \ldots, d_k, s_k = h). \tag{13}$$

The pseudo-code for the iterative computation of $\alpha_k(h)$ is:

- *Initialization* $\alpha_1(h) = \pi_0(h) f(d_1 \mid h) \quad \forall\, 1 \leq h \leq H.$
- *Recursion* $\alpha_k(h) = \left(\sum_{m=1}^{H} \alpha_{k-1}(m)\eta_{mh}\right) f(d_k \mid h) \quad \forall\, 1 \leq h, m \leq H,\ 1 \leq k \leq K.$
- *Termination* $p(\vec{d}) = \sum_{h=1}^{H} \alpha_K(h),$

where π_0 is a vector containing the initial probabilities for each of the H hidden states.

Backward pass. The probability of future observations given that the HMM is in state h in trial k is

$$\beta_k(h) := p(d_{k+1}, \ldots, d_K \mid s_k = h). \tag{14}$$

The pseudo-code for the iterative computation of $\beta_k(h)$ is:

- *Initialization* $\beta_K(h) = 1 \quad \forall\, 1 \leq h \leq H.$
- *Recursion* $\beta_k(h) = \sum_{m=1}^{H} \eta_{hm} f(d_{k+1} \mid m)\beta_{k+1}(m) \quad \forall\, 1 \leq h, m \leq H,\ 1 \leq k \leq K-1.$
- *Termination* $p(\vec{d}) = \sum_{h=1}^{H} \pi_0(h) f(d_1 \mid h)\beta_1(h).$

Forward-backward. The state occupation probability $\gamma_k(h)$ is

$$\gamma_k(h) := P(s_k = h \mid \vec{d}) = \frac{p(s_k = h, \vec{d})}{p(\vec{d})} = \frac{\alpha_k(h)\beta_k(h)}{\sum_{h'=1}^{H} \alpha_k(h')\beta_k(h')}. \tag{15}$$

The component and state occupation probability $\gamma_k(h, z)$ is the probability of component z at state h in trial k given the whole observation sequence $\vec{d}$:

$$\gamma_k(h, z) = P(s_k = h, z_k = z \mid \vec{d}) = \gamma_k(h)\frac{f(d_k, z \mid h)}{f(d_k \mid h)}. \tag{16}$$

We then estimate $\xi_k(h, m)$, the probability of transitioning from state h to m given all observations $\vec{d}$:

$$\xi_k(h, m) = P(s_k = h, s_{k+1} = m \mid \vec{d}) = \frac{p(s_k = h, s_{k+1} = m, \vec{d})}{p(\vec{d})} \tag{17}$$

$$= \frac{\alpha_k(h)\eta_{hm} f(d_{k+1} \mid m)\beta_{k+1}(m)}{\sum_{h'=1}^{H}\sum_{m'=1}^{H} \alpha_k(h')\eta_{h'm'} f(d_{k+1} \mid m')\beta_{k+1}(m')}. \tag{18}$$

For the M step, we update the transition and emission probabilities according to

$$\eta^*_{hm} = \frac{\frac{1}{K-1}\sum_{k=1}^{K-1} P(s_k = h, s_{k+1} = m \mid \vec{d})}{\frac{1}{K-1}\sum_{k=1}^{K-1} P(s_k = h \mid \vec{d})} = \frac{\sum_{k=1}^{K-1} \xi_k(h, m)}{\sum_{k=1}^{K-1} \gamma_k(h)}, \tag{19}$$

$$\phi^*_{hz} = \frac{\frac{1}{K}\sum_{k=1}^{K} P(z_k = z, s_k = h \mid \vec{d})}{\frac{1}{K}\sum_{k=1}^{K} P(s_k = h \mid \vec{d})} = \frac{\sum_{k=1}^{K} \gamma_k(h, z)}{\sum_{k=1}^{K} \gamma_k(h)}. \tag{20}$$

We then update the parameters of the BMM, (a_0, a_1, b_0, b_1), by maximizing the expected log-likelihood. First, we write down the likelihood of the BMM as

$$L(a_0, a_1, b_0, b_1) = \prod_{k=1}^{K}\sum_{s_k=1}^{H} p(d_k, z_k \mid s_k)p(s_k) = \prod_{k=1}^{K}\sum_{s_k=1}^{H} f(d_k, z_k \mid s_k)\pi_\infty(s_k), \tag{21}$$

where π_∞ represents the equilibrium probability for each of H hidden states, which can be computed using the estimated transition probabilities. The conditional distribution is subsequently determined by

$$r_{z_k} := P(z_k \mid d_k) = \frac{p(z_k, d_k \mid s_k)P(s_k)}{p(d_k)} \tag{22}$$

$$= \frac{\sum_{s_k=1}^{H} f(d_k, z_k \mid s_k)\pi_\infty(s_k)}{\sum_{z_k=0}^{1}\sum_{s_k=1}^{H} f(d_k, z_k \mid s_k)\pi_\infty(s_k)}. \tag{23}$$

