# Spectral envelopes of rhythmic facial movements predict intention and motor cortical representations

Richard Hakim[1,2], Gyuryang Heo[1,2], Akshay Jaggi[3], 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, Universityf of Bonn, Bonn, Germany*

September 15, 2025

## 1 Abstract

Animals, including humans, use coordinated facial movements to sample the environment, ingest nutrients, and communicate. To study these behaviors and the neural signals that underlie them, we introduce face-rhythm, a tool for quantitatively tracking, extracting, and interpreting facial movements. The approach utilizes markerless point tracking, spectral analysis, and tensor component analysis to extract demixed components of behavior from videos of facial movements. Face-rhythm is fully unsupervised and allows for the discovery of uninstructed behaviors; when applied to videos of facial behavior, face-rhythm identifies interpretable behaviors such as whisking, sniffing, and snout movements. Analysis of videos of mice in various behavioral conditions, including a classical conditioning protocol, a brain-machine interface (BMI) task, and natural behaviors outside of a task structure, revealed robust signatures of uninstructed facial movements in all regimes. The expression of these facial movements predicted internal belief states during classical conditioning and was correlated with instructed neural activity when the BMI was activated. Furthermore, facial behaviors identified by face-rhythm were highly represented in face-associated areas of primary motor cortex (M1). We found that M1 neural activity encodes a mixed representation of both the phase of facial movements as well as the phase-invariant spectral envelope of movement patterns, with higher-frequency facial movements being more likely to be represented as phase-invariant. Our results demonstrate that face-rhythm provides a novel and flexible approach for decomposing continuous face movements into natural behavioral motifs that are closely linked to neural activity patterns.

## 2 Introduction

The face is a special sensorimotor interface between the brain and the environment. Structurally, sensory organs and dozens of muscles are clustered in the face and possess higher acuity and dexterity than those in other regions. Functionally, facial motor control takes on a variety of ethological roles and, as a result, embodies a high-dimensional projection of internal state such that information about what an animal has experienced, is experiencing, or will do is often represented in how the face looks and moves. Scientific interest in facial behavior spans millennia (Galen 1968), and a variety of methods for analyzing facial behavior have been developed to better understand motor pattern generation and infer an animal's internal state. Currently, although several of the neural circuits and transformations involved in the production of specific facial behaviors have been well described, fundamental principles of how neural activity from

the motor cortex is represented by and transformed into facial behavior are still being uncovered.

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 technology and machine learning have resulted in the development of more performant, objective, and less invasive methods. In recent years, machine vision and unsupervised learning algorithms have taken leading roles in extracting useful representations of facial pose and movement and decomposing them into meaningful 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), a common bottleneck for these tools is that they often require extensive manual curation to track movements from 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). While *pose* description methods apply human interpretable labels to static poses, *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) and B-SOiD (Hsu and Yttri 2021), and others (Batty et al. 2019; Bohnslav et al. 2021; Luxem et al. 2022; Mackevicius et al. 2019), which have been applied to whole body poses. These works highlight the dominant trend in the systems neuroscience field towards more expressive machine learning methods that consider higher-dimensional movement patterns. However, one limitation of behavioral segmentation approaches is the assumption that an animal can only be producing one discrete behavior at a time. This assumption is not always appropriate, and researchers often have an interest in how multiple behaviors are orchestrated together.

Whole-face video recordings can also be used to associate facial movements with neural data. A method utilizing principal component analysis 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). Later, advancements made in markerless point tracking of rodent faces demonstrated that improved tracking and 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). In essence, FR aims to discover and demix facial behaviors that overlap in space, time, and spectral patterning to reveal latent "rhythms".

**BS suggested alternative to these last two paragraphs:** 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, lapping, sniffing, and more subtle motifs) or associated with cognitive task performance variables. Consistent with achieving this goal, the application of FR to facial video data

acquired during operant and classical conditioning 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. In fact, 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 $\sim 0.5$ Hz) are more likely to be represented in the cortex as a phase-invariant spectral envelope. In summary, FR provides a method that naturally decomposes spontaneous and operant facial behavior into interpretable movements that are represented in the activity of orofacial motor cortex.

The code to run the face-rhythm pipeline was refined for general use and is made 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 which points on the face move, a spectral factor that captures the dominant frequencies of the points participating in the movement component, and a temporal factor that describes the movement magnitude of the component for each video frame (**Figure 1B-C**).

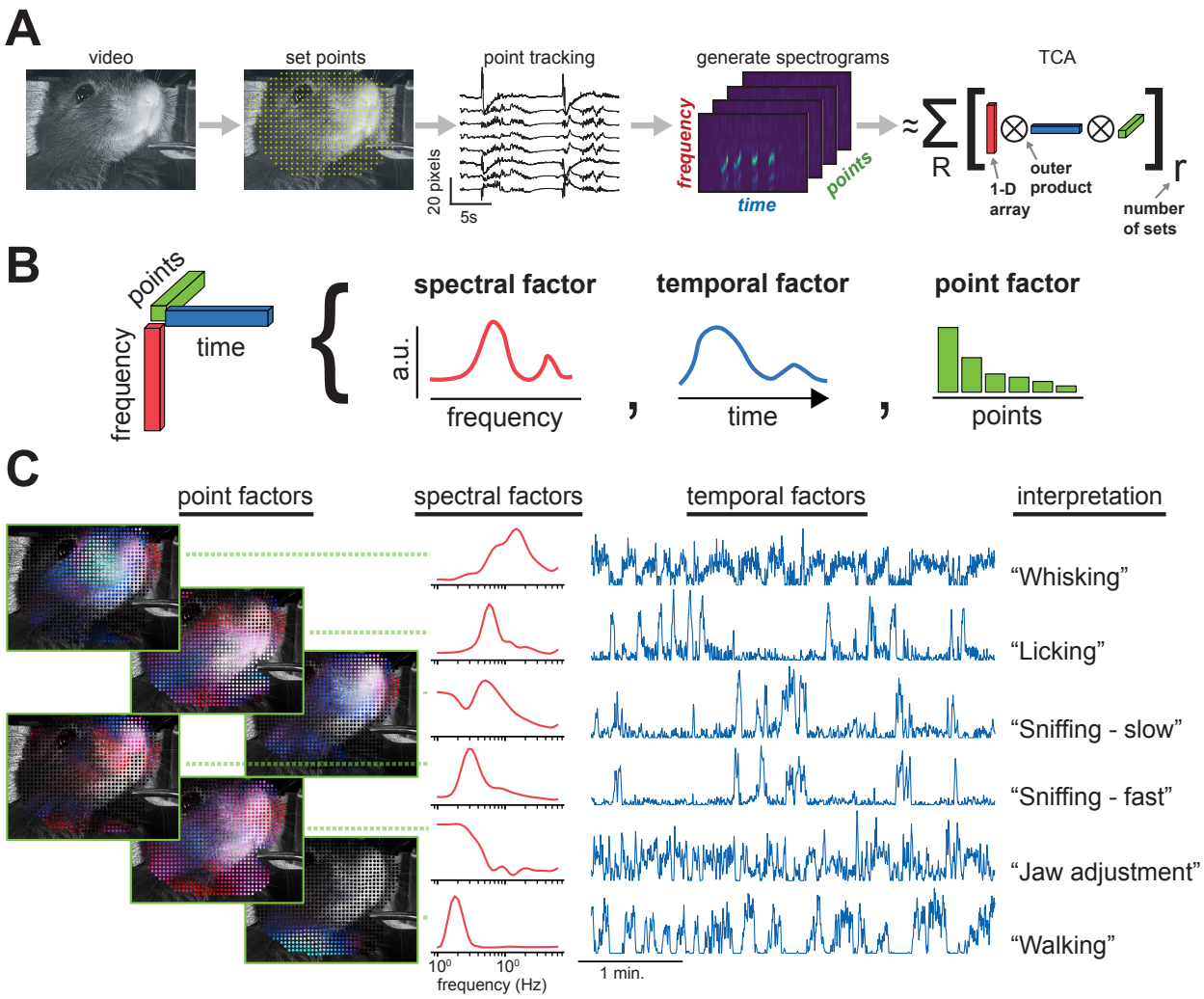


**Figure 1.** A) Diagram of face-rhythm pipeline. Video of a face is used as input. Point tracking is performed using a modified Lucas-Kanade algorithm. Spectrograms are generated for each point and each direction ('x' and 'y') using a variable-Q transform. Non-negative tensor component analysis is performed using non-negative PARAFAC. Output components are sets of vectors where each vector describes a dimension and the number of vector sets is determined by the rank of the decomposition. B) Representative examples of face-rhythm components. Colors match the dimensions in (A). 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 describing the amplitude of a component at that frequency. Middle-right: temporal factors describing the amplitude at each time point. Right: interpretation of what each component represents.

The specific algorithms that are used in the pipeline were selected to be high-performing unsupervised methods that require low 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 novel implementation of the variable-Q transform (Brown 1991; Chatterji et al. 2004), and the decomposition is performed using non-negative tensor component analysis (TCA, also known as 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 rodents, this is at least 30 Hz and ideally above 60 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 x num_points), (frequencies),* and *(binned_time)*. The size of the set, '$R$', is determined by the user and typically 6-12. Methods for rank selection vary depending on the use case (Williams et al. 2018) (see **Methods 5.1.4**). 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.

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 1B**) (**Supplementary Movie 1-2**). This approach does not require training of neural networks, maintains high interpretability, and can be used to analyze movements of individual mice during single video sessions.

We note that 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 running 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.

## 3.2 Comparison with motion energy principal component analysis

Our observation that FR components are well-aligned with natural facial behaviors motivated us to compare how well FR temporal factors could also explain neural activity. We therefore used a previously 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**); in addition, we used widefield imaging to simultaneously record cortex-wide neural activity in mice performing a visual discrimination task (**Figure 2 B left**, **Supplementary Movie 3**). The task structure included a visual stimulus, a subsequent delay, and a response epoch when mice could respond to the stimulus by licking one of two water spouts. The posterior visual cortical regions responded strongest to the visual stimulus, whereas medial regions responded persistently during the delay period, and frontal regions were most active during the response period (**Figure 2B right**).

