Spectral envelopes of facial movements predict intention, cortical representations, and neural prosthetic control

Richard Hakim[1,2], Akshay Jaggi[3], Gyuryang Heo[1,2], Hideyuki Matsumoto[7,8], Naoshige Uchida[8], Mitsuko Watabe-Uchida[8], Sandeep Robert Datta[3], Simon Musall[4,5,6], and Bernardo L. Sabatini[1,2]

[1]*Department of Neurobiology, Howard Hughes Medical Institute, Harvard Medical School, Boston*
[2]*Kempner Institute for the Study of Natural and Artificial Intelligence, Harvard University*
[3]*Department of Neurobiology, Harvard Medical School, Boston*
[4]*Institute of Biological Information Processing (IBI-3), Jülich Research Centre, Jülich, Germany*
[5]*Institute of Biology 2, RWTH Aachen University, Aachen, Germany*
[6]*Faculty of Medicine, Institute of Experimental Epileptology and Cognition Research, University of Bonn, Bonn, Germany*
[7]*Department of Physiology, Osaka Metropolitan University Graduate School of Medicine, Osaka, 545-8585, Japan*
[8]*Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA, USA*

April 22, 2026

1 Abstract

Animals, including humans, use coordinated facial movements to sample the environment, ingest nutrients, and communicate. Rodents, in particular, produce rhythmic facial movements during spontaneous behavior and cognitive tasks. Measuring these movements precisely and linking them to neural activity remains challenging. We introduce face-rhythm, an unsupervised pipeline that combines markerless point tracking, spectral analysis, and non-negative tensor component analysis to decompose facial video into a small set of interpretable components. Applied to videos of mice during a Pavlovian odor-reward task, a brain-machine interface (BMI) task, and free behavior, face-rhythm recovers human-interpretable behaviors such as whisking, sniffing, licking, and more subtle behaviors. The resulting components are consistent across animals, are sufficient to decode task variables or internal belief states, and explain cortical activity using a low-rank representation. We also find that the activity of neurons in face-associated primary motor cortex (M1) is predicted well by a phase-invariant spectral transformation of facial movements above $\sim 0.5\,\mathrm{Hz}$, while slower movements retain a phase-variant representation better predicted by the instantaneous position of the face; individual neurons can show either or both forms of tuning. A systematic comparison against deep-learning point-tracking models, contrastive-learning embeddings, and vision-transformer features places face-rhythm competitively across tasks while also achieving the goal of producing a low-dimensional, interpretable description of rodent facial behavior that is closely linked to cortical activity.

2 Introduction

A central question in motor neuroscience is how neural activity in the motor cortex is transformed into behavior. The face is a uniquely useful system for investigating this question: sensory organs and dozens of muscles are densely clustered in a compact area, enabling a rich repertoire of coordinated movements that serve many ethological roles (Galen 1968). Facial behavior also embodies a high-dimensional projection of internal state (Darwin 1872; Dolensek et al. 2020), such that information about what an animal has experienced, is experiencing, or will do is often represented in the appearance and movements of the face. Although several neural circuits that produce specific facial behaviors have been described, fundamental principles governing how cortical activity is represented in and transformed into coordinated facial move-

ments remain incompletely understood.

Early examples of facial behavior analysis used manual codification to quantify static facial expressions and actions (Ekman and Friesen 1976; Bartlett et al. 1999; Calder et al. 2001; Cohn and Ekman 2008). Although these approaches remain useful, advances in machine learning have since produced more automated, objective, and less invasive methods. Machine vision and unsupervised learning algorithms are now widely used to extract representations of facial pose and movement and to decompose them into lower-dimensional components. Focusing on tools applicable for nonhuman animals and rodents in particular, these methods can be divided into a few categories: pose tracking, pose description, and behavior description. Modern supervised facial pose tracking algorithms such as DeepLabCut (A. Mathis et al. 2018) and SLEAP (Pereira et al. 2022) use fine-tuned neural networks to perform markerless point tracking on defined body parts. The widespread adoption of pose tracking methods has democratized and rapidly advanced behavioral quantification methods, which are now commonly used in a large number of neuroscience studies. Despite ongoing efforts (Ye et al. 2024; Syeda et al. 2024), these tools often require extensive manual curation to track movements collected by different video sources across laboratories. Modern facial pose description methods, including most automatic facial action coding and facial expression recognition systems (Li and Deng 2022), are also highly useful; however, by construction, they do not analyze time-varying patterns. One example in rodents is the recent use of histograms of oriented gradients to describe emotional states (Dolensek et al. 2020). These *pose* description methods apply human-interpretable labels to static poses. In contrast, *behavioral* description methods typically chunk continuous behavior into interpretable 'syllables' of behavior, like rearing or scurrying. Notable models of this class include MoSeq (Wiltschko et al. 2015), B-SOiD (Hsu and Yttri 2021), and VAME (Luxem et al. 2022), among others (Batty et al. 2019; Bohnslav et al. 2021; Mackevicius et al. 2019), which have been applied to whole-body poses. These works highlight the dominant trend in systems neuroscience toward more expressive machine learning methods that consider higher-dimensional movement patterns. However, behavioral segmentation approaches represent movement as a sequence of discrete states, which does not capture the continuous and overlapping nature of the motor patterns that animals can express simultaneously.

Whole-face video recordings can also be used to associate facial movements with neural data. A method utilizing principal component analysis (PCA) of pixel intensity changes called "motion energy" decomposition revealed that a large fraction of neural variance correlates with uninstructed facial movements (Musall et al. 2019; Stringer et al. 2019; Salkoff et al. 2020). FaceMap later demonstrated that markerless point tracking of rodent faces combined with analysis methods that considered temporal movement patterns significantly improved predictions of single-unit neural activity (Syeda et al. 2024). These approaches provide valuable biological insight; however, the latent representations they produce tend to be high-dimensional and challenging to interpret. There remains a need for unsupervised models with greater inductive biases that allow for the generation of low-dimensional and interpretable components that match latent neural activity. Discovering methods with these properties greatly aids in our understanding of the motor system.

Here, we present face-rhythm (FR), an analysis framework and pipeline designed to meet this need. FR performs dense-mesh point tracking, spectrally transforms tracked point movements, and decomposes them into a sparse set of interpretable components. This approach is motivated by studies of the function and connectivity of central pattern generators for facial movements (Cramer and Keller 2006; Kaku et al. 2025; Kleinfeld et al. 2023; Moore et al. 2013), as well as the success of multi-way tensor factorization methods in finding interpretable components within single-unit neural data (Williams et al. 2018) and time-frequency representations of EEG signals (Mørup, Hansen, and Arnfred 2007). Spectral representations are especially natural for central pattern generator (CPG)-driven behaviors because CPGs produce intrinsically oscillatory output, and descending cortical commands may modulate the amplitude envelope of these oscillations rather than their instantaneous phase. In essence, FR aims to discover and demix facial behaviors that overlap in space, time, and spectral patterning to reveal latent "rhythms".

The architecture of the FR pipeline is designed to discover short temporal patterns (milliseconds to seconds) that can be mapped onto known behaviors in rodents (i.e., whisking, sniffing, licking, and more subtle motifs) or associated with performance variables in cognitive tasks. Consistent with achieving this goal, the application of FR to facial video data acquired during classical conditioning and brain-machine interface (BMI) tasks successfully decodes cognitive task variables and performance using a small set of FR

components. In addition, due to the existence of specialized motor circuitry for generating rhythmic facial behaviors in rodents, we hypothesize that the FR-identified patterns of facial movement map onto the natural modes of activity in the facial motor cortex. Supporting this hypothesis, we find that the temporal factors identified by FR align closely with the modes of neural population activity in the facial motor cortex. We explain these observations by building a regression model that includes an explicit spectral transformation of facial movements to predict facial motor cortex activity; we find that higher frequency movements (greater than ~ 0.5 Hz) are more likely to be represented in the cortex as a phase-invariant spectral envelope. We also compare FR against deep-learning methods for point tracking and continuous behavior embedding to characterize the conditions under which each approach excels. FR thus provides a method that decomposes spontaneous and operant facial behavior into interpretable movements that are consistent across animals and that are represented in the activity of facial motor cortex.

The code to run the face-rhythm pipeline was refined for efficiency and general use and is made publicly available here: https://github.com/RichieHakim/face-rhythm

3 Results

3.1 Face-rhythm methodology

The face-rhythm (FR) pipeline consists of 3 major components: 1) unsupervised and markerless keypoint tracking of a mesh of points distributed across the face; 2) transformation of the point-tracking signals into spectrograms; and 3) decomposition of a 3-dimensional tensor (spatial location by frequency by time) into components that describe discrete parts of behavior (**Figure 1A**, and see **Methods 5.1**). The standard output of the pipeline is a set of components that each describes a single movement pattern. Each FR 'component' is defined by three 'factors': a spatial factor corresponding to the points on the face that move, a spectral factor that captures the dominant frequencies of the movement of those points, and a temporal factor that describes the movement magnitude of the component in each video frame (**Figure 1B-C**).

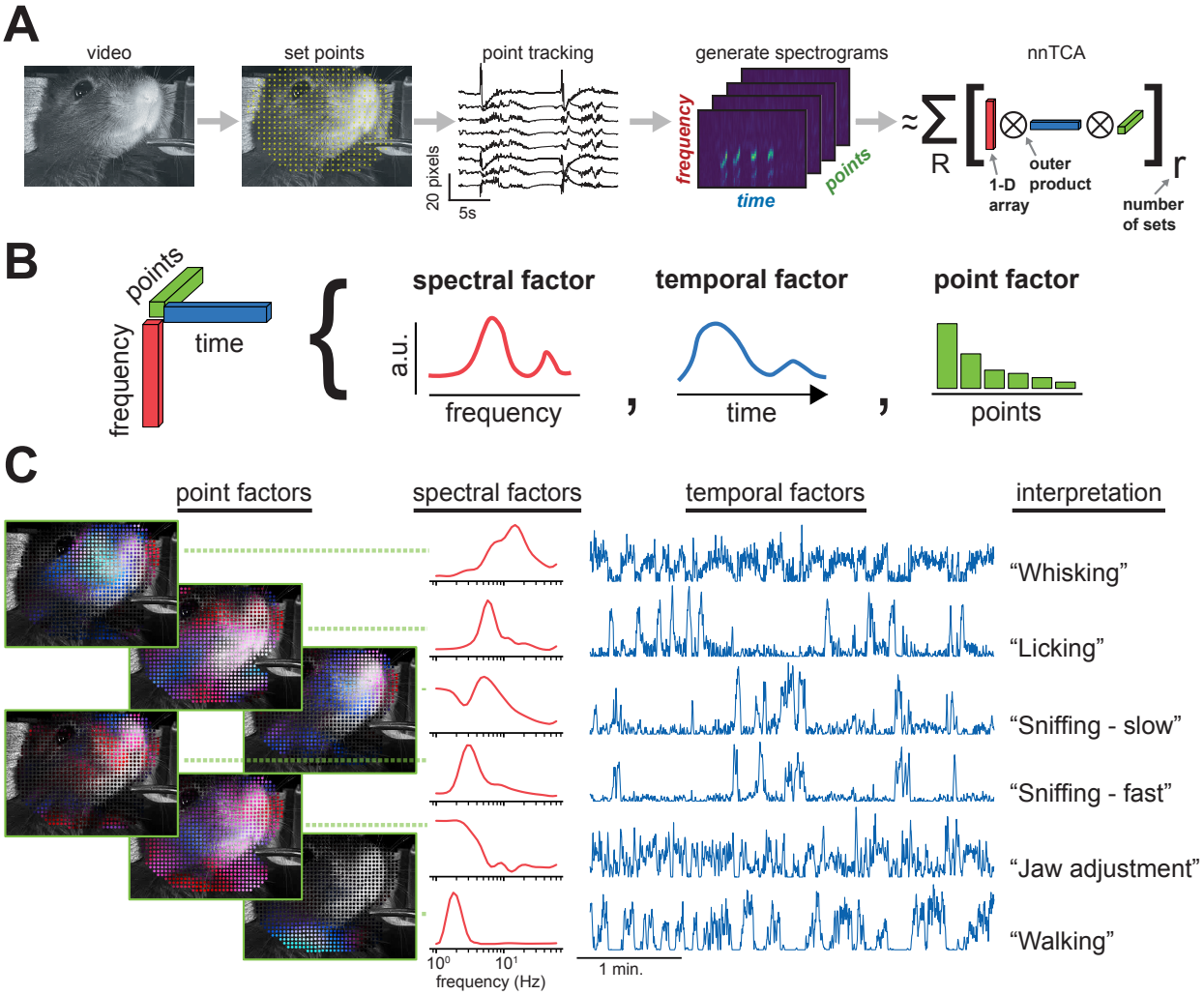


Figure 1. Face-rhythm pipeline. Overview of the face-rhythm pipeline and its output components. A) Schematic of face-rhythm pipeline for spectrotemporal decomposition of facial movements. Video of a face is used as input. Point tracking is performed using a modified Lucas-Kanade algorithm. Spectrograms are generated for each point and each direction ('x' and 'y') using a variable-Q transform. Non-negative tensor component analysis is performed using non-negative PARAFAC. B) Schematic of the output components. Each output component is comprised of 3 vectors: a spectral / frequency factor describing the spectral pattern of the movement, a temporal factor describing when the component is active, and a spatial / points factor describing where the component is active. A typical decomposition compresses the input tensor into a set of these components, which define the 'rank' of the reconstruction. C) Representative examples of face-rhythm components. Colors match the dimensions in (A) and (B). Left: Point / positional factors. Color saturation describes magnitude and color hue describes angle, where horizontal amplitude ('x') is blue and vertical amplitude ('y') is red. Middle-left: spectral / frequency factors. Middle-right: temporal factors. Right: interpretation of what each component represents.

The specific algorithms used in the pipeline are high-performing unsupervised methods that require minimal parameter tuning. In brief, markerless tracking is performed using a modified Lucas-Kanade algorithm (Lucas and Kanade 1981; Baker and Matthews 2004) further enhanced with mesh physics. The spectrograms are computed using a custom implementation of the variable-Q transform (Brown 1991; Chatterji et al. 2004), and the decomposition is performed using non-negative tensor component analysis (nnTCA, also known as non-negative PARAFAC) (Carroll and Chang 1970; Paatero and Tapper 1994; Harshman 1970;

Williams et al. 2018). Behavioral videos should ideally be captured at a frame rate greater than twice the frequency of the fastest movements of the subject (given the Nyquist-Shannon sampling theorem (Shannon 1949)). For rodent whisking, this is at least 30 Hz and ideally above 90 Hz. A description of the shape of the data as they pass through the pipeline is useful for understanding its processing: the input video data is of shape (*height, width, time*), the output of the point-tracking is of shape (*num_points*, 2, *time*) (where '2' indicates the 2 dimensions corresponding to x- and y-displacements), the output of the spectrograms is of shape (*num_points*, 2, *frequencies*, *binned_time*), and the output of the PARAFAC decomposition is a set of trios of vectors of lengths ($2\times$*num_points*), (*frequencies*), and (*binned_time*). The size of the set, R, is determined by the user; in practice for rodent facial behavior, we find that the number of discriminable and human-interpretable behavioral components ranges around 4–12. Quantitative methods for rank selection vary depending on the use case (Williams et al. 2018); here, we select rank based on a combination of reconstruction error, component consistency across random initializations, Bayesian information criterion, and human interpretability (see **Methods 5.1.5**), with diagnostics summarized in **Figure 1 supplement 1**. This process results in a discrete set of R components, each of which corresponds to a single behavioral pattern that describes a set of points on the face that move at the same time and at similar frequencies.

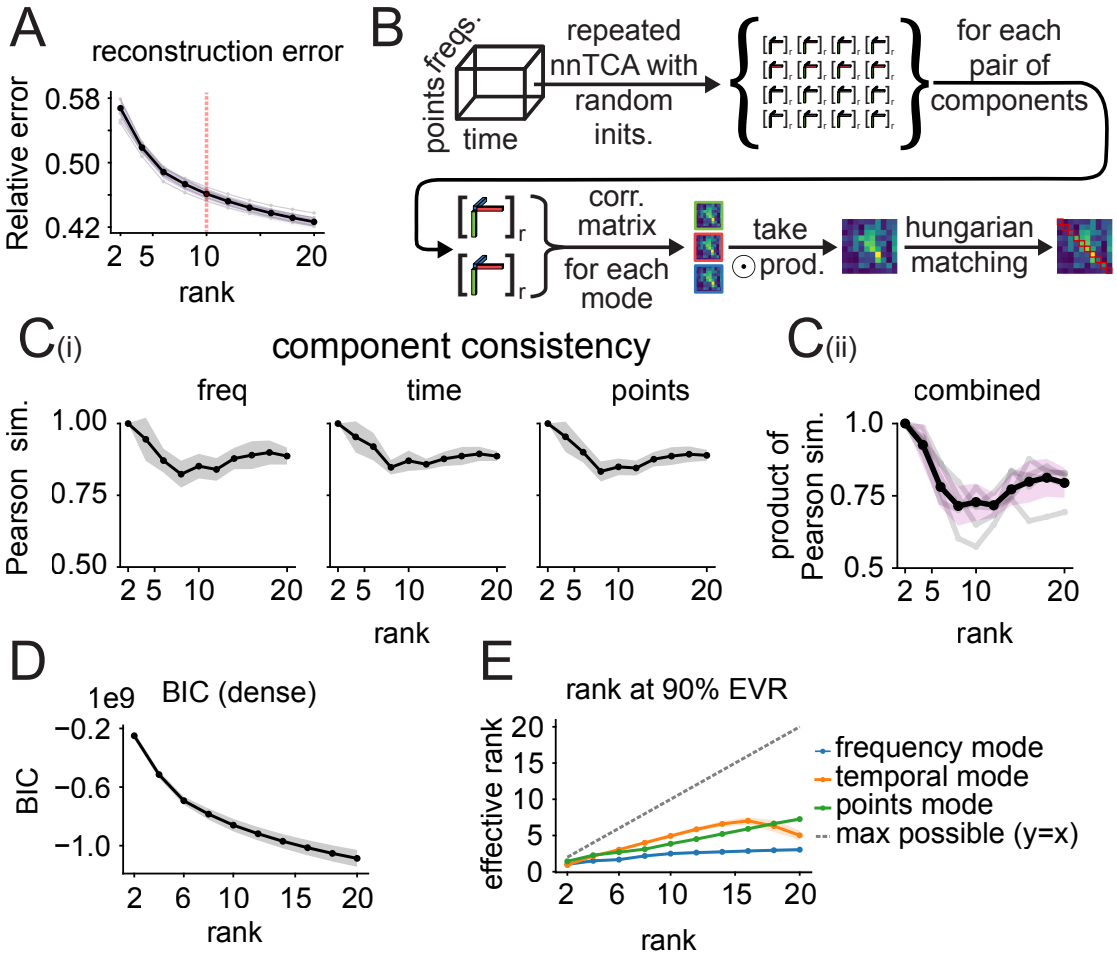


Figure 1 supplement 1. Tensor decomposition rank selection and component consistency. Rank-selection and component-consistency diagnostics for non-negative TCA. A) Relative reconstruction error versus rank for non-negative tensor component analysis (nnTCA) decomposition of the face-rhythm spectrogram tensor; dashed red line marks the rank used for downstream analyses. B) Schematic of the consistency analysis across random initializations. For each rank, non-negative TCA (nnTCA) was repeated with random starts; mode-wise component similarity matrices (frequency, temporal, and points) were computed, multiplied

across modes, and components were aligned with Hungarian matching. C) Component consistency across rank. Ci) Per-mode Pearson similarity between matched components (frequency, temporal, points). Cii) Combined consistency score computed as the product of mode-wise Pearson similarities. In (A) and (C), black lines show the mean across sessions and shaded regions show ± 1 s.d. ($N = 5$ mice, $n = 13$ sessions); light traces show individual sessions. D) Bayesian information criterion (BIC) versus rank under the dense Gaussian residual approximation. E) Effective rank/dimensionality of each factor matrix, defined as the number of principal components required to explain 90% of the variance in each mode; dashed gray line denotes the maximal possible value ($y = x$).

The output components of this pipeline have properties that are suitable for defining discrete behaviors. Spectral analysis results in non-negative data, which naturally allows for a non-negative decomposition constraint. Such matrix and tensor factorization methods are known to extract a 'parts-based' representation that tends to yield interpretable and 'demixed' components (D. D. Lee and Seung 1999). We find that, when applied to videos of mouse faces, FR components correspond to human-recognizable components of facial behaviors such as whisking, licking, and multiple forms of sniffing and twitching (**Figure 1C**) (**Supplementary Movie 1–2**). This approach does not require training neural networks, maintains high interpretability, and can be used to analyze movements of individual mice captured in single video sessions.

The non-negative tensor factorization method used does not impose an orthogonality constraint on the output components (as is seen in principal component analysis). An intuitive consequence of this is that multiple components can be simultaneously expressed and contribute to compound or correlated behaviors. For example, whisking and sniffing are often expressed together, but are typically decomposed as separate components that can be coactive. More precisely, the set of all components sums to approximate the magnitude spectrograms from each point tracking signal, and each component tends to align with a statistical mode of the data that is typically interpretable as a discrete behavioral pattern.

FR components can be matched across different animals in similar environmental conditions. We matched nnTCA components across 5 mice using Hungarian assignment (Williams et al. 2018) of FR spectral and spatial factors and found high cross-animal similarity significantly exceeding chance at all tested ranks (**Figure 1 supplement 2**; $p < 0.001$, permutation test; see **Methods 5.2**; 13 sessions; animals are head-fixed on a linear treadmill with infrequent random water rewards and random sensory stimuli; see **Methods 5.11.2**).

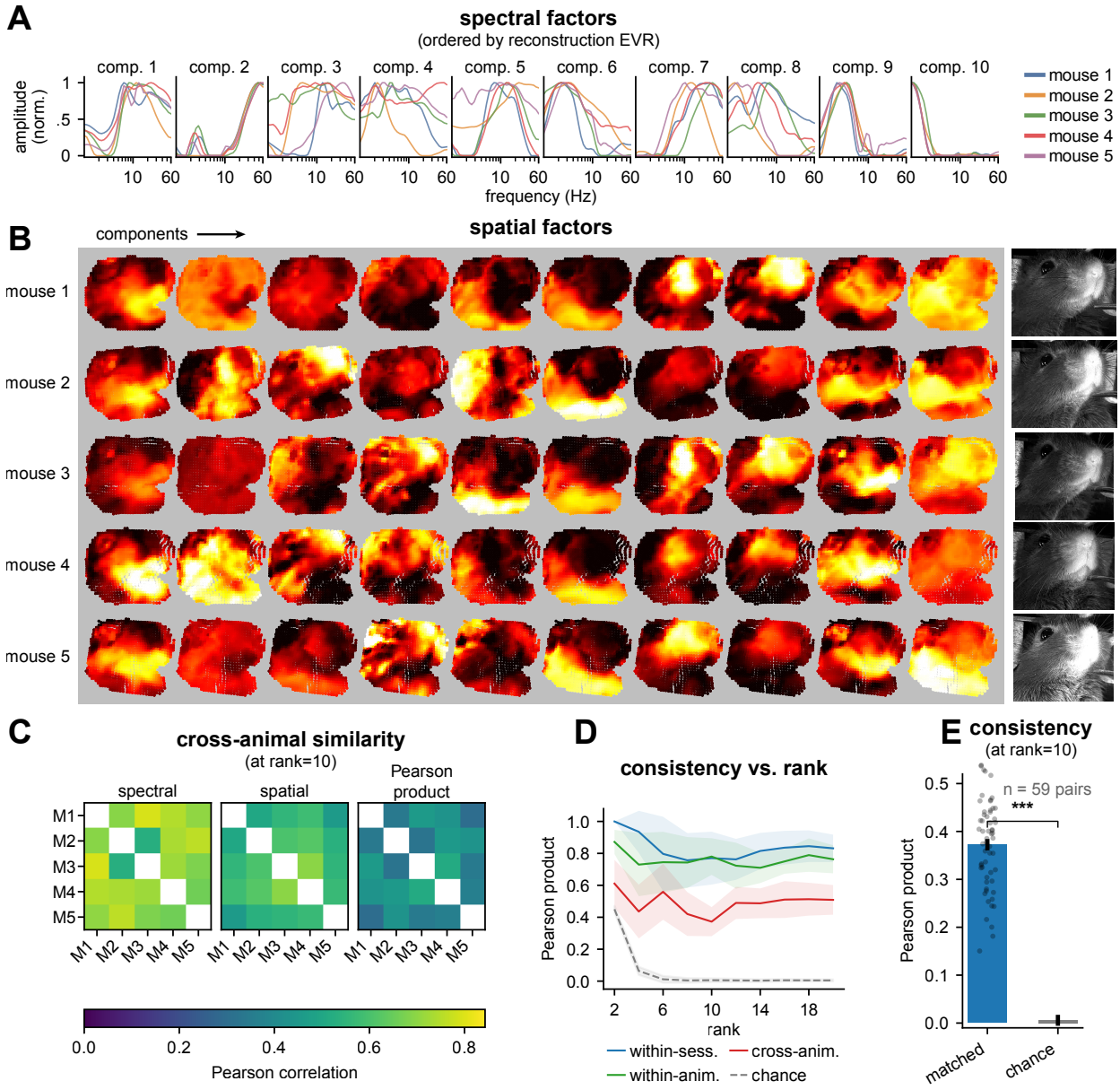


Figure 1 supplement 2. Cross-animal consistency of face-rhythm components. Cross-animal component matching analysis. The face-rhythm pipeline is run on each mouse independently, and the Hungarian algorithm is used to match components across animals using the product of spectral and spatial factors similarity matrices (see Methods 5.2). A) Matched spectral factors for all 10 components at rank 10 across mice, ordered (highest to lowest) by average marginal explained variance ratio (EVR) of the original spectrogram tensor. B) Matched spatial/point factors (magnitude of x and y amplitudes) for the same 10 components (columns matched to A) across all 5 mice (rows). On the right are reference camera images. C) Cross-animal pairwise similarity heatmaps at rank 10. Left: mean spectral Pearson correlation. Center: mean spatial Pearson correlation. Right: product of spectral and spatial correlations. Each element shows the average similarity over all session pairs between the indicated mouse pair. D) Pearson product score (spectral $\times$ spatial Pearson correlation) as a function of non-negative tensor component analysis (nnTCA) rank ($R \in \{2, 4, \ldots, 20\}$). Three comparison levels are shown: within-session over random initialization repetitions (blue), within-animal cross-session (green), and cross-animal (red). Dashed gray line: chance level from permutation null instead of using Hungarian matching indices. Shaded bands are standard deviations

(13 sessions, 5 mice). E) Consistency statistics at rank 10. Bar heights show the mean Pearson product score for optimally matched components and random component assignment. Error bars: s.e.m. for matched, s.d. for chance. Individual dots show per-pair scores ($n = 59$ cross-animal session pairs). ***: $p < 0.001$, permutation test (1000 shuffles per pair).

To examine the impact of the three corrections applied during Lucas-Kanade point tracking (mesh physics, slow relaxation, and displacement violation detection with point pausing), we performed an ablation analysis (**Figure 1 supplement 3**; see **Methods 5.3**). The complete FR model maintains stable tracking throughout ~60 min recordings, and each modification targets a distinct failure mode. Removing relaxation produces monotonic drift away from the initial point positions (**Figure 1 supplement 3A, D**; with-R {MRV, RV} versus without-R {MV, V} on final drift, $p = 0.002$, one-sided Wilcoxon signed-rank, $n = 9$ sessions from 7 mice). Removing mesh physics collapses local neighbor velocity coherence (**Figure 1 supplement 3C, F middle**; with-M {MRV, MV} versus without-M {RV, V}, $p = 0.002$). Removing violation detection allows catastrophic deformations from occlusive events such as grooming, and inflates the violation rate (**Figure 1 supplement 3E, F right**; with-V {MRV, MV, RV, V} versus without-V {MR, M, R, none} on violation rate, $p = 0.002$). Together, these results demonstrate that each correction makes a non-redundant contribution to long, stable point tracking.

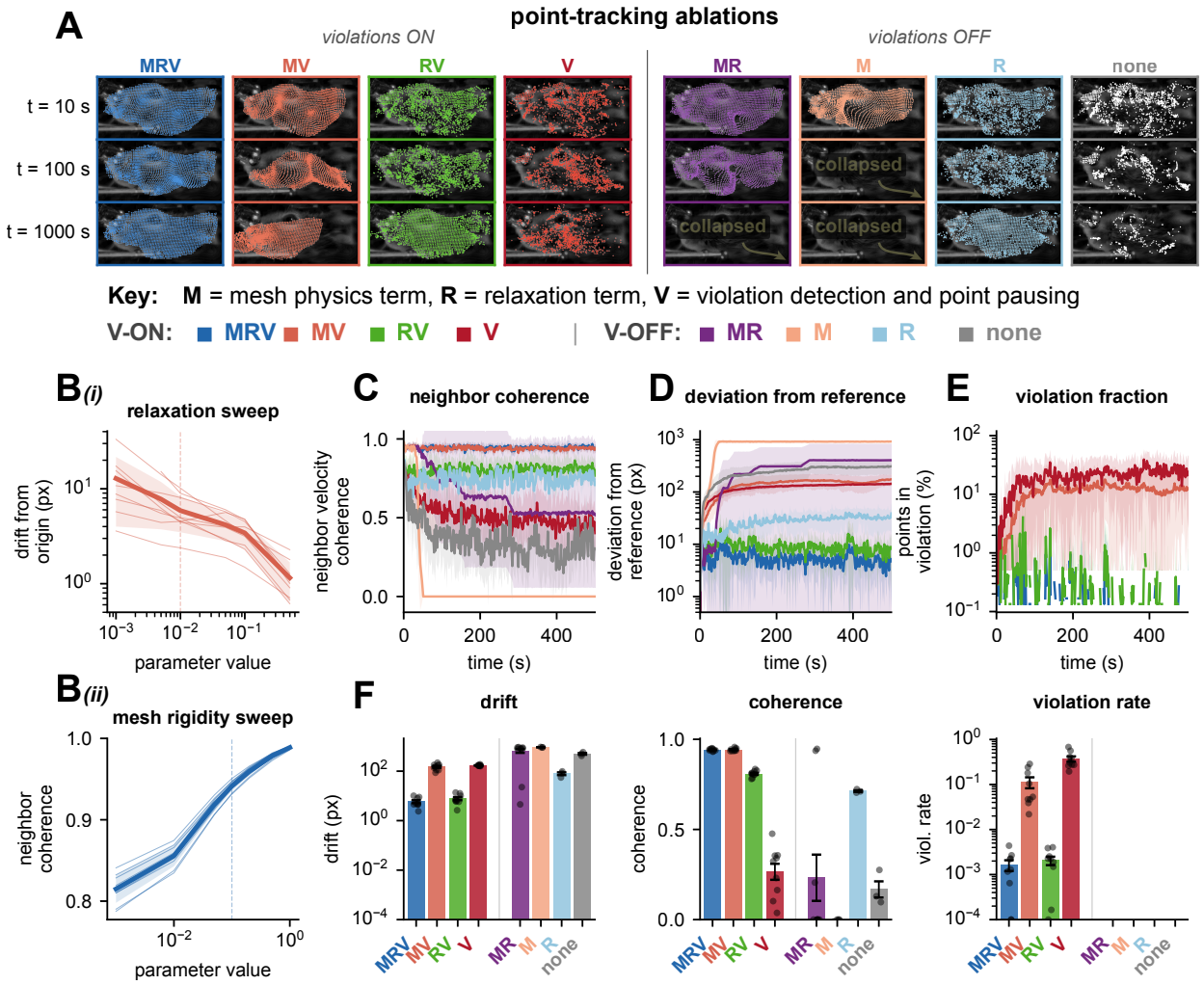


Figure 1 supplement 3. Ablation analysis of Lucas-Kanade point-tracking corrections. Ablation of Lucas-

Kanade point-tracking corrections. A) Face overlays showing tracked point positions at three timepoints ($t = 10$ s, 100 s, 1000 s) for each of eight ablation conditions (columns). Conditions describe different additions to the Lucas-Kanade integrated optical flow point-tracking algorithm where M = mesh physics constraint, R = relaxation, V = violation detection with point pausing. Conditions are grouped by violations ON (MRV, MV, RV, V) and violations OFF (MR, M, R, none); colored dots indicate tracked points overlaid on a reference frame from a representative session of the widefield calcium imaging dataset. B) Parameter sensitivity. Bi) Cumulative drift from origin as a function of relaxation strength (λ_{relax}; dashed value 0.01 was used for final analysis). Bii) Neighbor velocity coherence as a function of mesh rigidity (λ_{mesh}; dashed value 0.1 was used for final analysis). Thin lines: individual sessions; bold line: across-session mean; shaded band: ± 1 s.d. C) Local neighborhood point tracking time series coherence (mean cosine similarity of frame-to-frame displacement vectors between each point and its 4 nearest spatial neighbors) over the first 500 seconds of the recording. D) Deviation (Euclidean distance) from original point positions over time ($\log y$). E) Fraction of points in violation (see Methods 5.3) over time. F) Summary plots for cumulative drift (left), final neighbor coherence (middle), and violation rate (right). Bars are across-session means; dots are individual sessions; error bars are s.e.m. $n = 9$ sessions from 7 mice.

