OMiSO: Adaptive optimization of state-dependent brain stimulation to shape neural population states

Yuki Minai^{1,2,3}, **Joana Soldado-Magraner**^{1,3,4}, **Byron M. Yu**^{1,3,4,5*}, **Matthew A. Smith**^{1,3,5*}
^{1}Neuroscience Institute, Carnegie Mellon University
^{2}Machine Learning Department, Carnegie Mellon University
^{3}Center for the Neural Basis of Cognition
^{4}Department of Electrical and Computer Engineering, Carnegie Mellon University
^{5}Department of Biomedical Engineering, Carnegie Mellon University
`{yminai,jsoldado,byronyu,msmith}@andrew.cmu.edu`
*Denotes equal contribution.

Abstract

The coordinated activity of neural populations underlies myriad brain functions. Manipulating this activity using brain stimulation techniques has great potential for scientific and clinical applications, as they causally influence the nervous system. To improve the accuracy by which one can manipulate neural activity, it is important to (1) take into account the pre-stimulation brain state, which can influence the brain's response to stimulation, and (2) adaptively update stimulation parameters over time to compensate for changes in the brain's response to stimulation. In this work, we propose Online MicroStimulation Optimization (OMiSO), a brain stimulation framework that leverages brain state information to find stimulation parameters that can drive neural population activity toward specified states. OMiSO includes two key advances: i) training a stimulation-response model that leverages the pre-stimulation brain state, and inverting this model to choose the stimulation parameters, and ii) updating this inverse model online using newly-observed responses to stimulation. We tested OMiSO using intracortical microstimulation with a "Utah" array and found that it outperformed competing methods that do not incorporate these advances. Taken together, OMiSO provides greater accuracy in achieving specified activity states, thereby advancing neuromodulation technologies for understanding the brain and for treating brain disorders.

1 Introduction

Most brain functions are realized through the coordinated activity of neural populations [1–4]. Causal perturbations of the brain, for example with electrical stimulation, can influence brain function by modulating neural population activity [5]. This activity is variable from moment to moment, and such variability in part reflects the current state of the brain [1, 6]. The brain's state can influence how neural populations respond to incoming sensory stimuli [7, 8], and is therefore also important for understanding how the brain responds to causal perturbations.

Closed-loop brain stimulation tools iteratively test and adjust stimulation parameters based on observed neural responses to induce desired brain activity and/or behavior. Prior studies proposed closed-loop brain stimulation tools to drive neural population activity toward specific states [9, 10]. These methods learn and update the mapping between a wide range of potential brain stimulation parameters and brain responses during an experiment, but they do not take into account the state of the brain when choosing the stimulation parameters (but see [11] for a study testing this idea in simulation). Deep brain stimulation (DBS) approaches have leveraged brain state information

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

to update stimulation parameters for improving treatment efficacy [12, 13] or reducing the energy consumption of the stimulation device [14]. However, these methods were designed for manipulating low-dimensional (low-d) bio-markers (e.g., local field potentials [14]), rather than the activity of many simultaneously recorded neurons.

In this study, we develop Online MicroStimulation Optimization (OMiSO), a brain stimulation framework that manipulates neural population activity by incorporating the brain state to choose the stimulation parameters. OMiSO characterizes neural population activity in a low-d latent space identified using dimensionality reduction [15]. We refer to the low-d projection of the recorded population activity as a "brain state". The objective of OMiSO is to identify stimulation parameters that induce a specified target brain state given the pre-stimulation brain state. Specifically, OMiSO first fits a stimulation-response model, which predicts brain responses to different stimulation parameters given the pre-stimulation state. OMiSO then inverts the trained stimulation-response model, which is updated in real time (i.e., online), to output stimulation parameters for inducing a target brain state.