To link cortical activity to the behavioral video data, we used a Ridge regression model that explained neural activity in each pixel of the widefield imaging through a combination of different behavioral inputs, such as FR components or point tracking data (**Figure 2C**). Hierarchical regression showed that FR temporal factors encode a low-rank representation that was 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 was consistently high across much of the cortex, with particularly strong performance in anterior regions. This shows that behavioral video data provided a robust basis for predicting cortical activity (**Figure 2E**). In a direct comparison of performance, we found that the EVR obtained using just 10 FR components ($0.28 \pm 0.05$ *(std)*) substantially exceeded that of 10 ME PCA components ($0.14 \pm 0.08$ *(std)*) ($p = 0.0078$, Wilcoxon signed-rank test) and matched the performance achieved with 200 ME PCA components ($0.28 \pm 0.07$ *(std)*), demonstrating that FR components are better predictors of cortical dynamics.

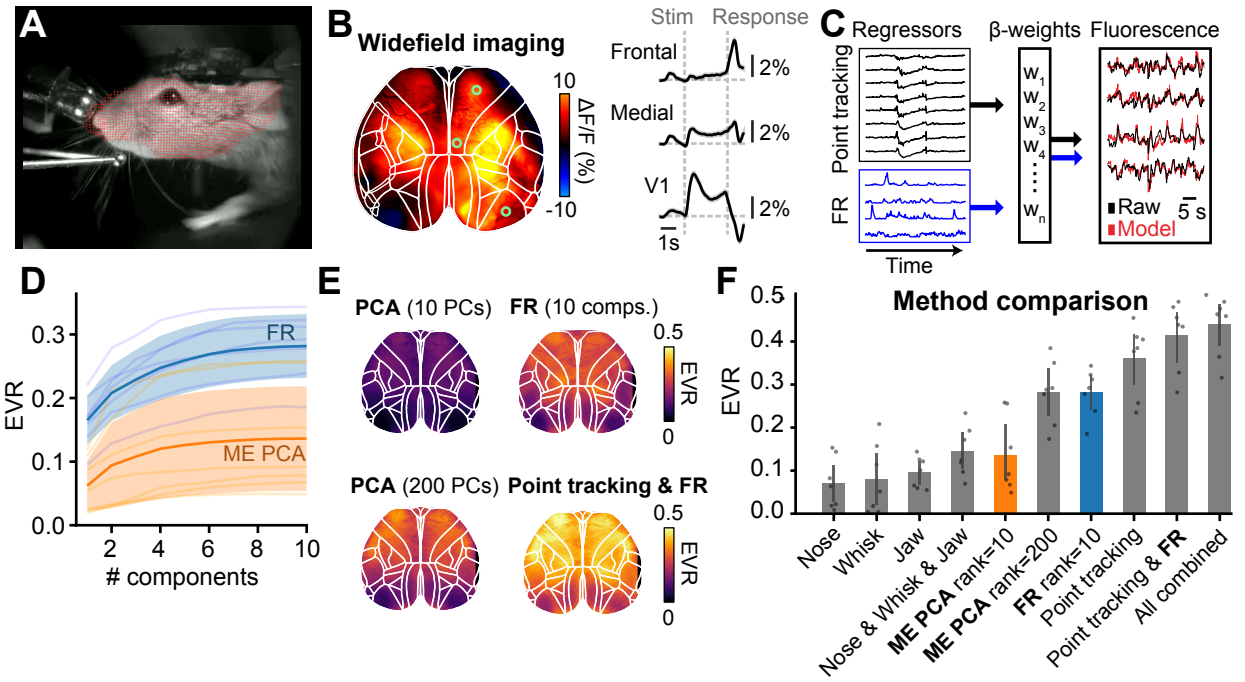


**Figure 2.** A) Left: Example image of a task-performing mouse on a running wheel with tracked points in red. B) Simultaneous widefield imaging was used to record cortex-wide neural activity. The left image shows a single example frame; the traces on the right show trial-averaged activity from the same session in different brain regions. The region locations (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; 4 animals, 2 sessions per animal. E) Explained variance ratio averaged across animals (N=4) 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.

A more detailed comparison of the EVR for specific behavioral variables, such as whisking or nose movements, further confirmed the superior predictive power of FR components (**Figure 2F**). A model based on 10 FR components was significantly more predictive than an alternative model using a combination of whisker, nose, and jaw movements ($0.14 \pm 0.05$ *(std)*) ($p = 0.0078$, Wilcoxon signed-rank test). Notably, appending the rank-10 FR components as additional predictors to the high-dimensional point tracking data further improved the EVR from $0.36 \pm 0.08$ to $0.42 \pm 0.7$ *(std)*, suggesting that the cortex maintains a separable representation of the spectral envelope of facial movement. However, further appending all signals together only modestly improved the EVR ($0.42 \pm 0.7$ to $0.44 \pm 0.07$ *(std)*), 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.

### 3.3 Conditioning of facial behaviors during reward-guided tasks

We examined uninstructed facial behaviors in the context of classical conditioning (**Figure 3**) using historical video data of thirsty mice undergoing Pavlovian association of odors with delivery of water or air puffs (Matsumoto et al., 2016)(Matsumoto et al. 2016) (see **Methods** 5.3). Each mouse was head-restrained

and presented one of three odorant conditioned stimuli (CS) that had been associated with the delivery of an unconditioned stimulus (US). The CS duration and the delay from the end of CS to US were each 1 second. The three CS classes are referred to as CS+ (associated with water reward), CS- (associated with air puff), or CS0 (CS only) (**Figure 3A**). We extracted FR components (**Figure 3B-C**) and attempted to predict the trial type using the amplitude of temporal factors during the CS epoch (i.e., before US delivery) using multinomial logistic multilinear regression. As above, the intermediate spectrograms were computed using an asymmetric spectrogram kernel with no information from future time points (see **Methods** 5.12.2). The temporal factor amplitudes of specific FR-identified components of facial movements were clearly modulated starting at the time of CS presentation and in a CS-dependent manner. The interpretable nature of FR components revealed that the animal began jaw and licking movements in response to CS+ presentations, reduced respiration in response to CS0 presentations, and twitched its ears during CS- presentations (**Figure 3D-E**). Using the amplitude of FR temporal factors during a period spanning the US presentation and ending 1 second before the CS, we were able to predict the trial class with high specificity and sensitivity (**Figure 3F**). This example illustrates the utility of FR as a method for screening, discovering, and interpreting animals' behavior during tasks. In particular, this includes uninstructed movements that are shaped by task contingencies or conditioned stimuli.

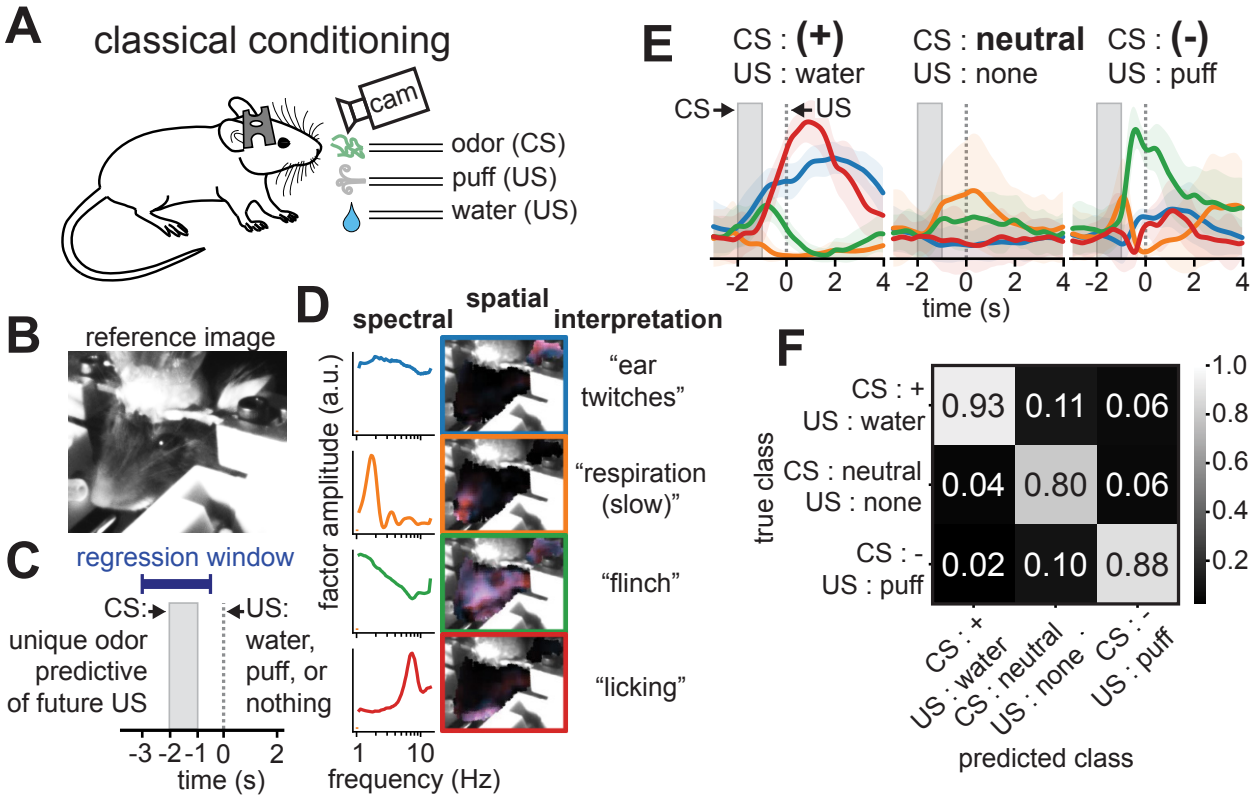


**Figure 3.** A) Diagram of experiment. Video of the face was recorded during classical conditioning with a 3 odors (CS) paired to either positive (water), neural (nothing), or aversive (air puff) stimuli (US). B) Field-of-view of camera. C) Diagram for how classification analysis is performed. Face-rhythm temporal factor dynamics during the time window of the CS are used to predict the identity of the CS/US class. Logistic tensor regression is used; see methods for details on model. D) Representative examples of extracted face-rhythm components. Left panels: frequency / spectral factors. Middle panels: Spatial / positional factors. Blue represents horizontal ('x') amplitude, Red represents vertical ('y') amplitude. Right panels: Experimenter's interpretation of what each component represents. E) Representative examples of temporal factor dynamics surrounding a CS presentation. Colors match components in (I). Error bars are standard deviation. F) Confusion matrix for classification.