3.2 Comparison with motion energy principal component analysis

Given that FR components align with natural facial behaviors, we examined how well FR temporal factors explain neural activity (**Figure 2**). We used an established experimental approach to quantify how well facial behavior data can explain neural activity data (Musall et al. 2019), and compared FR temporal factors against other common methods like video motion energy PCA (ME PCA). We used the standard FR pipeline to perform point tracking and decompose facial movement (**Figure 2A**) and monitored neuronal activity with widefield fluorescence imaging of calcium transients reported by GCaMP6s. Neural activity was recorded cortex-wide in mice performing a visual discrimination task (**Figure 2B**, **Supplementary Movie 3**). The task structure includes a visual stimulus, a subsequent delay, and a response epoch during which mice can respond to the stimulus by licking one of two water spouts. The posterior visual cortical regions respond strongest to the visual stimulus, whereas medial regions respond persistently during the delay period, and frontal regions are most active during the response period (**Figure 2B**).

To link cortical activity to the behavioral video data, we fit a ridge regression model that explains neural activity (as measured by fluorescence transients) in each pixel of the widefield imaging as a weighted combination of different behavioral inputs, such as FR components or point tracking data (**Figure 2C**). Hierarchical regression shows that FR temporal factors encode a low-rank representation that is significantly more aligned with cortical activity patterns than ME PCA (**Figure 2D**). In line with previous findings (Musall et al. 2019), the cross-validated explained variance ratio (EVR) of the model is consistently high across much of the cortex, with particularly strong performance in anterior regions. This shows that behavioral video data provides a robust basis for predicting cortical activity (**Figure 2E**). In a direct comparison of performance, the EVR obtained using only 10 FR components (0.245 ± 0.087 s.d.) exceeds that of 10 ME PCA components (0.144 ± 0.078 s.d.) ($p = 0.0039$, Wilcoxon signed-rank test, $n = 9$ sessions from 7 mice; linear mixed-effects model (LMM) accounting for repeated sessions within mice: $p = 4.27 \times 10^{-8}$) and approaches the performance achieved with 200 ME PCA components (0.279 ± 0.066 s.d.), indicating that FR components are more predictive of cortical dynamics at matched rank.

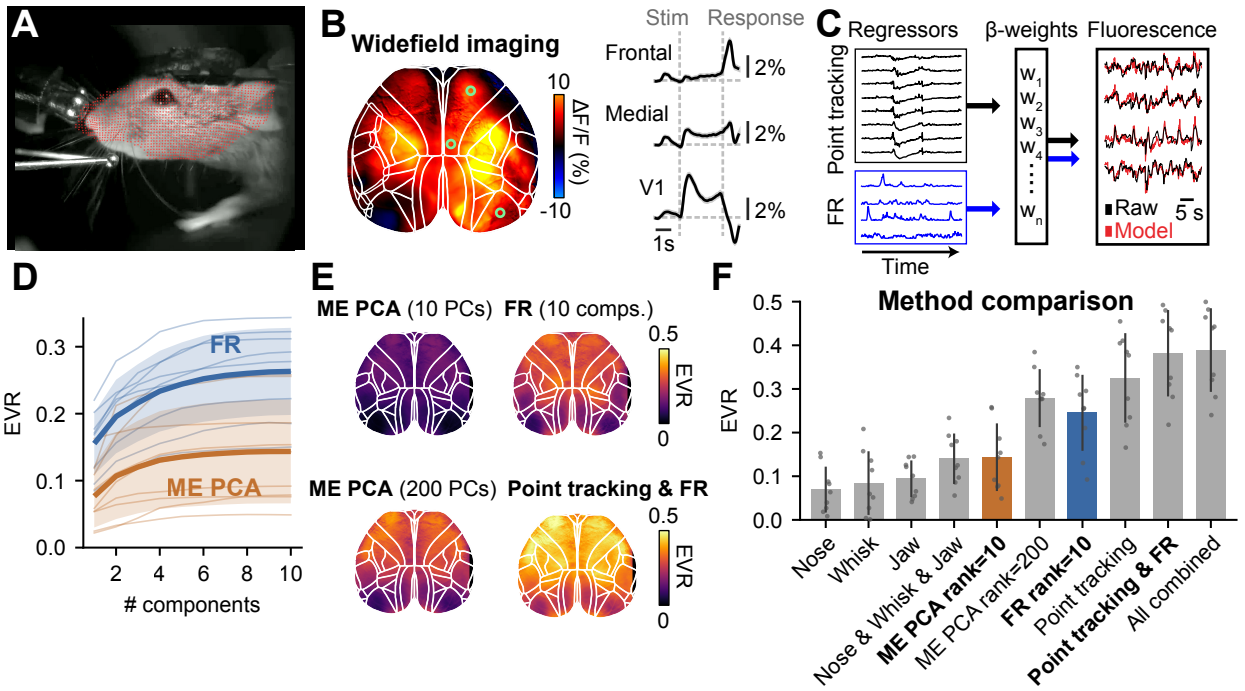


Figure 2. Face-rhythm factors predict cortex-wide widefield neural activity. Prediction of cortex-wide widefield activity from face-rhythm factors. A) Left: Example image of a task-performing mouse on a running wheel with tracked points in red. B) Simultaneous widefield imaging was used to record cortex-wide neural activity. The left image shows a single example frame; the traces on the right show trial-averaged activity from the same session in different brain regions. The region locations (primary visual cortex (V1), medial, and frontal cortex from posterior to anterior) are indicated as light green circles on the left image. C) Example schematic of the ridge regression model, showing how point tracking and FR factors can be combined to predict changes in cortical imaging data. D) Hierarchical regression of increasing numbers of factors. Factors are either video motion energy PCs (ME PCA) or FR temporal factors (FR). Error bars are 95% confidence intervals ($N = 7$ mice, 9 sessions total). E) Explained variance ratio averaged across mice ($N = 7$) for each pixel in the widefield imaging data based on ME PCA, FR, or a combination of FR and point tracking. F) Comparison of mean EVR across cortex for different regressor combinations. Error bars are s.e.m. across mice ($N = 7$).

A more detailed comparison of the EVR for specific behavioral variables, such as whisking or nose movements, further supports the low-rank explanatory power of FR components (**Figure 2F**). A model based on 10 FR components (0.245 ± 0.087 s.d.) is significantly more predictive than an alternative model using a combination of whisker, nose, and jaw movements (0.143 ± 0.062 s.d.; $p = 0.008$, Wilcoxon signed-rank test, $n = 9$ sessions from 7 mice). Appending the rank-10 FR components as additional predictors to the high-dimensional point tracking data further improves the EVR from 0.33 ± 0.10 to 0.38 ± 0.10 s.d., suggesting that the cortex maintains a separable representation of the spectral envelope of facial movement. However, further appending all signals together only modestly improves the EVR (0.38 ± 0.10 to 0.39 ± 0.10 s.d.), demonstrating that high-dimensional point-tracking and FR signals largely subsume the utility of ME PCA and specific facial body part signals. Together, these results demonstrate that spectral transformation and decomposition via tensor factorization provide useful low-dimensional behavioral components with high explanatory power that are closely aligned with the dominant modes of cortical activity.

To test the utility of different steps in the FR pipeline in producing a model that predicts neural activity, we reanalyzed the widefield imaging data after ablating portions of the FR pipeline (**Figure 2 supplement 1B**). These results indicate that spectrogram generation is a critical transformation (see **Methods 5.4.3**). Among low-rank decompositions (~10 components), the standard FR pipeline (non-negative TCA on VQT

spectrograms) achieves the highest cross-validated EVR. Spectrogram-domain decompositions consistently outperform point-tracking-domain methods at matched rank, confirming that the VQT spectral transformation is a critical step.

To further test whether FR's advantage depends on the specific tracker or on the spectral decomposition framework, we compared six input sources—FR, FaceMap (Syeda et al. 2024), CoTracker (Karaev et al. 2024) (a transformer-based dense tracker applied at FR mesh positions), CEBRA (Schneider, J. H. Lee, and M. W. Mathis 2023) (contrastive-learning embeddings of FR tracking data at dimensions 10 and 50), and frozen ViT-MAE embeddings (He et al. 2021) (768-dimensional CLS-token features per frame, motivated by BEAST (Y. Wang et al. 2025)). Each is processed through the same decomposition methods (**Figure 2 supplement 1C**).

We find that FR-derived decompositions perform favorably against other keypoint-tracking-based approaches across nearly all decomposition methods. Deep network approaches that are trained by reconstructing raw image/video representations such as frozen ViT-MAE also perform well; however, they can use extra information about the frame such as hand and body posture, can require extensive fine-tuning, and their embeddings are less directly relatable to fiducial points or muscle dynamics. We conclude that FR point-tracking provides a strong basis for decomposing facial behavior into a low-rank representation that is predictive of widefield cortical activity, and that spectral transformation plus non-negative TCA steps are critical for achieving this performance at low-rank.

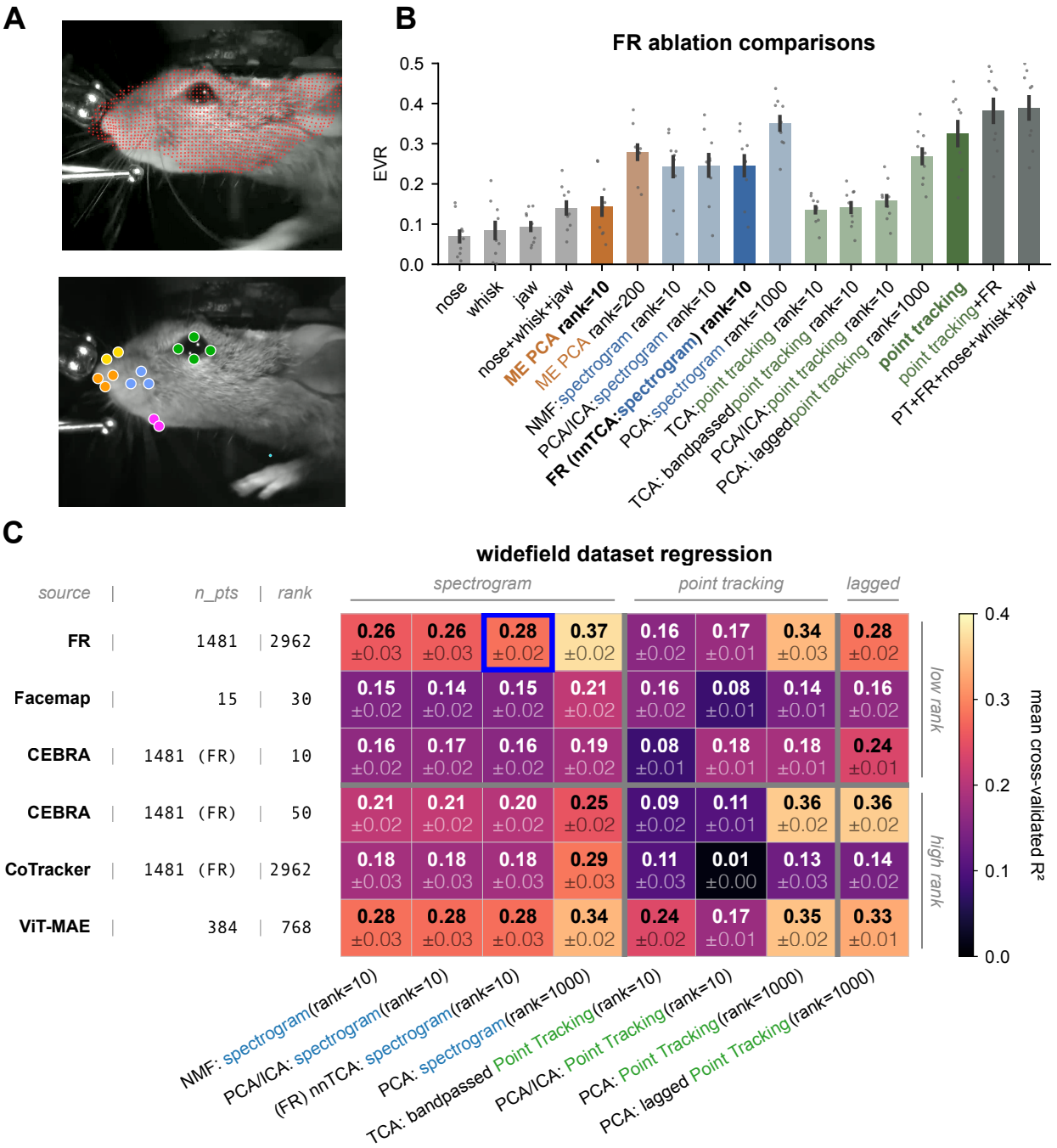


Figure 2 supplement 1. Comparison of keypoint-tracking and decomposition methods for predicting widefield cortical activity. Comparison of trackers and decomposition methods for predicting widefield cortical activity. A) Example frames showing face-rhythm (FR) dense-mesh point tracking (top; ∼1,481 points, red) and FaceMap keypoint tracking (bottom; 15 points, colored dots). B) FR pipeline comparison of decomposition methods against motion energy PCA (ME PCA). Bar height describes cross-validated explained variance ratio (EVR) of widefield GCaMP6s fluorescence time series data using different behavioral regressors derived from FR point-tracking data and motion energy PCA. Leftmost bars (nose, whisk, jaw) describe average ME of pixels within each area; blue bars describe different decompositions applied to FR point-tracking + spectrogram generation data; green bars describe different decompositions applied to the raw FR point-tracking data (matrix methods operate on the flattened $2P \times T$ displacement matrix after con-

catenating x- and y-coordinates; "TCA: point tracking" operates directly on the original 3-way $P \times 2 \times T$ tensor; see Methods 5.4.3 for details); rightmost bars describe combinations of regressors from previous bars. Dots indicate individual recording sessions; error bars are standard deviation ($N = 7$ mice, 9 sessions total). C) Peer method comparison of behavioral input representations for predicting widefield cortical activity. Rows indicate input source (n_pts: number of keypoints, if relevant; rank: input feature dimensionality, i.e., total feature count of the input source); note that ViT-MAE does not use keypoints. Columns indicate decomposition methods grouped by domain (spectrogram, point tracking). For columns, the displayed rank (e.g., rank 1000) is the requested upper bound on the number of components; the achieved rank is min(requested rank, input feature dimensionality), so for low-dimensional input rows the effective rank is capped at the input feature count rather than the requested column rank. Cell values show mean cross-validated $R^2 \pm$ standard error of the mean ($N = 7$ mice, 9 sessions; 1–2 sessions per mouse). Top row shows the standard FR pipeline, and the blue bordered cell describes the standard FR output. Input sources: FR (face-rhythm Lucas-Kanade tracker); FaceMap (Syeda et al. 2024); CEBRA using FR point-tracking data (Schneider, J. H. Lee, and M. W. Mathis 2023); CoTracker using FR point-tracking data (Karaev et al. 2024); ViT-MAE (frozen), CLS-token embeddings (768 dimensions) (He et al. 2021; Y. Wang et al. 2025).

To characterize computational performance, we benchmarked the speed/throughput and memory consumption of FR against FaceMap (**Figure 2 supplement 2**). FR runs at around 67 frames/s on a CPU (Intel i9-10980XE processor from 2019) and around 197–300 frames/s on a GPU (NVIDIA RTX 3090 GPU from 2020; the higher value is achieved when using hardware-accelerated NVDEC video decoding). Peak memory usage for FR is around 17 GB for a $\sim$60 min video (1280×1024 pixels); when using GPU video decoding, about 20 GB of VRAM is required (though this is user-configurable and smaller buffers work as well). Overall, time complexity scales linearly with video duration and is dominated by point tracking; the subsequent VQT and nnTCA steps operate on compact representations and are comparatively fast. Memory usage tends to be highest during the nnTCA step, but also runs well on consumer hardware (see **Methods 5.15**).

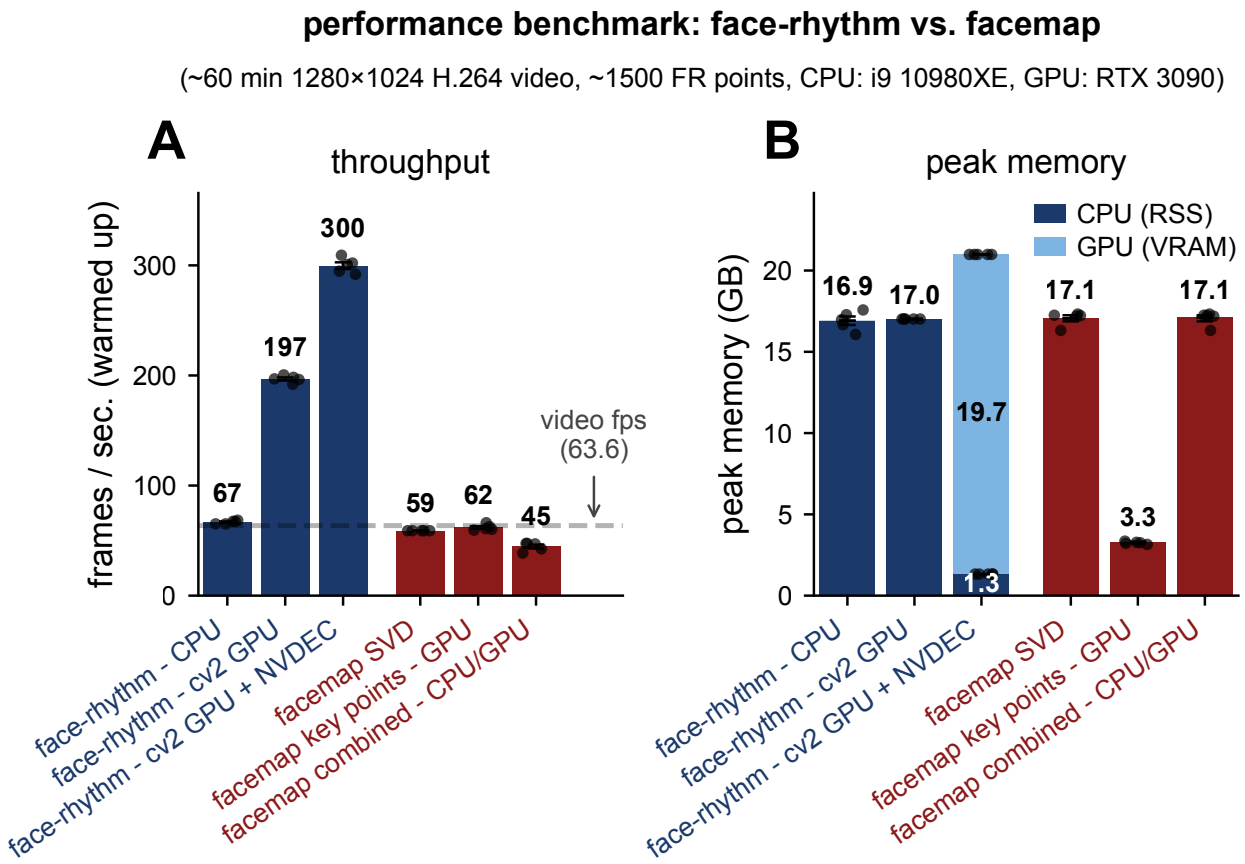


Figure 2 supplement 2. Speed and memory benchmarks. Throughput and memory benchmarks comparing face-rhythm and FaceMap. A) Throughput benchmark comparing face-rhythm (FR; blue) and FaceMap (red) on the same $\sim$60 min 1280×1024 H.264 video with the same hardware (Intel i9-10980XE CPU and NVIDIA RTX 3090 GPU). Bars show processing speed (frames/s) under a variety of conditions. Throughput does not consider any preprocessing / warm-up period and is more reflective of how these methods scale; for a $\sim$60 min video warm-up time is small. Conditions: FR-CPU is face-rhythm running without GPU support, FR-cv2 GPU uses the CUDA/GPU backend for OpenCV, FR-cv2 GPU+NVDEC adds rapid dataloading using the NVDEC codec on GPU, FaceMap-SVD uses only CPU, FaceMap key points-GPU leverages a neural network running on a GPU, and FaceMap combined-CPU/GPU is simply the combination of these two methods which can run together. Values are annotated above bars, black dots indicate individual sessions ($n = 4$), error bars are s.e.m. across sessions, and the dashed line marks source video rate (63.56 Hz). These data are from the same recording as in **Figure 2**. B) Peak memory usage for the same modes. Dark bars indicate CPU resident-set size (RSS) and light bars indicate GPU VRAM where applicable.

The utility of the non-negative TCA step is not immediately evident from the regression analysis; however, a more qualitative analysis of the extracted components reveals that nnTCA yields more interpretable components. By comparing nnTCA with standard PCA and matching the resulting component pairs (**Figure 2 supplement 3**), we observe that nnTCA factors are generally more population sparse, decorrelated, and demixed across all dimensions, and appear to be more behavioral-mode aligned than PCA factors (**Figure 2 supplement 3A-C**). For example, the high-frequency whisking factor in the bottom panel appears specifically over the whisker pad.

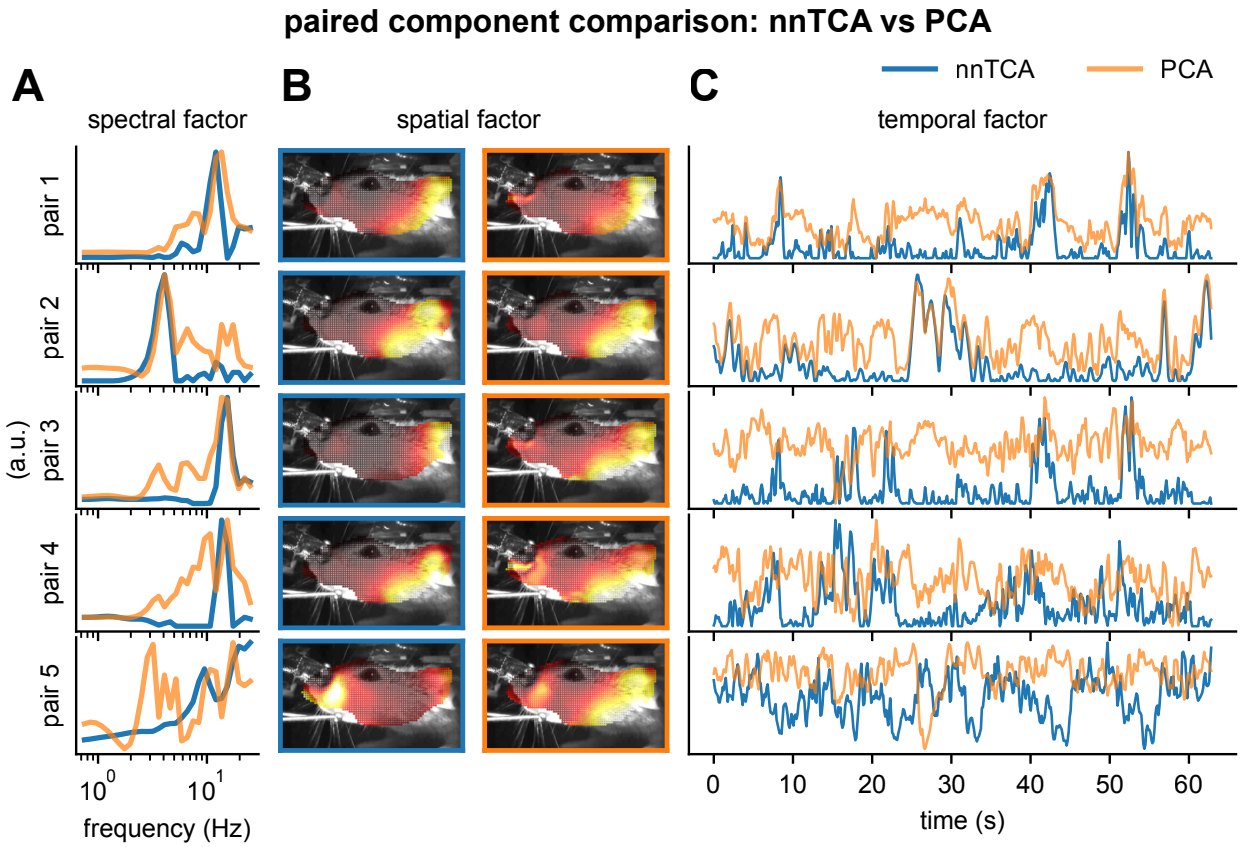


Figure 2 supplement 3. Comparison of non-negative TCA and PCA decompositions of the spectrogram tensor. Paired comparison of non-negative TCA and PCA factors on the spectrogram tensor. A) Spectral factors for five example paired components comparing face-rhythm non-negative TCA (blue) and standard PCA on the flattened spectrogram tensor (orange). Rows indicate component pairs that were matched via Hungarian-algorithm matching on their temporal factors. B) Spatial/point factors for the same component pairs shown in (A) and (C), overlaid on an example face frame. Left column: nnTCA spatial factors. Right column: PCA spatial factors. Colored borders indicate method (blue, nnTCA; orange, PCA). C) Temporal factors for the same five paired components over time (s). Data shown from a single representative session.

3.3 Conditioning of facial behaviors during reward-guided tasks

We examined whether FR captures uninstructed facial movements that are predictive of the conditioned-stimulus identity during Pavlovian/classical conditioning. We analyzed a separate facial video dataset acquired during the reward probability conditioning of a previously described task (Matsumoto et al. 2016) (**Figure 3**; see **Methods 5.5**). In this paradigm, head-restrained thirsty mice are presented with one of three conditioned stimuli (CS+, CS0, or CS-); after a 1 s odor presentation and 1 s delay, mice receive water, no outcome, or air puff, respectively (**Figure 3A**). FR components extracted from these videos show CS-dependent temporal modulation during the cue and delay epochs, including factors consistent with jaw/licking movements for CS+, respiration-related movement for CS0, and ear twitching for CS- (**Figure 3B**, **Figure 3D-E**). Using elastic net multinomial logistic regression on FR temporal factors across 20 sessions from 5 mice, CS identity is decoded with high accuracy: 0.92 ± 0.05, 0.87 ± 0.03, and 0.83 ± 0.01 (mean $\pm$ s.e.m.) for CS+, CS0, and CS-, respectively (all $p < 0.003$, Bonferroni-corrected across the 3 CS classes, one-sample t-test against chance level of $1/3$, $N = 5$ mice; **Figure 3F**; see **Methods 5.5**).