We tested OMiSO using electrical microstimulation (uStim) in a macaque monkey implanted with a multi-electrode Utah array in the prefrontal cortex (PFC, area 8Ar). OMiSO optimized the location of five stimulated electrodes on each trial, enabling the generation of complex spatial patterns of electric fields across a wide range of electrodes in the array. We found that taking into account brain state prior to uStim (pre-uStim) was beneficial to improve the prediction of the brain's response to uStim (Section 3.1). An inverse model, which incorporates brain state, successfully identified the stimulation parameters (in this case, electrode locations) necessary to achieve the specified brain states (Section 3.2). By adaptively updating the inverse model using newly-observed stimulation-response samples, OMiSO significantly improved the uStim parameter optimization performance (Section 3.3).

2 Methods

2.1 OMiSO overview

The goal of OMiSO is to find a set of stimulation parameters (termed "stimulation pattern") to drive neural population activity toward a targeted state given the pre-stimulation state (Fig. 1A, left). Specifically, OMiSO first collects and merges neural activity across multiple experimental sessions by using latent space alignment (Section 2.2). The merged sessions are used to fit a stimulation-response model, which predicts the post-stimulation brain state given the stimulation pattern and the pre-stimulation brain state (Section 2.3). OMiSO then inverts the trained stimulation-response model to output a stimulation pattern that can create a target state (Section 2.4). The inverse model is used to choose a stimulation pattern in a brain stimulation experiment. On each trial, OMiSO observes the pre-stimulation brain state $\boldsymbol{z}^{\text{Pre}}$ and uses the inverse model $\pi_\theta(\hat{\boldsymbol{s}}|\boldsymbol{z}^{\text{Pre}}, \boldsymbol{z}^{\text{Targ}})$, which is parameterized with θ, to choose a stimulation pattern $\hat{\boldsymbol{s}}$ that minimizes the error between the target state $\boldsymbol{z}^{\text{Targ}}$ and the induced brain state $\boldsymbol{z}_{\hat{\boldsymbol{s}}}$ (Section 2.5):

$$\underset{\hat{\boldsymbol{s}}}{\operatorname{argmin}} \, ||\boldsymbol{z}^{\text{Targ}} - \boldsymbol{z}_{\hat{\boldsymbol{s}}}||, \tag{1}$$

Then, using the newly-observed $\boldsymbol{z}^{\text{Pre}}$ and $\boldsymbol{z}_{\hat{\boldsymbol{s}}}$, OMiSO adaptively updates the inverse model π_θ to improve the selection of $\hat{s}$ (Section 2.6, Fig. 1A, right). The hyperparameters used in the model fitting and brain stimulation experiments are summarized in Section S1.

2.2 Latent space identification and alignment

When the stimulation parameter space is large, one can typically test only a small fraction of all possible stimulation patterns within an experimental session. This requires merging neural activity across sessions to create a large enough set of stimulation-response samples to learn their relationship. To merge neural activity across multiple experimental sessions, OMiSO identifies a low-d latent space of the high-d population activity in each session and aligns the latent spaces across sessions. Each session consists of two types of trials: "stimulation trials", in which we applied stimulation, and "no-stimulation trials", in which we did not apply stimulation. Following [10], for each session, we use Factor Analysis (FA) to identify a latent space that captures activity co-variations between neurons using only no-stimulation trials. To align the identified latent spaces across sessions, we solve the Procrustes problem to find an orthogonal transformation matrix that maximizes the alignment between two FA latent spaces [16].