We separately examined the ability of FR to detect facial behaviors in a mouse performing a reward-reinforcement task by assessing the utility of these extracted components in predicting task-related variables (**Figure 4**, see **Methods 5.9**). We performed this analysis in the context of a simple brain-machine interfacing (BMI) experiment (**Figure 4A**). An animal was tasked with moving the activity of a population of neurons along a particular direction in neural state space in order to receive a reward. The activities of $\sim$ 1000 Layer 2/3 neurons expressing GCaMP in the facial motor cortex (1.6 anterior, 1.9 lateral to bregma (Mayrhofer et al. 2019; Tamura et al. 2025)) were monitored via 2-photon imaging at 30 Hz of calcium transients as reported by changes in fluorescence. Population activity was analyzed in real-time using custom code for image stabilization and decoding. The BMI decoder output was defined using a linear combination of neuron weights selected using sparse principal component analysis of activity recorded on a baseline recording day. Instantaneous neural activity was projected onto the decoder axis and processed into a one-dimensional 'cursor' that was fed back to the animal as both the frequency of a tone and the location of an illuminated spot along an LED strip. The decoder amplitude threshold for reward delivery was determined by running simulations on the baseline recording data and optimizing the reward threshold to a value for which the animal would have received $\sim$ 1.25 rewards per minute. On experimental days with the BMI task engaged, when the cursor reached the threshold value, mice were rewarded with soy milk, and the feedback tone was turned off until the animal reduced the cursor back to baseline to start a new trial (**Figure 4C**).

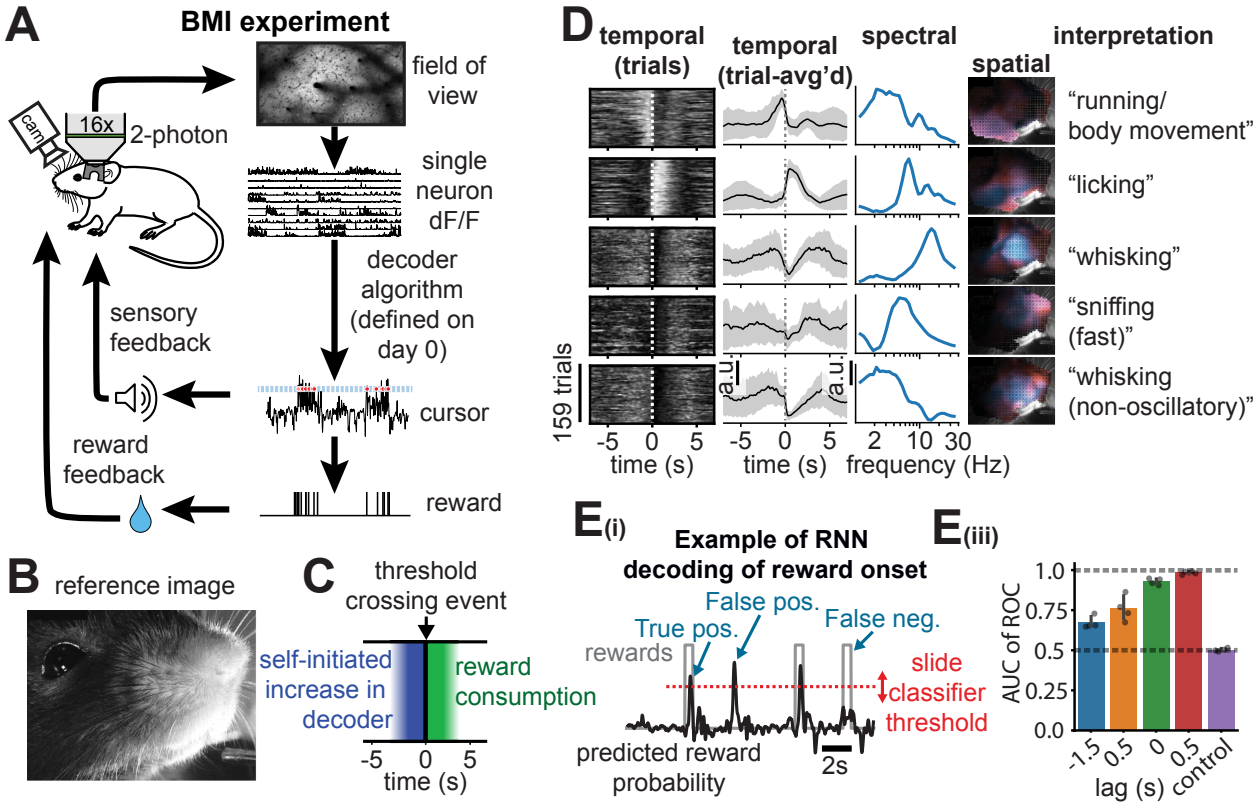


**Figure 4.** A) Diagram of experiment. Mouse is performing a brain-machine interfacing experiment (see methods). Animal is tasked with increasing the activity of a set of neurons that are based on spontaneously evoked activity during a baseline recording day ('day 0'); rewards are given when an animal 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 in-

dividual components from a face-rhythm extraction using an asymmetric (left-side only) sliding window (see methods). Left panels: rasters of temporal factors aligned to threshold-crossing events. Left-middle panels: averaged temporal factors; error bars are standard-deviation. Middle panels: frequency / spectral factors. Right-middle panels: Spatial / positional factors. Blue represents horizontal ('x') amplitude, Red represents vertical ('y') amplitude. Right panels: Experimenter's interpretation of what each component represents. E) Decoding analysis. Ei) Diagram for how decoding performance is defined. An RNN model is trained to output a prediction of the instantaneous probability that the animal is currently achieving a reward. Inputs to model are the instantaneous face-rhythm temporal factors. To asses 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 for models trained on input time series data that is temporally shifted / lagged relative to the output time series. Negative lags describe models that are predicting future events. Shuffles are randomly temporally shifted.

Continuous video of the face was acquired during the task and analyzed post-hoc to extract FR components (**Figure 4D**). Temporal alignment of each FR temporal factor to reward events revealed clear changes in the expression of specific components before and around the time of the reward threshold crossing. The changes in the amplitude of individual components were associated with the approach to threshold, onset of threshold, reward consumption, and the inter-trial interval (**Figure 4D**). Motivated by the clear structure of the FR temporal factors during the task, we quantified how well FR factors could predict instantaneous or future reward events. To maximize performance, a recurrent neural network (RNN) regression model was trained to predict threshold-crossing events using FR temporal factors. In this case, the intermediate spectrograms in the FR pipeline were computed using asymmetric spectrogram kernels (left-side only window) with no information about future time points (see **Methods** 5.12.2) (**Figure 4E**). The model was able to predict threshold-crossing events as they happened with high accuracy and could predict events in the near future (-1.5 to -0.5s) above chance (-1.5s: p=0.0016, -0.5s: p=0.0031, 0s: p=2.9e-5, +0.5s: p=8.7e-6, paired t-test against temporally shifted control). Thus, despite the task being operantly conditioned only on the neural activity, mice performed uninstructed facial movements that were predictive of the instructed neural patterns and reward expectation. Given that this operant conditioning task is goal-oriented and does not require any physical movements, we conclude that FR is capable of discovering temporal factors that carry a representation of goal-oriented intention from uninstructed movements.

## 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/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 M1 neural activity patterns and patterns of facial movements. We therefore examined if 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 a family of three regression models (**Figure 5C**). The three models varied in whether phase information from the behavior data could be used in predicting the neural data. We compared the quality of the fits to quantify the degree to which phase information is encoded in M1 neural activity.

These models all contain a convolution step in which the input is convolved with a learned convolutional kernel that is optimized by adjusting its shape 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. Such an operation approximates a wavelet transform via the convolution theorem. Our first model, referred to as the basic convolution model, simply

applies a standard real-valued convolution to map inputs to outputs. Importantly, 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. Our second model, called the 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, we define a third model, called the Hilbert-spectral convolution model, which is similar to the spectral convolution model but in which 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 computing the imaginary part as the Hilbert transform of the real part. 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.