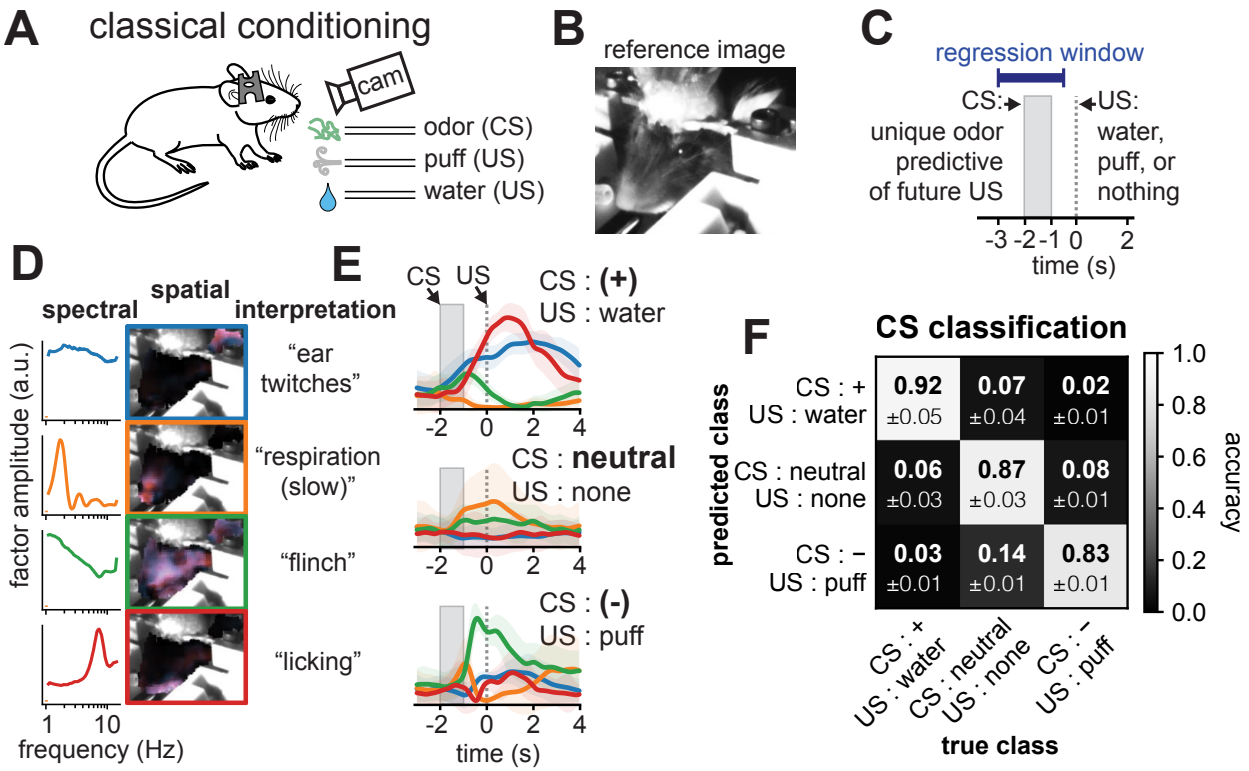


Figure 3. Facial dynamics classify conditioned stimulus identity during classical conditioning. Classification of conditioned stimulus identity from face-rhythm factors during classical conditioning. A) Diagram of the experiment. Face video was recorded during classical conditioning with three odors (conditioned stimulus; CS) followed by appetitive (water), neutral (nothing), or aversive (air puff) outcomes (unconditioned stimulus; US). B) Field-of-view of camera. C) Diagram for the classification analysis. Face-rhythm temporal factor dynamics during the CS and delay window are used in the regression model to predict CS class identity. Elastic net multinomial logistic regression is used; see Methods 5.5.2 for model details. D) Representative examples of extracted face-rhythm components. Left panels: frequency/spectral factors. Middle panels: spatial/positional factors. Blue represents horizontal ('x') amplitude and red represents vertical ('y') amplitude. Right panels: experimenter interpretation of each component. E) Representative examples of temporal factor dynamics surrounding CS presentation. Colors match the components in panel D. Error bars indicate standard deviation across trials. F) Confusion matrix for CS classification (20 sessions, 5 mice). Values indicate mean classification probability P(predicted | true) over animals ± s.e.m. Diagonal classification accuracies are significantly above the chance level of 1/3 (one-sample t-tests across animals, Bonferroni-corrected across the 3 CS classes; $p < 0.003$).

We also examined whether FR similarly captures uninstructed movements shaped by reward contingencies in an operant setting. We analyzed an operant brain-machine interface (BMI) experiment (**Figure 4**; see **Methods 5.11**), in which an animal was rewarded for increasing neural population activity along a predefined decoder axis (**Figure 4A,C**). During each session, activity from ∼1,000 layer 2/3 neurons expressing jGCaMP8m in facial motor cortex (+1.6 mm anterior, −1.9 mm lateral to bregma (Mayrhofer et al. 2019; Tamura et al. 2025)) is monitored with two-photon imaging at 30 Hz, while continuous face video is recorded for FR analysis (**Figure 4B**).

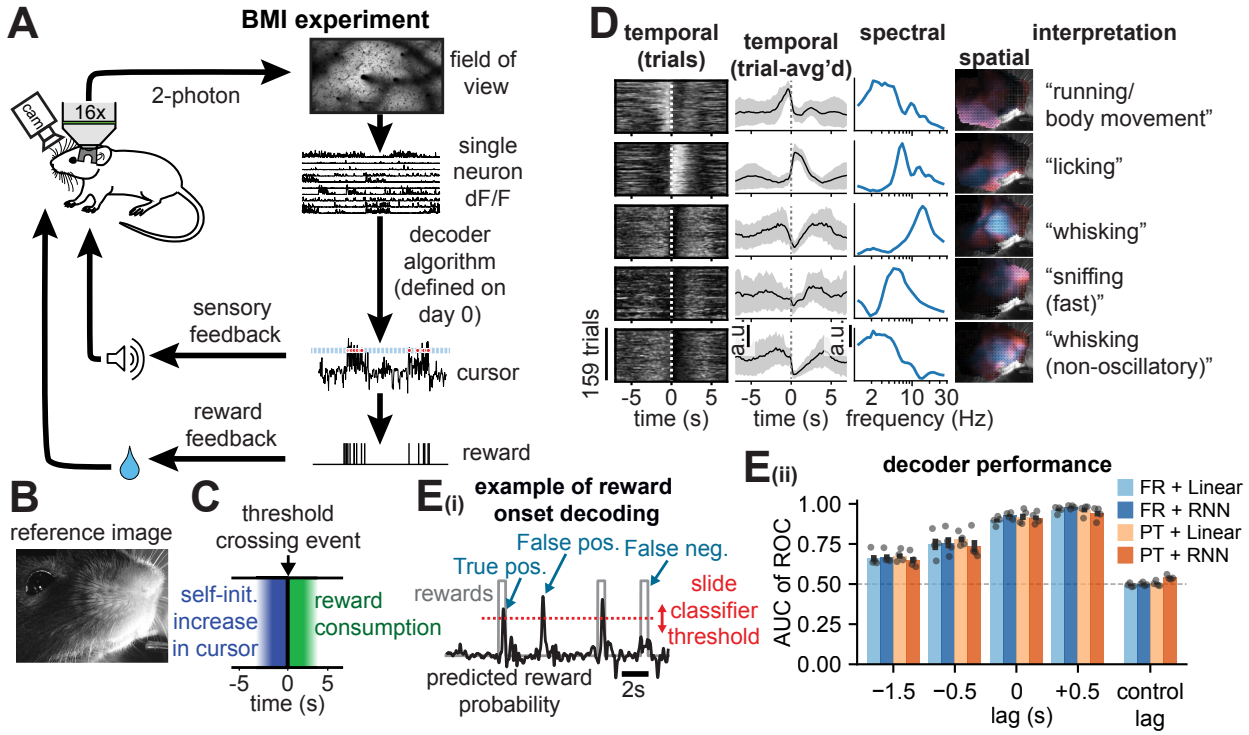


Figure 4. Uninstructed facial behavior predicts brain-machine interface reward events. Uninstructed facial behavior around BMI reward events and decoding analysis. A) Diagram of experiment. Mouse is performing a brain-machine interfacing experiment (see Methods 5.11). Animal is tasked with increasing the activity of a set of neurons that are based on spontaneously evoked activity during a baseline recording day ('day 0'); rewards are given when an animal successfully increases activity beyond a threshold. B) Field-of-view for face camera used for face-rhythm analysis. C) Diagram of a single threshold-crossing reward event. D) Representative example illustrating consistent uninstructed behavioral sequencing before and after threshold-crossing events. Rows are individual components from a face-rhythm extraction using an asymmetric (right-side only) sliding window (see Methods 5.14.2). Left panels: rasters of temporal factors aligned to threshold-crossing events. Left-middle panels: averaged temporal factors; error bars are standard deviation across reward events. Middle panels: frequency / spectral factors. Right-middle panels: Spatial / positional factors. Blue represents horizontal ('x') amplitude, red represents vertical ('y') amplitude. Right panels: experimenter interpretation of each component. E) Decoding analysis. Ei) Diagram for how decoding performance is defined. Decoders are trained to output a prediction of the instantaneous probability that the animal is currently achieving a reward. Inputs are either raw tracked-point time series (Points-linear) or the low-rank FR temporal factors (FR-linear and FR-RNN). To assess performance, receiver-operating-characteristic (ROC) analysis is performed by defining time periods just prior to reward events as desired positives and all other time periods as desired negatives. Eii) Area-under-curve (AUC) analysis comparing Points-linear, FR-linear, FR-RNN, and Points-RNN models trained on input time series data that are temporally shifted / lagged relative to the output time series. Negative lags describe models that are predicting future events. Controls are randomly temporally shifted. Points-RNN uses principal component analysis (PCA)-reduced raw tracked points (100 components) as input to the same long short-term memory (LSTM) architecture. Shaded regions are s.e.m. across mice ($N = 4$).

Continuous face video is analyzed post-hoc to extract FR components and align their temporal factors to reward-threshold events (**Figure 4D**). This reveals a reproducible sequence of component expression spanning approach to threshold, threshold crossing, reward consumption, and the inter-trial interval. Motivated by this structure, we compared four decoders of reward-threshold events (**Figure 4E**; see **Methods 5.6.3**, **Methods 5.6.2**, and **Methods 5.6.4**): a linear model on raw tracked points (Points-linear), a linear model on

FR temporal factors (FR-linear), an RNN model on FR temporal factors (FR-RNN), and an RNN model on PCA-reduced raw tracked points (Points-RNN). For this analysis, intermediate FR spectrograms are computed using causal asymmetric kernels (right-side-only window) with no information from future time points (see **Methods 5.14.2** and **Methods 5.6.3**). Predicted reward-probability traces align with reward events across sessions (example from the FR-RNN model in **Figure 4 supplement 1A-D**). All four models perform similarly across lag conditions (**Figure 4Eii**). At lag 0, the area under the ROC curve (AUC) is 0.90 ± 0.01 (FR-linear), 0.93 ± 0.01 (FR-RNN), 0.92 ± 0.02 (Points-linear), and 0.91 ± 0.02 (Points-RNN; mean $\pm$ s.e.m. across $N = 4$ mice), which is well above the temporally shifted controls (AUC 0.49–0.54). Precision-recall analysis corroborates the ROC results: the area under the precision-recall curve and the maximum F_1 for all four decoders exceed those of an out-of-context shifted control (lag $+20$ min) at every tested lag from -1.5 to $+0.5$ s (**Figure 4 supplement 1E-G**; paired t-test, Benjamini-Hochberg FDR-corrected across the 16 model $\times$ lag comparisons within each metric; AUCPR $p_{\text{adj}} \leq 0.044$, max-$F_1$ $p_{\text{adj}} \leq 0.019$; $N = 4$ mice; see **Methods 5.6.5**). The close agreement across all four decoders is consistent with much of the decodable reward-related signal being broadly accessible in both high-dimensional point trajectories and low-rank FR representations, and suggests that FR features already capture temporal structure that contributes to decoding performance while yielding a more compact and interpretable decoder. Thus, despite the task being operantly conditioned on neural activity alone, mice show stereotyped uninstructed facial movements that track task dynamics and reward expectation.

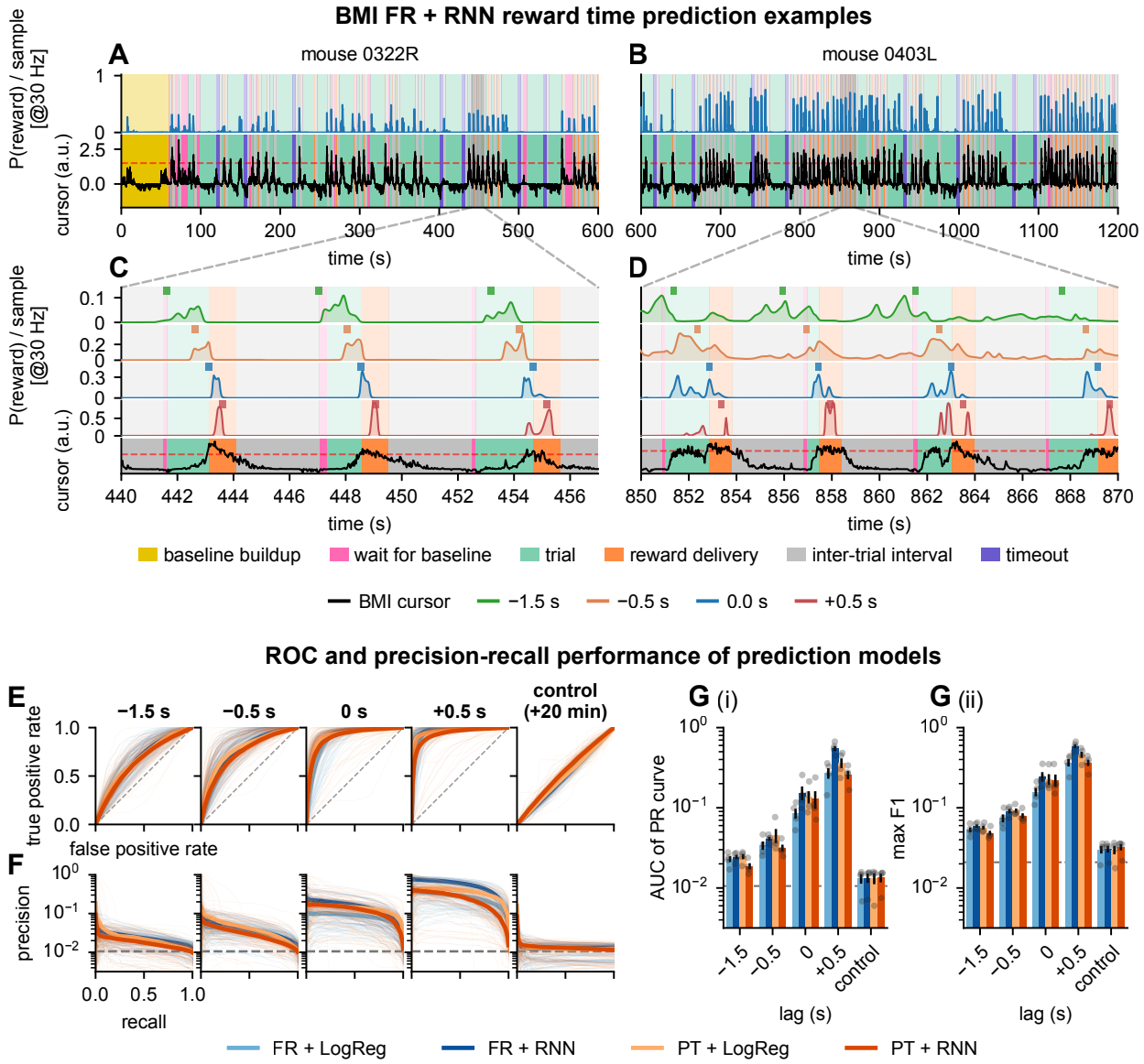


Figure 4 supplement 1. Reward-time prediction from facial behavior during the BMI task. BMI reward prediction model example traces and alternative accuracy metrics. A-B) Representative 600 s segments from BMI experiments showing the predicted reward probability $P(\text{reward})$ from the FR-RNN model (blue, top) and the online BMI cursor (black, bottom) for two example mice. Background shading indicates BMI task epochs: baseline buildup (yellow), wait for baseline (pink), trial (mint), reward delivery (orange), inter-trial interval (silver), and timeout (purple) (see Methods 5.6). C-D) Zoomed views ($\sim$20 s segments) of the same two sessions as in (A-B). Traces show $P(\text{reward})$ from the FR-RNN model at four lag offsets of the target signal (-1.5, -0.5, 0.0, $+0.5$ s; positive lags correspond to prediction into the future) stacked above the BMI cursor trace. E) ROC curves per lag (columns; -1.5, -0.5, 0.0, $+0.5$ s and a shifted control at $+20$ min) for the four prediction models (FR-linear, FR-RNN, Points-linear, Points-RNN; see Methods 5.6). Light traces show per-session ROCs; bold traces show the across-session mean with ± 1 s.e.m. shading. Dashed diagonal marks chance. F) Precision-recall curves for the same lag $\times$ model layout as in (E). Dashed horizontal line marks baseline (prevalence of positive frames). G) Grouped bar plots summarizing precision-recall performance across lags and models. Gi) Area under the precision-recall curve (AUCPR, log scale). Gii) Maximum F_1 (log scale) over a threshold sweep. Bars show the across-mice mean; error bars are s.e.m. across mice ($N = 4$); dots are individual mice. All 16 model $\times$ lag comparisons against the shifted con-

trol are significant after Benjamini-Hochberg false discovery rate correction ($p_{\text{adj}} \leq 0.044$ for AUCPR and $p_{\text{adj}} \leq 0.019$ for max-F_1; see Methods 5.6.5).

To interpret the contribution of different regressors to predicting BMI reward onset, we visualize the regression coefficients for an example session from one mouse (**Figure 4 supplement 2**). Under an FR-linear model trained at lag 0, coefficient mass concentrates on a small number of FR components, with lateral nose movement at ∼3 Hz emerging as the top predictor of reward onset (**Figure 4 supplement 2A**). The Points-linear decoder and model-free summaries (reward-triggered displacement, signal-to-noise ratio, and signed point-biserial correlation) provide a complementary view: similar facial regions (around the nose and jaw) carry the largest reward-associated point-displacement weights. However, these representations are mixed over points with different contributions and do not expose spectral patterning (**Figure 4 supplement 2B-D**; see **Methods 5.6.6**).

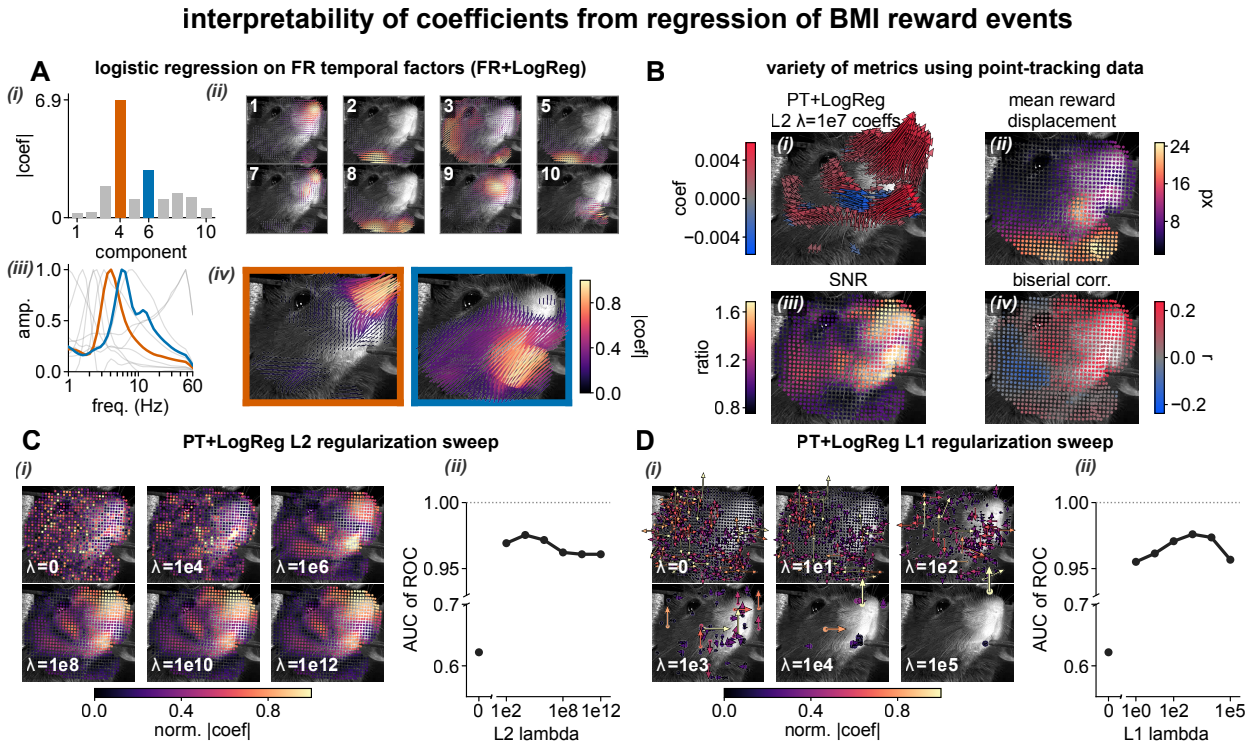


Figure 4 supplement 2. Interpretability of coefficients from regression of BMI reward events. Example comparison of regression coefficients from BMI reward-onset prediction models, for one representative session (see Methods 5.6.6). In the figure panels, FR-linear is labeled 'FR + LogReg' and Points-linear is labeled 'PT + LogReg'. A) FR-linear decoder coefficients. Ai) Per-component |coefficient| for the 10 FR temporal factors; the two components carrying most of the decoder weight are highlighted (component 4, orange; component 6, blue). Aii) Spatial loadings ('biquivers') for the 8 non-highlighted components; each point is drawn as a line whose angle and length describe the relative factor amplitude of the x and y displacement dimensions. Aiii) Spectral factors for all 10 components. Aiv) Enlarged biquivers for components 4 and 6. B) Four different per-point views of the raw point-tracking data (∼936 tracked points). Bi) Points-linear fit (L2 penalty $\lambda = 10^7$): each point's (x, y) regression coefficients drawn as an arrow. Bii) Mean reward-triggered point displacement. Biii) Signal-to-noise ratio: reward-window mean displacement magnitude divided by baseline-window mean displacement magnitude. Biv) Signed point-biserial correlation between the per-frame reward label and each point's per-frame signed displacement. C) Points-linear model L2 regularization sweep. Ci) |coefficient| maps at six values of λ (each face independently max-normalized). Cii) Test AUC as a function of λ (see Methods 5.6.6). D) Points-linear L1 regularization sweep. Di) Signed

coefficient maps at six values of λ, with quiver arrows at the L1-selected non-zero points. Dii) Test AUC as a function of λ.

3.4 Spectral transformation between facial behaviors and neural activity

The accurate decoding of task variables from FR temporal factors described above demonstrates that a spectral transformation of facial movements yields a description that is related to cognitive and task variables (i.e., conditioned output expectation) and cortical activity (i.e., neural trajectories controlling a BMI). In addition, the existence of central pattern generators that control rhythmic facial behaviors motivates the hypothesis that a spectral transformation may lie between patterns of M1 neural activity and of facial movements. We therefore examined whether the activity of neurons in M1 is better explained by a direct transformation of facial position or by a spectral transformation of facial movements (**Figure 5**).

To systematically assess the degree to which phase information is conserved or spectrally transformed between facial behavior and M1 neural activity, we developed three regression models (**Figure 5C**) that vary in whether phase information from behavior data can be used in predicting neural data. We compared the quality of the fits to quantify the degree to which phase information is encoded in M1 neural activity.

In all of these models, the input is convolved with a learned kernel whose shape is optimized to minimize the output prediction error. The models differ in how this convolution step proceeds and, as a result, whether this step performs a spectral transformation, which here denotes an operation in which a signal is convolved with a kernel (real or complex-valued) and then the magnitude of the resulting output is taken. The first "basic convolution" model simply applies a standard real-valued convolution to map inputs to outputs. This transformation is not capable of extracting the oscillatory power envelope from the input signals and can only translate, smooth, sharpen, and shape signals; therefore, this model is phase-variant. The second "spectral convolution" model employs a complex-valued kernel. In this model, both the real and imaginary parts of the kernel are optimized independently, and then the magnitude of the convolution output is taken, which discards explicit phase information and results in a phase-invariant spectral transformation. Finally, the third "Hilbert-spectral convolution" model is similar to the spectral convolution model, but the imaginary part of the complex kernel is constrained to be an exactly 90° phase-shifted version of the real part. This is achieved by optimizing only the real part and setting the imaginary part as its Hilbert transform, resulting in a complex spectral convolution that is matched in parameter number to the basic convolution. Assessing the performance of these three models provides a means to isolate the relative contributions of phase-variant and phase-invariant spectral representations in explaining the transformation from signals in one space to another (i.e., movement to neural space). We propose that this approach can be used to measure bounds on the relative amount of information that M1 encodes about the absolute position of the face versus the spectral content of movements in the face.

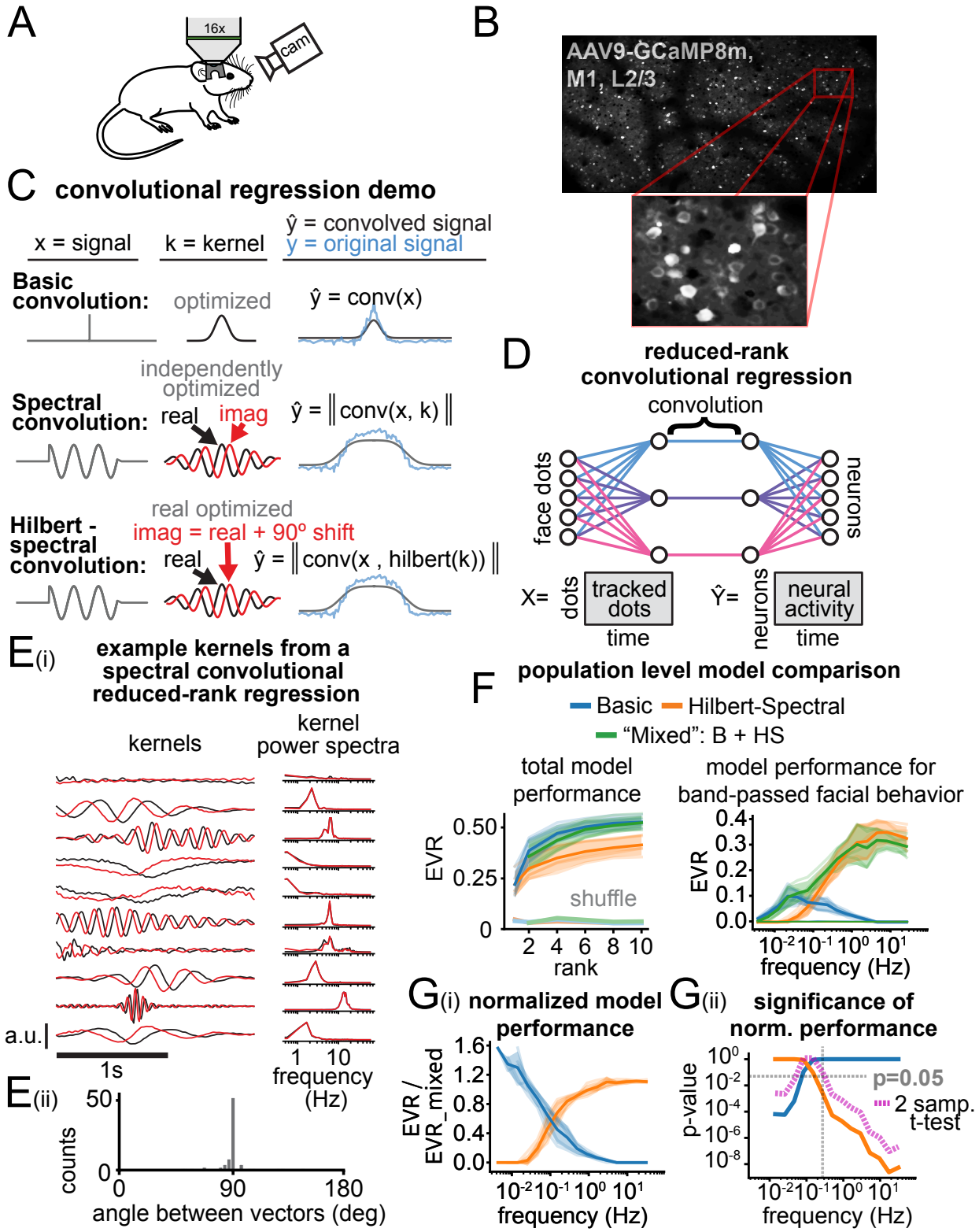


Figure 5. Convolutional reduced-rank regression model reveals frequency-dependent phase-variant and invariant coding of facial movement in M1. Convolutional reduced-rank regression of M1 activity from facial point tracking. A) Diagram of experimental setup. Mouse is not performing a task, but is in an en-

vironment with a linear treadmill, open-loop visual and auditory stimulation, and uncued infrequent soy milk droplets (see Methods 5.11.5). B) Example field-of-view from two-photon microscope. C) Demonstration example of convolutional regression models. Basic convolution: a signal (x) is convolved with a kernel (k) resulting in the convolved output ($\hat{y}$). During optimization, the shape of the kernel is optimized to minimize the error between the convolved output and ground truth output (y). Spectral convolution: a signal is convolved with a complex kernel where the real and imaginary components are independently optimized, and the magnitude of the output is taken. Hilbert-spectral convolution: same as spectral convolution, but the imaginary component is constrained to be the $90°$ phase shifted (Hilbert transformed) version of the real component. D) Diagram of reduced-rank convolutional regression. Inputs are tracked points from the face (2D time series with elements describing instantaneous positional displacement); note that input here has only gone through the first step of the face-rhythm pipeline. Inputs are encoded into a reduced dimensional space, convolved, then decoded to predict neural time series data. Optimization performed over the encoding matrix, convolution kernels, and decoding matrix (see Methods 5.7). E) Representative example of kernels derived from spectral convolution of neural data demonstrating that kernels naturally form $90°$ offset sinusoids after optimization. Ei) Raw kernels with real components in black and imaginary components in red. Fourier transform of kernel shown to the right. Eii) Angle between real and imaginary components (see Methods 5.7). F) Model performance comparison between basic convolution, Hilbert-spectral convolution, and a hybrid model where half of the reduced-rank components are from each model type. Shuffles are randomly temporally shifted. Error bars are s.e.m. across mice ($N = 5$ mice). G) Model performance comparison of different frequency bands of the output neural data (Y). Models are independently optimized for each frequency. Gi) Raw explained variance ratio (EVR) values for each model type. Gii) EVR ratios of the pure models with the hybrid model. Error bars in (G) are s.e.m. across mice ($N = 5$ mice). Significance in (Gii) is assessed by paired t-test at each frequency band with Bonferroni correction across bands.

Models are parameterized by an encoding matrix, a kernel for each "rank", and a decoding matrix. The inputs to the model are the raw point displacement time series from all points on the face/body derived from the face-rhythm optical flow analysis prior to any spectral processing. The outputs of the model are predictions of the simultaneously collected neural activity time series, which are the inferred spike/event rates of all jGCaMP8m-expressing neurons in layer 2/3 of M1 (see **Methods 5.7**). We use a reduced-rank regression architecture to wrap each type of convolution model (**Figure 5D**), which constrains the number of convolutional kernels to a small and user-defined number that corresponds to the "rank" of the model. This approach both regularizes the model and aids in interpretation by providing an input and output mapping for each kernel. This architecture resembles a linearized convolutional neural network, with the major difference being that the spectral convolutions are defined explicitly in a single layer without the nonlinear activations typically present in neural networks.

Fitting the spectral convolution model, which optimizes real and imaginary parts of the kernel independently, reveals complex convolutional kernels that are approximately $90°$-offset amplitude-modulated sinusoids tiling the frequency axis (**Figure 5E**). This observation illustrates the natural internal representations adopted by the model and further motivates a more direct comparison between basic (phase-variant) and spectral (phase-invariant) models. We therefore compared the predictive performance of the basic and Hilbert-spectral convolution models, which have the same number of parameters. We constrained our analysis to the dominant neural activity manifold by fitting the first 10 principal components. The basic convolution model explains more neural variance than the spectral model, and a hybrid model in which half of the ranks were basic and half were Hilbert-spectral (maintaining the same total number of parameters as the basic model) performs similarly to the Hilbert-spectral (**Figure 5F**). We conclude that the major dimensions of neural variance are generally better fit by a phase-variant transformation of behavior, and that there is a significant overlap in the specific dimensions of neural activity explained by the basic and spectral models (EVR of 'basic' vs. 'Hilbert-spectral' at rank-10, $p = 0.0015$, two-sided paired t-test, $N = 5$ mice).