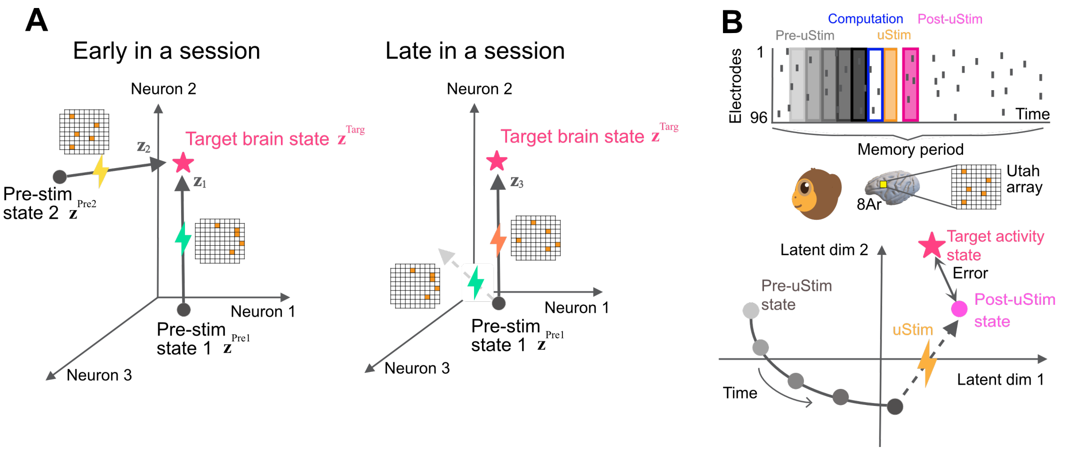


Figure 1: **Goal of OMiSO and experimental paradigm.** (A) OMiSO finds a stimulation pattern (grid representing an electrode array, where the stimulated electrodes are indicated by orange cells) to drive neural population activity toward a target state (pink star) given a pre-stimulation state (black dots). For different pre-stimulation states (Pre-stim state 1 vs. Pre-stim state 2), OMiSO selects different stimulation patterns (green bolt vs. yellow bolt) to achieve the same target state. For Pre-stim state 1, OMiSO updates the selected stimulation pattern from early (left panel, green bolt) to late (right panel, orange bolt) in an experimental session to achieve the target state. This is necessitated by the fact that the stimulation pattern that achieved the target state early in the session (left panel, green bolt) no longer does so late in the session (right panel, green bolt). (B) Experimental setup. Top: The monkey performed a memory-guided saccade task (Section S3) while spiking activity was recorded from a multi-electrode array implanted in PFC. Pre-uStim neural activity (gray bars) was recorded over five consecutive 50 ms bins. During the 50 ms computation period (blue unfilled bar), the system chose a uStim pattern, which was then applied for 40 ms (orange bar). Post-uStim neural activity (pink bar) was measured starting 10 ms after stimulation offset. Bottom: neural population activity was analyzed in a low-d (in our experiments, 5-d) latent space.

After collecting data over R experimental sessions, for the ith session, OMiSO extracts a list of n_i "usable" electrodes $\boldsymbol{e}_i \in \mathbb{R}^{n_i}$, where each element of $\boldsymbol{e}_i$ is an integer index of an electrode. An electrode is deemed "usable" if it satisfies criteria involving its mean firing rate, Fano factor, and coincident spiking with other electrodes (see Section S2). In our multi-electrode array stimulation set-up with 96 implanted electrodes (indexed 1 to 96), $\boldsymbol{e}_i$ represents the subset of electrode indices identified as usable in the ith session (e.g., $[1, 4, 10, 32, \ldots, 93]$).

For each time bin indexed by $j = 1, ..., J_i$ (where J_i is the total number of time bins used to fit FA for the ith session), OMiSO takes spike counts on each usable electrode $\boldsymbol{x}_{i,j} \in \mathbb{R}^{n_i}$ and fits the following FA model using the EM algorithm:

$$\begin{aligned} \boldsymbol{z}_{i,j} &\sim \mathcal{N}(\mathbf{0}, I) \\ \boldsymbol{x}_{i,j} | \boldsymbol{z}_{i,j} &\sim \mathcal{N}(\Lambda_i \boldsymbol{z}_{i,j} + \boldsymbol{\mu}_i, \Psi_i) \end{aligned} \tag{2}$$

where $\boldsymbol{z}_{i,j} \in \mathbb{R}^m$ ($m < n_i$) is the low-d brain state for the jth time bin, $\Lambda_i \in \mathbb{R}^{n_i \times m}$ is the loading matrix whose columns define the low-d latent space, $\boldsymbol{\mu}_i \in \mathbb{R}^{n_i}$ contains the mean spike counts for each electrode, and $\Psi_i \in \mathbb{R}^{n_i \times n_i}$ is a diagonal matrix capturing the independent variance of the spike counts for each electrode. In this work, the latent dimensionality m was chosen by first computing the optimal dimensionality separately for each session based on cross-validated data likelihoods, then finding the mode of the distribution of optimal dimensionalities across sessions ($m = 5$ in our implementation, Section S1).