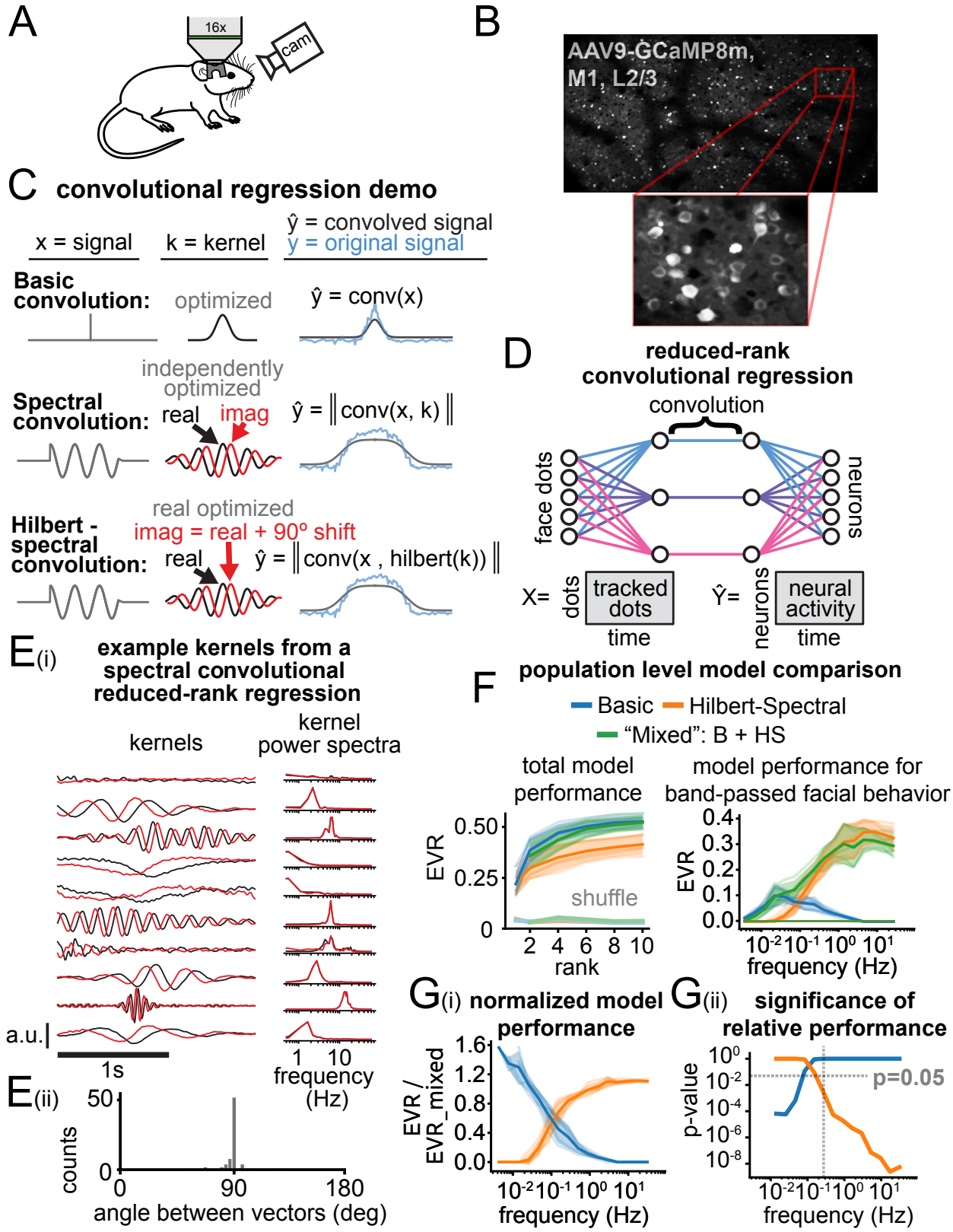


**Figure 5.** A) Diagram of experimental set up. Mouse is not performing a task, but is in an environment with a linear treadmill, open-loop visual and auditory stimulation, and uncued infrequent soy milk

droplets (see methods). B) Example field-of-view from 2-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). 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). 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. 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. Gil) EVR ratios of the pure models with the hybrid model.

Models are parameterized with: the encoding matrix, the kernel for each "rank", and the 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 GCaMP8m expressing neurons in layer 2/3 of M1 (see **Methods 5.5**). A reduced-rank regression architecture was used to wrap each type of convolution model (**Figure 5D**), which constrains the number of convolutional kernels that are created 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. We note that this architecture bears some resemblance to 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, resulted in the discovery of complex convolutional kernels that were approximately 90 degree 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 relative predictive performance between 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 explained more neural variance than the spectral model, and a hybrid model in which half of the ranks were basic and half were Hilbert-spectral performed 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 being explained between the basic and spectral models (EVR of 'basic' vs. 'Hilbert-spectral' at rank=10, p=0.0015).

Motivated by observations in sensory systems (Bendor and Wang 2007; Ewert, Vahle-Hinz, and Engel 2008), we tested whether the frequency of a movement was predictive of 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 showed high power at low frequencies (**Figure 5 supplement 1 E**) and exhibited a power-law decay ($\beta = -1.7, p < 1e-4$). We hypothesized that facial behaviors occurring at higher frequencies might be more likely to be represented in M1 activity by the Hilbert-spectral (phase-invariant) model, and so we refit the models using facial movements that were filtered into narrow frequency bands. We found that the Hilbert-spectral model significantly outperformed the basic model for facial movements in frequencies greater than $\sim 0.5$ Hz (**Figure 5G**) (stats). We conclude that higher frequency facial behavior is better represented in neural activity (up to the limits imposed by the 30 Hz 2-photon microscopy imaging rate) via a spectral transformation.

The tuning of individual neurons to either phase-variant or invariant facial movements was also examined. Many neurons show visually identifiable phase-locking with facial movements, and others show clear correlation with the envelope of oscillatory movements (**Figure 5 supplement B-G**). To quantify these relationships, we generated pairs of behavioral signals that correspond to roughly one behavior where one signal is phase-variant and the other is phase-invariant. This was accomplished by filtering raw point tracking data through a single FR component's spatial and spectral factors, resulting in the phase-variant signal, and then taking the magnitude of the Hilbert transform of that signal, resulting in the phase-invariant signal. Each neuron's activity trace was then compared to each FR component signal pair; the pairwise phase consistency was computed using the phase-variant signal, and the correlation was computed using the phase-invariant signal (**Figure 5 supplement G-H**). We found that most neurons show a significant correlation to the phase-invariant signal, and a smaller fraction are significantly phase-locked to the phase-variant signal. Notably, some neurons showed significant phase-locking with behavioral signals with peak frequencies near the Nyquist limit, roughly 15 Hz, for the neural data recordings. We also found that more neurons show phase-invariant tuning at higher movement frequencies (stats, relative to time-shifted control).

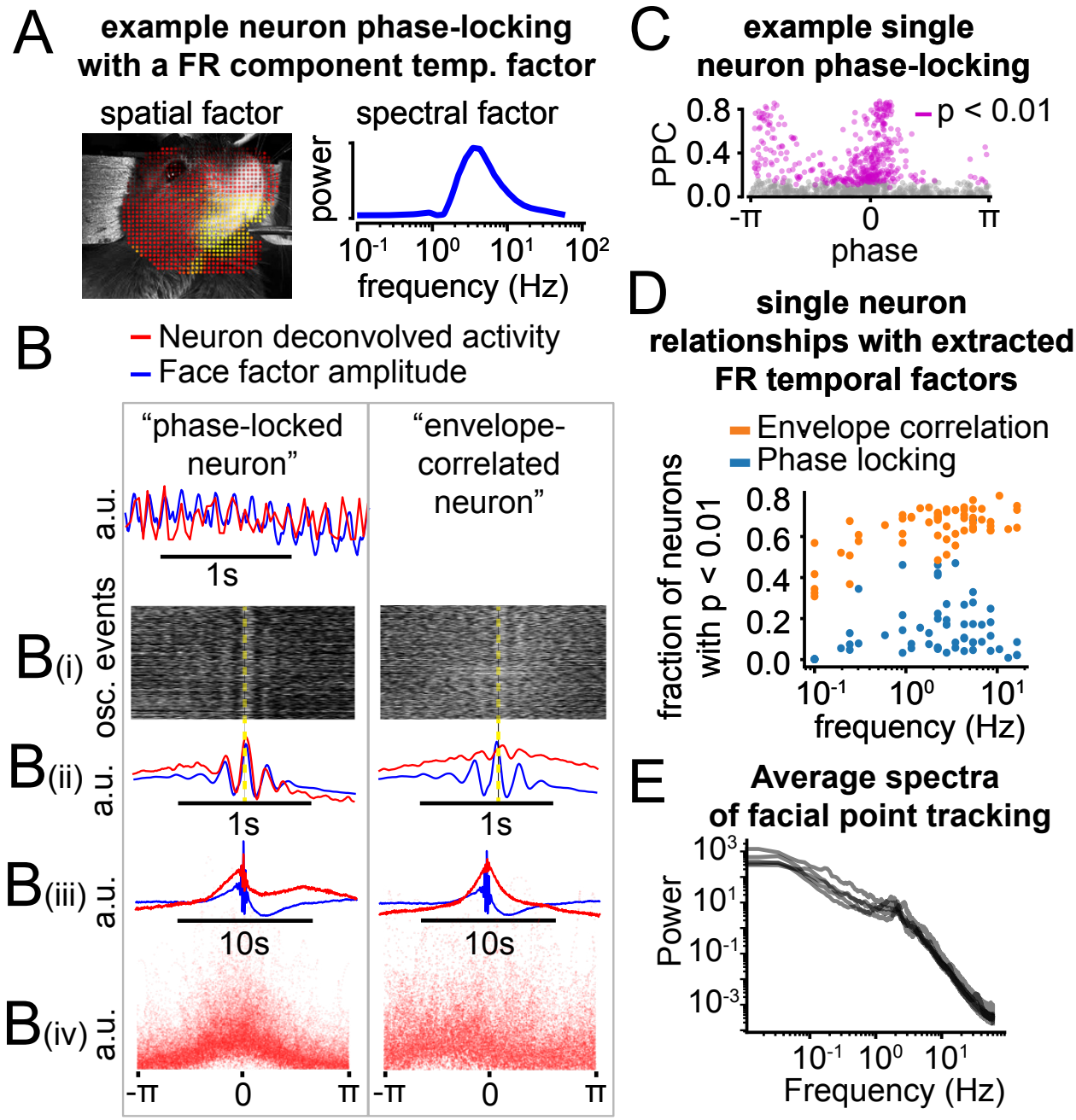


**Figure 5 supplement 1.** A) Example spatial/point factor and spectral factor from the same single component of an FR decomposition. B) Examples of how two neurons are associated with the temporal factor of the behavioral component in (A). One neuron is phase-locked to the instantaneous movement (left) 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). 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. 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

The anatomy of the facial motor system as well as the utility of spectral transformations in explaining neural variance led us to hypothesize that FR might be performing a set of transformations that 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 if there exists a one-to-one mapping between FR temporal factors and the temporal factors extracted from applying non-negative tensor decomposition to 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 2-photon calcium imaging and simultaneously recorded facial behavior using a camera. Mice were free to run on a linear treadmill and were given soy milk at random infrequent intervals. We found striking pairwise similarities in the temporal factors extracted from the two domains, despite the complete independence of the neural and facial movement analyses (**Figure 6B**). Furthermore, we 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 towards higher frequencies in the similarity between face-rhythm and neural components. Overall, we find that a large fraction of the matched factor pairs are highly correlated (**Figure 6D**). We conclude that the dominant modes of the recorded neural activity resemble the dominant modes of spectrally transformed facial movements.