Motivated by observations in sensory systems (Bendor and X. Wang 2007; Ewert, Vahle-Hinz, and Engel 2008), we tested whether the frequency of a movement predicts whether it is better represented in M1 via a phase-variant or invariant transformation. The spectra of natural phenomena, including various kinds of

animal behavior, tend to have higher power at lower frequencies (Bassingthwaighte, Liebovitch, and West 1994; Kafetzopoulos, Gouskos, and Evangelou 1997). Consistent with this, we observed that the spectra of raw point tracking signals from across the face also show high power at low frequencies (**Figure 5 supplement 1E**) and exhibit a power-law decay (slope in log-log space $\beta = -1.7$; $p < 10^{-4}$, two-sided Wald test on the regression slope against a null slope of zero, $N = 5$ mice). We hypothesized that facial behaviors occurring at higher frequencies are more likely to be represented in M1 activity by the Hilbert-spectral (phase-invariant) model. Therefore, we refit the models using facial movements filtered into narrow frequency bands. Indeed, the Hilbert-spectral model significantly outperforms the basic model for facial movements occurring at frequencies greater than ~ 0.5 Hz (**Figure 5G**) (paired t-test across $N = 5$ mice, Bonferroni-corrected $p < 0.05$ for each band above 0.5 Hz). We conclude that higher frequency facial behavior is better represented in neural activity (up to the limits imposed by the 30 Hz two-photon microscopy imaging rate) via a spectral transformation.

We also examined the tuning of individual neurons to phase-variant and phase-invariant facial movements. Many neurons show visually identifiable phase-locking with facial movements, and others show clear correlation with the envelope of oscillatory movements (**Figure 5 supplement 1A-B**). This single-neuron analysis uses the raw point tracking data (prior to the spectrogram step) to reconstruct phase-variant versions of each behavioral component, rather than using the phase-invariant FR temporal factors directly. To quantify these relationships, we generated pairs of behavioral signals that correspond to roughly one behavior in which one signal is phase-variant and the other is phase-invariant. Specifically, raw point tracking data are filtered through a single FR component's spatial and spectral factors to obtain the phase-variant signal, and the magnitude of the Hilbert transform of that signal yields the phase-invariant signal. Each neuron's activity trace is then compared to each FR component signal pair; the pairwise phase consistency is computed using the phase-variant signal, and the correlation is computed using the phase-invariant signal (**Figure 5 supplement 1C-D**). We find that 93.2% of neurons ($4{,}207/4{,}513$; $93.4 \pm 0.8\%$ s.e.m. across mice) show a significant envelope correlation to at least one phase-invariant component, and 65.4% ($2{,}951/4{,}513$; $66.2 \pm 5.9\%$ s.e.m.) are significantly phase-locked to at least one phase-variant component (Bonferroni-corrected across neurons, $p < 0.01$; $R = 12$ components; $N = 5$ mice). Some neurons show significant phase-locking with behavioral signals with peak frequencies near the Nyquist limit (~ 15 Hz) for the neural data recordings. We also find that more neurons show phase-invariant tuning at higher movement frequencies (permutation test against time-shifted null distribution, $p < 0.01$).

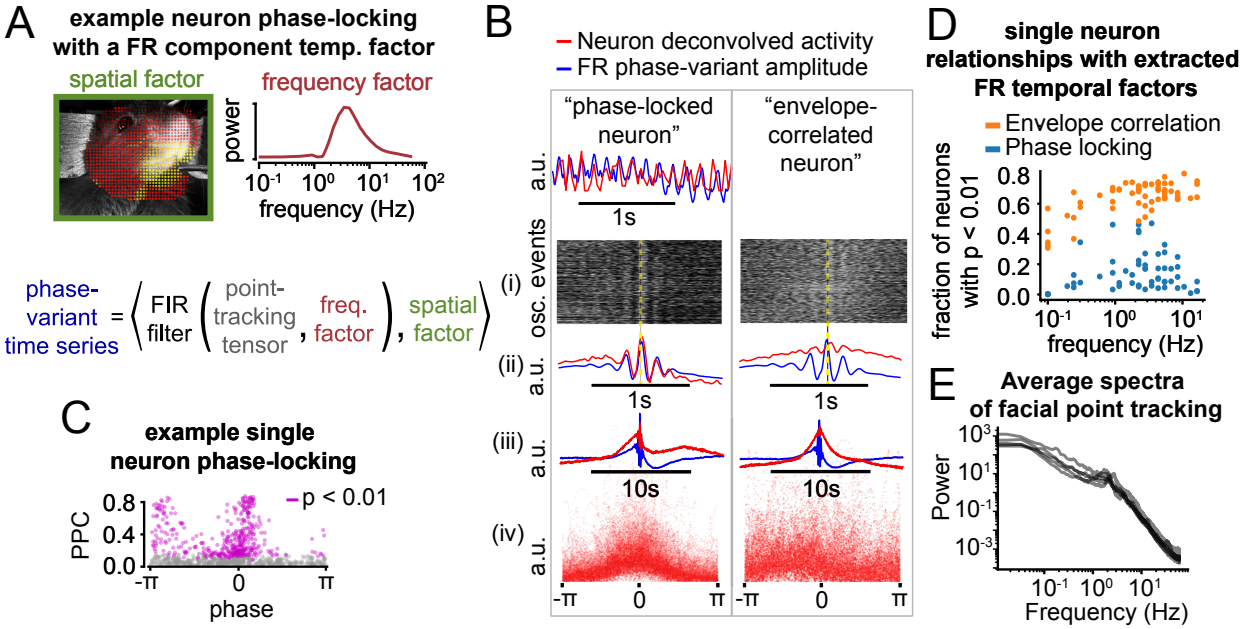


Figure 5 supplement 1. Phase-locking and envelope correlations between facial movements and single neurons. Single-neuron phase-locking and envelope-correlation analysis of facial components. A) Upper: Example spatial/point factor and frequency/spectral factor from the same single component of a face-rhythm (FR) decomposition. Lower: Equation for how phase-variant behavioral signals were generated. FR temporal factors normally describe spectral envelope time series; to make phase-variant signals, we spectrally filtered the raw point-tracking signals using the FR frequency/spectral factor via a finite-impulse response (FIR) filter; then, took the dot product of the resulting signals with the FR spatial factor yielding a single phase-variant signal for each FR component. See Methods 5.8 for details. B) Examples of how two neurons are associated with the temporal factor of the behavioral component in (A). One neuron is phase-locked to the instantaneous movement (left) and one is correlated with the spectral envelope (right). Bi) Raster plot of neural activity triggered by oscillatory events in the behavioral time series; brightness describes the instantaneous neural activity amplitude. Yellow line describes when the behavioral time series phase equals zero. Bii) Average of (Bi) over oscillatory events. The overlap of red and blue traces shows the phase-locking of the neuron to the behavioral signal. Biii) Same as in (Bii), but zoomed out in time to show the increased envelope correlation in the right panel. Biv) Scatter plot of time points in neural activity, where the horizontal/x-axis describes the instantaneous phase within the behavioral time series and the vertical/y-axis describes the instantaneous amplitude of neural activity. C) Phase-locking of all recorded neurons to the same example behavioral component in (A) and (B). Pairwise phase consistency (PPC) describes the phase-locking strength, and phase describes the average instantaneous phase difference between the behavioral signal and the neural activity signal after bandpassing it using the same spectral factor. Dots are colored purple if the PPC value is significantly greater than a null distribution of PPC values generated by randomly shifting/rolling the behavioral time series relative to the neural time series. D) Summary of the fraction of all neurons with a significant PPC value or envelope correlation score. Each dot describes a behavioral component (rank-12 tensor decompositions in FR from 5 mice resulting in 60 total behavioral components). The horizontal/x-axis describes the peak frequency of the spectral factor for the behavioral component that is being used. Significance for both PPC and envelope correlation is assessed at $p < 0.01$ with Bonferroni correction across neurons (see Methods 5.8). E) Example of the raw power spectra from the entire behavioral time series for each of the 5 mice.

3.5 Similarity between face-rhythm components and neural activity components

Based on the anatomy of the facial motor system as well as the utility of spectral transformations in explaining neural variance, we hypothesized that FR transformations may approximate the inverse of the transformations present in the facial motor pathway. This hypothesis predicts that the signals observed along the temporal dimension of the behavior data prior to undergoing the tensor decomposition might be closely related to time-varying neural activity observed in cortex. Therefore, the application of a similar tensor decomposition method to neural data might yield temporal factors that are closely related to FR temporal factors.

We examined whether a one-to-one mapping exists between FR temporal factors and the temporal factors extracted by non-negative tensor decomposition of the simultaneously recorded neural data (in 2-dimensional data, this is better known as non-negative matrix factorization, or NMF) (**Figure 6A**). We recorded M1 activity using two-photon calcium imaging and simultaneously recorded facial behavior using a camera. Mice were free to run on a linear treadmill and given soy milk at random infrequent intervals. We found pairwise similarities in the temporal factors extracted from the two domains, despite the complete independence of the neural and facial movement analyses (**Figure 6B**). We also observed greater correspondence, as measured with correlation coefficients, between neural temporal factors and FR temporal factors for FR components with a higher peak frequency in their spectral factor (**Figure 6C**), demonstrating a bias toward higher frequencies in the similarity between face-rhythm and neural components. Overall, 41 of 50 matched factor pairs (82%; 70–90% per mouse, $N = 5$ mice) show significant correlations (Bonferroni-corrected $p < 0.001$ against $25{,}000$ circular time-shift shuffles; median $r = 0.35$, range 0.03–0.69) (**Figure 6D**). Thus, the dominant modes of the recorded neural activity resemble the dominant modes of spectrally transformed facial movements.

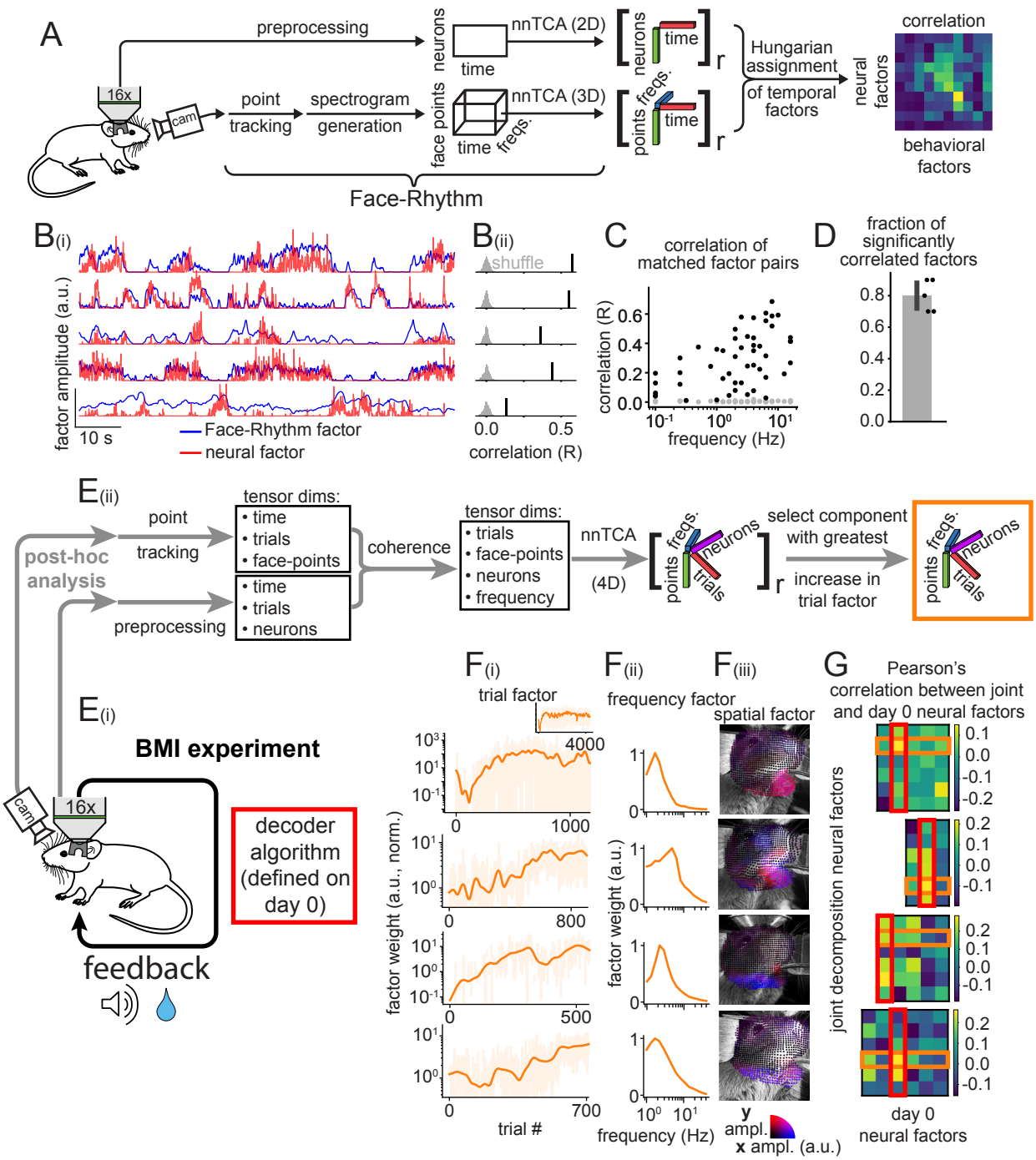


Figure 6. Joint neural-behavioral decomposition reveals learning-associated face-brain coupling. Joint decomposition of simultaneous neural and facial recordings during spontaneous behavior and multi-day BMI learning. A) Diagram of analysis pipeline for comparing independently extracted tensor components from neural data (top arrow path) and behavior / face data (bottom arrow path). Mouse is not performing a task, but is in an environment with a linear treadmill, open-loop visual and auditory stimulation, and uncued infrequent soy milk droplets (see Methods 5.11.5). Neural data is collected from primary motor cortex (M1) and is preprocessed and 2D non-negative tensor component analysis (nnTCA) is performed

(which is equivalent to NMF). Face data is run through the standard face-rhythm pipeline. Temporal factors extracted from the neural and facial datasets are then 1-to-1 matched using linear sum assignment ("the Hungarian algorithm"), resulting in a maximized correlation along the diagonal of the correlation matrix. B) Representative example comparing the matched temporal factors from one mouse. Bi) Time series of matched temporal factors, showing visual correspondence. Bii) Pearson correlation between each matching temporal factor pair. Shuffles are randomly temporally shifted. C) Relationship between the correlation between matched temporal factors (y-axis) and the peak frequency of the frequency factor from the corresponding facial component. Each black dot is a single component, each gray dot is a temporally shuffled single component. D) Fraction of significantly correlated matching temporal components ($N = 5$ mice; Bonferroni-corrected $p < 0.001$ against $25{,}000$ circular time-shift shuffles). Error bars are s.e.m. across animals. E) Experiment and analysis pipeline for joint neural-behavioral decomposition during a multi-day BMI task. Ei) BMI experiment setup. On a baseline recording day ("day 0") when sensory feedback is open-loop, recorded neural activity from M1 is decomposed to generate a set of neural factors; one factor is selected as the BMI decoder/cursor dimension (red box; see Methods 5.11.6). During subsequent BMI sessions, the animal receives real-time auditory and visual feedback proportional to its neural activity along the decoder dimension, with soy milk reward delivered upon reaching a threshold amplitude. Eii) Post-hoc neural-behavioral joint factorization analysis (see Methods 5.9). Coherence is computed between all neuron:face-point pairs, producing a 4D tensor (trials $\times$ frequencies $\times$ face points $\times$ neurons) that is decomposed using non-negative TCA (nnTCA). Each resulting component has four factors: trial (expression across sessions), frequency, spatial (points), and neuron. The component whose trial factor shows the greatest increase across sessions is highlighted for illustration (orange box). F) Factors of the learning-associated joint component for each of 4 mice (one row per mouse). Fi) Trial factor showing increase in expression magnitude across BMI training sessions; the orange trace is the selected component and lighter traces show remaining components for context. Fii) Spectral / frequency factor. Fiii) Spatial factor; blue indicates horizontal (x) amplitude and red/magenta indicates vertical (y) amplitude. G) Pearson correlation between all joint component neural factors (rows; 6 joint components per mouse, from a rank-6 joint decomposition) and all day-0 neural factors (columns), shown for each of the 4 mice ($N = 4$). The red box marks the day-0 factor selected as the BMI decoder axis; the orange box marks the joint component with the greatest increase in trial factor amplitude (same component as in F). In 3 of 4 mice, the intersection of the orange row and red column corresponds to the highest value in the orange row, indicating that the learning-associated component's neural factor most closely matches the decoder axis despite being selected without knowledge of the decoder.

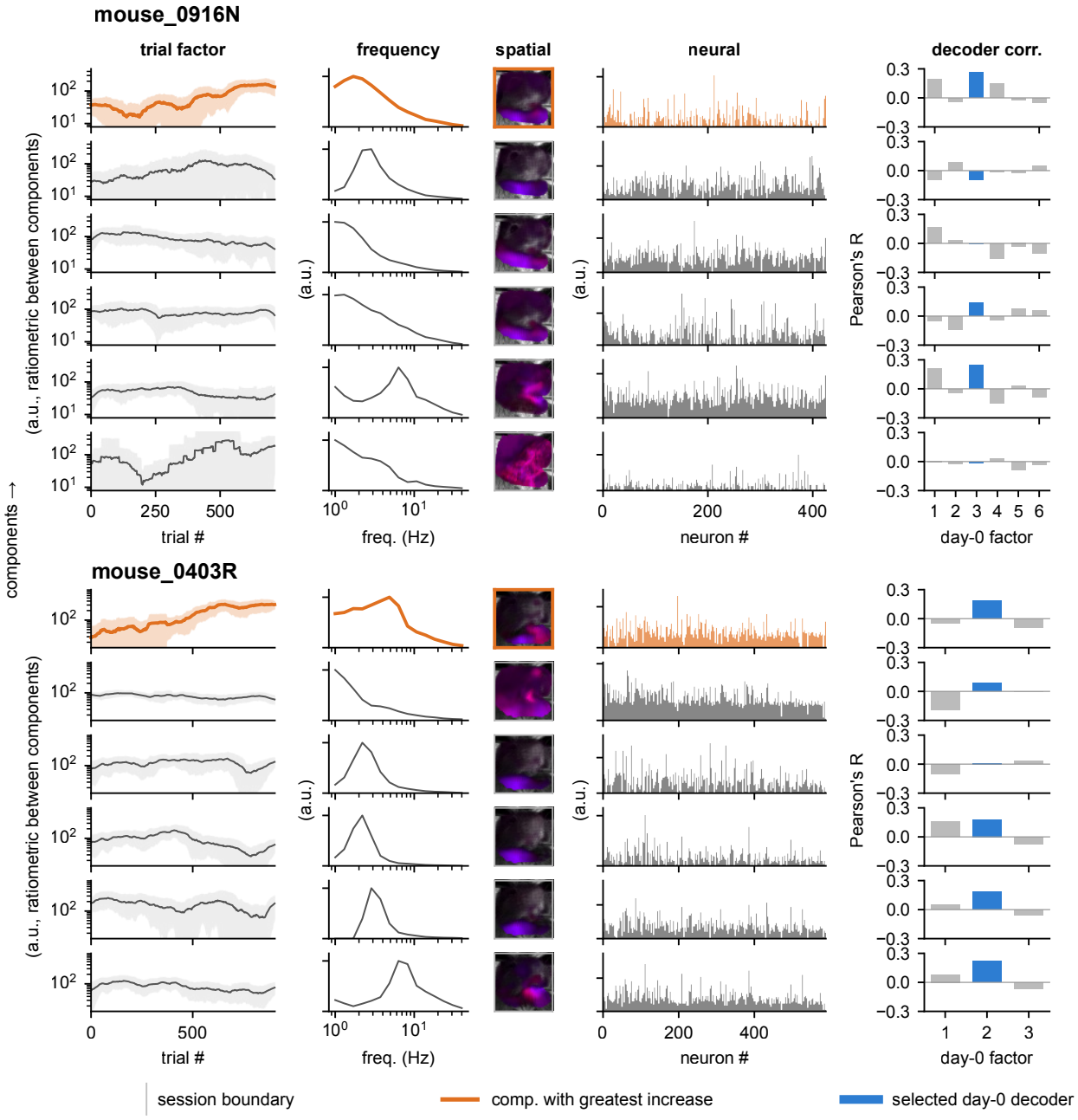


Figure 6 supplement 1. All joint neural-behavioral components for representative mice. All joint neural-behavioral components for two representative mice. Each row is one component extracted from the joint coherence tensor decomposition described in **Figure 6Eii**. Columns show, from left to right: trial factor (expression magnitude over trials), spectral/frequency factor, spatial/points factor (blue: horizontal amplitude, red: vertical amplitude), neural factor (loading on each neuron), and the decoder correlation. Pearson's R between that component's neural factor and each day-0 neural factor; blue bar indicates the day-0 factor selected as the BMI decoder. The component whose trial factor shows the greatest increase across training sessions is highlighted in orange (same component shown in **Figure 6F**); other joint components are shown in gray. Shaded regions in the trial factor column indicate the interquartile range of a sliding window smooth.

The above analysis indicates a natural similarity between the latent spaces of FR-transformed facial behavior and neural activity. We therefore developed an approach to jointly decompose the shared space between neural activity and behavior, identifying components that capture correlated neural-behavioral motifs. Inspired by canonical correlation analysis, we sought to define a low-dimensional representation of *coherence* between two datasets. We defined a joint decomposition by calculating the coherence between all neurons and all face-tracking points, producing a 4-dimensional tensor (trials $\times$ frequencies $\times$ face points $\times$ neurons), and decomposing this tensor using non-negative tensor factorization (**Figure 6Eii**). Each resulting component is defined by four factors: a trial factor (expression magnitude for each trial across sessions), a spectral factor (frequencies), a spatial factor (points), and a neuron factor (which neurons participate).

We applied this joint decomposition to data from 4 mice performing a multi-day BMI task (**Figure 6Ei**). In this experiment, a BMI decoder axis is defined on a baseline recording day ("day 0") by decomposing neural activity using sparse PCA and selecting one neural factor as the cursor dimension (red boxes, **Figure 6G**; see **Methods 5.11.6**); this selection occurs before the animal begins BMI training. The joint decomposition is then applied post-hoc to neural and behavioral data collected across subsequent training sessions. The decomposition has no knowledge of which day-0 factor was chosen as the decoder; all components are extracted in an unsupervised manner.

To identify which neural-behavioral motifs are shaped by learning, we selected the joint component whose trial factor showed the greatest increase over training (orange lines, **Figure 6Fi**); all components and their factors are shown in **Figure 6 supplement 1**. The Pearson correlation between each joint component's neural factor and each day-0 neural factor is shown in **Figure 6G**; this matrix allows one to assess whether the learning-associated component (orange box) aligns with the decoder axis (red box) despite being selected without any knowledge of the decoder. In 3 of 4 mice, the learning-associated component had the highest correlation with the decoder axis among all 6 joint components; in the fourth mouse, it was the second-highest match (probability ≈ 0.004 under a null in which each of the 6 components is equally likely to be the top match in each mouse). The behavioral factors of these learning-associated components indicate movements consistent with preparatory licking and running (**Figure 6Fii-Fiii**). Thus, the joint decomposition independently discovers that the neural-behavioral motif most shaped by learning is aligned with the task-relevant neural dimension (decoder). This illustrates how FR-based joint decomposition can reveal specific behavioral strategies that become coupled to task-relevant neural activity during learning.

4 Discussion

This study introduces face-rhythm, an unsupervised analysis pipeline that combines markerless point tracking, spectral analysis, and non-negative tensor component analysis to extract interpretable facial behaviors. The output components of face-rhythm are predictive of task and internal variables, and their time courses align closely with demixed neural activity components. Alongside the pipeline, we developed several supporting computational methods including a joint decomposition for discovering shared neural-behavioral motifs. In addition, we report that for facial movements above ~ 0.5 Hz, primary motor cortex activity is better described by a phase-invariant spectral transformation of behavior than by the instantaneous point positions themselves. Together, these results support spectral analysis as a natural featurization for relating ongoing facial behavior to motor cortical activity.

4.1 Comparison to other methods

Automated facial and whole-body behavior analysis has advanced rapidly, and several tools are now in common use. Many new methods exploit machine learning architectures like deep neural networks and semi-supervised frameworks (e.g., DeepLabCut-Live! (Kane et al. 2020), SLEAP (Pereira et al. 2022), VAME (Luxem et al. 2022), CEBRA (Schneider, J. H. Lee, and M. W. Mathis 2023)) to analyze and decompose movements. Although powerful, large neural networks pose challenges for data-constrained experiments. Supervised approaches require manual labeling, and all networks trained on specific datasets are typically only usable on new datasets that are similar to the training data. In contrast, the FR pipeline is end-to-end unsupervised and can be run on small and even '$N = 1$' datasets; the individual tracking, transformation,

and decomposition methods are not trained models, and thus have no dataset-specific learned biases. The standard FR pipeline provides a principled approach to behavioral decomposition and is appropriate for use within defined conditions. In particular, FR works best when the behavior of interest moves like a deformable mesh without occlusions, a spectral transformation is a reasonable representation of underlying movement, and a sparse parts-based continuous-time description of behavior is desirable.

4.2 Interchangeability with other methods

The FR pipeline is composed of three modular analysis steps: unsupervised point tracking, spectral transformation, and tensor decomposition. Although we present specific, optimized implementations for each stage, FR is designed as an extensible framework that can be integrated with other behavioral analysis pipelines. This modularity offers several avenues for customization. For instance, the initial modified Lucas-Kanade-based point tracking could be replaced with outputs from deep-learning-based pose estimation tools like DeepLabCut, SLEAP, CoTracker, FaceMap, or other methods, allowing researchers to feed pre-existing keypoint data directly into FR's spectral analysis stage. Alternatively, FR's point-tracking output could be used for different decompositions, as was shown in this work. Finally, the final nnTCA-derived temporal factors from FR can serve as low-dimensional inputs for subsequent behavioral segmentation algorithms such as MoSeq or B-SOiD. The codebase was developed with this interchangeability in mind and is designed to be a flexible, rather than monolithic, tool.

4.3 Relationship of FR factors to neural activity

Our findings echo the concept of motor primitives (Shadmehr and Mussa-Ivaldi 1994; Hogan 1984; Botvinick, Niv, and Barto 2009) and align with recent evidence on central pattern generators in rhythmic behavior (Dempsey et al. 2021; Moore et al. 2013; Kaku et al. 2025). Together, these results suggest that M1 employs a multiplexed encoding strategy, integrating phase-variant and phase-invariant representations, reminiscent of representations in sensory systems like the auditory and somatosensory cortices (Peterson and Heil 2019; Bendor and X. Wang 2007; Ewert, Vahle-Hinz, and Engel 2008). Practically, closed-loop brain-machine interfaces and neural decoder designs may benefit from the introduction of spectral transformations into models.

Previous studies have also shown that a large fraction of the total observed variance in neural activity can be explained by spontaneous and uninstructed behavior (Musall et al. 2019; Stringer et al. 2019), and that point-tracking and convolutional neural networks can improve estimations of neural variance (Syeda et al. 2024). The observations that higher frequency behaviors are more likely to be represented in M1 via a spectral transformation, and that the major axes of M1 activity align with spectrally transformed behavior, are consistent with existing theories for how high-frequency rhythmic motor behavior is produced. Our findings are consistent with the hypothesis that motor cortical regions propagate phase-invariant movement signals to lower central pattern generators, where absolute amplitude is converted into the amplitude of oscillatory components (Cramer and Keller 2006; Hill et al. 2011; Friedman, Zeigler, and Keller 2012).

The dominance of spectral-envelope representations of higher frequency movements in cortex has several implications for coding efficiency and the architecture of the motor system. First, because spectral envelopes have lower frequency content than time-domain phase-variant signals, they require less bandwidth and impose lower bit-rate demands; such representations are also easier to maintain and propagate through the noisy low-rank dynamics thought to characterize motor cortex (Mastrogiuseppe and Ostojic 2018; Omrani et al. 2017; Laughlin, De Ruyter Van Steveninck, and Anderson 1998). Second, a related implication is that the rhythmic motor patterning itself may be compartmentalized, with discrete central pattern generators acting as low-plasticity reservoir networks that offload specialized computation to specialized anatomical structures (Kaku et al. 2025). Finally, if the primary output of motor cortical regions encoding higher frequency behaviors is a phase-invariant spectral envelope sent to CPGs, then fine cortical control of instantaneous movement phase may be limited for these behaviors, or a secondary descending signal carrying explicit phase information may interact with brainstem CPGs.

4.4 Analysis of conditioned uninstructed movements

Our observation of subtle behaviors that are robustly associated with, and predictive of, brain-machine interface performance highlights the need for careful behavioral tracking during BMI tasks. In addition, it suggests that micromovements, while not explicitly required to perform a BMI task, may be present in many commonly studied BMI tasks and calls into question the utility of using the lack of grossly observable movement as a common inclusion criterion. In the BMI decoding analysis, the similar performance of all four decoders (Points-linear, FR-linear, FR-RNN, and Points-RNN) is consistent with the interpretation that much of the decodable reward-related signal is broadly accessible across both high-dimensional and low-rank feature representations, and is retained after FR dimensionality reduction. This supports the interpretation that FR temporal factors provide a compact representation that preserves behaviorally relevant information while still capturing temporal structure important for decoding. A study may demonstrate that BMI-cursor-related neural activity is output-null and volitionally controlled, which can provide evidence of sufficiency to claim that embodied computations are not involved. However, the corollary that output-potent neural activity implies the existence of task-critical non-local computation is not necessarily true. Instead, a more thorough attempt to identify uninstructed BMI-associated behaviors is warranted to better interpret the strategy taken by brains and animals to achieve control over cursor dynamics. Similar considerations apply for the analysis of signals acquired during tasks with waiting periods, such as the Pavlovian odor association task in **Figure 3**. Uninstructed movements and associated neural activity present during the waiting period can cause spurious correlations between movements, cognitive signals, and task variables (Mohebi et al. 2024). These observations argue for monitoring subtle uninstructed movements during BMI experiments.