Finally, the expected log-likelihood of the BMM is

$$\mathbb{E}[\log L(a_0, a_1, b_0, b_1)] = \mathbb{E}\left[\log\left(\prod_{k=1}^{K}\sum_{s_k=1}^{H} f(d_k, z_k \mid s_k)\pi_\infty(s_k)\right)\right] \tag{24}$$

$$= \sum_{k=1}^{K}\mathbb{E}\left[\log\left(\sum_{s_k=1}^{H} f(d_k, z_k \mid s_k)\pi_\infty(s_k)\right)\right] \tag{25}$$

$$= \sum_{k=1}^{K}\sum_{z_k=0}^{1} P(z_k \mid d_k)\log\left(\sum_{s_k=1}^{H} f(d_k, z_k \mid s_k)\pi_\infty(s_k)\right) \tag{26}$$

$$= \sum_{k=1}^{K}\sum_{z_k=0}^{1} r_{z_k}\cdot\log\left(\sum_{s_k=1}^{H} f(d_k, z_k \mid s_k)\pi_\infty(s_k)\right). \tag{27}$$

In practice, we find $(a_0^*, a_1^*, b_0^*, b_1^*)$ that maximize the quantity in Eq 27 through numerical optimization.

6.2.2 Oracle BMM-HMM

In each session, the oracle BMM-HMM substitutes the ground truth observed behaviors $\vec{y}$ for $\vec{z}$, treating $\vec{z}$ as a known quantity. This allows us to learn the underlying data-generating mechanism that produces the decoder outputs $\vec{d}$. The process consists of the following steps:

1. Train a discrete-state HMM on the ground truth observed behaviors $\vec{y}$ to estimate the oracle model parameters, including transition probabilities η_{hm} and emission probabilities ϕ_{hz} for each session.

2. Apply a BMM to the decoder outputs $\vec{d}$, treating the mixture assignment variable $\vec{z}$ as a known quantity by substituting $\vec{z}$ with the ground truth observed behaviors $\vec{y}$. This step provides the correct assignment of mixture components. The learned oracle BMM parameters, (a_0, a_1, b_0, b_1), capture the true probabilistic relationship between $\vec{d}$ and $\vec{z}$.

3. Use the learned oracle model parameters to initialize and fit the BMM-HMM using the EM algorithm described in the section "EM algorithm for BMM-HMM" for the corresponding session. During model fitting, fix the oracle parameters $(\eta_{hm}, \phi_{hz}, a_0, a_1, b_0, b_1)$.

This procedure allows us to deduce the latent behavioral states $\vec{s}$ and latent behaviors $\vec{z}$ as if we know the true data generation process.

6.2.3 Learning empirical priors of state-space model parameters

To learn empirical priors for the multi-session BMM-HMM, we fit a variational HMM [57] to the ground truth observed behavior $\vec{y}$ from non-target sessions. This allows us to learn an empirical prior of the trial-to-trial correlations inherent in the true behavioral data. We impose Dirichlet priors on the initial state distribution π_0, rows of the transition probability matrix $\eta_{h\cdot}$, and rows of the emission probability matrix $\phi_{h\cdot}$ as follows:

$$p(\pi_0) = \mathrm{Dir}(\{\pi_0(1), \dots, \pi_0(H)\}; \{u_1^{(\pi_0)}, \dots, u_H^{(\pi_0)}\}), \tag{28}$$

$$p(\eta) = \prod_{h=1}^{H}\mathrm{Dir}(\{\eta_{h1}, \dots, \eta_{hH}\}; \{u_{h1}^{(\eta)}, \dots, u_{hH}^{(\eta)}\}), \tag{29}$$

$$p(\phi) = \prod_{h=1}^{H}\mathrm{Dir}(\{\phi_{h0}, \phi_{h1}\}; \{u_{h0}^{(\phi)}, u_{h1}^{(\phi)}\}), \tag{30}$$

where $(u^{(\pi_0)}, u^{(\eta)}, u^{(\phi)})$ are the Dirichlet distribution concentration parameters, learned by fitting a variational HMM on the ground truth observed behaviors $\vec{y}$ from the training sessions using the Python package *hmmlearn*. The resulting posterior distributions serve as priors for the multi-session BMM-HMM parameters, constraining their updates during the EM algorithm's M step.