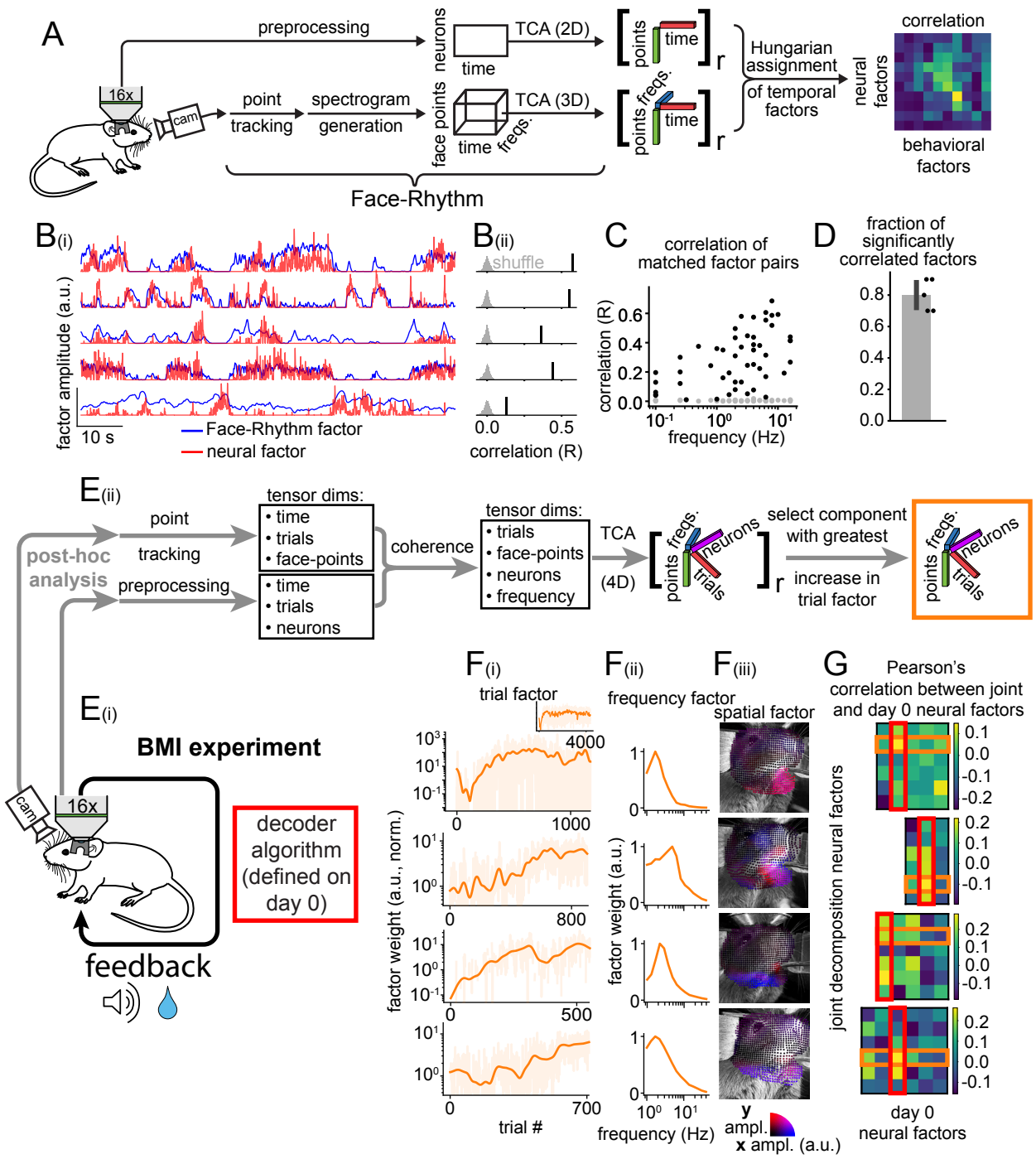


**Figure 6.** A) Diagram of analysis pipeline for comparing independently extracted tensor components from neural data (top) and behavior / face data (bottom). Mouse is not performing a task, but is in an environment with a linear treadmill, open-loop visual and auditory stimulation, and uncued infrequent soy milk droplets (see methods). Neural data is collected from M1 and is preprocessed and 2D non-negative tensor component analysis 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 temporal factors and showing visual correspondence.

Bii) Pearson's 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). E) Diagram of experiment and analysis pipeline for brain-machine interfacing task and post-hoc joint factorization analysis (see methods). Neural data is recorded in M1. Ei) BMI decoder is defined on "day 0" by recording the animal in an open-loop environment and then decomposing the neural data to generate a set of day 0 neural factors. One day 0 factor is selected to be the decoder / cursor dimension for the BMI (red box). During BMI experiment, animal receives real-time feedback of the current magnitude of its neural activity along the decoder dimension in the form of visual and auditory cues, and if activity reaches a threshold then the a soy milk reward is given. Eii) Data from the same animal performing the same task with the same decoder is analyzed by preprocessing the neural and behavioral / facial data similar to (A), but prior to performing TCA the coherence is calculated between all pairs of time series between the two datasets. The component with the largest maximal increase in the amplitude is then selected for illustration purposes (orange box). F) Illustrative examples for joint factor analysis from 4 mice. Each factor dimension from the single selected component is shown. Fi) Trial factor showing increase in amplitude of component expression over the course of adaptation to the BMI. Fii) Frequency factor. Fiii) Spatial factor. Blue shows horizontal ('x') amplitude and red shows vertical ('y') amplitude. Fiv) Pearson's correlation between all extracted joint factorization neural factors and all day 0 neural factors. Red box shows selected day 0 component (arbitrarily selected, see methods) and orange box shows selected joint component (selected based on increase in trial factor).

The above analysis indicates that there is a natural similarity in the latent spaces of FR transformed facial behavior and neural activity. We therefore explored the development of an approach to jointly decompose the shared space between neural activity and behavior in order to identify joint components representing correlated neural-behavioral motifs. Inspired by canonical correlation analysis, we sought to discover low dimensional representations of *coherence* between two datasets. We define a joint decomposition by first combining the signals from the two spaces into one, then performing dimensionality reduction. Here, we combine neural activity signals and point tracking signals by calculating the coherence between all neurons and all face points (**Figure 6E**). The resulting tensor is then decomposed using non-negative tensor factorization, resulting in a set of components with spectral (frequencies), trial, spatial (points), and neuron factors. These components describe distinct neural-behavioral actions, and in the context of a BMI experiment, they show evolution over the course of a multi-day experiment. As an illustrative example, we sought to reveal the joint neural-behavioral strategy employed by 4 mice during a multi-day BMI experiment. We found that, in 3 out of 4 mice, the component that showed the greatest increase over the course of the experiment (labeled with orange boxes and lines) had a neural factor that best matched the neural factor that was the decoder for the BMI task. We interpret the discovered behaviors in these animals to look like preparatory licking and running based on the frequency and spatial components. This component had been independently selected on a baseline recording day prior to starting the BMI task (**Figure 6F-G**). We conclude that the neural-behavioral motif that saw the greatest increase in expression over time was the one most closely aligned with the neural decoder axis.

# 4 Discussion

This study introduces face-rhythm, a novel unsupervised analysis pipeline that leverages markerless point tracking, spectral analysis, and tensor component analysis to extract interpretable facial behaviors. The output components of face-rhythm are predictive of various internal states and their time courses exhibit a striking similarity to those of demixed neural activity components. In addition, this work introduces several new computational methods, reveals a strong phase-invariant spectral transformation between higher frequency (>0.5 Hz) facial movements and neural activity in the primary motor cortex, and provides a novel method for discovering jointly defined neuro-behavioral motifs. In addition, this work highlights spectral transformations as a powerful and natural featurization for relating ongoing facial

behavior to motor cortical activity.

## 4.1 Translational applications

We designed FR for the analysis of facial movements in the mouse as a research tool. However, the tools and findings presented in this study likely have utility in clinical and industrial facial analysis applications. The approach described here offers the ability to discover perceptually subtle facial behavior deficits, define robust early diagnostic biomarkers, or define more reproducible endpoints for therapeutics development. In addition, approaches for discovering and monitoring an individual's unique motor dysfunction can allow for objective evaluation of responses to interventions. Facial behaviors may also be predictive of future epileptic episodes, or other episodic conditions, such that their detection could allow for early intervention. Conditions in which facial behaviors are perturbed also stand to benefit from the identification of improved facial biomarkers. These include conditions like neurodegenerative disorders with motor deficits like Parkinson's disease; progressive supranuclear palsy; Bell's palsy, and multiple system atrophy; neuromuscular disorders like bulbar-involving myasthenia gravis; muscular dystrophies, and myopathies; and movement disorders like essential tremor, dystonia, and tic disorder. Adapting FR for use in humans and other subjects is a future direction with promise for use cases that involve subtle or uncharacterized movements and data constraints.