4.5 Computational tools

Several computational approaches were developed here to improve on existing methods: a mesh-physics-constrained modification of the Lucas-Kanade algorithm that reduces noise and drift; an exact formulation of the variable-Q transform with more flexible filter design; convolutional reduced-rank regression models that build a spectral transformation into the inductive bias; and a coherence-based joint tensor decomposition for identifying shared neural-behavioral components. The face-rhythm software package and related code is made available here: `https://github.com/RichieHakim/face-rhythm`. See **Methods 5.16** for details on software dependencies and all other code made available for this work.

4.6 Conclusion

Face-rhythm (FR) is an unsupervised, data-efficient method for decomposing facial movements into interpretable behavioral components. Applied to rodent facial video, the standard pipeline recovers common behaviors such as whisking, sniffing, and licking, along with subtler movements that track internal cognitive states, task variables, and dominant modes of motor cortical activity. These components are consistent across mice, indicating that the decomposition captures stereotyped motor patterns rather than animal-specific artifacts, and they highlight the value of measuring subtle uninstructed movements in behavioral analyses, particularly during BMI experiments. In the recording conditions used here, most M1 neurons showed a significant relationship to the spectral envelope of higher frequency facial movements, and a smaller fraction were phase-locked to instantaneous movement; individual neurons could show both. Together, these results support a role for phase-invariant spectral representations of movement in rodent facial motor cortex, and motivate the use of explicit spectral transformations in models that relate cortical activity to ongoing behavior.

References

Baker, Simon and Iain Matthews (Feb. 2004). "Lucas-Kanade 20 Years On: A Unifying Framework". en. In: *International Journal of Computer Vision* 56.3, pp. 221–255. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000011205.11775.fd. URL: http://link.springer.com/10.1023/B:VISI.0000011205.11775.fd (visited on 03/04/2025).

Bartlett, Marian Stewart et al. (Mar. 1999). "Measuring facial expressions by computer image analysis". en. In: *Psychophysiology* 36.2, pp. 253–263. ISSN: 0048-5772, 1469-8986. DOI: 10.1017/S0048577299971664. URL: https://onlinelibrary.wiley.com/doi/10.1017/S0048577299971664 (visited on 03/02/2025).

Bassingthwaighte, James B., Larry S. Liebovitch, and Bruce J. West (1994). *Fractal Physiology*. en. New York, NY: Springer New York. ISBN: 978-1-4614-7572-9. DOI: 10.1007/978-1-4614-7572-9. URL: http://link.springer.com/10.1007/978-1-4614-7572-9 (visited on 03/14/2025).

Batty, Eleanor et al. (2019). "BehaveNet: nonlinear embedding and Bayesian neural decoding of behavioral videos". en. In: *Advances in Neural Information Processing Systems*.

Bendor, Daniel and Xiaoqin Wang (June 2007). "Differential neural coding of acoustic flutter within primate auditory cortex". en. In: *Nature Neuroscience* 10.6, pp. 763–771. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/nn1888. URL: https://www.nature.com/articles/nn1888 (visited on 03/14/2025).

Bohnslav, James P et al. (Sept. 2021). "DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels". en. In: *eLife* 10, e63377. ISSN: 2050-084X. DOI: 10.7554/eLife.63377. URL: https://elifesciences.org/articles/63377 (visited on 03/04/2025).

Botvinick, Matthew M., Yael Niv, and Andew G. Barto (Dec. 2009). "Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective". en. In: *Cognition* 113.3, pp. 262–280. ISSN: 00100277. DOI: 10.1016/j.cognition.2008.08.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S0010027708002059 (visited on 08/12/2025).

Brown, Judith C. (Jan. 1991). "Calculation of a constant *Q* spectral transform". en. In: *The Journal of the Acoustical Society of America* 89.1, pp. 425–434. ISSN: 0001-4966, 1520-8524. DOI: 10.1121/1.400476. URL: https://pubs.aip.org/jasa/article/89/1/425/678890/Calculation-of-a-constant-Q-spectral (visited on 03/04/2025).

Calder, Andrew J et al. (Apr. 2001). "A principal component analysis of facial expressions". en. In: *Vision Research* 41.9, pp. 1179–1208. ISSN: 00426989. DOI: 10.1016/S0042-6989(01)00002-5. URL: https://linkinghub.elsevier.com/retrieve/pii/S0042698901000025 (visited on 03/02/2025).

Carroll, J. Douglas and Jih-Jie Chang (1970). "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition". en. In: *Psychometrika*.

Chatterji, S et al. (Oct. 2004). "Multiresolution techniques for the detection of gravitational-wave bursts". en. In: *Classical and Quantum Gravity* 21.20, S1809–S1818. ISSN: 0264-9381, 1361-6382. DOI: 10.1088/0264-9381/21/20/024. URL: https://iopscience.iop.org/article/10.1088/0264-9381/21/20/024 (visited on 03/04/2025).

Cohn, Jeffrey F. and Paul Ekman (Mar. 2008). "Measuring Facial Action". In: *The New Handbook of Methods in Nonverbal Behavior Research*. Ed. by Jinni Harrigan, Robert Rosenthal, and Klaus Scherer. Oxford University Press, pp. 9–64. ISBN: 978-0-19-852962-0. DOI: 10.1093/acprof:oso/9780198529620.003.0002. URL: https://academic.oup.com/book/25991/chapter/193829763 (visited on 03/02/2025).

Cramer, Nathan P. and Asaf Keller (July 2006). "Cortical Control of a Whisking Central Pattern Generator". en. In: *Journal of Neurophysiology* 96.1, pp. 209–217. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.00071.2006. URL: https://www.physiology.org/doi/10.1152/jn.00071.2006 (visited on 03/04/2025).

Darwin, Charles (1872). *The expression of the emotions in man and animals.* en. London: John Murray. DOI: 10.1037/10001-000. URL: https://content.apa.org/books/10001-000 (visited on 08/12/2025).

Dempsey, Bowen et al. (Nov. 2021). "A medullary centre for lapping in mice". en. In: *Nature Communications* 12.1, p. 6307. ISSN: 2041-1723. DOI: 10.1038/s41467-021-26275-y. URL: https://www.nature.com/articles/s41467-021-26275-y (visited on 04/30/2025).

Dolensek, Nate et al. (Apr. 2020). "Facial expressions of emotion states and their neuronal correlates in mice". en. In: *Science* 368.6486, pp. 89–94. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aaz9468. URL: https://www.science.org/doi/10.1126/science.aaz9468 (visited on 03/02/2025).

Ekman, Paul and Wallace V. Friesen (1976). "Measuring facial movement". en. In: *Environmental Psychology and Nonverbal Behavior* 1.1, pp. 56–75. ISSN: 0361-3496, 1573-3653. DOI: 10.1007/BF01115465. URL: http://link.springer.com/10.1007/BF01115465 (visited on 08/12/2025).

Ewert, Tobias A. S., Christiane Vahle-Hinz, and Andreas K. Engel (May 2008). "High-Frequency Whisker Vibration Is Encoded by Phase-Locked Responses of Neurons in the Rat's Barrel Cortex". en. In: *The Journal of Neuroscience* 28.20, pp. 5359–5368. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.0089-08.2008. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.0089-08.2008 (visited on 03/14/2025).

Friedman, Wendy A., H. Philip Zeigler, and Asaf Keller (Jan. 2012). "Vibrissae motor cortex unit activity during whisking". en. In: *Journal of Neurophysiology* 107.2, pp. 551–563. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.01132.2010. URL: https://www.physiology.org/doi/10.1152/jn.01132.2010 (visited on 03/04/2025).

Galen (1968). *On the Usefulness of the Parts of the Body* (*M. T. Tallmadge May, Trans.*)

Harshman, Richard A (1970). "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis". en. In: *UCLA working papers in phonetics*.

He, Kaiming et al. (Dec. 2021). *Masked Autoencoders Are Scalable Vision Learners*. en. arXiv:2111.06377 [cs]. DOI: 10.48550/arXiv.2111.06377. URL: http://arxiv.org/abs/2111.06377 (visited on 04/01/2026).

Hill, Daniel N. et al. (Oct. 2011). "Primary Motor Cortex Reports Efferent Control of Vibrissa Motion on Multiple Timescales". en. In: *Neuron* 72.2, pp. 344–356. ISSN: 08966273. DOI: 10.1016/j.neuron.2011.09.020. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627311008713 (visited on 06/29/2025).

Hogan, N (Nov. 1984). "An organizing principle for a class of voluntary movements". en. In: *The Journal of Neuroscience* 4.11, pp. 2745–2754. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.04-11-02745.1984. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.04-11-02745.1984 (visited on 08/11/2025).

Hsu, Alexander I. and Eric A. Yttri (Aug. 2021). "B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors". en. In: *Nature Communications* 12.1, p. 5188. ISSN: 2041-1723. DOI: 10.1038/s41467-021-25420-x. URL: https://www.nature.com/articles/s41467-021-25420-x (visited on 03/03/2025).

Kafetzopoulos, Evangelos, Stelios Gouskos, and Spiros N. Evangelou (Dec. 1997). "1/f noise and multifractal fluctuations in rat behavior". en. In: *Nonlinear Analysis: Theory, Methods & Applications* 30.4, pp. 2007–2013. ISSN: 0362546X. DOI: 10.1016/S0362-546X(96)00233-7. URL: https://linkinghub.elsevier.com/retrieve/pii/S0362546X96002337 (visited on 03/14/2025).

Kaku, Heet et al. (Jan. 2025). *A brainstem map of orofacial rhythms*. en. DOI: 10.1101/2025.01.27.635041. URL: http://biorxiv.org/lookup/doi/10.1101/2025.01.27.635041 (visited on 03/04/2025).

Kane, Gary A et al. (Dec. 2020). "Real-time, low-latency closed-loop feedback using markerless posture tracking". en. In: *eLife* 9, e61909. ISSN: 2050-084X. DOI: 10.7554/eLife.61909. URL: https://elifesciences.org/articles/61909 (visited on 07/08/2025).

Karaev, Nikita et al. (Oct. 2024). *CoTracker: It is Better to Track Together*. en. arXiv:2307.07635 [cs]. DOI: 10.48550/arXiv.2307.07635. URL: http://arxiv.org/abs/2307.07635 (visited on 04/01/2026).

Kleinfeld, David et al. (Dec. 2023). "Low- and high-level coordination of orofacial motor actions". en. In: *Current Opinion in Neurobiology* 83, p. 102784. ISSN: 09594388. DOI: 10.1016/j.conb.2023.102784. URL: https://linkinghub.elsevier.com/retrieve/pii/S0959438823001095 (visited on 03/04/2025).

Laughlin, Simon B., Rob R. De Ruyter Van Steveninck, and John C. Anderson (May 1998). "The metabolic cost of neural information". en. In: *Nature Neuroscience* 1.1, pp. 36–41. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/236. URL: https://www.nature.com/articles/nn0598_36 (visited on 08/04/2025).

Lee, Daniel D. and H. Sebastian Seung (Oct. 1999). "Learning the parts of objects by non-negative matrix factorization". en. In: *Nature* 401.6755, pp. 788–791. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/44565. URL: https://www.nature.com/articles/44565 (visited on 03/02/2025).

Li, Shan and Weihong Deng (July 2022). "Deep Facial Expression Recognition: A Survey". en. In: *IEEE Transactions on Affective Computing* 13.3, pp. 1195–1215. ISSN: 1949-3045, 2371-9850. DOI: 10.1109/TAFFC.2020.2981446. URL: https://ieeexplore.ieee.org/document/9039580/ (visited on 03/03/2025).

Lucas, Bruce D and Takeo Kanade (1981). "An Iterative Image Registration Technique with an Application to Stereo Vision". en. In: *Proceedings of the International Joint Conference on Artificial Intelligence* 7th.

Luxem, Kevin et al. (Nov. 2022). "Identifying behavioral structure from deep variational embeddings of animal motion". en. In: *Communications Biology* 5.1, p. 1267. ISSN: 2399-3642. DOI: 10.1038/s42003-022-04080-7. URL: https://www.nature.com/articles/s42003-022-04080-7 (visited on 03/04/2025).

Mackevicius, Emily L et al. (Feb. 2019). "Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience". en. In: *eLife* 8, e38471. ISSN: 2050-084X. DOI: 10.7554/eLife.38471. URL: https://elifesciences.org/articles/38471 (visited on 03/03/2025).

Mastrogiuseppe, Francesca and Srdjan Ostojic (Aug. 2018). "Linking Connectivity, Dynamics, and Computations in Low-Rank Recurrent Neural Networks". en. In: *Neuron* 99.3, 609–623.e29. ISSN: 08966273. DOI: 10.1016/j.neuron.2018.07.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627318305439 (visited on 07/08/2025).

Mathis, Alexander et al. (Sept. 2018). "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning". en. In: *Nature Neuroscience* 21.9, pp. 1281–1289. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-018-0209-y. URL: https://www.nature.com/articles/s41593-018-0209-y (visited on 03/04/2025).

Matsumoto, Hideyuki et al. (Oct. 2016). "Midbrain dopamine neurons signal aversion in a reward-context-dependent manner". en. In: *eLife* 5, e17328. ISSN: 2050-084X. DOI: 10.7554/eLife.17328. URL: https://elifesciences.org/articles/17328 (visited on 03/13/2025).

Mayrhofer, Johannes M. et al. (Sept. 2019). "Distinct Contributions of Whisker Sensory Cortex and Tongue-Jaw Motor Cortex in a Goal-Directed Sensorimotor Transformation". en. In: *Neuron* 103.6, 1034–1043.e5. ISSN: 08966273. DOI: 10.1016/j.neuron.2019.07.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627319306348 (visited on 03/04/2025).

Mohebi, Ali et al. (Apr. 2024). "Dopamine transients follow a striatal gradient of reward time horizons". en. In: *Nature Neuroscience* 27.4, pp. 737–746. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-023-01566-3. URL: https://www.nature.com/articles/s41593-023-01566-3 (visited on 07/08/2025).

Moore, Jeffrey D. et al. (May 2013). "Hierarchy of orofacial rhythms revealed through whisking and breathing". en. In: *Nature* 497.7448, pp. 205–210. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature12076. URL: https://www.nature.com/articles/nature12076 (visited on 03/04/2025).

Mørup, Morten, Lars Kai Hansen, and Sidse M. Arnfred (Apr. 2007). "ERPWAVELAB". en. In: *Journal of Neuroscience Methods* 161.2, pp. 361–368. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2006.11.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S016502700600567X (visited on 03/04/2025).

Musall, Simon et al. (Oct. 2019). "Single-trial neural dynamics are dominated by richly varied movements". en. In: *Nature Neuroscience* 22.10, pp. 1677–1686. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-019-0502-4. URL: https://www.nature.com/articles/s41593-019-0502-4 (visited on 03/04/2025).

Omrani, Mohsen et al. (Sept. 2017). "Perspectives on classical controversies about the motor cortex". en. In: *Journal of Neurophysiology* 118.3, pp. 1828–1848. ISSN: 0022-3077, 1522-1598. DOI: 10.1152/jn.00795.2016. URL: https://www.physiology.org/doi/10.1152/jn.00795.2016 (visited on 07/08/2025).

Paatero, Pentti and Unto Tapper (June 1994). "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values". en. In: *Environmetrics* 5.2, pp. 111–126. ISSN: 1180-4009, 1099-095X. DOI: 10.1002/env.3170050203. URL: https://onlinelibrary.wiley.com/doi/10.1002/env.3170050203 (visited on 03/04/2025).

Pereira, Talmo D. et al. (Apr. 2022). "SLEAP: A deep learning system for multi-animal pose tracking". en. In: *Nature Methods* 19.4, pp. 486–495. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-022-01426-1. URL: https://www.nature.com/articles/s41592-022-01426-1 (visited on 07/08/2025).

Peterson, Adam J. and Peter Heil (May 2019). "Phase Locking of Auditory-Nerve Fibers Reveals Stereotyped Distortions and an Exponential Transfer Function with a Level-Dependent Slope". en. In: *The Journal of Neuroscience* 39.21, pp. 4077–4099. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.1801-18.2019. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.1801-18.2019 (visited on 03/14/2025).

Salkoff, David B et al. (Jan. 2020). "Movement and Performance Explain Widespread Cortical Activity in a Visual Detection Task". en. In: *Cerebral Cortex* 30.1, pp. 421–437. ISSN: 1047-3211, 1460-2199. DOI: 10.1093/cercor/bhz206. URL: https://academic.oup.com/cercor/article/30/1/421/5618750 (visited on 03/04/2025).

Schneider, Steffen, Jin Hwa Lee, and Mackenzie Weygandt Mathis (May 2023). "Learnable latent embeddings for joint behavioural and neural analysis". en. In: *Nature* 617.7960, pp. 360–368. ISSN: 0028-0836,

1476-4687. DOI: 10.1038/s41586-023-06031-6. URL: https://www.nature.com/articles/s41586-023-06031-6 (visited on 03/03/2025).

Shadmehr, R and Fa Mussa-Ivaldi (May 1994). "Adaptive representation of dynamics during learning of a motor task". en. In: *The Journal of Neuroscience* 14.5, pp. 3208–3224. ISSN: 0270-6474, 1529-2401. DOI: 10.1523/JNEUROSCI.14-05-03208.1994. URL: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.14-05-03208.1994 (visited on 08/12/2025).

Shannon, C.E. (Jan. 1949). "Communication in the Presence of Noise". en. In: *Proceedings of the IRE* 37.1, pp. 10–21. ISSN: 0096-8390. DOI: 10.1109/jrproc.1949.232969. URL: http://ieeexplore.ieee.org/document/1697831/ (visited on 07/13/2025).

Stringer, Carsen et al. (Apr. 2019). "Spontaneous behaviors drive multidimensional, brainwide activity". en. In: *Science* 364.6437, eaav7893. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aav7893. URL: https://www.science.org/doi/10.1126/science.aav7893 (visited on 03/04/2025).

Syeda, Atika et al. (Jan. 2024). "Facemap: a framework for modeling neural activity based on orofacial tracking". en. In: *Nature Neuroscience* 27.1, pp. 187–195. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-023-01490-6. URL: https://www.nature.com/articles/s41593-023-01490-6 (visited on 03/04/2025).

Tamura, Keita et al. (Feb. 2025). "Cell-class-specific orofacial motor maps in mouse neocortex". en. In: *Current Biology*, S0960982225001198. ISSN: 09609822. DOI: 10.1016/j.cub.2025.01.056. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960982225001198 (visited on 03/04/2025).

Wang, Yanchen et al. (July 2025). *Animal behavioral analysis and neural encoding with transformer-based self-supervised pretraining*. en. arXiv:2507.09513 [q-bio]. DOI: 10.48550/arXiv.2507.09513. URL: http://arxiv.org/abs/2507.09513 (visited on 04/01/2026).

Williams, Alex H. et al. (June 2018). "Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis". en. In: *Neuron* 98.6, 1099–1115.e8. ISSN: 08966273. DOI: 10.1016/j.neuron.2018.05.015. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627318303878 (visited on 03/04/2025).

Wiltschko, Alexander B. et al. (Dec. 2015). "Mapping Sub-Second Structure in Mouse Behavior". en. In: *Neuron* 88.6, pp. 1121–1135. ISSN: 08966273. DOI: 10.1016/j.neuron.2015.11.031. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627315010375 (visited on 03/03/2025).

Ye, Shaokai et al. (June 2024). "SuperAnimal pretrained pose estimation models for behavioral analysis". en. In: *Nature Communications* 15.1. ISSN: 2041-1723. DOI: 10.1038/s41467-024-48792-2. URL: https://www.nature.com/articles/s41467-024-48792-2 (visited on 07/13/2025).

5 Methods

5.1 Face-rhythm pipeline

The standard face-rhythm pipeline consists of three major steps: (1) point-tracking, (2) spectrogram generation, (3) dimensionality reduction via tensor factorization. These steps are separable and composable.

A user-oriented library has been developed and is available here: https://github.com/RichieHakim/face-rhythm

5.1.1 Defining the region of interest / point positions

A region of the face is circumscribed, and equidistantly spaced points are placed to fill the region. The interpoint distance is set such that there are around 500–3000 points.

For analyses in which multiple sessions or multiple animals are used, a set of points is defined on one mouse face, rigid image registration is performed to align the fields of view, and then the points are warped to matched positions across all recording sessions. Rigid registration is performed using the `roicat.tracking.alignment` module from the `ROICaT` library: https://github.com/RichieHakim/ROICaT

5.1.2 Point-tracking

Point tracking is performed using the Lucas-Kanade sparse optical-flow algorithm (Lucas and Kanade 1981; Baker and Matthews 2004) and recursively integrating velocities calculated for each point after each frame. Two additional forces were applied to each point after every frame. First, a 'mesh physics' force similar to Hooke's law was used to shape the way individual points could move. Each point was 'pushed' and 'pulled' by its neighbors to maintain the local inter-point distances. Second, a gentle 'relaxation' force was applied to each point toward its initial position to counteract drift.

Optical-flow was chosen for its efficiency and lack of domain priors (i.e., it is not trained on a specific dataset). We use OpenCV's (Bradski 2000) implementation which exposes a variety of parameters to the user. The optical-flow local integration window size (parameter name: 'winSize') is set to be equivalent to roughly 1–3 mm in the field of view, which is typically around 20–60 pixels in our recordings. The optimization parameters were chosen to minimize drift and jump errors ('maxLevel' ≈ 3, 'maxCount' ≈ 3).

Contrast-limited adaptive histogram equalization (CLAHE) is applied to each frame of the input video prior to point tracking using OpenCV's `createCLAHE` class. This normalization step improves robustness to occlusions (such as whisker and tongue movements), changes in lighting, and drift. The face-rhythm pipeline exposes arguments for CLAHE and other parameters.

In each frame t, given the prior positions x_p^t and the original positions x_p^0, each point p is updated in four steps:

Step 1. **Lucas-Kanade prediction.**

$$\mathrm{x}_p^{\mathrm{LK}} = \mathrm{x}_p^t + \Delta\tilde{\mathrm{x}}_p^{\mathrm{LK}}, \tag{5.1}$$

where $\Delta\tilde{\mathrm{x}}_p^{\mathrm{LK}}$ is the displacement vector (i.e., 'velocity') obtained from the Lucas-Kanade optical-flow algorithm using OpenCV's `calcOpticalFlowPyrLK` function.

Step 2. **Violation freezing.**

$$\mathrm{x}_p^{\mathrm{frozen}} = \begin{cases} \mathrm{x}_p^t, & f_{\mathrm{viol}}(p,t) = 1, \\ \mathrm{x}_p^{\mathrm{LK}}, & f_{\mathrm{viol}}(p,t) = 0, \end{cases} \tag{5.2}$$

$$f_{\mathrm{viol}}(p,t) = \begin{cases} 1, & |\mathrm{x}_p^{\mathrm{LK}} - \mathrm{x}_p^t| > d_{\mathrm{thresh}}, \\ 0, & \text{otherwise}. \end{cases} \tag{5.3}$$

Step 3. **Mesh correction.**

$$\Delta\mathrm{x}_p^{\mathrm{mesh}} = \lambda_{\mathrm{mesh}} \frac{1}{|\nu(p)|} \sum_{j \in \nu(p)} \left((\mathrm{x}_p^t - \mathrm{x}_j^t) - \mathrm{d}_{0,pj} \right), \tag{5.4}$$

$$\mathrm{x}_p^{\text{mesh}} = \mathrm{x}_p^{\text{frozen}} - \Delta\mathrm{x}_p^{\text{mesh}}. \tag{5.5}$$

Step 4. **Relaxation.**

$$\Delta\mathrm{x}_p^{\text{relax}} = \lambda_{\text{relax}}\,(\mathrm{x}_p^{\text{mesh}} - \mathrm{x}_p^{0}), \tag{5.6}$$

$$\mathrm{x}_p^{t+1} = \mathrm{x}_p^{\text{mesh}} - \Delta\mathrm{x}_p^{\text{relax}}. \tag{5.7}$$

Variable definitions:

- x_p^t: position of point p on frame t.
- $\Delta\tilde{\mathrm{x}}_p^{\text{LK}}$: displacement vector (i.e., 'velocity') from the Lucas-Kanade algorithm.
- $\nu(p)$: neighbor indices of point p in the mesh.
- $\mathrm{d}_{0,pj}$: initial / 'resting' displacement vector (i.e., 'relative position') between point p and neighbor j.
- $\lambda_{\text{mesh}}, \lambda_{\text{relax}}$: mesh rigidity and relaxation scaling factors.
- x_p^0: initial position of point p.

Output
The resulting point-tracking tensor is denoted as $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T\times P\times 2}$, where T is the number of frames, P is the number of points, and the last dimension contains the coordinates (x and y) of each point. The elements of the output tensor contain the tracked position of each point for each time frame.

5.1.3 Spectrogram generation

Spectrograms are computed using a custom implementation of the variable-Q transform (VQT) (Brown 1991; Chatterji et al. 2004) (see Section 5.14 for details), followed by normalization. A spectrogram is computed for each point from the point-tracking output separately for the x and y dimensions and is denoted as $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{T\times P\times F\times 2}$, where F is the number of VQT filters/frequencies. For consistency in future steps, the spectrogram is reshaped to $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{2P\times F\times T}$, by concatenating the x- and y-displacements along the points dimension. The time dimension is typically downsampled ($\sim$15 Hz) relative to the original point-tracking tensor ($\sim$250 Hz).

Normalization
The spectrogram is normalized in several ways. For subsequent decomposition steps, it is useful to partially equalize along the frequency, temporal, and point dimensions. This normalization boosts the variance allotted to higher frequencies, time periods with less movement, and points with more subtle movements.

The normalization step includes three sequential operations controlled by user-set hyperparameters `spectrogram_exponent` α, `normalization_factor` γ, and `one_over_f_exponent` β:

1. **Exponentiation**

$$\boldsymbol{\mathcal{S}}_{\text{norm}} \leftarrow \boldsymbol{\mathcal{S}}^{\,\alpha}. \tag{5.8}$$

2. **Frame-wise power normalization** The total power at each time point t is

$$\mathbf{m}(t) = \frac{1}{P}\sum_{p=1}^{P}\sum_{k=1}^{F}\boldsymbol{\mathcal{S}}_{\text{norm}}(p,k,t)\ , \tag{5.9}$$

where $\mathbf{m} \in \mathbb{R}^T$, and each time point is rescaled via

$$\boldsymbol{\mathcal{S}}_{\text{norm}}(p,k,t) \leftarrow \frac{\boldsymbol{\mathcal{S}}_{\text{norm}}(p,k,t)}{\gamma\,\mathbf{m}(t) + \big(1-\gamma\big)}, \qquad 0 \le \gamma \le 1. \tag{5.10}$$

Thus, $\gamma = 0$ leaves the spectrogram unchanged, while $\gamma = 1$ enforces identical total power at each time bin.

3. $1/f$ **correction (optional)** To attenuate the natural $1/f$ decay in movement spectra, every frequency slice is multiplied by a power of its center frequency:

$$\boldsymbol{\mathcal{S}}_{\text{norm}}(p, k, t) \leftarrow \boldsymbol{\mathcal{S}}_{\text{norm}}(p, k, t)\ f_k^{\ \beta}, \tag{5.11}$$

where $\beta =$`one_over_f_exponent`. $\beta = 0$ disables the correction; positive β flattens the spectrum.

5.1.4 Dimensionality reduction via tensor factorization

Non-negative tensor component analysis (nnTCA) is applied to the spectrogram tensor (which is naturally non-negative) to obtain a set of components. Each output component describes a unique 'part' or 'kind' of behavior, as defined by each component's unique combination of spatial, spectral, and temporal factors.

Tensor factorization is performed using CANDECOMP/PARAFAC (CP) decomposition (Harshman 1970; Paatero and Tapper 1994; Carroll and Chang 1970; Mørup, Hansen, and Arnfred 2007; Williams et al. 2018). The primary optimization algorithm used is non-negative hierarchical alternating least-squares (nn-CP-HALS); though for very large tensors, alternative algorithms are used, such as custom mini-batch implementation of nn-CP-HALS or tensorly's Alternating Optimization Alternating Direction Method of Multipliers (AO-ADMM) algorithm. The tensorly library was used to perform all tensor factorization operations (Kossaifi et al. 2018).

The input spectrogram tensor is denoted (same as above) as $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{2P \times F \times T}$. The output of the tensor factorization is a set of components, each with a spatial, spectral, and temporal factor. The number of components in the decomposition is defined by the user-defined 'rank' R parameter. Each component can be denoted as a set of vectors (sometimes called a Kruskal-tensor), one for each factor: $[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \{\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r\}_{r=1}^{R}$, with the following loadings:

- $\mathbf{a}_r \in \mathbb{R}^{2P}$ or $\mathbf{A} \in \mathbb{R}^{2P \times R}$: 'Spatial factor' loads onto face points (both x and y).
- $\mathbf{b}_r \in \mathbb{R}^{F}$ or $\mathbf{B} \in \mathbb{R}^{F \times R}$: 'Spectral factor' loads onto VQT filters.
- $\mathbf{c}_r \in \mathbb{R}^{T}$ or $\mathbf{C} \in \mathbb{R}^{T \times R}$: 'Temporal factor' loads onto time bins.

The components can be combined to reconstruct a rank-R approximation of the original spectrogram tensor. The reconstruction is given by the following equation using the outer product representation:

$$\hat{\boldsymbol{\mathcal{S}}} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r \tag{5.12}$$

where $\otimes$ denotes the outer product operation. Alternatively, it can be expressed using Einstein summation notation (see Table 6):

$$\hat{\boldsymbol{\mathcal{S}}}_{\,2P,F,T} = \mathbf{A}_{2P,R}\, \mathbf{B}_{F,R}\, \mathbf{C}_{T,R} \tag{5.13}$$

Non-negative CP decomposition
Optimization with non-negative constraints is performed over the component vectors to minimize the reconstruction error (mean squared error, MSE) between the original spectrogram tensor and the reconstructed tensor:

$$\min_{\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r} \text{MSE}(\hat{\boldsymbol{\mathcal{S}}}, \boldsymbol{\mathcal{S}}) \tag{5.14}$$

In most cases, the initialization of the component vectors is done using truncated singular value decomposition (SVD) on the reshaped spectrogram tensor, which provides deterministic output components.

5.1.5 Rank selection

Rank is a user-selected hyperparameter that we chose by balancing reconstruction quality, decomposition stability, downstream utility, and interpretability. To evaluate rank systematically, we computed a set of complementary diagnostics across different ranks $R \in \{2, 4, 6, \ldots, 20\}$. Unless otherwise noted, high-order SVD (HOSVD) initialization was used. These diagnostics are summarized in **Figure 1 supplement 1** (panels A-E): panel A shows reconstruction error versus rank, panel B outlines the cross-initialization matching workflow, panel C reports per-mode and combined component consistency, panel D reports BIC, and panel E reports effective dimensionality of each factor matrix.