OMiSO then defines the latent space from one of the R sessions as a reference latent space $\Lambda_0 \in \mathbb{R}^{n_0 \times m}$ and aligns the latent space of the i-th session Λ_i to Λ_0. In contrast to our previous work [10], which aligns the latent spaces using the common "usable" electrodes, OMiSO introduces an extra step to identify the common "stable" electrodes among usable electrodes to make the alignment more robust to neural recording instabilities across sessions (see Algorithm 1 in [16]). Concretely, for the

ith session, OMiSO identifies an orthogonal transformation matrix $\hat{O}_i \in \mathbb{R}^{m \times m}$ that fulfills:

$$\hat{O}_i = \underset{O:OO^{\intercal}=I}{\operatorname{argmin}} \|\Lambda_0(\boldsymbol{e}_{\text{stable}}, :) - \Lambda_i(\boldsymbol{e}_{\text{stable}}, :)O^{\intercal}\|_F^2 \tag{3}$$

where $\boldsymbol{e}_{\text{stable}}$ is a list of n_{stable} stable electrodes that are common to the reference session and the ith session, and $\|\cdot\|_F$ is the Frobenius Norm. This optimization can be solved in closed-form [17]. The $\hat{O}_i$ found is applied to Λ_i to obtain the aligned latent space $\tilde{\Lambda}_i \in \mathbb{R}^{n_i \times m}$:

$$\tilde{\Lambda}_i = \Lambda_i \hat{O}_i^{\intercal} \tag{4}$$

The brain states in the pre-stimulation (Fig. 1B, bottom panel, gray dots) and post-stimulation periods (Fig. 1B, bottom panel, pink dot) are estimated as the posterior mean from the FA model (Eq. 2) using the loading matrix $\tilde{\Lambda}_i$:

$$\mathbb{E}[\boldsymbol{z}_{i,j}|\boldsymbol{x}_{i,j}] = \beta_i(\boldsymbol{x}_{i,j} - \boldsymbol{\mu}_i) \tag{5}$$

where $\beta_i = \tilde{\Lambda}_i^{\intercal}(\tilde{\Lambda}_i\tilde{\Lambda}_i^{\intercal} + \Psi_i)^{-1}$. By using $\tilde{\Lambda}_i$, the induced brain state $\mathbb{E}[\boldsymbol{z}_{i,j}|\boldsymbol{x}_{i,j}]$ resides in a common latent space across all sessions.

2.3 Stimulation-response model fitting

Using the merged neural activity across sessions, OMiSO fits a stimulation-response model to predict a post-stimulation brain state given a stimulation pattern and pre-stimulation brain state. In this work, we used a combination of a CNN and a LSTM (Fig. S1). To train the stimulation-response model, OMiSO uses the stimulation trials of R experimental sessions collected in Section 2.2, which involves stimulating with randomly chosen patterns among all possible stimulation patterns defined by the user. OMiSO creates a dataset that comprises the pre-stimulation and post-stimulation brain states (where the latent space is defined using latent space alignment, Section 2.2), and the stimulation patterns tested across the R sessions. For the kth trial in the ith session, the pre-stimulation brain state over t time bins $\boldsymbol{Z}_{i,k}^{\text{Pre}} \in \mathbb{R}^{m \times t}$ and induced post-stimulation brain state $\boldsymbol{z}_{i,k}^{\text{Post}} \in \mathbb{R}^m$ are computed using Eq. 5. Note that $\boldsymbol{Z}$ denotes pre-stimulation brain states across multiple timesteps (a matrix), whereas $\boldsymbol{z}$ denotes a post-stimulation brain state at a single timestep (a vector). The entries of $\boldsymbol{Z}_{i,k}^{\text{Pre}}$ and $\boldsymbol{z}_{i,k}^{\text{Post}}$ corresponding to the user-defined l target dimensions within the m dimensional aligned latent space ($l \leq m$) are then subselected and collected in new vectors $\check{\boldsymbol{Z}}_{i,k}^{\text{Pre}} \in \mathbb{R}^{l \times t}$ and $\check{\boldsymbol{z}}_{i,k}^{\text{Post}} \in \mathbb{R}^l$ (see Section S3 for how l target dimensions are chosen). Given a planar grid of electrodes of size $h \times v$ ($h, v \in \mathbb{R}$), the stimulation pattern $S_{i,k} \in \mathbb{R}^{h \times v}$ tested in the kth trial during the ith session is encoded using a value of 1 for stimulated electrodes and 0 for all other electrodes.