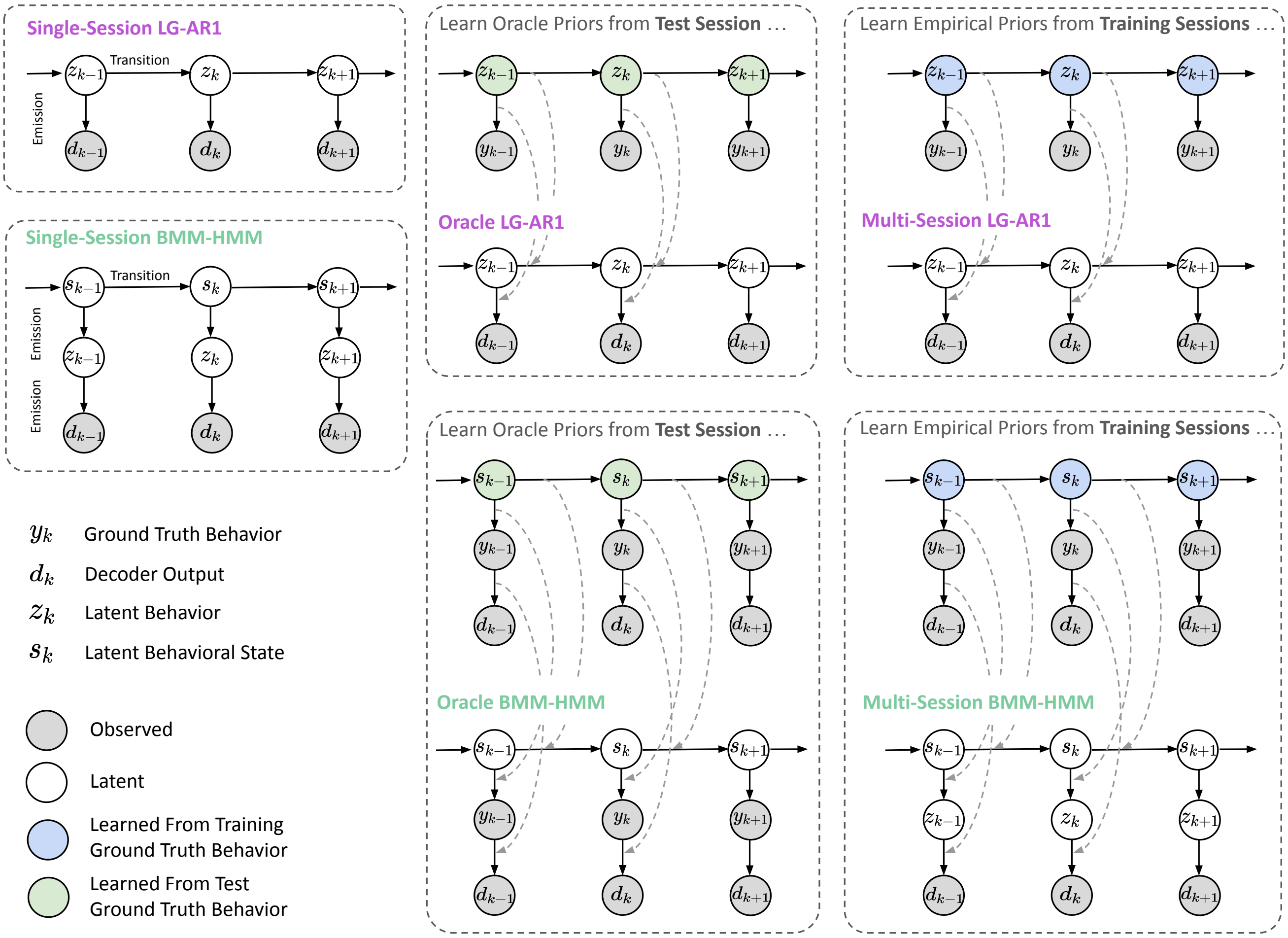
Single-Session LG-AR1
Transition
Emission
Single-Session BMM-HMM
Transition
Emission
Emission
Learn Oracle Priors from Test Session ...
Oracle LG-AR1
Learn Empirical Priors from Training Sessions ...
Multi-Session LG-AR1
Learn Oracle Priors from Test Session ...
Oracle BMM-HMM
Learn Empirical Priors from Training Sessions ...
Multi-Session BMM-HMM
z_{k-1} z_k z_{k+1}
s_{k-1} s_k s_{k+1}
y_{k-1} y_k y_{k+1}
d_{k-1} d_k d_{k+1}
y_k Ground Truth Behavior
d_k Decoder Output
z_k Latent Behavior
s_k Latent Behavioral State
Observed
Latent
Learned From Training Ground Truth Behavior
Learned From Test Ground Truth Behavior

Figure 10: **Graphical models for single-session, oracle, and multi-session LG-AR1 and BMM-HMM models.** The single-session LG-AR1 and BMM-HMM models learn parameters directly from the test session. In contrast, the oracle versions of these models use a two-step learning process. They first derive oracle priors from the test set's ground truth behavior, then use these priors to guide parameter learning on the test data, as indicated by gray arrows. The multi-session variants follow a similar approach. They learn empirical priors from the training data's ground truth behaviors, which then constrain parameter learning on the test set, also shown by gray arrows. This approach allows the multi-session models to leverage information from multiple training sessions while adapting to the current test data.