Reconstruction error and explained variance (panel A): For each rank, we computed the relative reconstruction error

$$\epsilon_{\text{rel}} = \frac{\|\boldsymbol{\mathcal{S}} - \hat{\boldsymbol{\mathcal{S}}}\|_F}{\|\boldsymbol{\mathcal{S}}\|_F}, \tag{5.15}$$

and the explained variance ratio

$$\text{EVR} = 1 - \frac{\|\boldsymbol{\mathcal{S}} - \hat{\boldsymbol{\mathcal{S}}}\|_F^2}{\|\boldsymbol{\mathcal{S}}\|_F^2}, \tag{5.16}$$

where $\boldsymbol{\mathcal{S}}$ is the input tensor and $\hat{\boldsymbol{\mathcal{S}}}$ is the CP reconstruction (Eq. 5.12).

Component consistency metric: To quantify cross-run reproducibility, we used the consistency framework described previously for tensor decomposition (Williams et al. 2018). For this analysis, we used the BMI dataset (see Section 5.6) and ran 30 random initializations per rank. For a pair of runs u and v, with factor matrices $\mathbf{A}^{(u)}, \mathbf{B}^{(u)}, \mathbf{C}^{(u)}$ and $\mathbf{A}^{(v)}, \mathbf{B}^{(v)}, \mathbf{C}^{(v)}$, we first computed per-mode cosine similarities between component r in run u and component s in run v:

$$s_{r,s}^{(m)} = \frac{\mathbf{M}_{:,r}^{(u)} \cdot \mathbf{M}_{:,s}^{(v)}}{\|\mathbf{M}_{:,r}^{(u)}\|_2 \, \|\mathbf{M}_{:,s}^{(v)}\|_2}, \quad m \in \{A, B, C\}, \tag{5.17}$$

where $\mathbf{M}_{:,r}^{(u)}$ denotes a single factor vector from mode m and initialization u. These per-mode similarities were combined into a joint cross-similarity matrix by taking the product across modes,

$$S_{r,s} = \prod_{m \in \{A,B,C\}} s_{r,s}^{(m)}, \tag{5.18}$$

and the Hungarian algorithm was used to find the assignment π that maximized $\sum_{r=1}^{R} S_{r,\pi(r)}$. After matching, we evaluated consistency using Pearson correlation ρ between matched factor vectors. We report two summary statistics as a function of rank: (1) the per-mode consistency, defined as the Pearson correlation between matched components averaged over components and run pairs, reported separately for each mode; and (2) a combined product consistency score,

$$\text{ProdScore} = \frac{1}{|\mathcal{P}|\, R} \sum_{(u,v) \in \mathcal{P}} \sum_{r=1}^{R} \prod_{m \in \{A,B,C\}} \rho_{r,(u,v)}^{(m)}, \tag{5.19}$$

where $\rho_{r,(u,v)}^{(m)}$ is the Pearson correlation for matched component r in mode m for run pair (u, v) and $\mathcal{P}$ is the set of all $\binom{30}{2} = 435$ run pairs. The product score provides a stricter measure of agreement that is sensitive to poor consistency in any individual mode.

BIC (panel D): Under a Gaussian residual model, we computed

$$\text{BIC} = N \ln(\text{RSS}/N) + k \ln N, \tag{5.20}$$

with

$$\text{RSS} = \|\boldsymbol{\mathcal{S}} - \hat{\boldsymbol{\mathcal{S}}}\|_F^2, \quad N = \prod_{n=1}^{3} d_n, \quad k = R(d_1 + d_2 + d_3 - 2). \tag{5.21}$$

Here, RSS is the residual sum of squares, N is the total number of observed tensor entries, and k is the effective number of free parameters in the rank-R CP model. In this setting, d_1, d_2, d_3 are the mode dimensions of $\boldsymbol{\mathcal{S}}$ (spatial, spectral, temporal), so $N = d_1 d_2 d_3$. The expression for k includes two redundant scaling degrees of freedom per component. Lower BIC indicates a better fit-complexity tradeoff. Because non-negative spectrogram residuals violate IID Gaussian assumptions, and because temporal/spatial correlations reduce effective sample size, we treated BIC as a supplementary heuristic rather than a formal model-selection test. We report BIC due to its stronger complexity penalty, but we note that AIC (Akaike information criterion) gives nearly identical scores.

Effective dimensionality of factor matrices (panel E): To quantify redundancy across extracted components, we performed PCA on each factor matrix $\mathbf{F}^{(n)} \in \mathbb{R}^{d_n \times R}$ (spatial/points, spectral/frequencies, temporal) and computed the effective dimensionality/rank required to explain 90% variance:

$$d_{\text{eff}}^{(m)} = \min \left\{ q : \frac{\sum_{i=1}^{q} \lambda_i^{(m)}}{\sum_{i=1}^{R} \lambda_i^{(m)}} \geq 0.9 \right\}, \tag{5.22}$$

where $\lambda_i^{(m)}$ are PCA eigenvalues for mode m. Values near R indicate decorrelated components in that mode, and values below R indicate increasing collinearity/redundancy.

In practice, final rank selection recognizes these diagnostics, but is balanced against human interpretability of component factors and downstream utilities like regression.

5.2 Cross-animal component consistency

To assess whether FR components generalize across individuals, we extended the within-session consistency framework (Section 5.1.5) to the cross-animal setting. Spectrogram tensors were computed for 13 baseline recording sessions across 5 mice using the standard FR pipeline. For each session, 30 non-negative CP decompositions were run with random initializations at each rank $R \in \{2, 4, 6, \ldots, 20\}$, and the run with the lowest reconstruction error was selected for cross-animal analysis. Point positions were aligned across mice using rigid image registration (Section 5.1.1).

Because temporal factors are session-specific, cross-animal component matching used only the spectral and spatial factors, and not the temporal factors. For each cross-animal session pair, per-mode cosine similarities were computed and combined into a joint cross-similarity matrix $S_{r,s} = s_{r,s}^{(A)} \cdot s_{r,s}^{(B)}$, following the same procedure as Section 5.1.5 but restricted to two modes. The Hungarian algorithm was used to find the optimal one-to-one assignment π^* maximizing $\sum_r S_{r,\pi(r)}$.

After matching, similarity was evaluated using Pearson correlation ρ between matched factor vectors. The product score for a session pair was defined as:

$$\text{ProdScore} = \frac{1}{R} \sum_{r=1}^{R} \rho_r^{(A)} \cdot \rho_r^{(B)}. \tag{5.23}$$

To assess whether cross-animal similarity exceeded chance, a one-sided permutation test was performed. For each of the $\binom{13}{2} = 78$ session pairs, 1000 random component assignments were drawn and the mean product score was computed. The p-value was the fraction of permutations in which the mean null score (averaged across all pairs) exceeded the actual mean cross-animal product score.

5.3 Ablation of point-tracking corrections

To quantify the contribution of each correction applied during point tracking (Section 5.1.2), we performed a full $2 \times 2 \times 2$ factorial ablation across three factors: mesh correction (M), relaxation (R), and violation detection with point pausing (V). This yielded eight conditions (Table 1), run on $n = 9$ sessions from 7 mice ($\sim$60 min each, 63.56 Hz, 1985 tracked points per session).

Label	Condition	λ_{mesh}	λ_{relax}	d_{thresh}
MRV	full model	0.1	0.01	300
MV	no relaxation	0.1	0	300
RV	no mesh	0	0.01	300
V	pure LK	0	0	300
MR	no violations	0.1	0.01	off
M	M only	0.1	0	off
R	R only	0	0.01	off
none	none	0	0	off

Table 1: Ablation conditions. λ_{mesh}: mesh rigidity; λ_{relax}: relaxation strength; d_{thresh}: displacement threshold for violation detection (px; "off" = disabled). Production parameters: $\lambda_{\text{mesh}} = 0.1$, $\lambda_{\text{relax}} = 0.01$, $d_{\text{thresh}} = 300$ px.

The following Lucas-Kanade parameters were used: `winSize`$= [40, 40]$, `maxLevel`$= 3$, `maxCount`$= 3$, mesh neighbor count $|\nu(p)| = 24$.

Ablation metrics

Tracking quality was evaluated using four per-frame metrics: (1) *cumulative drift*: $\|\mathrm{x}_p^t - \mathrm{x}_p^0\|_2$, displacement of each point from its initial position; (2) *deviation from reference*: Euclidean distance between instantaneous point positions and the original/initial reference points; (3) *neighbor velocity coherence*: mean cosine similarity of unit-normalized velocity vectors between each point and its 4 nearest mesh neighbors (velocity estimated as derivative between point positions from consecutive frames), where values near 1 indicate locally coherent motion; and (4) *violation rate*: fraction of points exceeding d_{thresh} on any given frame.

5.4 Widefield imaging regression comparison analysis

5.4.1 Video motion energy and PCA analysis

To capture facial movements from the behavioral video, we computed the absolute difference between pixels across each frame to generate the video motion energy (video ME). To isolate movements of specific parts of the face, we manually selected a rectangular area of 50 pixels around the nose, jaw, or whiskers and created the average ME across all pixels within each area. To reduce the dimensionality of the behavioral data, we then used principal component analysis (PCA) to compute the 200 highest-variance dimensions. These dimensions accounted for at least 90% of the total variance in the data, indicating that 200 dimensions were sufficient to describe the main temporal dynamics in the video. We then used the PCA scores as temporal components of animal behavior and used either 10 or all 200 components in the linear encoding model to predict cortical dynamics.

5.4.2 Widefield imaging and ridge regression analysis

Experimental details of the cortical widefield imaging are described previously in Musall et al. 2019 (Musall, Kaufman, et al. 2019). For data processing, we reduced the dimensionality of the video data with singular value decomposition (SVD) and retained the top 200 components, which explained $\geq 95\%$ of the total variance. SVD yielded spatial components (U; pixels $\times$ components), temporal components (VT; components $\times$ frames), and singular values (S). Subsequent analyses were performed on the product SVT to reduce computational load. SVT was high-pass filtered above 0.1 Hz using a zero-phase lag, second-order Butter-

worth filter. To project results back into pixel space, outputs were multiplied with U. All widefield data were rigidly aligned to the Allen common coordinate framework.

To link behavioral video components to cortical activity, we used a linear encoding model with a ridge penalty where the regularization penalty λ was estimated separately for each column of the widefield data (Couto et al. 2021; Musall, Sun, et al. 2023). The same model structure was used for all comparisons of cross-validated EVR in **Figure 2**, using either ME PCA, FR components, or point tracking data as model regressors to predict cortical activity.

To account for the nested structure of sessions within mice, we additionally fit a linear mixed-effects model (LMM) to the cross-validated EVR values. The model was specified as EVR $\sim$ method + (1|mouse), with method (FR vs. ME PCA) as a fixed effect and mouse as a random intercept, fit using restricted maximum likelihood (REML) via the `statsmodels` `MixedLM` implementation in Python. Significance is assessed with the Wald z-test on the estimated coefficient (`statsmodels` default). Animal-level paired t-tests were computed on per-mouse mean EVR values. We also report the non-parametric Wilcoxon signed-rank test outputs alongside each main parametric comparison.

5.4.3 Decomposition ablation analysis

To evaluate the contribution of each pipeline stage, we compared the standard FR decomposition (rank 10 non-negative TCA on VQT magnitude spectrograms) against alternative decompositions spanning three families. Spectrogram-domain methods operate on the VQT spectrogram tensor with the original shape $\mathbb{R}^{2\times P\times F\times T}$, which is flattened along the x- and y- coordinate dimensions to give $\mathbb{R}^{2P\times F\times T}$. The nnTCA-spectrogram (rank 10) decomposition operates directly on this 3D tensor, while all 2D decompositions operate on a flattened version of this tensor ($\mathbb{R}^{2PF\times T}$); these include PCA (rank 10 and 1000), NMF (rank 10), and related decompositions.

Point-tracking-domain methods operate on the raw displacement traces (prior to any spectral transformation) with the original shape ($\mathbb{R}^{2\times P\times T}$). For the "TCA: 3D unflattened point tracking rank=10" decomposition only, the original 3-way point-tracking tensor was used, preserving the separation between spatial and coordinate dimensions. For matrix decomposition methods like PCA, the x- and y-coordinate dimensions were first concatenated along the spatial dimension to form a 2D matrix ($\mathbb{R}^{2P\times T}$). For the "PCA: lagged point tracking" decomposition, time-shifted versions of the 2D matrix were concatenated to construct a truncated Hankel (time-delay embedding) matrix with 200 lags ($\sim$3.1 s), then PCA (rank 1000) was performed; anticausal, causal, and centered lag variants were tested and the center lag is presented in the main text. For the "TCA: bandpassed point tracking rank=10" decomposition, the 2D flattened point tracking matrix was bandpass filtered using the same spectral filters as the VQT spectrograms, then the resulting bandpassed tensor was reshaped to $\mathbb{R}^{2P\times F\times T}$ and nnTCA (rank 10) was performed on this tensor.

All decompositions were aligned to the widefield imaging timebase and evaluated using the same cross-validated ridge regression framework (Section 5.4.2). The same set of decompositions was applied to each alternative input source (FaceMap, CoTracker, CEBRA, and ViT-MAE; see below), enabling controlled comparisons across tracking and embedding methods. For decompositions (like 'PCA with rank=1000'), the requested rank serves as an upper bound on the number of extracted components: the achieved rank equals min(requested rank, input feature dimensionality). For example, using CEBRA embeddings with 10 dimensions as input, requesting rank 1000 yields the same result as requesting rank 10.

5.4.4 FaceMap comparison

To compare FR's dense optical-flow tracking against sparse keypoint tracking, we ran FaceMap (Syeda et al. 2024) on the same widefield-session videos (7 mice, 1–2 sessions per mouse, 9 sessions total). FaceMap uses a neural-network-based detector to track 15 facial keypoints, yielding a point-tracking tensor $\boldsymbol{\mathcal{X}}_{\text{FM}} \in \mathbb{R}^{T\times 15\times 2}$. We finetuned the FaceMap keypoint tracking model through an iterative process of manual labeling (15-frame batches), retraining (finetuning), and visual inspection on held-out sessions. After six iterations (90 total frames; $\sim$180 active labeling minutes), performance stabilized across all seven mice and nine sessions. We then deployed this final model to extract keypoints across all sessions. These trajectories were

passed through the same decomposition ablation pipeline used for FR: VQT spectrograms followed by each decomposition method (PCA, ICA, NMF, nnTCA, and Hankel-PCA variants) at matched ranks. FaceMap keypoints were also evaluated directly as regressors (without spectral transformation) via the same ridge regression framework described in Section 5.4.2, allowing a controlled comparison of point-tracking front-ends while holding all downstream analysis steps constant.

5.4.5 CoTracker point-tracking comparison

To test whether FR's custom Lucas-Kanade tracker could be replaced by a modern deep-learning tracker, we applied CoTracker v3 (Karaev, Rocco, et al. 2024; Karaev, Makarov, et al. 2024) to the same behavioral videos used in the widefield regression analysis. CoTracker is a transformer-based dense point tracker; we used the pretrained offline model (`scaled_offline.pth`) with 6 refinement iterations. Query points were initialized at the FR mesh positions ($\sim$1,481 points per session). Videos were processed in non-overlapping 300-frame chunks, with tracked positions carried across chunk boundaries.

CoTracker outputs absolute pixel coordinates; to obtain displacement signals comparable to FR, the temporal mean position was subtracted from each point. The resulting displacement tensor was processed through the full decomposition ablation pipeline (Section 5.4.3), aligned to the widefield timebase, and evaluated via the same ridge regression framework (Section 5.4.2).

5.4.6 CEBRA comparison

To evaluate whether a learned nonlinear embedding could substitute for the explicit spectral transformation in FR, we applied CEBRA (Schneider, Lee, and Mathis 2023), a contrastive-learning method that optimizes an InfoNCE objective (Oord, Li, and Vinyals 2019) to map high-dimensional time series into low-dimensional embeddings. We used the self-supervised CEBRA-Time variant (positive pairs defined by temporal proximity, no behavioral labels). The encoder was a single-frame MLP (`offset1-model`, 128 hidden units, GELU activations, L^2-normalized output), trained with batch size 1024, learning rate 10^{-4}, for 5,000 iterations. Input to CEBRA was the first 50 PCA components of the flattened FR point-tracking data. Embeddings were computed at dimensions 10 and 50.

Each embedding matrix was reshaped to $T \times d \times 1$ (where d is the embedding dimension) and processed through the full decomposition ablation pipeline, including VQT spectrogram computation, all decomposition methods, temporal alignment, and ridge regression evaluation.

5.4.7 ViT-MAE feature extraction

Motivated by BEAST (Wang et al. 2025), which demonstrated that vision-transformer representations of behavioral video predict neural activity, we evaluated frozen ViT-MAE embeddings as an alternative behavioral representation. We used a ViT-Base/16 model pretrained with masked autoencoding (He et al. 2021; Dosovitskiy et al. 2021) (checkpoint: `facebook/vit-mae-base`). To restrict the encoder to the same facial region used by FR, video frames were masked to the face region defined for FR point tracking (Section 5.1.1); pixels inside the region retained their original values, and pixels outside were set to zero. Masked frames were then resized to 224×224 pixels, normalized to ImageNet statistics, and passed through the encoder with masking ratio 0. The CLS-token embedding (768 dimensions) was extracted per frame using batched inference (batch size 128, float32).

CLS embeddings were aligned to the widefield timebase by linear interpolation. For compatibility with the decomposition pipeline, the 768-dimensional embedding was reshaped to 384×2 pseudo-channels; this reshaping is numerically transparent because all decomposition methods except TCA operate on a flattened 2D matrix and are therefore invariant to the $(768,)$ versus $(384, 2)$ representation. VQT spectrograms were computed on all 768 channels using the same parameters as for FR data. The full decomposition ablation suite was then applied. We additionally evaluated 200-dimensional SVD and 10-dimensional PCA reductions of the CLS embeddings directly via ridge regression.

5.5 Pavlovian conditioning analysis

To analyze the utility of FR components in describing uninstructed behaviors during Pavlovian conditioning, we used a separate facial video dataset acquired during the high- and low-reward probability classical conditioning tasks described previously (Matsumoto et al. 2016). These facial video data were not analyzed in that study and are distinct from the 60-Hz eye-video dataset used there for eye-blinking analysis.

5.5.1 Behavioral paradigm and data acquisition

Thirsty mice were head-fixed and randomly presented with one of three conditioned stimuli (CS) odors on each trial:

- CS+: odor predictive of a water reward,
- CS0: odor predictive of no outcome,
- CS-: odor predictive of an aversive air puff.

In the high-reward probability task, reward was delivered in 90% of CS+ trials, whereas in the low-reward probability task, reward was delivered in 20% of CS+ trials; air puff was delivered in 90% of CS- trials in both task variants, as described previously (Matsumoto et al. 2016). Each trial consisted of a $1\,\mathrm{s}$ CS odor presentation, followed by a $1\,\mathrm{s}$ delay epoch and then the unconditioned stimulus (US). Analyses were restricted to cued trials, yielding 240 analyzed trials per session (80 trials per CS class), across 20 sessions from 5 animals. Behavioral event times were taken from the original experimental records for alignment of FR temporal factors to task events. Continuous video of the face was recorded at $\sim 30\,\mathrm{Hz}$. The standard face-rhythm pipeline was used to extract 10 temporal factors using right-asymmetric ('causal') kernels/filters (see Section 5.14.2 for details). The four temporal factors shown in **Figure 3D-E** were selected based on visual inspection for display purposes only; all 10 temporal factors were used for the classification analysis.

5.5.2 Elastic net CS classifier

CS identity was predicted from FR temporal factors using multinomial logistic regression with elastic net regularization (L1+L2 penalty). For each trial, the 10 temporal factor traces were concatenated across all time bins in a $[-2.5,\ +1.9]$ s window relative to CS onset (≈25 bins per factor, ≈250 features total), ending one time bin before US onset ($2.0\,\mathrm{s}$) to exclude the US-evoked response. Regularization strength was selected via cross-validation within each session, and classifier performance was evaluated using stratified k-fold cross-validation ($k = 5$, 20 repeats).

For each session, a confusion matrix (rows: true class, columns: predicted class) was computed by summing across folds and row-normalized so that each entry gives $P(\hat{c} \mid c)$. The null expectation for each cell is the marginal prediction frequency $P(\hat{c})$, estimated from the column sums of the raw confusion matrix; under a classifier with no information about the true class, all rows of the confusion matrix are identical and equal to these marginal frequencies (sometimes referred to as a 'mean expectation model'). Per-animal means were computed by averaging across sessions within each animal. Significance for each cell was assessed with a two-sided one-sample t-test ($N = 5$ animals), testing whether the observed confusion-matrix entry differed from the corresponding null expectation $P(\hat{c})$ of 33.3%. A multiple-comparison Bonferroni correction over classes would give $\alpha/3 = 0.017$.

5.6 Decoding of BMI reward events

We compared four decoders that were trained to predict reward events (onset times of reward-triggering threshold crossings). These models make up a 2×2 matrix of input data type $\times$ architecture: input data include (FR temporal factors vs. raw point tracking) and model classes include (linear vs. RNN). We reference each as FR-RNN, FR-linear, Points-linear, and Points-RNN. All four share a common analysis framework, and we describe model-specific details below. See Section 5.11 for the behavioral paradigm and data acquisition.

5.6.1 Shared procedures

The goal of each decoder is to classify the instantaneous probability of a reward event at each video frame from either the FR temporal factors or the raw point-tracking data. Reward times are converted to a Boolean target vector $\mathbf{y} \in \{0,1\}^T$ with value 1 at each reward-event frame, widened by 200 ms in the backward (causal) direction. To test the temporal dependence of decoding on reward events, the target vector is shifted/lagged by $\Delta\tau$ before each training run: four 'in-context' lags $\Delta\tau \in \{-1.5, -0.5, 0, +0.5\}$ s relative to reward delivery, and an 'out-of-context' control at $\Delta\tau = +1200$ s ($+20$ min) that displaces the target far outside its reward window. For all FR-based runs, a right-asymmetric ('causal') VQT kernel is used (see Section 5.14.2), ensuring each temporal factor depends only on past information.

Cross-validation uses a group-shuffle-split protocol: each session is divided into contiguous, non-overlapping $\sim$10 min blocks randomly assigned to training (70%) and test (30%) partitions. Classifier performance is summarized by the area under the receiver-operating-characteristic curve (AUC-ROC), the area under the precision-recall curve (AUCPR), and the maximum F_1 score; see Section 5.6.5 for details.

5.6.2 FR-RNN

The FR-RNN receives rank-R temporal factors $\mathbf{C} \in \mathbb{R}^{T\times R}$ from the FR pipeline. The architecture is a single-layer LSTM with hidden size $h = 5$ and input dimensionality $R = 5$, trained on $\sim$5 s batches using the Adam optimizer with cross-entropy loss. Regularization is not used given the small network size. The model is implemented in PyTorch.

$$\mathbf{C} \in \mathbb{R}^{T\times R} \xrightarrow{\text{LSTM}} \hat{\mathbf{Y}} \in \mathbb{R}^{T\times 2}, \tag{5.24}$$

$$\min_{\text{RNN params}} \quad \texttt{cross-entropy}(\hat{\mathbf{Y}}, \mathbf{Y}), \text{ where } \mathbf{Y} \in \{0,1\}^{T\times 2}. \tag{5.25}$$

5.6.3 Linear decoders (FR-linear and Points-linear)

Both linear models are L2-regularized logistic regression classifiers with class weights inversely proportional to class frequency, implemented in scikit-learn.

FR-linear. Receives the same rank-R temporal factors as the FR-RNN, min-max scaled to $[0,\ 1]$ and mean-centered. `LogisticRegressionCV` with the L-BFGS solver is used; regularization strength C is selected via 5-fold cross-validation over $\sim$30 logarithmically spaced values in $[10^{-6},\ 10^{3}]$, optimizing AUC.

Points-linear. Operates on raw tracked-point positions ($P \approx 930$ keypoints $\times$ 2 coordinates, yielding $\mathbf{X} \in \mathbb{R}^{T'\times 2P}$ at 120 Hz), each column mean-centered. `LogisticRegression` with the SAGA solver is used; C is tuned via Bayesian optimization (Optuna; up to 20 trials over $[10^{-6},\ 10^{3}]$).

5.6.4 Points-RNN

Operates on the same raw tracked-point positions as Points-linear, reduced via per-session PCA to $K = 100$ components (computed on the full session prior to train/test splitting, analogous to how nnTCA is pre-computed for FR models). The model uses the same single-layer LSTM architecture as the FR-RNN ($h = 5$), with input dimensionality $K = 100$. Because reward events are rare ($\sim$1.1% of frames), the loss is weighted binary cross-entropy via PyTorch's `BCEWithLogitsLoss` (pos_weight $= n_{\text{neg}}/n_{\text{pos}}$), trained with Adam (lr $= 10^{-3}$) on random chunks of $\sim$5.3 s (636 frames at 120 Hz). Early stopping terminates training when smoothed validation loss improvement falls below 10^{-4} over a 30-check window, or after 1500 epochs; the checkpoint with highest test-set AUC is retained. At test time, inference is chunked, carrying the LSTM hidden state across chunks. Within each session, 10 independent repeats are run; the median AUC across repeats is computed per session, averaged per mouse, and summarized as mean $\pm$ s.e.m. ($N = 4$).

5.6.5 Precision-recall evaluation of reward-time decoders

We sweep a threshold θ over the model's output probabilities $\hat{p}_t \in [0, 1]$ using held-out test data and compute receiver operating characteristic (ROC) and precision-recall (PR) curves relative to the binary reward-event target $y_t \in \{0, 1\}$. To pool across heterogeneous test sets, curves are linearly interpolated onto common FPR and recall grids (200 points on $[0, 1]$), then averaged hierarchically (sessions within mouse, then across mice). Chance performance is the identity line for ROC and the positive-class rate π for PR.

The AUCPR metric is computed directly from probabilities and labels via `sklearn.metrics.average_precision_score`, which uses the step-function estimator.

$$\text{AUCPR} \;=\; \sum_k \big(\text{Recall}_k - \text{Recall}_{k-1}\big)\,\text{Precision}_k, \tag{5.26}$$

(this avoids the linear-interpolation bias of the trapezoidal rule on PR curves). The maximum F_1 score is the best achievable F_1 over all classifier thresholds, taken on the same PR curve,

$$F_1^{\max} \;=\; \max_k \frac{2\,\text{Precision}_k\,\text{Recall}_k}{\text{Precision}_k + \text{Recall}_k}. \tag{5.27}$$

At prevalence π the chance baselines are $\text{AUCPR}_{\text{chance}} \approx \pi$ and $F^{\max}_{1,\text{chance}} = 2\pi/(1+\pi)$. Values are averaged across cross-validation repeats and sessions.

To test whether in-context decoding exceeds the out-of-context (long-shifted control) baseline, for each (model, lag) condition, the per-mouse metric value m_i was paired with the same mouse's per-mouse value c_i at $\Delta\tau = +1200$ s, and the paired differences $d_i = m_i - c_i$ were tested for $\mathrm{E}[d] > 0$ using a one-sided paired t-test. Effect size is reported as paired Cohen's $d_z = \bar{d}/s_d$. Multiple comparisons are corrected using the Benjamini-Hochberg FDR procedure.

5.6.6 Decoder coefficient interpretability

To visualize and compare the coefficient structure of the two linear decoders (**Figure 4 supplement 2**), FR-linear and Points-linear were re-fit on single representative sessions using the same protocol as above. FR-linear receives the 10 FR temporal factors; Points-linear receives the raw point displacements.

Model-free summaries in **Figure 4 supplement 2Bii-Biv** are: per-point mean reward-triggered displacement (mean absolute per-point displacement in a window around reward events minus the baseline-window mean); per-point SNR (ratio of reward-window mean to baseline-window mean); and per-point signed point-biserial correlation (`scipy.stats.pointbiserialr` between the per-frame reward label and each point's per-frame displacement, signed by whichever axis has the larger absolute correlation).

5.7 Convolutional reduced-rank regression (convRRR)

Here, we describe a custom model that is used to predict neural activity from raw point-tracking data. This model combines convolution with reduced-rank regression (RRR) to learn a mapping from raw facial movements to neural activity. The overall structure of the model is related to a small linear convolutional neural network; however, the convolutional layer is implemented differently and is described below.

5.7.1 Model architecture

Let $\mathbf{X} \in \mathbb{R}^{2P \times T}$ be the behavioral data matrix (raw displacements of points on the face) and $\mathbf{Y} \in \mathbb{R}^{N \times T}$ the deconvolved neural activity matrix, where $2P$ is twice the number of points, N is the number of neurons, and T is the number of time samples. For a temporal window $\tau = 0, \ldots, L-1$ that describes the index of the convolutional kernel or the temporal lag between the kernel and the signal, we define a rank-R convolutional reduced-rank regression model (convRRR) in three stages.

(i) Encoding Project the raw face point displacement signals onto R spatial weight vectors. For each rank in $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{2P \times R}$ we can calculate an encoding vector $\mathbf{e}_r$:

$$\mathbf{e}_r = \mathbf{w}_{\text{enc},r}^{\top} \mathbf{X}, \quad r = 1, \ldots, R, \tag{5.28}$$

resulting in the intermediate matrix $\mathbf{E} \in \mathbb{R}^{R \times T}$.

(ii) Convolution The R encoded channels are convolved with a set of R convolution kernels (Fourier convolution is used in code, see Section 5.12 for details). The convolution operation proceeds in one of three ways, depending on the model type:

- (ii.a) Basic convolution

$$\mathbf{z}_r = \mathbf{k}_r * \mathbf{e}_r, \tag{5.29}$$

  where $\mathbf{k}_r \in \mathbb{R}^L$ is the convolution kernel for component r.

- (ii.b) Spectral convolution

$$\mathbf{z}_r = \left\| \mathbf{k}_r * \mathbf{e}_r \right\|, \tag{5.30}$$

  where $\mathbf{k}_r \in \mathbb{C}^L$ is the convolution kernel for component r, and optimization independently learns the real and imaginary parts of the kernel.

- (ii.c) Hilbert-spectral convolution

$$\mathbf{z}_r = \left\| (\mathcal{H}[\mathbf{k}_r]) * \mathbf{e}_r \right\|, \tag{5.31}$$

  where $\mathbf{k}_r \in \mathbb{R}^L$ is the convolution kernel for component r, and $\mathcal{H}[\cdot]$ is the Hilbert transform operator.