To train the stimulation-response model $f(\hat{\check{\boldsymbol{z}}}_{i,k}^{\text{Post}}|S_{i,k}, \check{\boldsymbol{Z}}_{i,k}^{\text{Pre}})$, which maps the stimulation pattern $S_{i,k}$ and pre-stimulation state $\check{\boldsymbol{Z}}_{i,k}^{\text{Pre}}$ to the predicted post-stimulation state $\hat{\check{\boldsymbol{z}}}_{i,k}^{\text{Post}} \in \mathbb{R}^l$, OMiSO minimizes the mean squared error (MSE) between the predicted $\hat{\check{\boldsymbol{z}}}_{i,k}^{\text{Post}}$ and observed $\check{\boldsymbol{z}}_{i,k}^{\text{Post}}$ post-stimulation brain states. OMiSO performs R-fold cross validation, where each session is used as a validation set once (Section S4). The final prediction $\hat{\check{\boldsymbol{z}}}_{i,k}^{\text{Post}}$ is obtained by averaging the predictions of the R models.

2.4 Inverting the stimulation-response model

Using the trained stimulation-response model, OMiSO estimates a stimulation-response inverse model. The goal of the inverse model is to identify a stimulation pattern that can reach a target brain state given the pre-stimulation brain state. More specifically, a stimulation-response inverse model $\pi_\theta(\hat{\boldsymbol{s}}_{i,k}|\check{\boldsymbol{z}}^{\text{Targ}}, \check{\boldsymbol{Z}}_{i,k}^{\text{Pre}})$, parameterized by θ, receives the target brain state $\check{\boldsymbol{z}}^{\text{Targ}} \in \mathbb{R}^l$ and pre-stimulation state $\check{\boldsymbol{Z}}_{i,k}^{\text{Pre}} \in \mathbb{R}^{l \times t}$. Then, it returns a u-dimensional vector of stimulation suitabilities for each electrode $\hat{\boldsymbol{s}}_{i,k} \in \mathbb{R}^u$, where u is the total number of candidate electrodes to be stimulated whose value is determined by the user and $u \leq h \times v$ (Section S3). Each entry in $\hat{\boldsymbol{s}}_{i,k}$ is a number between 0 and 1 representing the suitability of stimulating each electrode for achieving the target brain state, where a value closer to 1 indicates greater suitability. Note that the vector is not normalized to sum to one.

OMiSO generates synthetic data using the trained stimulation-response model to train an inverse model π_θ (Fig. S1). The stimulation-response model outputs the average post-stimulation state over R models $\hat{\check{\boldsymbol{z}}}^{\text{Post}} \in \mathbb{R}^l$ for any given combination of a stimulation pattern $S \in \mathbb{R}^{h \times v}$ and pre-stimulation state $\check{\boldsymbol{Z}}^{\text{Pre}} \in \mathbb{R}^{l\times t}$. By considering the predicted post-stimulation state $\check{\boldsymbol{z}}^{\text{Post}}$ as a target state $\check{\boldsymbol{z}}^{\text{Targ}}$, OMiSO constructs synthetic data tuples $(\check{\boldsymbol{Z}}^{\text{Pre}}, \check{\boldsymbol{z}}^{\text{Targ}}, \boldsymbol{s})$, where $\boldsymbol{s} \in \{0,1\}^u$ is a vector representation of S including only the u candidate electrodes for stimulation. The value for each candidate electrode in $\boldsymbol{s}$ is 1 if stimulated and 0 if not. The generated synthetic data is used to train the inverse model by minimizing the binary cross entropy over multiple epochs with respect to θ (Section S5). In this study, we used a Multi Layered Perceptron (MLP, Fig. S1) as the stimulation-response inverse model.