To set empirical priors for the BMM parameters, we assume d_k follows a mixture of beta distributions from the exponential family, expressed as:

$$\mathrm{Beta}(d; \{a_z, b_z\}) = h(d)c(a_z, b_z)\exp\left(w(a_z, b_z)^\top t(d)\right), \tag{31}$$

$$h(d) = 1,\ c(a_z, b_z) = 1/B(a_z, b_z), \tag{32}$$

where

$$B(a_z, b_z) = \Gamma(a_z)\Gamma(b_z)/\Gamma(a_z + b_z), \tag{33}$$

$$w(a_z, b_z) = (a_z - 1, b_z - 1)^\top,\ t(d) = (\ln d, \ln(1-d))^\top. \tag{34}$$

For exponential family members, the conjugate prior is

$$p(a_z, b_z \mid \nu_1, \nu_2, \psi) \propto c(a_z, b_z)^\psi \exp(w(a_z, b_z)^\top (\nu_1, \nu_2)^\top). \tag{35}$$

Therefore, a suitable conjugate prior distribution for (a_z, b_z) is

$$p(a_z, b_z \mid \nu_1, \nu_2, \psi) \propto \frac{1}{B(a_z, b_z)^\psi \exp(-(\nu_1 a_z + \nu_2 b_z))}. \tag{36}$$

Setting the natural conjugate prior ψ parameter to zero yields independent exponential priors for (a_z, b_z), which have proven effective empirically. We apply a hierarchical BMM on the decoder outputs $\vec{d}$, using the

Python package *pymc3*. We assume that the mixture assignment $\vec{z}$ can be empirically determined a priori, and substitute $\vec{z}$ with the observed behaviors $\vec{y}$ from the training sessions. The posterior distributions for $(\nu_1^{(z)}, \nu_2^{(z)})$ then serve as priors for the multi-session BMM-HMM parameters, constraining their updates during the EM algorithm's M step.

6.2.4 Multi-session BMM-HMM

Following the approach in Eq 28-30, we impose Dirichlet priors on the BMM-HMM dynamic parameters $(\pi_0, \eta_{h\cdot}, \phi_{h\cdot})$. We modify the EM algorithm in the section "EM algorithm for BMM-HMM" by using Maximum A Posteriori (MAP) estimation [58] to learn the posterior distributions of these parameters. The E step remains unchanged, while the M step incorporates the new prior terms when updating the HMM parameters with fixed latent s_k and z_k. The posterior means of the HMM parameters become

$$\pi_0(h) = \frac{\tilde{u}_h^{(\pi_0)} + \gamma_0(h)}{\sum_{h'=1}^{H} \tilde{u}_{h'}^{(\pi_0)} + \gamma_0(h')}, \quad \eta_{hm} = \frac{\tilde{u}_{hm}^{(\eta)} + \sum_{k=1}^{K-1} \xi_k(h,m)}{\sum_{m'=1}^{H} \tilde{u}_{hm'}^{(\eta)} + \sum_{k=1}^{K-1} \gamma_k(h)}, \tag{37}$$

$$\phi_{hz} = \frac{\tilde{u}_{hz}^{(\phi)} + \sum_{k=1}^{K} \gamma_k(h,z)}{\sum_{z'=0}^{1} \tilde{u}_{hz'}^{(\phi)} + \sum_{k=1}^{K} \gamma_k(h)}, \tag{38}$$

where $(\tilde{u}^{(\pi_0)}, \tilde{u}^{(\eta)}, \tilde{u}^{(\phi)})$ are the posterior concentration parameters from fitting the variational HMM on the training sessions. When updating BMM parameters, we add the Dirichlet prior term $\log p(\pi_0, \eta, \phi)$ to the complete-data log-likelihood in Eq 24 and solve for (a_0, a_1, b_0, b_1) that maximize this new objective function.

We constrain BMM parameters (a_0, a_1, b_0, b_1), using empirical priors, $(\nu_1^{(0)}, \nu_2^{(0)}, \nu_1^{(1)}, \nu_2^{(1)})$, learned from the training sessions; see details in the section "Learning empirical priors of state-space model parameters". Incorporating the log-prior term (Eq 36) into the complete log-likelihood involves adding the following penalty to the right-hand side of Eq 27:

$$\sum_{z=0}^{1} \log p(a_z, b_z; \nu_1^{(z)}, \nu_2^{(z)}, \psi = 0) = -\sum_{z=0}^{1} (\nu_1^{(z)} a_z + \nu_2^{(z)} b_z) + \text{const.}. \tag{39}$$

Numerically solving the penalized objective yields MAP estimates for the BMM parameters instead of the standard maximum likelihood estimation (MLE) solutions.

6.3 LG-AR1: Model details

For scalar-valued $y_k \in \mathbb{R}$, we assume the decoder output $d_k \in \mathbb{R}$ linearly depends on the latent behavior $z_k \in \mathbb{R}$. To incorporate trial-to-trial correlations, the transitions of z_k between trials are modeled using a first-order autoregressive process. The objective aligns with that of a Kalman smoother [19], which is to infer the state of a dynamical system (z_k) given a sequence of noisy observations (d_k). The formal data generating model is described as

$$d_k = \theta z_k + \mu + \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, \sigma_\epsilon^2), \tag{40}$$

$$z_k = \rho z_{k-1} + \tau_k, \quad \tau_k \sim \mathcal{N}(0, \sigma_\tau^2). \tag{41}$$

Intuitively, as ρ approaches 1, z_k in the current trial is expected to exhibit minimal deviation from z_{k-1} in the preceding trial, as per Eq 41. As θ approaches 1, d_k is expected to closely track the pattern of z_k according to Eq 40. In practice, the values of θ and ρ are determined by fitting the LG-AR1 model to the observed $\vec{d}$.