(iii) Decoding Linearly combine convolved channels:

$$\hat{\mathbf{Y}} = \mathbf{W}_{\text{dec}} \mathbf{Z}, \tag{5.32}$$

Unified form We combine the encoding and decoding steps that make up the RRR with the single notation summarizing the three versions of the convolution step using the following equation:

$$\hat{\mathbf{Y}} = \mathbf{W}_{\text{dec}} \Big(\mathbf{K} * (\mathbf{W}_{\text{enc}}^{\top} \mathbf{X}) \Big), \tag{5.33}$$

$$\mathbf{X} \in \mathbb{R}^{2P \times T} \xrightarrow{\text{enc}} \mathbf{E} \in \mathbb{R}^{R \times T} \xrightarrow{\text{conv}} \mathbf{Z} \in \mathbb{R}^{R \times T} \xrightarrow{\text{dec}} \hat{\mathbf{Y}} \in \mathbb{R}^{N \times T}. \tag{5.34}$$

where $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{2P \times R}$, $\mathbf{K} \in \mathbb{R}^{R \times L}$ or $\mathbb{C}^{R \times L}$, $\mathbf{W}_{\text{dec}} \in \mathbb{R}^{N \times R}$, and the result $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times T}$. The convolution operator $[*]$ represents one of the three convolution types described above and is performed on each rank independently.

5.7.2 Optimization

The model is trained to minimize the mean squared error (MSE) between the predicted neural activity $\hat{\mathbf{Y}}$ and the true neural activity $\mathbf{Y}$:

$$\min_{\mathbf{W}_{\text{enc}}, \mathbf{W}_{\text{dec}}, \mathbf{k}_r} \text{MSE}(\hat{\mathbf{Y}}, \mathbf{Y}) \tag{5.35}$$

All parameters $\{\mathbf{W}_{\text{enc}}, \mathbf{W}_{\text{dec}}, \mathbf{K}\}$ are optimized jointly with ridge regularization using either the Adam or L-BFGS optimizer (using PyTorch's built-in implementations). Hyperparameter optimization is performed using cross-validation and the `optuna` library over the ridge regularization strength. Model performance evaluation is performed on a fully held-out test set that is not used for training or hyperparameter optimization.

5.7.3 Model performance

We calculate model performance as the explained variance ratio (EVR) between the true neural activity signals for each neuron $\mathbf{y}_n$ and a rescaled version of the predicted signals from the model $\beta_n \hat{\mathbf{y}}_n$. The rescaling is performed using ordinary least squares (OLS); this step results in the final EVR metric being invariant to differences in the scale of the true and predicted signals.

$$\beta_n = (\hat{\mathbf{y}}_n^\top \hat{\mathbf{y}}_n)^{-1} \hat{\mathbf{y}}_n^\top \mathbf{y}_n, \tag{5.36}$$

where $\hat{\mathbf{y}}_n$ is the predicted signal for neuron n and $\mathbf{y}_n$ is the true signal for neuron n.

$$\text{EVR} = 1 - \frac{\sum_{n=1}^{N} \|\mathbf{y}_n - \beta_n \hat{\mathbf{y}}_n\|_2^2}{\sum_{n=1}^{N} \|\mathbf{y}_n - \bar{y}_n\|_2^2}, \tag{5.37}$$

where $\bar{y}_n$ is the mean of the true signal for neuron n, and $\|\cdot\|_2$ is the L^2 norm.

Statistical comparisons between model types (**Figure 5F**) use the two-sided paired Student's t-test, across $N = 5$ mice.

5.7.4 Band-pass filtering

To examine frequency-specific performance, we additionally rerun analyses using band-pass filtered behavioral input signals. A set of 10–20 band-pass filters is designed with center frequencies logarithmically spaced between 0.01–40 Hz, a Q-factor of 5, and Hamming windowing. FIR filtering is used to apply these filters (see Section 5.13). The filtered behavioral signal is then used in the same convRRR pipeline as above to compute the band-specific EVR. Statistical comparisons between model types at each frequency band are corrected for multiple comparisons using the Bonferroni method across bands.

5.8 Single-neuron coupling to face-rhythm components

We quantify how individual M1 neurons relate to rhythmic facial-movement components in two complementary ways: (i) phase locking (phase-variant) and (ii) envelope correlation (phase-invariant). For this analysis, we use rank-$R = 12$ FR decompositions per mouse across $N = 5$ mice (yielding 60 total behavioral components).

5.8.1 Phase-coupling analysis

For each FR component r we generate a matched pair of time series, starting with the raw point-tracking displacement matrix $\mathbf{X} \in \mathbb{R}^{2P \times T}$ and the deconvolved neural activity matrix $\mathbf{Y} \in \mathbb{R}^{N \times T}$, where N is the number of neurons and T is the number of time samples.

We generate phase-variant behavior signals by projecting the $\mathbf{X}$ matrix onto individual FR spatial factors $\mathbf{a}_r \in \mathbb{R}^{2P}$, and then spectrally filtering the result by treating the corresponding spectral factor $\mathbf{b}_r \in \mathbb{R}^F$ as a finite-impulse response (FIR) filter kernel (see Section 5.13 for details on FIR filtering):

$$\mathbf{x}_r^{\text{filt}} = \text{FIR}[\mathbf{a}_r^\top \mathbf{X}, \mathbf{b}_r], \tag{5.38}$$

By taking the angle of the analytic signal (via a Hilbert transform over the time dimension), yielding $\mathbf{x}_r^{\text{H}} \in \mathbb{C}^T$, we obtain the instantaneous phase of the behavioral signal:

$$\mathbf{x}_r^{\text{H}} = \mathcal{H}[\mathbf{x}_r^{\text{filt}}], \tag{5.39}$$

$$\mathbf{x}_r^{\phi} = \angle \mathbf{x}_r^{\text{H}}, \tag{5.40}$$

where $\mathcal{H}[\cdot]$ is the Hilbert transform operator, $\angle$ is the complex angle/phase operator, and $\mathbf{x}_r^{\phi} \in [-\pi, \pi]^T$ is the phase of the filtered behavioral signal.

A similar procedure is applied to the deconvolved neural activity $\mathbf{Y}$. The same spectral filter $\mathbf{b}_r$ is used to match the frequency content of the neural activity to that of the behavioral signal. Combining the above 3 steps, we obtain the phase-variant coupling signal for neuron n and component r:

$$\mathbf{y}^{\phi}_{n,r} = \angle\mathcal{H}[\text{FIR}[\mathbf{y}_n, \mathbf{b}_r]], \tag{5.41}$$

Pairwise phase consistency (PPC)
For neuron n and component r, we define the phase difference between the neural activity and the behavioral signal as:

$$\Delta\boldsymbol{\phi}_{n,r} = \mathbf{y}^{\phi}_{n,r} - \mathbf{x}^{\phi}_{r} \ , \tag{5.42}$$

We can then measure the phase-variant coupling using a simplified implementation of the *pairwise phase consistency* (PPC) metric (Vinck, Van Wingerden, et al. 2010; Vinck, Battaglia, et al. 2012):

$$\text{PPC}_{n,r} = \frac{(\sum_t \cos\Delta\boldsymbol{\phi}_{n,r})^2 + (\sum_t \sin\Delta\boldsymbol{\phi}_{n,r})^2 - T}{T(T-1)}, \tag{5.43}$$

We assessed the statistical significance of each observed PPC value using a parametric z-score approach: PPC was computed repeatedly on 60 randomly circularly time-shifted behavioral signals to estimate the null mean μ_{null} and standard deviation σ_{null}, and a z-score for the observed PPC value was calculated as $z_{n,r} = (\text{PPC}_{n,r} - \mu_{\text{null}})/\sigma_{\text{null}}$. Two-sided p-values were obtained from the standard normal CDF. These p-values are corrected for multiple comparisons across the N neurons tested using the Bonferroni method (N comparisons per component).

5.8.2 Envelope correlation

To obtain a phase-invariant behavioral signal, we take the magnitude of the analytic signal of the filtered behavioral signal (from Eq. 5.39), commonly referred to as computing the spectral envelope:

$$\mathbf{x}^{\text{env}}_{r} = \left\|\mathbf{x}^{\text{H}}_{r}\right\| . \tag{5.44}$$

Phase-invariant coupling is measured with Pearson's correlation (ρ) between the filtered behavioral envelope and the unfiltered neural activity:

$$\rho_{n,r} = \text{corr}\left(\mathbf{x}^{\text{env}}_{r}, \mathbf{y}_n\right). \tag{5.45}$$

Significance is determined using the same parametric z-score procedure as for PPC, with Bonferroni correction applied across the N neurons (N comparisons per component).

5.9 Independent factorization of neural and facial activity

Here, we describe the approach used to model the joint space of facial movements during head-restrained free locomotion and simultaneously recorded neural activity.

5.9.1 Data acquisition and preprocessing

Data were acquired from the baseline recording sessions of the BMI experiment described in Section 5.11 and Section 5.11.5. This includes widefield facial video and two-photon calcium imaging of layer 2/3 neurons in the primary motor cortex (M1) of head-fixed mice during free locomotion and sparse reward consumption.

Facial videos are processed through the first two steps of the face-rhythm pipeline (Section 5.1) to produce spectrogram tensors $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{2P\times F\times T}$. Calcium imaging data are processed to produce a deconvolved neural activity matrix $\mathbf{Y} \in \mathbb{R}^{N\times T}$. Spectrogram tensors are linearly interpolated to synchronize with the neural time series.

5.9.2 Decomposition of temporal factors

We note that non-negative CP decomposition (as used in the standard FR pipeline) is equivalent to non-negative matrix factorization (NMF) when the input data are two-dimensional (i.e., a matrix).

We applied non-negative CP decomposition (using nn-CP-HALS, see Section 5.1.4) to both the neural activity and spectrogram tensors independently, producing temporal factors for each modality. The rank R was set to 10 for both decompositions.

$$\boldsymbol{\mathcal{S}} \approx \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r, \qquad \mathbf{Y} \approx \sum_{r=1}^{R} \mathbf{u}_r \otimes \mathbf{v}_r, \tag{5.46}$$

where $\mathbf{a}_r \in \mathbb{R}^{2P}$, $\mathbf{b}_r \in \mathbb{R}^{F}$, $\mathbf{c}_r \in \mathbb{R}^{T}$ are the spatial, spectral, and temporal factors of the behavioral spectrogram tensor; and $\mathbf{u}_r \in \mathbb{R}^{N}$, $\mathbf{v}_r \in \mathbb{R}^{T}$ are the neural and temporal factors, respectively.

5.9.3 Matching temporal factors

The temporal factors of each modality $\mathbf{V} \in \mathbb{R}^{T\times R}$ and $\mathbf{C} \in \mathbb{R}^{T\times R}$ are then compared by finding a one-to-one pairing that maximizes the summed Pearson correlation.

$$\pi^* \;=\; \arg\max_{\pi} \;\mathrm{trace}\Big(\mathrm{corr}\big(\mathbf{V},\, \mathbf{C}^{(\pi)}\big)\Big)\,. \tag{5.47}$$

where $\mathbf{C}^{(\pi)}$ denotes the column-permutation of $\mathbf{C}$ by π. The Hungarian algorithm (implemented using the `scipy.optimize.linear_sum_assignment` function) is used to find π^*, producing the optimally matched columns of $\mathbf{V}$ and $\mathbf{C}$.

Pearson's correlation is calculated between the matched temporal factors, and significance is assessed via a parametric z-score approach: the null mean and standard deviation are estimated from $n_{\text{shuf}} = 25{,}000$ circularly time-shifted controls, and one-sided p-values are obtained from the standard normal CDF. Bonferroni correction is applied across all matched pairs pooled across mice.

5.10 Joint factorization of neural-behavioral coherence

Here, we describe the approach used to decompose the joint space of facial movement and neural activity during the performance of the BMI task (see Section 5.11 for details on the experimental setup).

5.10.1 Data acquisition and preprocessing

Data are acquired from the experimental sessions described in Section 5.11. This includes widefield facial video and two-photon calcium imaging of layer 2/3 neurons in the primary motor cortex (M1) of head-fixed mice.

Data are synchronized and peri-reward epochs are extracted from both datasets. The peri-reward epochs are defined as a $[-5\,\mathrm{s},\ 0\,\mathrm{s}]$ window relative to the reward onset time. The facial video is processed through the face-rhythm pipeline (Section 5.1) to produce the peri-reward point-tracking tensor $\boldsymbol{\mathcal{X}}^{(M)} \in \mathbb{R}^{2P\times T\times M}$. Neural activity data are processed to produce a similar tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N\times T\times M}$, where N is the number of neurons, T is the number of time samples, and M is the number of trials.

We compute the coherence between the point-tracking signals of all the facial points and the neural activity of all neurons.

$$\boldsymbol{\Phi} = \mathrm{coh}(\mathbf{X}, \mathbf{Y}), \tag{5.48}$$

where $\mathrm{coh}(\cdot)$ is the coherence function, which is computed along the temporal dimension, producing $\boldsymbol{\Phi} \in \mathbb{R}^{2P\times N\times F\times M}$.

We then factorize the coherence tensor using non-negative CP decomposition (mini-batch nn-CP-HALS, see Section 5.1.4) to obtain joint factors:

$$\boldsymbol{\Phi} \approx \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r \otimes \mathbf{d}_r, \tag{5.49}$$

where $\mathbf{a}_r \in \mathbb{R}^{2P}$ is the spatial factor for facial points, $\mathbf{b}_r \in \mathbb{R}^N$ is the neural factor, $\mathbf{c}_r \in \mathbb{R}^F$ is the frequency factor, and $\mathbf{d}_r \in \mathbb{R}^M$ is the trial factor. The rank R is set to 6 to approximately match the number of factors extracted in the baseline of the BMI experiment (see Section 5.11.6 for details).

5.10.2 Component selection and comparison to BMI decoder

We compare the neural loading that each animal is tasked with activating during the BMI experiment $\mathbf{w}_{\text{dec}} \in \mathbb{R}^N$ (the same weighting for all sessions; see Section 5.11.5 and Section 5.11.6 for details) against the neural factors extracted using the joint neural-behavioral factorization. Pearson's correlation is calculated between each 'joint neural' factor $\mathbf{b}_{r_1}$ in $\mathbf{B} \in \mathbb{R}^{N \times R_1}$ and each factor extracted during the baseline/day 0 recording sessions of the BMI experiment ('BMI baseline factors') $\mathbf{w}_{r_2}$ in $\mathbf{W} \in \mathbb{R}^{N \times R_2}$, which includes $\mathbf{w}_{\text{dec}}$. We define one of the joint neural factors as associated with the animal's 'strategy' $\mathbf{b}_{\text{strat}}$ by looking at the trial factors $\mathbf{D} \in \mathbb{R}^{M \times R}$ and selecting the component r_{strat} with the largest proportional increase in magnitude over all trials (concatenated across all sessions). We find that in 3 out of 4 mice, $\mathbf{w}_{\text{dec}}$ is most strongly correlated with $\mathbf{b}_{\text{strat}}$.

5.11 BMI experiment data

5.11.1 Animals and surgery

6-week-old male C57BL/6J mice underwent surgery and were group housed in a reverse light/dark cycle. Craniotomies and intracranial injections were performed under isoflurane anesthesia (5% for induction, 0.5–1.5% for maintenance). Mice were implanted with a titanium headplate and a glass window over the right motor cortex (M1) for two-photon imaging. Stereotactic coordinates were (AP +1.6 mm, ML −1.9 mm) relative to bregma. The headplate was secured to the skull using dental cement, and the glass window was glued to the skull using cyanoacrylate glue. AAV injections of AAV9-hsyn-jGCaMP8m-WPRE or PHP.eB-syn-Ribo-L1-jGCaMP8s were made using a beveled pulled glass micropipette needle; 6–14 injections at 200 nl each were made near the center of the craniotomy based on vascular anatomy with an infusion rate of 50–75 nl per minute. Virus was diluted to 1.5×10^{12} vg/mL. Mice were allowed to recover for at least 10 days before any experimental procedures were performed. Buprenorphine, ketoprofen, and saline were given for post-surgical analgesia and recovery. After the recovery period, mice were partially water restricted to 85–90% body weight and given 1.0–1.5 mL of water for maintenance. All procedures were approved by the Institutional Animal Care and Use Committee (IACUC) at Harvard Medical School.

5.11.2 Data acquisition

Mice are head-fixed, placed on a linear treadmill, and allowed to run freely. Widefield video of the face is acquired at 120 Hz, 1 ms exposure time, and MPEG-2 compression, using a FLIR Flea3-13Y3M-C camera illuminated with infrared light. Videos of calcium activity in layer 2/3 neurons of M1 are acquired at 30 Hz. A custom-built resonant-galvo two-photon microscope is controlled using ScanImage. A 16× 0.8-NA Nikon (N16XLWD-PF) objective is used, the excitation laser wavelength is 960 nm from a Ti:sapphire laser (Coherent Chameleon Ultra), the power is set to at most 70 mW, and the resolution is set to 512×1024 pixels. Data are also collected from a treadmill velocity sensor and a lick spout sensor. A synchronization pulse is sent to and recorded by each camera, and is used to align all data streams.

5.11.3 Neural data preprocessing

Neural data are preprocessed using the Suite2p (Pachitariu et al. 2016) and ROICaT pipelines. For each region-of-interest (ROI), Suite2p produces: an ROI mask, neuropil mask, ROI fluorescence time series, neu-

ropil time series, and deconvolved calcium event time series. For some analyses, the deconvolved calcium event time series was used, while for others the dF/F is computed as:

$$\mathrm{dF/F} = \Delta\mathbf{F}/\mathbf{F} = \frac{\dot{\mathbf{F}} - \mathbf{F}_{\text{baseline}}}{\mathbf{F}_{\text{baseline}}} \tag{5.50}$$

where $\dot{\mathbf{F}}$ is the neuropil-corrected fluorescence time series computed as:

$$\dot{\mathbf{F}} = \mathbf{F} - 0.7 \cdot \mathbf{F}_{\text{neuropil}}, \tag{5.51}$$

and $\mathbf{F}_{\text{baseline}}$ is the baseline fluorescence time series computed as the rolling 30th percentile of the fluorescence time series over a 10-minute window.

$$\mathbf{F}_{\text{baseline}} = \texttt{rolling_percentile}(\dot{\mathbf{F}},\ q = 30,\ window = 10\,\text{min}). \tag{5.52}$$

ROI classification is performed using the ROICaT library. Image embeddings are computed for each ROI mask, visualized in a 2D space using UMAP (McInnes, Healy, and Melville 2020), and a region of the embedding projection is selected to include all ROIs that look like neurons. ROIs that look like dendrites, axons, non-neuronal structures, or out-of-focal-plane neurons are excluded. Subsequent quality inclusion criteria are applied to each ROI based on their dF/F time series. These include: (i) minimum ROI to neuropil variance ratio, (ii) maximum explained variance of ROI by neuropil, (iii) minimum baseline ROI fluorescence, (iv) minimum signal-to-noise ratio (SNR), (v) maximum dF/F value, and (vi) maximum variance of baseline fluorescence time series. Only ROIs that pass all inclusion criteria and are also classified as neurons are included in subsequent analyses.

5.11.4 BMI design

Mice are trained to control their neural activity to achieve rewards in a closed-loop brain-machine interface (BMI) experiment. The data gathered during this experiment fall into two categories: (i) baseline recordings of natural behavior prior to training, and (ii) BMI training sessions in which mice are provided with real-time feedback of their neural activity and are rewarded for achieving a target.

During BMI training sessions, the neural activity of the animal is processed, decoded, and converted into a controllable auditory and visual cursor that is played back to the animal in real-time to allow the animal to learn to control the cursor's dynamics (see Section 5.11.6 and Section 5.11.7 for details on decoder design and Section 5.11.8 on the decoder forward algorithm). The input to the decoder is the instantaneous activity of all neurons being monitored in the field-of-view (FOV). The output, referred to as the 'cursor', is a bounded scalar value $c \in [0, 1]$ representing the animal's instantaneous performance. The cursor value is converted into a tone frequency that is played back to the animal, as well as the position of an illuminated LED on a linear LED strip. If the cursor value is above a certain threshold, the animal is also rewarded with a drop of soy milk. The tone frequency is logarithmically mapped to the cursor value between $2\,\text{kHz}$ and $18\,\text{kHz}$, with the reward threshold at $18\,\text{kHz}$. The blue LED strip is oriented from left to right across the mouse's visual field, and $\sim$1 light on the LED strip is illuminated at a time; the position is linearly mapped to the cursor value with the reward threshold at the right-most position.

- **Auditory feedback:** A sine wave / pure tone is generated on each frame with tone frequency $f_{\text{tone}}(t)$:

$$f_{\text{tone}}(t) \;=\; 2\,\text{kHz} \times \left(\frac{18\,\text{kHz}}{2\,\text{kHz}}\right)^{c}.$$

- **Visual feedback:** A pixel on a linear LED strip containing $L_{\text{LED}} = 72$ contiguous segments is illuminated on each frame, where the illuminated segment index $i_{\text{LED}}(t)$ is computed as:

$$i_{\text{LED}}(t) \;=\; \lceil c \times L_{\text{LED}} \rceil,$$

  which maps left-to-right across segments 1–72.

5.11.5 Baseline session recordings

Baseline session Baseline recordings are collected for 1–4 days / sessions prior to BMI training. During these 1-hour sessions, mice are played back the auditory and visual cursor feedback from a randomly selected previous BMI session of a trained animal. Sparse soy milk rewards are given by setting the cursor threshold value so that they receive ∼0.5 rewards per minute. This playback is fully open-loop and independent of the animal's behavior or neural activity. Neural data from these days are preprocessed using Suite2p; region-of-interest (ROI) classification and cross-session tracking are performed using ROICaT, and custom pipelines for quality control are used to obtain the final dF/F or deconvolved calcium event time series of high-quality neurons tracked across the baseline sessions. The number of neurons tracked that passed the inclusion criteria is roughly 200–800 neurons per session. For some analyses, single-day recordings are used without tracking, with the number of included neurons being roughly 700–1500 per session.

Reference image acquisition A three-dimensional reference image is acquired on the first day of the baseline sessions to enable precise spatial alignment across all experimental days. A dense z-stack is acquired with 41 optical slices at a $0.8\,\mu$m step size, spanning a total axial range of $32\,\mu$m centered around the chosen focal plane. Each slice is averaged across 10 acquisition iterations with 60 frames each to create a high-quality reference image for each z-position. The final z-stack that is used for online motion correction is a subset of the dense z-stack, consisting of 5 evenly spaced slices at $4\,\mu$m intervals for computational efficiency during online motion correction.

5.11.6 Decoder axis selection process

We use the neural activity recorded from the baseline sessions to define a set of up to 6 neural factors $\mathbf{W} \in \mathbb{R}^{N\times F}$. One of these factors $\mathbf{w}_{\text{dec}}$ is used as the 'decoder axis' for the real-time decoder, which maps the neural activity to the instantaneous cursor value.

Pre-orthogonalization
To reduce the influence of task-irrelevant factors on the selected decoder axis, we orthogonalized out signals like slow temporal trends, motion artifacts, structured noise, and population-averaged neural activity from the baseline neural activity matrix $\mathbf{X}$ prior to any factorization. These included:

1. Average neural activity
2. Average neuropil activity
3. Slow temporal trends
4. Session indicator variables (for concatenated baseline sessions)

The average neural (1) and neuropil (2) activity is split into bandpassed components using a set of FIR band-pass filters with boundaries at $[0, 0.003, 0.05, 0.5, 5, \infty]$ Hz, and Kaiser windowing with $\beta = 3.0$. The slow temporal trends (3) are modeled using a linear time trend and a nonlinear M-spline basis function over the duration of the temporal dimension with order 3 and 4 basis functions. The session indicator variables (4) are used to remove any spurious 'jumps' in offsets introduced by concatenating multiple sessions of neural activity. These variables are placed into a design matrix $\mathbf{D} \in \mathbb{R}^{T\times Z}$, where Z is the total number of featurized time series variables.

Orthogonalization is performed by subtracting off the projections of the design matrix from the neural activity matrix $\mathbf{X} \in \mathbb{R}^{T\times N}$ (transposed from other sections), where N is the number of neurons and T is the number of time samples. First, to improve stability, PCA is performed on the design matrix:

$$\Omega = \text{PCA}(\mathbf{D}) \,, \qquad \Psi = \mathbf{D}\Omega \,, \tag{5.53}$$

where $\Omega \in \mathbb{R}^{Z\times R}$ are principal components (loadings) and $\Psi \in \mathbb{R}^{T\times R}$ are scores. Then, we compute the

optimal projection of the design matrix onto the neural activity matrix using ordinary least squares (OLS):

$$\hat{\mathbf{X}} = \Psi\Big(\big(\Psi^{\mathsf{T}}\Psi\big)^{-1}\Psi^{\mathsf{T}}\mathbf{X}\Big) \,, \tag{5.54}$$

where $\hat{\mathbf{X}} \in \mathbb{R}^{T\times N}$ is the reconstructed neural activity matrix based on the design matrix. The orthogonalized neural activity matrix is then computed as:

$$\mathbf{X}^{\text{orth}} = \mathbf{X} - \hat{\mathbf{X}} \,, \tag{5.55}$$

where $\mathbf{X}^{\text{orth}}$ is the orthogonalized neural activity matrix. For simplicity, the above three steps can be notated as:

$$\mathbf{X}^{\text{orth}} = \mathbf{X} \perp \mathbf{D} \,, \tag{5.56}$$

SPCA
Following pre-orthogonalization, a low-dimensional neural subspace $\mathbf{W}^{\text{SPCA}} \in \mathbb{R}^{N\times F}$ is generated using sparse-PCA (sklearn's `sklearn.decomposition.SparsePCA` implementation). We used a grid search over the sparsity hyperparameter (L1 penalty) α such that the number of non-zero neuron loadings is approximately 100, averaged across factors. The rank R is set to 6, and any factors with less than 30 non-zero neuron loadings are discarded. A small L2 penalty is also used to ensure the stability of the optimization. The resulting factors are ordered by their variance explained in the pre-orthogonalized neural data. Sparse PCA was selected because it discovers factors that are aligned to the latent distributional modes better than standard PCA.

Decoder weight fine-tuning
To define the final factor space $\mathbf{W}$, we then process the SPCA factors with the following steps:

- **Reprojection:** The SPCA factors $\mathbf{W}^{\text{SPCA}}$ are reprojected using Ridge regression to regularize and distribute the weight representation across more neurons so as to increase robustness to changes in single-neuron signal-to-noise or tuning. This is done by defining a new weight matrix $\tilde{\mathbf{W}} \in \mathbb{R}^{N\times F}$ to predict the SPCA scores $\mathbf{X}\mathbf{W}^{\text{SPCA}}$ from the pre-orthogonalized neural activity $\mathbf{X}^{\text{orth}}$:

$$\tilde{\mathbf{W}} \leftarrow \text{Ridge}(\mathbf{X}^{\text{orth}}, \mathbf{X}\mathbf{W}^{\text{SPCA}}) \,, \tag{5.57}$$

- **Orthogonalization:** The weight matrix $\tilde{\mathbf{W}}$ is then orthogonalized to ensure that the factors are orthogonal to each other. This is done using a Gram-Schmidt-like process, where the orthogonalization operator $\perp$ is derived from Eq. 5.56

$$\tilde{\mathbf{W}} \leftarrow \{\mathbf{w}_1,\ \mathbf{w}_2 \perp \mathbf{w}_1,\ \mathbf{w}_3 \perp (\mathbf{w}_1,\ \mathbf{w}_2),\ \ldots,\ \mathbf{w}_F \perp (\mathbf{w}_1, \ldots, \mathbf{w}_{F-1})\} \,, \tag{5.58}$$

  and $\mathbf{w}_i$ is the ith factor in $\tilde{\mathbf{W}}$.

- **Mean subtraction:** The mean of each factor in $\tilde{\mathbf{W}}$ is then subtracted to center the factors around zero (effectively orthogonalizing the factors to the mean of the neural population activity).

- **Sign flipping:** The sign of each factor in $\tilde{\mathbf{W}}$ is flipped, if necessary, so that the skewness of the factor distribution is positive. In nearly all cases, no flipping was necessary, as SPCA factors tend to align with the natural positive skewness of the neural activity distribution.

  The final decoder weight matrix is then defined as $\mathbf{W} \in \mathbb{R}^{N\times F} \leftarrow \tilde{\mathbf{W}}$.

5.11.7 Neural decoder definition

The neural decoder is designed to incentivize the animal to voluntarily control its neural activity specifically along the decoder axis. The decoder is implemented as:

$$c_t = |\theta_{\mathbf{w}_{\text{dec}}}(\mathbf{s}_t)|^p \cdot \left\| \text{proj}_{\mathbf{w}_{\text{dec}}}(\mathbf{s}_t) \right\|_2 , \tag{5.59}$$

where $\mathbf{w}_{\text{dec}} \in \mathbb{R}^N$ is the decoder weight vector ('decoder axis'), which is selected as a column from the $\mathbf{W}$ matrix. $\mathbf{s}_t \in \mathbb{R}^N$ is the instantaneous neural activity vector (see Section 5.11.9). $\theta_{\mathbf{w}_{\text{dec}}}(\cdot)$ is the cosine similarity between $\mathbf{w}_{\text{dec}}$ and another vector, and $\text{proj}_{\mathbf{w}_{\text{dec}}}(\cdot)$ is the projection operator onto $\mathbf{w}_{\text{dec}}$. Lastly, a scalar power coefficient p is used to control the angular selectivity of the decoder.

An offline simulation of BMI performance is conducted on the selected factor using the baseline recording dataset to validate the decoder's performance and tune any task-related parameters. We typically select the factor with the second highest explained variance, dependent on the simulation being able to achieve a satisfactory number of rewards. We avoid choosing $\mathbf{w}_1$, the first factor / principal component, as a target factor, as a previous report suggests that the first factor is significantly related to a "neural engagement" during the trial period (Hennig et al. 2021). The main parameter that is tuned is τ_R, which defines the cursor reward threshold; this value is tuned so that the simulated reward rate is ~ 1.25 rewards per minute.

Qualitatively, the cursor output increases linearly with increases along the axis of the selected factor (i.e., the 'decoder axis'), but falls off sublinearly with angle changes away from the decoder axis. When the power coefficient is zero ($p = 0$), the decoder becomes a simple projection magnitude decoder, and when $p \to \infty$, the cursor output is only non-zero when the neural activity is perfectly aligned with the factor direction. For the BMI task, we use $p = 2$ to balance selectivity and linearity.

5.11.8 Task structure

Here, we describe the methods used to control the task's state-transitions and the delivery of rewards during the BMI task. The task is structured as a state machine with several states: (i) Trial periods, when the sensory feedback is on and the animal is eligible to receive a reward, (ii) reward periods, when the feedback is briefly paused at the current value and a $\sim 3.5\,\mu$l soy milk droplet is delivered, (iii) the inter-trial-interval, when feedback is off and rewards are not available, (iv) timeout periods when the animal has not achieved a reward over the maximum trial period duration (~ 20 s), and (v) 'waiting for quiescence' periods where the animal must reduce its neural activity along the decoder axis in order to start a new trial period.