2.5 Stimulation pattern selection on each trial

To find the stimulation pattern for achieving a target brain state with the inverse model, each brain stimulation session starts with latent space identification trials, where no stimulation is applied. OMiSO extracts a list of usable electrodes $\boldsymbol{e_i}$, which includes n_i electrodes, using these trials. The observed spike count vectors $\boldsymbol{x}_{i,j} \in \mathbb{R}^{n_i}$ for all time bins j across these trials are used to fit the FA parameters $\Lambda_i \in \mathbb{R}^{n_i\times m}$, $\boldsymbol{\mu}_i \in \mathbb{R}^{n_i}$, and $\Psi_i \in \mathbb{R}^{n_i\times n_i}$. The identified latent space Λ_i is aligned to the reference latent space Λ_0 using the methods described in Section 2.2, yielding $\tilde{\Lambda}_i \in \mathbb{R}^{n_i\times m}$. To start the optimization, OMiSO loads the user defined target brain state $\check{\boldsymbol{z}}^{\text{Targ}} \in \mathbb{R}^l$, the trained inverse model π_θ, and the number of electrodes used for each stimulation pattern $n_{\text{Stim}} \in \mathbb{Z}$. Note that out of $h \times v$ total electrodes (used for the stimulation pattern representation S), not all may be suitable for stimulation (e.g., some may have no effect on the neural population activity, Section S3). This results in u $(\leq h \times v)$ candidate electrodes (used for the stimulation pattern representation $\boldsymbol{s}$). Of those, OMiSO chooses n_{Stim} electrodes to form each stimulation pattern, where $n_{\text{Stim}} \leq u$.

On each trial, OMiSO selects the stimulation pattern to perform given the observed pre-stimulation brain state using the inverse model π_θ. On the kth trial, OMiSO estimates the pre-stimulation brain state $\boldsymbol{Z}_{i,k}^{\text{Pre}} \in \mathbb{R}^{m\times t}$ in real time using Eq. 5. The entries of $\boldsymbol{Z}_{i,k}^{\text{Pre}}$ corresponding to the target dimensions are subselected as $\check{\boldsymbol{Z}}_{i,k}^{\text{Pre}} \in \mathbb{R}^{l\times t}$, and $\check{\boldsymbol{Z}}_{i,k}^{\text{Pre}}$ and $\check{\boldsymbol{z}}^{\text{Targ}}$ are passed to π_θ to get a u-dimensional vector $\hat{\boldsymbol{s}}_{i,k}$ of stimulation suitabilities for each candidate electrode. Using $\hat{\boldsymbol{s}}_{i,k}$, OMiSO chooses a stimulation pattern with an epsilon greedy algorithm. With probability $1 - \varepsilon_{i,k}$, where $0 \leq \varepsilon_{i,k} \leq 1$, it chooses the n_{Stim} electrodes with the highest predicted suitabilities in $\hat{\boldsymbol{s}}_{i,k}$. With probability $\varepsilon_{i,k}$, it chooses n_{Stim} electrodes stochastically following the softmax transformed probability $p_{i,k}^w \in \mathbb{R}$ computed for each electrode indexed by $w = 1, ..., u$ as:

$$p_{i,k}^w = \frac{\exp(\hat{s}_{i,k}^w)}{\sum_{v=1}^{u} \exp(\hat{s}_{i,k}^v)} \tag{6}$$