Similar to BMM-HMM, the main idea is to replace the original decoder estimate d_k, based solely on neural activity X_k, with a smoothed estimate $\tilde{d}_k$ derived from the inferred latent state z_k. $\tilde{d}_k$ incorporates trial-to-trial correlations from $\vec{d} = \{d_1, d_2, \ldots, d_k\}$, as $\vec{d}$ is used to infer the latent states $\vec{z} = \{z_1, z_2, \ldots, z_k\}$. This process potentially improves $\tilde{d}_k$'s accuracy over the original d_k in estimating the true behavior. While $\vec{d}$ is used for model fitting and latent state inference, $\tilde{d}_k$ is the improved (smoothed) decoder estimate for the held-out trial k given the entire $\vec{d}$. To obtain $\tilde{d}_k$, we sample from its posterior predictive distribution

$$p(\tilde{d}_k \mid \vec{d}) = \int p(\tilde{d}_k \mid \Lambda) p(\Lambda \mid \vec{d}) d\Lambda, \tag{42}$$

after placing prior distributions on the model parameters $\Lambda = (\theta, \mu, \rho, \sigma_\epsilon^2, \sigma_\tau^2)$, which can be estimated using Markov chain Monte Carlo (MCMC) sampling [59].

To fit LG-AR1 on single-session data, we use a Bayesian approach, treating model parameters $\Lambda = (\theta, \mu, \rho, \sigma_\epsilon^2, \sigma_\tau^2)$ as random variables with joint prior $p(\Lambda)$:

$$\theta, \mu, \rho \sim \mathcal{N}(0, 1), \quad \sigma_\epsilon^2, \sigma_\tau^2 \sim \text{Half-}\mathcal{N}(0, 1). \tag{43}$$

In practice, we use the Python package *pymc3* to fit the hierarchical LG-AR1 and learn the posterior distribution of session-specific parameters Λ via MCMC sampling.

To implement the multi-session LG-AR1, we begin by learning the dynamic model parameters (ρ, σ_τ^2). This estimation is performed using the observed behaviors $\vec{y}$ from the training sessions, under the assumption that these dynamic model parameters can be empirically determined a priori. Next, we estimate observation model parameters $(\theta, \mu, \sigma_\epsilon^2)$ using decoder outputs $\vec{d}$ and corresponding observed $\vec{y}$ from training sessions. After estimating model parameters from the training data, we use the posterior means of these multi-session LG-AR1 parameters $\Lambda = (\theta, \mu, \rho, \sigma_\epsilon^2, \sigma_\tau^2)$ to initialize the hierarchical LG-AR1 model (Eq 40-41) for the held-out session, with Λ fixed during model fitting. For this held-out session, where true behaviors are unknown, we infer the latent behaviors z_k and obtain improved decoder outputs $\tilde{d}_k$ via MCMC sampling.

We also implement an oracle LG-AR1 model to emulate the ground-truth data-generating process for $\vec{d}$. This oracle model is constructed by estimating model parameters using the ground truth observed $\vec{y}$ from the target session, under the assumption that the true values of the variable $\vec{z}$ are known. For the oracle model, we learn dynamic AR1 parameters (ρ, σ_τ^2) and observation model parameters $(\theta, \mu, \sigma_\epsilon^2)$ using true $\vec{y}$ and observed $\vec{d}$ from the test session. We initialize the hierarchical LG-AR1 model using these oracle solutions and hold them fixed while inferring the latent z_k and improved decoder outputs $\tilde{d}_k$, as if we know the true data-generating mechanism.

6.4 Data details

For choice, we align trials to the stimulus onset, considering neural activity from 0.5 seconds before to 1.5 seconds post-onset. For prior, we also align trials to the stimulus onset, including neural activity from 0.6 seconds to 0.1 seconds pre-onset. The prior represents the mice's estimate of the stimulus side probability. We use the same decoding window as in the previous study [18], focusing on the period with minimal wheel movements. Within each trial, we segment neural activity into 50-millisecond non-overlapping time bins. For each time bin, we bin spike counts using all neurons, sorted by Kilosort 2.5 [60], from each session. For continuous behaviors, we select an alignment event – first movement onset for wheel speed, motion energy and pupil diameter – and decode the target starting at the alignment event and ending at 1 second after the alignment event. The neural activity within each trial is binned into non-overlapping 20 ms bins. For each time bin, we similarly bin spike counts using all neurons from each session. For static behaviors (choice and prior), we use a 50 ms time bin size following [11], and for continuous behaviors, we use a 20 ms time bin size as in [27].

6.5 Hyperparameter selection

For choice and prior, baseline decoders (linear, MLP, LSTM) decode both behaviors in a single-trial, single-session context, where each trial's target behavior is decoded using the corresponding neural activity within that trial and session. For continuous behaviors, the target value for a time bin ending at time t is decoded using spikes from all time bins within a trial. To share neural data, we use a multi-session reduced-rank model for choice and continuous behaviors, and a multi-trial, multi-session reduced-rank model for prior (see the section "Multi-trial reduced-rank model for prior decoding"). To share behavioral data, we employ a multi-session BMM-HMM for choice and a multi-session LG-AR1 for prior. Decoder performance is evaluated using AUC for choice, Pearson's correlation for prior, and R^2 for continuous behaviors.