Algorithm 1 Finite-state controller for the closed-loop BMI task

Require: incoming neural vector $\mathbf{s}_t$ (30 Hz)
Require: decoder axis $\mathbf{w}_{\text{dec}}$, exponent p
Require: reward threshold θ_{reward}, quiescence threshold θ_{quiet}
Require: quiescent duration n_{quiet}, maximum trial length n_{trial}
Require: maximum session frames T_{session}, maximum reward number R_{max}

State names
  WaitQuiescent: feedback off, waiting for low activity
  TrialActive: feedback on, reward eligible
  DeliverReward: feedback frozen, soy-milk delivery
  Timeout: enforced pause after an over-long trial

initialize frame index $t \leftarrow 0$, reward count $r \leftarrow 0$
initialize counters $q \leftarrow 0$ ▷ quiescence duration counter
initialize $trialTimer \leftarrow 0$ ▷ trial duration counter
state ← WaitQuiescent ▷ initialize the state machine
while $t < T_{\text{session}}$ **and** $r < R_{\text{max}}$ **do** ▷ run until session end or maximum rewards
  acquire $\mathbf{s}_t$; $t \leftarrow t + 1$ ▷ extract and preprocess neural activity from incoming image frames
  $m_t \leftarrow \left\|\text{proj}_{\mathbf{w}_{\text{dec}}}(\mathbf{s}_t)\right\|_2$ ▷ decoder magnitude
  $c_t \leftarrow \left|\theta_{\mathbf{w}_{\text{dec}}}(\mathbf{s}_t)\right|^p m_t$ ▷ cursor value
  if state = WaitQuiescent **then** ▷ wait and check for quiescence
    if $m_t < \theta_{\text{quiet}}$ **then** ▷ if decoder magnitude is below quiescence threshold
      $q \leftarrow q + 1$
    else
      $q \leftarrow 0$
    end if
    if $q \geq n_{\text{quiet}}$ **then** ▷ if quiescence achieved
      state ← TrialActive; $trialTimer \leftarrow 0$ ▷ change state to start a trial
    end if
  else if state = TrialActive **then**
    provide audiovisual feedback proportional to c_t
    $trialTimer \leftarrow trialTimer + 1$
    if $c_t \geq \theta_{\text{reward}}$ **then** ▷ if cursor value reaches reward threshold
      deliver reward; $r \leftarrow r + 1$
      state ← DeliverReward; $d \leftarrow 0$ ▷ change state to deliver reward
    else if $trialTimer \geq n_{\text{trial}}$ **then** ▷ if no reward is achieved within the maximum trial length
      state ← Timeout; $to \leftarrow 0$ ▷ change state to timeout
    end if
  else if state = DeliverReward **then**
    freeze feedback; $d \leftarrow d + 1$
    if $d \geq 0.5\,\text{s} \times 30$ **then**
      state ← WaitQuiescent; $q \leftarrow 0$
    end if
  else if state = Timeout **then**
    $to \leftarrow to + 1$
    if $to \geq 3\,\text{s} \times 30$ **then**
      state ← WaitQuiescent; $q \leftarrow 0$
    end if
  end if
end while

5.11.9 BMI software system

The software suite consists of an offline analysis pipeline written in Python for defining the decoder weights and preparing ROI masks, and a real-time integration and control system written in MATLAB and controlled via a ScanImage user function for online decoding and feedback. Code is available at https://github.com/RichieHakim/BWAIN.

On 'day 0' (defined as the day prior to the first BMI session with feedback), the decoder weights are computed, ROI masks are defined, and a simulation is conducted to set the cursor reward threshold. The decoder weights are then used to compute the cursor value in real-time during the BMI sessions.

Closed-loop latency was found to be 10–25 ms, defined as the time from the end of scanning a frame to the output of new feedback output signals.

Non-rigid image transformation
To compensate for tissue deformation over days/sessions and maintain spatial correspondence of neural decoder masks across sessions, we apply a non-rigid transformation to the ROI masks, neuropil masks, z-stack, and spatial decoder weights on each experiment day. To compute the warping field, we use MATLAB's `imregdemons` function with 500 iterations across 4 pyramid levels and an accumulated field smoothing parameter of 2. The reference image is acquired on 'day 0' and each experiment day we acquire a 10-minute-long calibration recording and generate an image to warp to the reference. During this time, the animal is in an idle state without sensory feedback.

On each experiment day, after manually aligning the current field of view to coordinates from the day 0 session, a mean reference image is computed using an average of the most stable frames (identified by minimal motion artifacts). Then, we apply a non-rigid image warp to ensure robust coordinate alignment between sessions.

Motion correction
Online motion correction has two components: a fast planar correction to the current imaging frame and a slower axial correction to a rolling average of the imaging plane.

Planar correction Online motion correction during a BMI experiment is implemented using a custom phase-correlation algorithm. To improve robustness, we apply bandpass filtering to the input images using a third-order Butterworth filter with cutoff frequencies at $[1/64, 1/4]$ cycles/pixel (image resolution is cropped from 512×1024 to 512×512).

Axial correction For online axial ('z-axis') drift correction, we use the warped reference z-stack (from day 0) to compare the current image frame in order to determine the current z-slice position. A rolling average of 60 frames is correlated against the warped z-stack, and the estimate of current z-position is determined by maximum phase-correlation. When the estimated current z-position is different from the center plane, we move the imaging plane by $0.5\,\mu$m in the direction of the center plane. Axial drift correction is only performed during the inter-trial-interval (ITI) period of the task.

Online BMI
During BMI sessions, neural signals are extracted, classified, and used to update the current cursor value in real time.

Online neural data preparation Online neural signal extraction is performed in real-time at a 30 Hz frame acquisition rate. The online extraction process is designed to match, as closely as possible, the extraction process used by Suite2p during offline ROI time series signal extraction. Raw fluorescence signals $F \in \mathbb{R}^{N \times T}$ are calculated as weighted sums of pixel intensities from each flattened frame $\mathbf{a}_t$ in the video $\mathbf{A} \in \mathbb{R}^{H \cdot W \times T}$ (where H and W are the height and width of the image frame) and ROI masks $\mathbf{M} \in \mathbb{R}^{N \times H \cdot W}$:

$$\mathbf{F} = \mathbf{MA} \ , \tag{5.60}$$

where $\mathbf{F} \in \mathbb{R}^{N \times T}$ is the fluorescence time series for N ROIs and T time samples. The ROI masks are non-rigidly transformed to match the current image frame using the non-rigid transformation field computed during the motion correction step.

A similar process is used to compute the neuropil signal $\mathbf{F}_{\text{neuropil}} \in \mathbb{R}^{N \times T}$, using the non-rigidly transformed neuropil masks. The neuropil signal is then subtracted from the ROI signal using a scaling factor of 0.7, matching the offline dF/F calculation (similar to Eq. 5.50):

Baseline fluorescence $\mathbf{F}_{\text{baseline}} \in \mathbb{R}^{N \times T}$ is updated every 10 frames using a rolling 30th percentile calculation over a 10-minute sliding window. For online quality inclusion criteria, the minimum baseline ROI fluorescence and the maximum dF/F value are applied with parameters that match the offline process (see Section 5.11.3). Any ROIs that do not meet the inclusion criteria are replaced with NaN and ignored in subsequent calculations.

Trial types BMI training sessions included two types of catch trials, which are sparsely distributed throughout the session. These are not used in any of the analyses in this study. In the first type of catch trial, feedback is decoupled and randomly played back from a different session, and rewards could occur. In the second type, neither the feedback nor the reward is available.

Task parameters Each trial had a maximum duration of 20 s, with minimum inter-trial intervals (ITIs) of 3 s. Any threshold crossing event is discarded if it lasted less than 3 frames. The reward is calibrated to be $\sim 3.5\,\mu$l per delivery.

5.12 Fourier convolution

For several analyses, we used Fourier convolution ('FFTconv') in place of direct convolution for efficiency and simplicity. We note that the convolution theorem provides the following equality:

$$\text{FFTconv}(\mathbf{x}, \mathbf{k}) = \mathcal{F}^{-1}\Big[\mathcal{F}[\mathbf{x}] \odot \mathcal{F}[\text{flip}[\mathbf{k}]]\Big] \qquad = \qquad \text{conv}(\mathbf{x}, \mathbf{k}) = \mathbf{x} * \mathbf{k} \tag{5.61}$$

where $\mathcal{F}[\cdot]$ is the Fourier transform operator, $\mathcal{F}^{-1}[\cdot]$ is the inverse Fourier transform operator, $\odot$ is element-wise multiplication, flip$[\cdot]$ is the flip operator (which reverses the order of the elements along the time dimension), and $l \in \{0, \ldots, L-1\}$ is the index of the convolution kernel $\mathbf{k}$. For specific applications, we checked for equivalence with direct convolution to avoid instances of non-negligible ringing and edge artifacts.

5.13 Finite-impulse response (FIR) filtering

For several analyses, we used a zero-phase finite-impulse response (FIR) spectral filter. Filtering is applied via forward-backward convolution to prevent phase distortion.

We implemented the spectral filter using Fourier convolution, similar to above (see Section 5.12):

$$\text{FIR}(\mathbf{x}, \mathbf{k}) = \mathcal{F}^{-1}\Big[\mathcal{F}[\mathbf{x}] \odot \mathcal{F}[\mathbf{k}] \odot \mathcal{F}[\text{flip}[\mathbf{k}]]\Big] \tag{5.62}$$

5.14 Variable-Q transform (VQT)

The VQT is a time-frequency analysis method that is similar to the short-time Fourier transform (STFT) but with a logarithmic frequency scale and a tunable frequency-dependent bandwidth. The VQT is often used to allow for low-frequency signals to be represented with high temporal-resolution and low frequency-resolution ($Q \approx 4$), while high-frequency signals can be represented with low temporal-resolution and high frequency-resolution ($Q \approx 20$). The VQT is related to a Morlet wavelet transform, and is similar to the constant-Q transform (CQT) but with a variable Q-factor.

Our implementation computes spectrograms via a convolution between the signal and a set of pre-computed filters, whereas the original VQT implementation uses recursive down-sampling to approximate lower-frequency filters (Brown 1991; Chatterji et al. 2004). Our implementation results in an exact solution without artifacts, more flexibility in the choice of parameters, and the ability to arbitrarily customize filter kernels.

5.14.1 VQT implementation

First, a set of filters/kernels is generated. Let f_s be the sampling rate, F the number of filters, and $\{f_k\}_{k=1}^{F}$ the center frequencies. We specify two endpoints Q_{low} and Q_{high} and interpolate each filter's quality factor logarithmically:

$$Q_k = Q_{\text{low}} \left(\frac{Q_{\text{high}}}{Q_{\text{low}}} \right)^{(k-1)/(F-1)}, \quad k = 1, \ldots, F. \tag{5.63}$$

The window length for filter k is

$$L = \left\lceil \frac{Q_k \, f_s}{f_k} \right\rceil , \tag{5.64}$$

We define the filter kernel / impulse response function $\mathbf{h}_k$ of a single filter k as a complex-valued windowed sinusoid:

$$\mathbf{h}_k[n] = \mathbf{w}_k[n] e^{2\pi i f_k n / f_s}, \quad n = 0, \ldots, L-1, \tag{5.65}$$

where $\mathbf{w}_k$ is a (possibly asymmetric) window function (like Hanning) of length L. The VQT coefficients can be computed by convolution (denoted by $[*]$) between the input signal $\mathbf{x}$ and the filter kernel $\mathbf{h}_k$. Optionally, convolution can be implemented using Fourier convolution for efficiency (see Section 5.12 for details):

$$\mathbf{z}_k = \mathbf{x} * \mathbf{h}_k \ , \qquad \mathbf{Z} = \{\mathbf{z}_k\}_{k=1}^{F} \ , \qquad \mathbf{z}_k \in \mathbb{C}^T, \ \mathbf{Z} \in \mathbb{C}^{F \times T}, \ \mathbf{x} \in \mathbb{R}^T, \ \mathbf{h}_k \in \mathbb{C}^L, \tag{5.66}$$

where $\mathbf{Z}$ is the complex spectrogram. Finally, the magnitude spectrogram $\mathbf{S}$ is obtained via:

$$\mathbf{S} = \left\| \mathbf{Z} \right\| , \qquad \mathbf{S} \in \mathbb{R}^{F \times T}. \tag{5.67}$$

5.14.2 Asymmetric windows

Asymmetric windows can be used to zero out one side/half of the filter. For several analyses, we used right-asymmetric (i.e., 'causal') kernels/filter windows, which only encode past information (i.e., 'trailing signal') at each time point. Our implementation applies a Heaviside window (on top of any other tapering window functions, like Hanning), and optionally tapers/smooths the step transition to reduce ringing artifacts.

5.15 Computational complexity

Symbol	Definition
F	Number of video frames
H, W	Frame height and width (pixels)
P	Number of tracked points
K	Mesh neighbors per point
W_{lk}	Lucas-Kanade window size
L	Lucas-Kanade pyramid levels
B_f	VQT frequency bins
D	Temporal downsampling factor
T	Downsampled time frames ($= F/D$)
R	nnTCA rank
I_{max}	nnTCA maximum iterations

Table 2: Variables used in computational complexity analysis.

Point tracking is the dominant computational step in the entire pipeline with a per-frame cost $O(HW + P \cdot W_{\text{lk}}^2 \cdot L)$ for frame preprocessing (grayscale conversion, CLAHE) and sparse pyramidal optical flow, plus $O(PK)$ for mesh regularization (where K is the number of mesh neighbors). This complexity then scales linearly with total number of frames. Peak memory is determined by the video read buffer, which is user-configurable. In practice, for maximum speed, this value is around 2–20 GB depending on the size of the video and speed of the processor (faster processors benefit from larger buffers to reduce I/O overhead).

Spectral analysis (VQT) applies B_f bandpass filters to each of $2P$ displacement traces via FFT convolution, with total time complexity $O(P \cdot B_f \cdot F \log F)$ per video.

Tensor decomposition (nnTCA) uses Hierarchical Alternating Least Squares (HALS) with per-iteration cost $O(R \cdot P \cdot B_f \cdot T)$, operating on the downsampled spectrogram tensor. Total time is $O(I_{\text{max}} \cdot R \cdot P \cdot B_f \cdot T)$. Twice the size in memory of the spectrogram tensor is required for this step; in practice, spectrogram tensors are often 1–20 GB in size depending on the user-configured parameters and video length. Therefore, some consideration is required when choosing to use a GPU for this step since GPU memory is often more limited than CPU RAM.

In practice, point tracking accounts for the majority of wall-clock time. For a one-hour recording on a typical consumer computer, VQT typically completes in well under 10 min on CPU and under 1 min on GPU; nnTCA typically completes in well under an hour on CPU and a few minutes on a GPU. Peak memory usage is typically determined by the nnTCA decomposition.

5.16 Software dependencies and availability

The face-rhythm pipeline is implemented in Python and depends on standard scientific computing packages including NumPy, SciPy, PyTorch, and OpenCV. GPU acceleration requires an NVIDIA GPU with CUDA support. The complete list of dependencies and version requirements is specified in the package repository. All code repositories developed for this work include open-source licenses:

- face-rhythm: https://github.com/RichieHakim/face-rhythm
- ROICaT: https://github.com/RichieHakim/ROICaT
- Incremental PCA in PyTorch: https://github.com/RichieHakim/incremental_pca
- FastICA implemented in PyTorch: https://github.com/RichieHakim/FastICA_torch
- Variable-Q transform: https://github.com/RichieHakim/vqt
- BMI control software: https://github.com/RichieHakim/BWAIN

5.17 Best practices

5.17.1 Video capture and acquisition

Although FR is fairly robust to acquisition conditions, several choices can improve the discovery and signal-to-noise ratio (SNR) of the extracted components. Acquisition frame rates should satisfy the Nyquist criterion for the highest frequency of interest; in practice we use 60–250 Hz to capture the full range of rodent facial rhythms (e.g., whisking up to ∼25 Hz, licking ∼5–10 Hz, sniffing ∼3–15 Hz). Spatial resolution should provide ∼10–30 px/mm over the region of interest, ensuring that the Lucas-Kanade integration window (∼1–3 mm, Section 5.1.2) contains adequate local texture. Global-shutter sensors are preferred over rolling-shutter sensors to avoid intra-frame motion artifacts, and short exposure times (e.g., <4 ms) reduce motion blur.

Heavy video compression, especially codecs with temporal compression (e.g., low-bitrate H.264), can introduce block artifacts that propagate through point tracking, and thus should be avoided. In our experiments, we use MPEG-2 compression at a high bitrate (e.g., >50 Mbps), though many other settings may be suitable.

Stable, high-contrast illumination is the second major determinant of tracking quality. We illuminate the face with bright and diffuse infrared light (two $50\,\mathrm{W}$, $\sim 850\,\mathrm{nm}$ LED chips, invisible to rodents) positioned at a distance to produce shadowing over the face.

5.17.2 Processing hardware

The face-rhythm pipeline is designed to run on consumer-grade hardware. For video processing, a modern multi-core CPU with at least $16\,\mathrm{GB}$ of RAM is recommended for the kinds of datasets presented in this work (~ 1500 points, $\sim 1\,\mathrm{hr}$ duration, $\geq 600 \times 600$ pixels). For spectral analysis and tensor decomposition, any CPU is sufficient, but a GPU with at least $8\,\mathrm{GB}$ of VRAM is recommended for efficient processing.

5.18 Notation guide and style conventions

Index	Length	Description
t	T	time bins
p	P	points
k	F	spectrogram kernels or frequency bins
n	N	neurons
r	R	rank of CP decomposition or convRRR
l	L	convolutional kernel index (samples)

Table 3: Glossary of dimensions.

Symbol	Number of dimensions	Description	Convention
a	0	scalar	lower-case, plain
$\mathbf{a}$	1	vector	lower-case, bold
$\mathbf{A}$	2	matrix	upper-case, bold
$\boldsymbol{\mathcal{A}}$	$3+$	tensor	calligraphic, bold

Table 4: Glossary of common matrix notation convention.

Operator	Description
$*$	discrete convolution along the time axis
$\otimes$	outer product
$\odot$	element-wise multiplication
$\mathcal{H}[\cdot]$	Hilbert transform (analytic signal, complex valued)
$\mathcal{F}[\cdot]$	Fourier transform
$\mathcal{F}^{-1}[\cdot]$	inverse Fourier transform
$\lvert \cdot \rvert$	absolute value
$\left\lVert \cdot \right\rVert$	magnitude of a complex value
$\lVert \cdot \rVert_2$	L^2 norm (Euclidean distance)
$\lceil \cdot \rceil$	ceiling function (round up)
$\angle[\cdot]$	angle (phase of complex value)
$\texttt{cross-entropy}(\hat{\mathbf{Y}}, \mathbf{Y})$	log-softmax followed by negative log likelihood.

Table 5: Glossary of operators.

Example	Description
$\mathbf{A}_{i,j} = \boldsymbol{\mathcal{B}}_{i,j,k,m}\ \mathbf{C}_{k,m}$	Inner product contraction/summation over dimensions k and m
$\boldsymbol{\mathcal{D}}_{i,j,k,m} = \mathbf{a}_i\ \mathbf{b}_j\ \mathbf{C}_{k,m}$	Outer product expansion over dimensions i, j, k, and m
$h = \boldsymbol{\mathcal{L}}_{i,j,k}$	Summation contraction over dimensions i, j and k into a scalar
$\mathbf{m}_i = \mathbf{n}_i\ \mathbf{p}_i$	Element-wise product along dimension i
$\boldsymbol{\mathcal{E}}_{i,j,k} = \boldsymbol{\mathcal{F}}_{i,k,m,n}\ \boldsymbol{\mathcal{G}}_{j,k,m}$	Combination of inner product over m, outer products over i, j, element-wise product over k, and summation over n

Table 6: Glossary of Einstein summation notation examples.

Methods References

Baker, Simon and Iain Matthews (Feb. 2004). "Lucas-Kanade 20 Years On: A Unifying Framework". en. In: *International Journal of Computer Vision* 56.3, pp. 221–255. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000011205.11775.fd. URL: http://link.springer.com/10.1023/B:VISI.0000011205.11775.fd (visited on 03/04/2025).

Bradski, G. (2000). "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools*.

Brown, Judith C. (Jan. 1991). "Calculation of a constant Q spectral transform". en. In: *The Journal of the Acoustical Society of America* 89.1, pp. 425–434. ISSN: 0001-4966, 1520-8524. DOI: 10.1121/1.400476. URL: https://pubs.aip.org/jasa/article/89/1/425/678890/Calculation-of-a-constant-Q-spectral (visited on 03/04/2025).

Carroll, J. Douglas and Jih-Jie Chang (1970). "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition". en. In: *Psychometrika*.

Chatterji, S et al. (Oct. 2004). "Multiresolution techniques for the detection of gravitational-wave bursts". en. In: *Classical and Quantum Gravity* 21.20, S1809–S1818. ISSN: 0264-9381, 1361-6382. DOI: 10.1088/0264-9381/21/20/024. URL: https://iopscience.iop.org/article/10.1088/0264-9381/21/20/024 (visited on 03/04/2025).

Couto, Joao et al. (July 2021). "Chronic, cortex-wide imaging of specific cell populations during behavior". en. In: *Nature Protocols* 16.7, pp. 3241–3263. ISSN: 1754-2189, 1750-2799. DOI: 10.1038/s41596-021-00527-z. URL: https://www.nature.com/articles/s41596-021-00527-z (visited on 07/15/2025).

Dosovitskiy, Alexey et al. (June 2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. en. arXiv:2010.11929 [cs]. DOI: 10.48550/arXiv.2010.11929. URL: http://arxiv.org/abs/2010.11929 (visited on 04/01/2026).

Harshman, Richard A (1970). "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis". en. In: *UCLA working papers in phonetics*.

He, Kaiming et al. (Dec. 2021). *Masked Autoencoders Are Scalable Vision Learners*. en. arXiv:2111.06377 [cs]. DOI: 10.48550/arXiv.2111.06377. URL: http://arxiv.org/abs/2111.06377 (visited on 04/01/2026).

Hennig, Jay A. et al. (May 2021). "Learning is shaped by abrupt changes in neural engagement". en. In: *Nature Neuroscience* 24.5, pp. 727–736. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-021-00822-8. URL: https://www.nature.com/articles/s41593-021-00822-8 (visited on 06/13/2025).

Karaev, Nikita, Iurii Makarov, et al. (Oct. 2024). *CoTracker3: Simpler and Better Point Tracking by Pseudo-Labelling Real Videos*. en. arXiv:2410.11831 [cs]. DOI: 10.48550/arXiv.2410.11831. URL: http://arxiv.org/abs/2410.11831 (visited on 04/01/2026).

Karaev, Nikita, Ignacio Rocco, et al. (Oct. 2024). *CoTracker: It is Better to Track Together*. en. arXiv:2307.07635 [cs]. DOI: 10.48550/arXiv.2307.07635. URL: http://arxiv.org/abs/2307.07635 (visited on 04/01/2026).

Kossaifi, Jean et al. (May 2018). *TensorLy: Tensor Learning in Python*. en. arXiv:1610.09555 [cs]. DOI: 10.48550/arXiv.1610.09555. URL: http://arxiv.org/abs/1610.09555 (visited on 06/13/2025).

Lucas, Bruce D and Takeo Kanade (1981). "An Iterative Image Registration Technique with an Application to Stereo Vision". en. In: *Proceedings of the International Joint Conference on Artificial Intelligence* 7th.

Matsumoto, Hideyuki et al. (Oct. 2016). "Midbrain dopamine neurons signal aversion in a reward-context-dependent manner". en. In: *eLife* 5, e17328. ISSN: 2050-084X. DOI: 10.7554/eLife.17328. URL: https://elifesciences.org/articles/17328 (visited on 03/13/2025).

McInnes, Leland, John Healy, and James Melville (Sept. 2020). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. en. arXiv:1802.03426 [stat]. DOI: 10.48550/arXiv.1802.03426. URL: http://arxiv.org/abs/1802.03426 (visited on 06/13/2025).

Mørup, Morten, Lars Kai Hansen, and Sidse M. Arnfred (Apr. 2007). "ERPWAVELAB". en. In: *Journal of Neuroscience Methods* 161.2, pp. 361–368. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2006.11.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S016502700600567X (visited on 03/04/2025).

Musall, Simon, Matthew T. Kaufman, et al. (Oct. 2019). "Single-trial neural dynamics are dominated by richly varied movements". en. In: *Nature Neuroscience* 22.10, pp. 1677–1686. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-019-0502-4. URL: https://www.nature.com/articles/s41593-019-0502-4 (visited on 03/04/2025).

Musall, Simon, Xiaonan R. Sun, et al. (Jan. 2023). "Pyramidal cell types drive functionally distinct cortical activity patterns during decision-making". en. In: *Nature Neuroscience*. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-022-01245-9. URL: https://www.nature.com/articles/s41593-022-01245-9 (visited on 07/15/2025).

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals (Jan. 2019). *Representation Learning with Contrastive Predictive Coding*. en. arXiv:1807.03748 [cs]. DOI: 10.48550/arXiv.1807.03748. URL: http://arxiv.org/abs/1807.03748 (visited on 07/22/2025).

Paatero, Pentti and Unto Tapper (June 1994). "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values". en. In: *Environmetrics* 5.2, pp. 111–126. ISSN: 1180-4009, 1099-095X. DOI: 10.1002/env.3170050203. URL: https://onlinelibrary.wiley.com/doi/10.1002/env.3170050203 (visited on 03/04/2025).

Pachitariu, Marius et al. (June 2016). *Suite2p: beyond 10,000 neurons with standard two-photon microscopy*. en. DOI: 10.1101/061507. URL: http://biorxiv.org/lookup/doi/10.1101/061507 (visited on 06/13/2025).

Schneider, Steffen, Jin Hwa Lee, and Mackenzie Weygandt Mathis (May 2023). "Learnable latent embeddings for joint behavioural and neural analysis". en. In: *Nature* 617.7960, pp. 360–368. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06031-6. URL: https://www.nature.com/articles/s41586-023-06031-6 (visited on 03/03/2025).

Syeda, Atika et al. (Jan. 2024). "Facemap: a framework for modeling neural activity based on orofacial tracking". en. In: *Nature Neuroscience* 27.1, pp. 187–195. ISSN: 1097-6256, 1546-1726. DOI: 10.1038/s41593-023-01490-6. URL: https://www.nature.com/articles/s41593-023-01490-6 (visited on 03/04/2025).

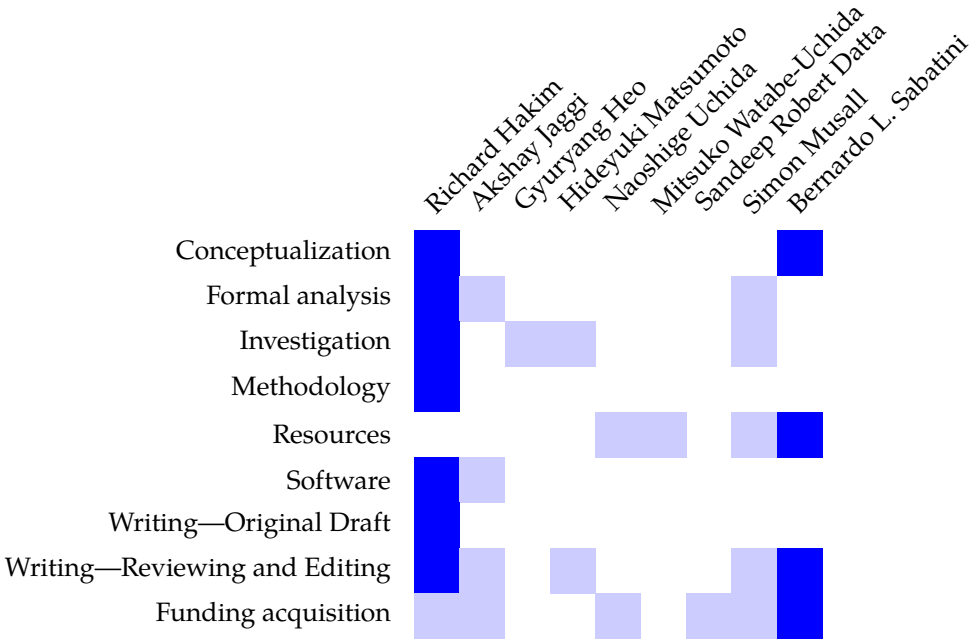
Vinck, Martin, Francesco Paolo Battaglia, et al. (Aug. 2012). "Improved measures of phase-coupling between spikes and the Local Field Potential". en. In: *Journal of Computational Neuroscience* 33.1, pp. 53–75. ISSN: 0929-5313, 1573-6873. DOI: 10.1007/s10827-011-0374-4. URL: http://link.springer.com/10.1007/s10827-011-0374-4 (visited on 06/13/2025).

Vinck, Martin, Marijn Van Wingerden, et al. (May 2010). "The pairwise phase consistency: A bias-free measure of rhythmic neuronal synchronization". en. In: *NeuroImage* 51.1, pp. 112–122. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2010.01.073. URL: https://linkinghub.elsevier.com/retrieve/pii/S1053811910000959 (visited on 06/13/2025).

Wang, Yanchen et al. (July 2025). *Animal behavioral analysis and neural encoding with transformer-based self-supervised pretraining*. en. arXiv:2507.09513 [q-bio]. DOI: 10.48550/arXiv.2507.09513. URL: http://arxiv.org/abs/2507.09513 (visited on 04/01/2026).

Williams, Alex H. et al. (June 2018). "Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis". en. In: *Neuron* 98.6, 1099–1115.e8. ISSN: 08966273. DOI: 10.1016/j.neuron.2018.05.015. URL: https://linkinghub.elsevier.com/retrieve/pii/S0896627318303878 (visited on 03/04/2025).

6 Contributions



7 Acknowledgements

We thank: Alex Williams and Scott Linderman for helpful discussions; and Christopher Harvey, John Assad, and Mark Harnett for guidance.

8 Funding attribution

RH is supported by the Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University, and NSF Graduate Research Fellowship 1000244863.

AJ is supported by award Number T32GM007753 and T32GM144273 from the National Institute of General Medical Sciences. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of General Medical Sciences or the National Institutes of Health.

BLS is supported by the Howard Hughes Medical Institute, the Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University, and NIH grants R37NS046579 (Action and interaction of ionotropic and metabotropic neurotransmission), and R35NS137336 (Action and Interaction of Neurotransmitters and Neuromodulators).

BLS, SRD, and NU are supported by NIH grant U19NS113201 (Team DOPE: Towards a unified framework for dopamine signaling and basal ganglia function).

9 Disclosures

The authors declare no competing interests.