## 4.2 Comparison to other methods

The field of facial and whole body behavior analysis has seen significant improvement and adoption in recent years. In particular, 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 data that is 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 learned biases. Instead, the standard FR pipeline provides a principled approach to behavioral decomposition and is appropriate to 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.

## 4.3 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 Lucas-Kanade-based point tracking could be substituted out with outputs from deep learning-based pose estimation tools like DeepLabCut, SLEAP, or FaceMap, allowing researchers to feed pre-existing keypoint data directly into FR's spectral analysis stage. Conversely, the final TCA-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.

## 4.4 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 sensitive and phase invariant representations, reminiscent of representations in sensory systems like the auditory and somatosensory cortices (Peterson and Heil 2019; Bendor and Wang 2007; Ewert, Vahle-Hinz, and Engel 2008). Practically, closed loop brain machine interfaces and neural decoder designs stand to improve 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, reduce uncertainty in existing theories for how high frequency rhythmic motor behavior is produced. In particular, our work adds evidence that motor cortical regions broadly propagate phase-invariant movement signals to lower central pattern generators, where absolute amplitude is converted into oscillatory amplitude.

The dominance of spectral-envelope representations of higher frequency movements in cortex impacts our understanding of coding efficiency as well as the architecture of the motor system. Spectral envelope encoding has more modest bandwidth and bit-rate requirements for encoding due to a significantly lower frequency content compared to time-domain/phase-variant encoding. These signal properties are advantageous for maintaining and propagating representations within noisy low-rank dynamical systems, which bear resemblance to the motor cortex (Mastrogiuseppe and Ostojic 2018; Omrani et al. 2017; Laughlin, De Ruyter Van Steveninck, and Anderson 1998). Second, modularity and compartmentalization in rhythmic motor patterning, likely via the utilization of discrete central pattern generators as reservoir networks with low plasticity, provide an efficient means of offloading specialized computation to specialized anatomical structures (Kaku et al. 2025). Third, if the primary output of motor cortical regions for higher frequency behaviors is a phase-invariant spectral envelope sent to CPGs, then this predicts that dextrous control of movement phase is either nonexistent (and possibly not ethologically critical) or there exists a secondary descending signal containing an explicit representation of instantaneous phase that interacts with brainstem CPGs.

## 4.5 Analysis of conditioned uninstructed movements

Our observation of subtle behaviors that are robustly associated with, and predictive of, brain-machine interface performance underscores the importance of careful behavioral tracking during BMI tasks. In addition, it suggests that micromovements 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. 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 discussed in Figure 4. 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). Taken together, these observations emphasize the need to monitor subtle uninstructed movements during BMI experiments.

## 4.6 Computational tools

Several computational approaches were developed here to improve on existing methods. A novel implementation of the Lucas-Kanade algorithm was introduced that includes a mesh-physics constraint that reduces noise and drift. An exact formulation of the variable-Q transform was introduced with improve-

ments to filter design flexibility and accuracy. Several novel regression models were also introduced that integrate the concept of spectral convolution, providing an inductive bias for problems that include a spectral transformation between inputs and outputs. Finally, a novel approach to cross decomposition was introduced by combining coherence and tensor decompositions. The code that implements these methods is made available here: https://github.com/RichieHakim/face-rhythm.

## 4.7 Conclusion

Face-rhythm (FR) is an unsupervised, data-efficient method of decomposing facial movements to uncover behavioral components. The standard FR pipeline extracts interpretable common behavioral patterns like whisking, sniffing, and lapping, as well as more nuanced behaviors that associate with internal cognitive states, task variables, and dominant motor cortical activity. These results emphasize the need to consider subtle, uninstructed movements in behavioral analyses, and particularly in BMI paradigms. Phase-invariant spectral transformations of higher frequency facial movement explain a large fraction of variance in motor cortical activity, and individual neurons show tuning to both phasic and envelope representations of movement. These findings demonstrate the importance of spectral transformations in relating motor cortical control and behavior. Face-rhythm and related approaches also show promise for a variety of applications, including discovering uninstructed behaviors, providing objective biomarkers for motor dysfunctions, finding predictive indicators for neurological conditions, and the development of face-computer-interfaces. Taken together, this work underscores the importance and effectiveness of integrating spectral transformations into behavioral analysis, identifies a strong phase-invariant representation of movement within M1, and highlights potential advancements for neural decoding and behavioral quantification.

## 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.

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).

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).

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*.

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).

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. Publisher: Institute of Electrical and Electronics Engineers (IEEE), 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).

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. Publisher: Springer Science and Business Media LLC. 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 calculation, (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' $\approx 3$, 'maxCount' $\approx 3$).

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 $\mathrm{x}_p^t$ and the original positions $\mathrm{x}_p^0$, 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.**

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

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

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:**

- $\mathrm{x}_p^t$: 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.
- $\mathrm{x}_p^0$: 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 5.12 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 (∼15 Hz) relative to the original point-tracking tensor (∼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_exponen` $\alpha$, `normalization_factor` $\gamma$, and `one_over_f_exponent` $\beta$:

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

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

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 $\beta$ flattens the spectrum.

### 5.1.4 Dimensionality reduction via tensor factorization

Non-negative Tensor component analysis (TCA) 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: $\boldsymbol{\mathcal{X}} = \{\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r\}_{r=1}^{R}$, 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 4):

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

**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:

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

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

**Rank selection.**
Rank is selected differently for each analysis. In some cases, a rough estimate of the rank and factor consistency is made by visual inspection on repeated runs of randomly initialized tensor factorizations (Williams et al. 2018). For final analyses, truncated SVD is used for initialization.

## 5.2 Widefield imaging regression comparison analysis

### 5.2.1 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 percent 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.

### 5.2.2 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 $\geq 95$ percent of the total variance. SVD yielded spatial components (U; pixels × components), temporal components (VT; components × 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 Butterworth 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 $\lambda$ 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 Fig. 2, using either ME PCA, FR components, or point tracking data as model regressors to predict cortical activity.

## 5.3 Pavlovian conditioning analysis

To analyze the utility of FR components in describing uninstructed behaviors during a Pavlovian conditioning task, we obtained a video dataset from a previous study (Matsumoto et al., 2016)(Matsumoto et al. 2016). FR temporal factors are extracted, and a custom multinomial tensor regression framework is used to predict the identity of the conditioned stimulus (CS) during delay epochs. We reference the methods used in that study for details on the behavioral paradigm.

We note that this recording was made under different conditions and without the initial intention of performing facial analysis; it is also captured at a relatively lower frame rate (30 Hz) and resolution. While this naturally constrains the spectrum and magnitude of resolvable movements, FR was successful in tracking and extracting robust components.

### 5.3.1 Behavioral paradigm and data acquisition

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

- $CS^+$: odor predictive of a water reward ($US^+$),
- $CS^0$: odor predictive of neither reward nor punishment (no US),
- $CS^-$: odor predictive of an aversive air puff ($US^-$).

Each trial consisted of a 1 s CS odor presentation, followed by a 1 s delay epoch, followed by the unconditioned stimulus (US). The total number of trials per session was $\sim 120$. Continuous video of the face was recorded at $\sim 30$ Hz. The standard face-rhythm pipeline is used to extract 10 temporal factors using left-asymmetric ('causal') kernels/filters (see 5.12.2 for details). The 4 temporal factors selected for visualization are based on visual inspection.

### 5.3.2 Input and output tensor design

To predict CS identity, we construct an input feature tensor containing the amplitude of each temporal factor at each time bin for each trial.

$$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{L \times T \times R} \tag{5.15}$$

where $L$ is the number of trials, $T$ is the number of time bins in the CS epoch and the delay period ($T \approx 2s$ or $15\ time\ bins$), and $R$ is the number of temporal factors ('rank').

We encode the output prediction as a one-hot matrix.

$$\mathbf{Y} \in \{0, 1\}^{L \times C} \tag{5.16}$$

where $C$ is the number of CS classes ($C = 3$), each row corresponds to a trial, and each column corresponds to a CS class.

### 5.3.3 Multinomial tensor regression model

To predict class identity from FR temporal factors, we employed a multinomial tensor regression (MTR) model. The model resembles multinomial logistic regression with the added benefit of an extra temporal dimension ($T$) to allow specific temporal patterns learned by the model to be used to predict class identity.

We construct the forward pass of the model as a tensor multiplication and contraction. Using Einstein summation notation (see 4), the model can also be expressed in one step:

$$\hat{\mathbf{Y}}_{L,C} = \boldsymbol{\mathcal{X}}_{L,T,R}\, \boldsymbol{\mathcal{W}}_{T,R,C} + \mathbf{b}_C, \tag{5.17}$$

where $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{T \times R \times C}$ is a weight tensor and $\mathbf{b} \in \mathbb{R}^C$ is a bias vector. To reduce overfitting and enhance interpretability, $\mathbf{W}$ is parameterized as a low-rank tensor, and a dense tensor is generated only during the forward pass. This is done to allow the ability to constrain certain dimensions to be non-negative. The tensor is constructed as a sum of outer products of three vectors, one for each dimension. Using Einstein summation notation, the weight tensor can be expressed as

$$\boldsymbol{\mathcal{W}}_{T,R,C} = \boldsymbol{\alpha}_{T,D}\, \boldsymbol{\beta}_{R,D}\, \boldsymbol{\gamma}_{C,D}, \tag{5.18}$$

where $\boldsymbol{\alpha}$, which describes the temporal dimension, is constrained to be non-negative, while the other two dimensions are not. The rank $D$ is set to 3, which does not reduce the rank of the model since the number of output classes is also 3. Non-negativity is imposed by passing $\boldsymbol{\alpha}$ through a softplus activation function during the forward pass.