Baseline linear decoders use L2-penalized logistic regression for choice and ridge regression for prior and continuous behaviors, implemented with *scikit-learn* in Python. Regularization coefficients are cross-validated over $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$. Reduced-rank, MLP, and LSTM decoders are trained using gradient descent in PyTorch with Adam optimizer and cosine annealing learning rate scheduler. Learning rate, weight decay, and batch size are selected via grid search over $\{10^{-2}, 10^{-3}\}$, $\{10^{-1}, 10^{-3}\}$, and $\{8, 16, 32\}$, respectively, based on validation set performance. Models are trained until convergence, and the best-performing model with the lowest validation loss is used for test set decoding. Hyper-parameter search is conducted using *Ray Tune* in Python.

For reduced-rank models, the optimal rank for each behavior is determined via grid search over $R \in \{2, 5, 10, 15, 20, 25, 30\}$ based on validation performance. Fig 2C summarizes the sensitivity of choice decoding performance to the rank. For the MLP decoder, we explore architectures: $\{(256, 128, 64), (512, 256, 128, 64)\}$, where each number represents the hidden size of a layer, and the tuple length indicates the number of hidden layers. For the LSTM decoder, we investigate hidden sizes $\{32, 64\}$ and depths $\{1, 3, 5\}$. The LSTM is followed by MLP layers for predicting the target behavior, with architectures: $\{(64, 32), (64), (32)\}$. MLP and LSTM architectures are selected based on decoding performance, avoiding overly complex architectures due to limited training data, which could lead to overfitting and convergence issues.

For the multi-region reduced-rank model, we fix hyperparameters based on pilot studies due to the extensive training time required for fitting over 400 sessions. The rank of the region-specific temporal basis V^j is set to 2, and the rank of the global temporal basis B is set to 5. We use gradient descent with Adam optimizer, a learning rate and weight decay of 10^{-3}, and a batch size of 8. The model is trained for 500 epochs or until the loss does not decrease for 50 consecutive epochs to ensure convergence.

6.6 Differences between RRM, PCA, CCA and demixed PCA

The reduced rank model (RRM) is similar to dimensionality reduction techniques like PCA and CCA, but with different objectives. As shown in Eq 10, RRM maximizes the correlation between the centered predictor X and the centered response D, as well as the variance of D:

$$\text{RRM} : \text{Corr}(X, D)^2 \text{Var}(D). \tag{44}$$

According to [61], PCA and CCA aim to maximize:

$$\text{PCA} : \text{Var}(X), \quad \text{CCA} : \text{Corr}(X, D)^2. \tag{45}$$

PCA captures the major variations in neural activity X but ignores the variations in behavior D, while CCA considers the correlation between X and D but doesn't prioritize modeling the variations in D. RRM balances both the correlation between X and D and the variance of D, making it more suitable for decoding tasks where capturing behavioral variations is crucial for prediction.

RRM is closely related to demixed-PCA [33], which minimizes the loss

$$\mathcal{L}_{\text{demixed-PCA}} = \|X_s - FEX\|^2, \tag{46}$$

where X is the centered data matrix, with each row representing the neural activity of each neuron across all trials and task conditions. The reconstruction target, X_s, is a matrix of stimulus averages, with each data point replaced by the average neural activity for the corresponding stimulus. The solutions for F and E can be analytically obtained using reduced-rank regression through singular value decompositions. The main difference is the reconstruction target: the behavior D in our model (Eq. 4) vs. the task-condition averaged neural activity X_s in demixed PCA. Intuitively, demixed-PCA maximizes the correlation between the neural activity X and the task-condition averaged neural activity X_s, while also maximizing the variance of X_s.

6.7 Assessing statistical significance

In Section 4.7 "Mapping behaviorally-relevant timescales across the brain," we measure the increased information decoded from each region using the multi-region reduced-rank model compared to the baseline linear decoder. To control for potential spurious correlations, we conduct an additional experiment, following the approach in [12]. We generate null distributions to test the significance of our decoding results according to the procedure described in the caption of Figure S1.