Optimization is performed by minimizing the cross-entropy loss between the predicted class logits $\hat{\mathbf{Y}} \in \mathbb{R}^{L \times C}$ and the true one-hot labels $\mathbf{Y} \in \{0, 1\}^{L \times C}$. PyTorch's L-BFGS optimizer is used, and L2 regularization is manually applied to reduce overfitting.

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \mathbf{b}} \quad \texttt{cross-entropy}(\hat{\mathbf{Y}}, \mathbf{Y}) \tag{5.19}$$

The performance of the model is evaluated using a held-out test set.

## 5.4 RNN decoding of BMI reward events

To predict the onset times of reward-triggering threshold crossings, a lightweight recurrent neural network (RNN) is trained using FR temporal factors that are sequentially fed into the RNN one time step at a time. The analysis is performed in four steps: feature generation, target generation, model training & cross-validation, and performance evaluation. See Section 5.9 for details on the behavioral paradigm and data acquisition.

### 5.4.1 Feature and target generation

Temporal factors are obtained from a rank-$R$ tensor decomposition and are used as time-series inputs to the RNN, $\mathbf{C} \in \mathbb{R}^{T \times R}$. Critically, for these FR runs, a left-asymmetric ('causal') window is used for the VQT kernels (see Section 5.12.2), meaning that each element of the temporal factor only describes past information and no future information. Reward times are first converted to a Boolean vector $\mathbf{y} \in \{0, 1\}^T$ with a value of 1 for each video frame corresponding to a reward event.

To explore causal relationships and future information embedded in the FR temporal factors, the target vector is circularly shifted (or 'rolled') by $\Delta\tau \in$ {-1.5, -0.5, 0, +0.5} seconds (as well as 60 large random shifts as control/null distribution rolls) before each training run.

### 5.4.2 Network architecture

The RNN is designed to function as a binary classifier for the instantaneous probability of a reward event occurring at each video frame using the FR temporal factors as inputs. The architecture is a single-layer LSTM with hidden size $h = 5$, input dimensionality equal to the rank ($R = 5$) of the temporal factors. The input batches are $\sim 5$ seconds long. Optimization is performed using the Adam optimizer; cross-entropy loss is used to compare the 2D boolean one-hot matrix $\mathbf{Y}$ to the model's output 2D logit matrix $\hat{\mathbf{Y}}$. Regularization is not used given the small size of the network. The model is implemented using PyTorch.

$$\mathbf{C} \in \mathbb{R}^{T \times R} \xrightarrow{\text{LSTM}} \hat{\mathbf{Y}} \in \mathbb{R}^{T \times 2}, \tag{5.20}$$

$$\min_{\text{RNN params}} \quad \texttt{cross-entropy}(\hat{\mathbf{Y}}, \mathbf{Y}), \text{ where } \mathbf{Y} \in \{0, 1\}^{T \times 2}. \tag{5.21}$$

### 5.4.3 Cross-validation

For cross-validation, a group-shuffle-split approach is used. The full session is divided into contiguous, non-overlapping $\sim$10 minute blocks, and these blocks are used to segment the data into training, validation, and test sets.

### 5.4.4 Evaluation metrics

For the held-out test frames, we report the area under the curve (AUC) of the receiver operating characteristic (ROC) curve. This is calculated by sweeping a threshold over the model's output probabilities $\hat{\mathbf{y}}$ and computing the true positive rate (TPR) and false positive rate (FPR) at each threshold.

## 5.5 Convolutional regression

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.

### 5.5.1 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 $\mathbf{e}_r$:

$$\mathbf{e}_r = \mathbf{w}_{\text{enc},r}\, \mathbf{X}, \quad r = 1, \ldots, R, \tag{5.22}$$

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 5.10 for details). The convolution operation proceeds in one of three ways, depending on the model type:

- (ii.a) Basic convolution
$$\mathbf{z}_r = \mathbf{k}_r * \mathbf{e}_r, \tag{5.23}$$
where $\mathbf{k}_r \in \mathbb{R}^L$ is the convolution kernel for component $r$.
- (ii.b) Spectral convolution
$$\mathbf{z}_r = \left\| \mathbf{k}_r * \mathbf{e}_r \right\|, \tag{5.24}$$
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
$$\mathbf{z}_r = \left\| (\mathcal{H}[\mathbf{k}_r]) * \mathbf{e}_r \right\|, \tag{5.25}$$
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:

$$\hat{\mathbf{Y}} = \mathbf{W}_{\text{dec}}\, \mathbf{Z}, \tag{5.26}$$

**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:

$$\hat{\mathbf{Y}} = \mathbf{W}_{\text{dec}} \Big( \mathbf{K} * (\mathbf{W}_{\text{enc}}^{\top} \mathbf{X}) \Big), \tag{5.27}$$

$$\mathbf{X} \in \mathbb{R}^{2P \times T} \xrightarrow{\text{enc}} \mathbf{E} \in \mathbb{R}^{R \times T} \xrightarrow{\text{conv}} \mathbf{Z} \in \mathbb{R}^{R \times T} \xrightarrow{\text{dec}} \hat{\mathbf{Y}} \in \mathbb{R}^{N \times T}. \tag{5.28}$$

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.

### 5.5.2 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 $\mathbf{Y}$:

$$\min_{\mathbf{W}_{\text{enc}}, \mathbf{W}_{\text{dec}}, \mathbf{k}_r} MSE(\hat{\mathbf{Y}}, \mathbf{Y}) \tag{5.29}$$

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.

### 5.5.3 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.

$$\beta_n = (\hat{\mathbf{y}}_n^\top \hat{\mathbf{y}}_n)^{-1} \hat{\mathbf{y}}_n^\top \mathbf{y}_n, \tag{5.30}$$

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

$$\text{EVR} = 1 - \frac{\sum_{n=1}^{N} \|\mathbf{y}_n - \beta_n \hat{\mathbf{y}}_n\|_2^2}{\sum_{n=1}^{N} \|\mathbf{y}_n - \bar{y}_n\|_2^2}, \tag{5.31}$$

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

### 5.5.4 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 5.11). The filtered behavioral signal is then used in the same convRRR pipeline as above to compute the band-specific EVR.

## 5.6 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).

### 5.6.1 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 an FIR filter kernel (see 5.11 for details on FIR filtering):

$$\mathbf{x}_r^{\text{filt}} = FIR[\mathbf{a}_r \mathbf{X}, \mathbf{b}_r], \tag{5.32}$$

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:

$$\mathbf{x}_r^{\text{H}} = \mathcal{H}[\mathbf{x}_r^{\text{filt}}], \tag{5.33}$$

$$\mathbf{x}_r^{\phi} = \angle \mathbf{x}_r^{\text{H}}, \tag{5.34}$$

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 $r$:

$$\mathbf{y}^{\phi}_{n,r} = \angle\mathcal{H}[FIR[\mathbf{Y}, \mathbf{b}_r]], \tag{5.35}$$

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

$$\Delta\boldsymbol{\phi}_{n,r} = \mathbf{y}^{\phi}_{n,r} - \mathbf{x}^{\phi}_{r} \ , \tag{5.36}$$

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):

$$\mathrm{PPC}_{n,r} = \frac{(\sum_t \cos\Delta\boldsymbol{\phi}_{n,r})^2 + (\sum_t \sin\Delta\boldsymbol{\phi}_{n,r})^2 - T}{T(T-1)}, \tag{5.37}$$

We determined the statistical significance of an observed PPC value by comparing it to a null distribution obtained by recomputing PPC on (60) randomly temporally shifted behavioral signals ('shuffled'), yielding a p-value for each neuron against each behavioral signal $p_{n,r}$.

### 5.6.2 Envelope correlation

To obtain a phase-invariant behavioral signal, we take the magnitude of the analytic signal of the filtered behavioral signal (from eq. 5.33), commonly referred to as computing the spectral envelope:

$$\mathbf{x}^{\mathrm{env}}_{r} = \left\| \mathbf{x}^{\mathrm{H}}_{r} \right\| . \tag{5.38}$$

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

$$\rho_{n,r} = \mathrm{corr}\left(\mathbf{x}^{\mathrm{env}}_{r}, \mathbf{y}_n\right). \tag{5.39}$$

Significance is determined using the same shuffling procedure as for PPC.

## 5.7 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.

### 5.7.1 Data acquisition and preprocessing

Data was acquired from the baseline recording sessions of the BMI experiment described in Section 5.9 and 5.9.5. This includes wide-field 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 was 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.

### 5.7.2 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 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.

$$\mathcal{S} \approx \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r, \qquad \mathbf{Y} \approx \sum_{r=1}^{R} \mathbf{u}_r \otimes \mathbf{v}_r, \tag{5.40}$$

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.

### 5.7.3 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.

$$\pi^* \;=\; \arg\max_{\pi} \;\operatorname{trace}\Big(\operatorname{corr}\big(\mathbf{V},\, \mathbf{C}^{(\pi)}\big)\Big)\,. \tag{5.41}$$

where $\mathbf{C}^{(\pi)}$ denotes the column-permutation of $\mathbf{C}$ by $\pi$. The Hungarian algorithm (implemented using the `scipy.optimize.linear_sum_assignment` function) is used to find $\pi^*$, 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 against a null distribution of $n_{\text{shuf}} = 25000$ circularly time-shifted controls.

## 5.8 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 5.9 for details on the experimental setup).

### 5.8.1 Data acquisition and preprocessing

Data are acquired from the experimental sessions described in Section 5.9. This included wide-field facial video and two-photon calcium imaging of layer 2/3 neurons in the primary motor cortex (M1) of head-fixed mice.

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

$$\boldsymbol{\Phi} = \operatorname{coh}(\mathbf{X}, \mathbf{Y}), \tag{5.42}$$

where $\operatorname{coh}(\cdot)$ 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:

$$\boldsymbol{\Phi} \approx \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r \otimes \mathbf{d}_r, \tag{5.43}$$

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 5.9.6 for details).