To assess the significance of our decoding results, we analyze brain regions PO, LP, DG, CA1, and VISa as representative examples. Figure S1 displays the adjusted scores, with the original scores for choice and prior decoding corresponding to those in Figure 9D. While the percentage increase in decoding

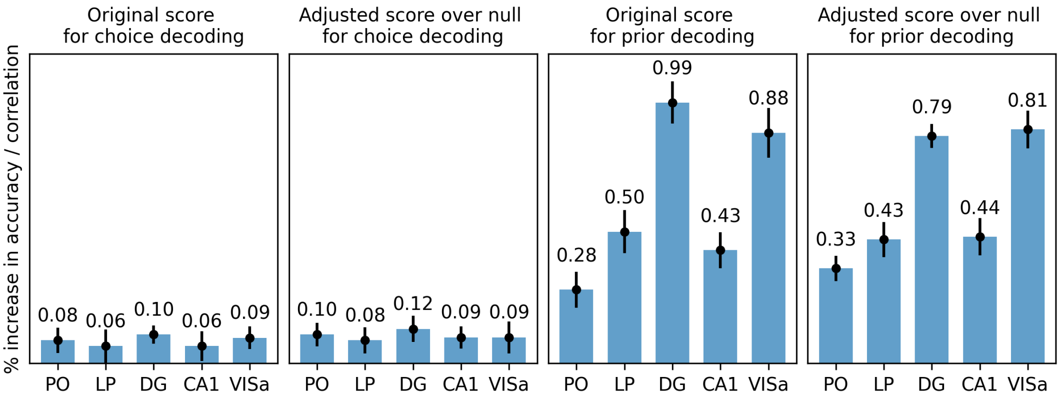

Figure S1: **Assessing the significance of decoding improvement achieved by multi-region reduced-rank model relative to null distributions generated from imposter sessions.** For each session with probe insertions in PO, LP, DG, CA1, and VISa, we create 10 "imposter sessions" from behaviors (choice and prior) of other mice in different sessions, as in [12]. These are generated by concatenating trials across all analyzed sessions, excluding the session under consideration, then randomly selecting a chunk of N consecutive trials (where N matches the original session length) from the concatenated sessions. We obtain the original score from the real session, while the adjusted score is calculated by subtracting the decoding accuracy (or correlation) of the imposter sessions from the original score. Each bar shows the mean score from 10 imposter sessions, with error bars indicating one standard deviation of these scores.

metrics varies slightly between original and adjusted scores, the relative ranking of brain regions, based on decoding improvement, remains largely consistent. For instance, DG shows the highest improvement in decodable information for choice, both before and after null distribution adjustment. This analysis demonstrates the reliability of the decoding improvement offered by our proposed model.