### 5.8.2 Component selection and comparison to BMI decoder

We compare the neural loading that each animal is tasked with activating during the BMI experiment $\mathbf{w}_{dec} \in \mathbb{R}^N$ (the same weighting for all sessions; see 5.9.5 and 5.9.6 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 $\mathbf{w}_{dec}$. We define one of the joint neural factors as associated being associated with the animal's 'strategy' $\mathbf{b}_{strat}$ 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, $\mathbf{w}_{dec}$ is most strongly correlated with $\mathbf{b}_{strat}$.

## 5.9 BMI experiment data

### 5.9.1 Animals and surgery

6 week-old male C57BL/6J mice were surgerized and group housed in a reverse light/dark cycle. Craniotomies and intracranial injections were performed under isofluorane 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, ML -1.9) 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 to 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 $1.5 \times 10^{12}$ 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 analgesic 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.

### 5.9.2 Data acquisition

Mice are head-fixed, placed on a linear treadmill, and allowed to run freely. Wide-field video of the face is acquired at 120 Hz, 1s 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 16x 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 512x1024 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.

### 5.9.3 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, neuropil 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:

$$\mathrm{dF/F} = \Delta\mathbf{F}/\mathbf{F} = \frac{\dot{\mathbf{F}} - \mathbf{F}_{\text{baseline}}}{\mathbf{F}_{\text{baseline}}} \tag{5.44}$$

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

$$\dot{\mathbf{F}} = \mathbf{F} - 0.7 \cdot \mathbf{F}_{\text{neuropil}}, \tag{5.45}$$

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.

$$\mathbf{F}_{\text{baseline}} = \texttt{rolling_percentile}(\dot{\mathbf{F}},\ q = 30,\ window = 10\,\text{min}). \tag{5.46}$$

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.

### 5.9.4 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 + visual cursor that is played back to the animal in real-time so that they could learn to control the cursor's dynamics (see 5.9.6 and 5.9.7 for details on decoder design and 5.9.8 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/geometrically mapped to the cursor value between 2 kHz and 18 kHz, with the reward threshold at 18 kHz. The blue LED strip is oriented from left to right in around 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 $f_{\text{tone}}(t)$:

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

- **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:

$$i_{\text{LED}}(t) \;=\; \lceil c \times L_{\text{LED}} \rceil \quad \text{0-72 maps left-right.}$$

### 5.9.5 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.

### 5.9.6 Decoder axis selection process

We use the neural activity recorded from the baseline sessions to define a set of $\sim$8 neural factors $\mathbf{W} \in \mathbb{R}^{N \times F}$. One of these factors $\mathbf{w}_{dec}$ 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:

$$\Omega = \mathrm{PCA}(\mathbf{D}) \ , \qquad \Psi = \mathbf{X}\Omega \ , \tag{5.47}$$

where $\Omega \in \mathbb{R}^{N \times R}$ are principal components 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):

$$\hat{\mathbf{X}} = \mathbf{D}\Big(\left(\Psi^{\mathsf{T}}\Psi\right)^{-1}\Psi^{\mathsf{T}}\mathbf{X}\Big) \ , \tag{5.48}$$

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:

$$\mathbf{X}^{\mathrm{orth}} = \mathbf{X} - \hat{\mathbf{X}} \ , \tag{5.49}$$

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

$$\mathbf{X}^{\text{orth}} = \mathbf{X} \perp \mathbf{D} \ , \tag{5.50}$$

**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) $\alpha$ 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 as a method as 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 $\mathbf{X}^{\text{orth}}$:

$$\tilde{\mathbf{W}} \leftarrow \text{Ridge}(\mathbf{X}^{\text{orth}}, \mathbf{X}\mathbf{W}^{\text{SPCA}}) \ , \tag{5.51}$$

- **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 orthgonalization operator $\perp$ is derived from eq. 5.50

$$\tilde{\mathbf{W}} \leftarrow \{\mathbf{w}_1,\ \mathbf{w}_2 \perp \mathbf{w}_1,\ \mathbf{w}_3 \perp (\mathbf{w}_1,\ \mathbf{w}_2),\ \ldots,\ \mathbf{w}_F \perp (\mathbf{w}_1, \ldots, \mathbf{w}_{F-1})\} \ , \tag{5.52}$$

  and $\mathbf{w}_i$ is the $i$th 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 $\mathbf{W} \in \mathbb{R}^{N \times F} \leftarrow \tilde{\mathbf{W}}$.

### 5.9.7 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:

$$c_t = |\theta_{\mathbf{w}_{dec}}(\mathbf{s}_t)|^p \cdot \left\| \text{proj}_{\mathbf{w}_{dec}}(\mathbf{s}_t) \right\|_2 \ , \tag{5.53}$$

where $\mathbf{w}_{dec} \in \mathbb{R}^N$ 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 5.9.9). $\theta_{\mathbf{w}_{dec}}(\cdot)$ is the cosine similarity between $\mathbf{w}_{dec}$ and another vector, and $\text{proj}_{\mathbf{w}_{dec}}(\cdot)$ is the projection operator onto $\mathbf{w}_{dec}$. 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 $\tau_R$, which defines the cursor reward threshold; this value is tuned so that the simulated reward rate is $\sim 1.25$ 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 (more rapidly than linearly) 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.

#### 5.9.8 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 ($\sim$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 $\theta_{\text{reward}}$, quiescence threshold $\theta_{\text{quiet}}$
**Require:** quiescent duration $n_{\text{quiet}}$, maximum trial length $n_{\text{trial}}$
**Require:** maximum session frames $T_{\text{session}}$, maximum reward number $R_{\text{max}}$

**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**

#### 5.9.9 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 applied a non-rigid image warp to ensure robust coordinate alignment between sessions.

**Motion correction**

**Planar correction** Online motion correction during a BMI experiment is implemented using a custom phase-correlation algorithm. To improve robustness, we applied 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 512x1024 to 512x512).

**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**

**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 $\mathbf{M} \in \mathbb{R}^{N \times H \cdot W}$:

$$\mathbf{F} = \mathbf{M}\mathbf{A} \ , \tag{5.54}$$

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. 5.44):

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 5.9.3). 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 seconds, with minimum inter-trial intervals (ITIs) of 3 seconds. Any threshold crossing event is discarded if it lasted less than 3 frames. The reward is calibrated to be $\sim 3.5$ l per delivery.

## 5.10 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:

$$\text{FFTconv}(\mathbf{x}, \mathbf{k}) = \mathcal{F}^{-1}\Big[\mathcal{F}[\mathbf{x}] \odot \mathcal{F}[\text{flip}[\mathbf{k}]]\Big] \qquad = \qquad \text{conv}(\mathbf{x}, \mathbf{k}) = \mathbf{x} * \mathbf{k} \tag{5.55}$$

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.

## 5.11 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 5.10):

$$\text{FIR}(\mathbf{x}, \mathbf{k}) = \mathcal{F}^{-1}\Big[\mathcal{F}[\mathbf{x}] \odot \mathcal{F}[\mathbf{k}] \odot \mathcal{F}[\text{flip}[\mathbf{k}]]\Big] \tag{5.56}$$

## 5.12 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 with-

out artifacts, more flexibility in the choice of parameters, and the ability to arbitrarily customize filter kernels.

### 5.12.1 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 $Q_{\text{low}}$ and $Q_{\text{high}}$ and interpolate each filter's quality factor logarithmically:

$$Q_k = Q_{\text{low}} \left( \frac{Q_{\text{high}}}{Q_{\text{low}}} \right)^{(k-1)/(F-1)}, \quad k = 1, \ldots, F. \tag{5.57}$$

The window length for filter $k$ is

$$L = \left\lceil \frac{Q_k \, f_s}{f_k} \right\rceil , \tag{5.58}$$

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

$$\mathbf{h}_k[n] = \mathbf{w}_k[n] e^{2\pi i f_k n / f_s}, \quad n = 0, \ldots, L-1, \tag{5.59}$$

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 5.10 for details):

$$\mathbf{z}_k = \mathbf{x} * \mathbf{h}_k \ , \qquad \mathbf{Z} = \{\mathbf{z}_k\}_{k=1}^{F} \ , \qquad \mathbf{z}_k \in \mathbb{C}^T, \ \mathbf{Z} \in \mathbb{C}^{F \times T}, \ \mathbf{x} \in \mathbb{R}^T, \ \mathbf{h}_k \in \mathbb{C}^L, \tag{5.60}$$

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

$$\mathbf{S} = \left\| \mathbf{Z} \right\| , \qquad \mathbf{S} \in \mathbb{R}^{F \times T}. \tag{5.61}$$

### 5.12.2 Asymmetric windows

Asymmetric windows can be used to zero out one side/half of the filter. For several analyses, we used left-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.

### 5.12.3 Software availability

A user-oriented library was developed. The backend is written using PyTorch, allowing for GPU acceleration and automatic differentiation. Convolution is performed either directly or using Fourier-convolution, and complex filters are generated using a Hilbert transform. Code is available here: https://github.com/RichieHakim/vqt/.

## 5.13 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 1: 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 2: 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 3: 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 4: 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. Publisher: Springer Science and Business Media LLC, 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).

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*.

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).

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*. Publisher: Springer Science and Business Media LLC. 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).

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).

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).

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

Conceptualization: RH, BLS. Computational methodology: RH. Experiments: RH, GH, SM. Analysis: RH, SM. Software development: RH, AJ. Funding: BLS, RSD. Writing—draft: RH, BLS. Writing—editing: RH, GH, SM, BLS.

## 7 Acknowledgements

We thank: Hideyuki Matsumoto, Ju Tian, Naoshige Uchida, and Mitsuko Watabe-Uchida for contributing the classical conditioning dataset; Alex Williams and Scott Linderman for helpful discussions; and Christopher Harvey, John Assad, and Mark Harnett for guidance.