6.8 Supplemental figures

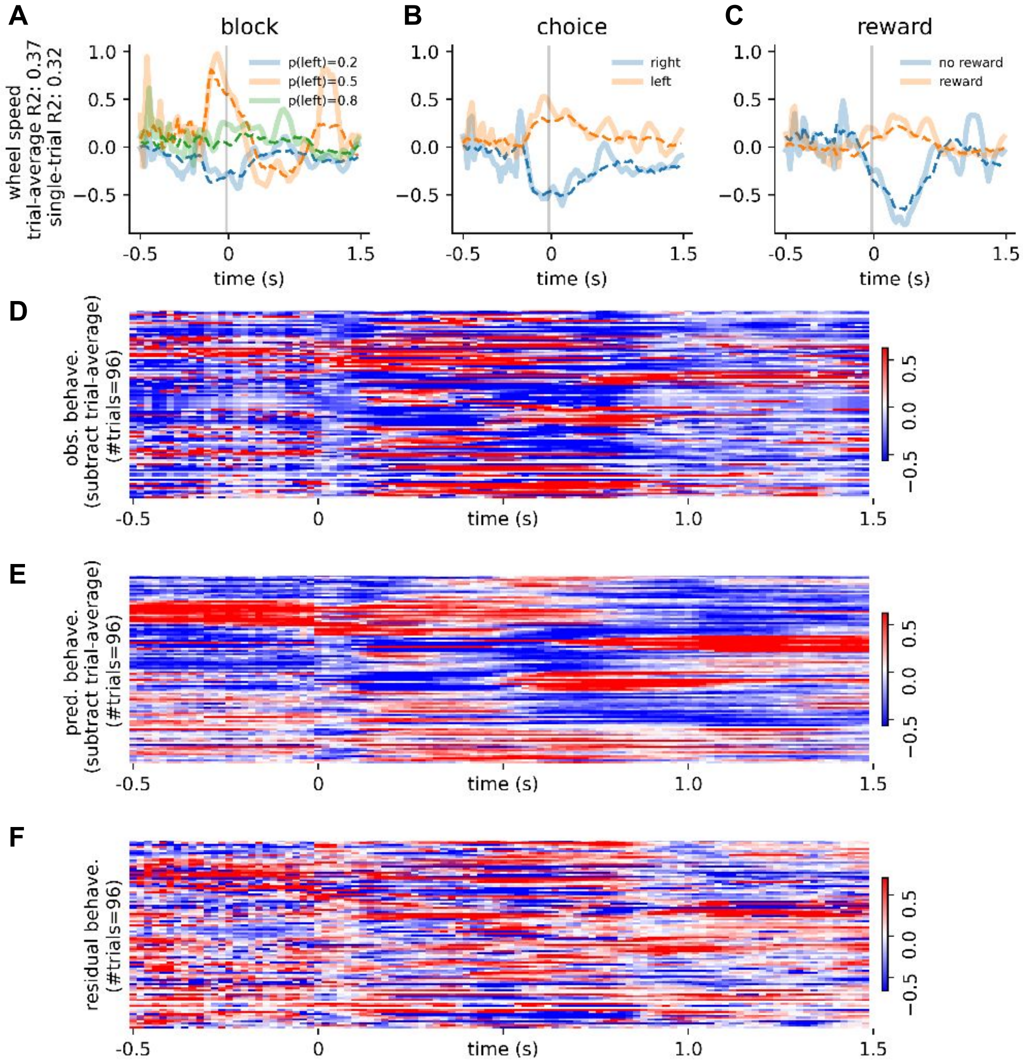

Figure S2: **Evaluating wheel speed decoding quality using spiking activity from 1313 neurons in a RE dataset session.** **(A-C)** Comparison between the reduced-rank model's predicted wheel speed (dotted curves) and observed ground truth behavior (solid curves) across different block (A), choice (B) and reward (C) conditions. Refer to Fig 3 (A-C) for conventions. **(D-F)** illustrate observed behavior, predicted behavior (D) from the reduced-rank model (E), and residual behavior (F) from individual experimental trials in this session. Refer to Fig 3 (D-F) for conventions.

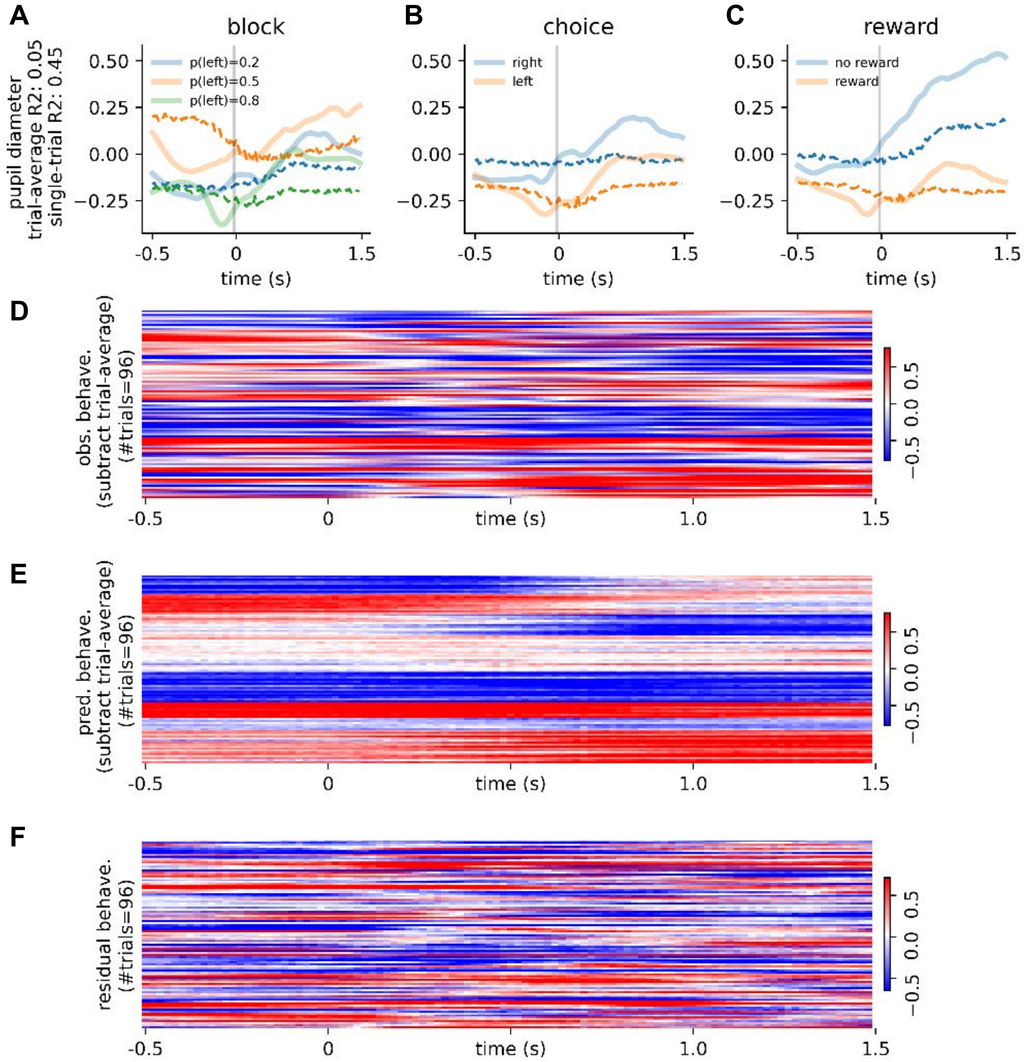

Figure S3: **Evaluating pupil diameter decoding quality using spiking activity from 1313 neurons in a RE dataset session.** **(A-C)** Comparison between the reduced-rank model's predicted pupil diameter (dotted curves) and observed ground truth behavior (solid curves) across different block (A), choice (B) and reward (C) conditions. Refer to Fig 3 (A-C) for conventions. **(D-F)** illustrate observed behavior, predicted behavior (D) from the reduced-rank model (E), and residual behavior (F) from individual experimental trials in this session. Refer to Fig 3 (D-F) for conventions.