where $\hat{s}_{i,k}^w \in \mathbb{R}$ is the predicted stimulation suitability for the wth electrode. In this way, electrodes with low predicted stimulation suitability can be occasionally chosen to induce exploration, but the electrodes with higher stimulation suitabilities are chosen more often. OMiSO uses a time-varying $\varepsilon_{i,k}$ to initially encourage exploration and gradually shift to exploitation:

$$\varepsilon_{i,k} = \max(\varepsilon_{\text{init}} \cdot \gamma^k,\ \varepsilon_{\text{floor}}) \tag{7}$$

where $0 \leq \varepsilon_{\text{init}} \leq 1$ is an initial value of $\varepsilon_{i,k}$, $0 \leq \gamma \leq 1$ is a discount factor, $0 \leq \varepsilon_{\text{floor}} \leq 1$ is the smallest allowable value of $\varepsilon_{i,k}$ $(\varepsilon_{\text{floor}} \leq \varepsilon_{\text{init}})$, and k is the current trial index. OMiSO uses $\varepsilon_{\text{floor}}$ to maintain some amount of exploration throughout a session. Note that the inverse model π_θ estimates the suitability of each electrode independently, without explicitly modeling potential interactions between electrodes stimulated simultaneously. Consequently, the selection of the top n_{Stim} electrodes is based on their individual contributions to producing the target state. This design choice was motivated by our empirical observation that, using our stimulation parameters (25 µA, 50 Hz, 40 ms), stimlating a single electrode primarily affected nearby recording electrodes, with minimal interaction with other concurrently stimulated electrodes.

Once the stimulation pattern is selected with the epsilon greedy algorithm, OMiSO stimulates the selected n_{Stim} electrodes and measures post-stimulation spike counts $\boldsymbol{x}_{i,k}^{\text{Post}} \in \mathbb{R}^{n_i}$ to compute $\check{\boldsymbol{z}}_{i,k}^{\text{Post}} \in \mathbb{R}^l$ (Eq. 5). The set of $\check{\boldsymbol{z}}^{\text{Targ}}$, $\check{\boldsymbol{Z}}_{i,k}^{\text{Pre}}$, $\check{\boldsymbol{z}}_{i,k}^{\text{Post}}$, and the tested stimulation pattern $\boldsymbol{s}_{i,k} \in \{0,1\}^u$ are stored to update the inverse model in a batched manner, as explained in the next section.

2.6 Adaptive updating of the stimulation-response inverse model

During the brain stimulation experiment, OMiSO adaptively updates the stimulation-response inverse model π_θ using the recent stored observations. In this study, we implemented OMiSO's adaptive model update using the clipped policy gradient objective function used in Proximal Policy Optimization (PPO) [18]. PPO is a widely-used RL method to update an action selection strategy. OMiSO uses a PPO-inspired objective function to update the stimulation-response inverse model to improve the stimulation pattern selection strategy. Compared to other policy update algorithms such as REINFORCE [19] and Actor Critic [20], PPO offers a more stable policy update by using a clipped objective function, which might be particularly well-suited to handle trial-by-trial variance in neural activity.

At each update, OMiSO increases the suitability of stimulating electrodes that produced a post-stimulation state closer to the target state and decreases it for electrodes that did not. Specifically, for the kth stored trial in a batch update of the inverse model (a batch is five trials in our implementation, Section S1), OMiSO computes the advantage $A_{i,k} \in \mathbb{R}$ for the stimulation pattern used in the trial:

$$A_{i,k} = -||\check{\boldsymbol{z}}^{\text{Targ}} - \check{\boldsymbol{z}}^{\text{Post}}_{i,k}||_1 + b \tag{8}$$

where $b \in \mathbb{R}$ is a fixed baseline value. In contrast to the original PPO implementation, which uses a separate model (often called a value network) to estimate the expected outcome at each state, OMiSO uses a fixed baseline b shared across all pre-uStim states. By avoiding the simultaneous online updating of two interacting models (the inverse model and the value network), this design reduces the inverse model update instability especially when only a limited number of observations are available. The value of b was chosen by running a simulation (Section S6). With the computed advantage $A_{i,k}$, OMiSO updates the parameters θ in the stimulation-response inverse model π_θ by performing gradient ascent over multiple epochs on the following PPO-inspired objective function:

$$\begin{gathered}\mathcal{L}_{\text{PPO}} = \textstyle\sum_{e=1}^{n_{\text{Stim}}} \min\left(r^e_{i,k}(\theta)A_{i,k}, \operatorname{clip}\left(r^e_{i,k}(\theta), 1-\varepsilon_{\text{clip}}, 1+\varepsilon_{\text{clip}}\right)A_{i,k}\right), \\ \text{where} \quad r^e_{i,k}(\theta) = \frac{\pi_\theta(\hat{s}^e_{i,k}|\check{\boldsymbol{z}}^{\text{Targ}}, \check{\boldsymbol{Z}}^{\text{Pre}}_{i,k})}{\pi_{\theta_{\text{old}}}(\hat{s}^e_{i,k}|\check{\boldsymbol{z}}^{\text{Targ}}, \check{\boldsymbol{Z}}^{\text{Pre}}_{i,k})},\end{gathered} \tag{9}$$

$e = 1, ..., n_{\text{Stim}}$ is the index of the electrode among n_{Stim} electrodes used for the stimulation pattern $\boldsymbol{s}_{i,k}$, and the clipping parameter $0 \leq \varepsilon_{\text{clip}} \leq 1$ defines upper and lower bounds of $r^e_{i,k}(\theta)$ through the clip function, limiting the magnitude of the gradient update to avoid taking a too large update step (Section S1). $\pi_\theta(\hat{s}^e_{i,k}|\check{\boldsymbol{z}}^{\text{Targ}}, \check{\boldsymbol{Z}}^{\text{Pre}}_{i,k})$ represents the predicted stimulation suitability for the eth stimulated electrode under the updated stimulation-response inverse model. Similarly, $\pi_{\theta_{\text{old}}}(\hat{s}^e_{i,k}|\check{\boldsymbol{z}}^{\text{Targ}}, \check{\boldsymbol{Z}}^{\text{Pre}}_{i,k})$ is the predicted suitability under the original stimulation-response inverse model before updating θ. The ratio $r^e_{i,k}(\theta)$ is the change in suitability for selecting the eth electrode. Computing this ratio separately for each electrode rather than for an entire stimulation pattern is a design choice, motivated by the same empirical observation as in Section 2.5 for the selection of which electrodes to stimulate. To maximize this objective function, the model updates θ to increase $r^e_{i,k}(\theta)$ when the advantage is positive and decrease it when the advantage is negative. By updating θ, OMiSO not only updates the predictions of the stimulation patterns tested in the stored observations, it also updates the predictions of stimulation patterns that were not tested.

2.7 Experimental details

We tested OMiSO using uStim in a rhesus macaque monkey with a 96-electrode Utah array implanted in the PFC. In each experimental session, the monkey performed a memory-guided saccade task (Section S3). Each trial began when the monkey fixated a central dot, followed by a memory target briefly appearing at a peripheral location. After the memory target disappeared, the monkey needed to remember the target location during a “memory period” (Fig. 1B). The disappearance of the central dot served as the “go cue”, after which the monkey made a saccade to the remembered location. On about 80% of trials, we applied uStim for 40 ms during the memory period (Fig. 1B). The stimulation patterns tested in this study consisted of all possible $n_{\text{Stim}} = 5$ electrode spatial patterns chosen from $u = 20$ candidate electrodes (15,504 possible patterns, Fig. S2, see Section S3 for the selection criteria of the candidate electrodes). Each experimental session started with 120 no-uStim trials for latent space identification. We identified $m = 5$ latent dimensions by applying FA to spike counts taken in 50 ms bins. Among them, we chose $l = 2$ target dimensions (Section S3). Post-uStim spike counts (50 ms bin) were calculated starting 10 ms after uStim offset (Fig. 1B, pink bar), which we empirically verified to be sufficient to recover from stimulation artifacts induced by our uStim set-up. The number of sessions used in each analysis is summarized in Section S4.

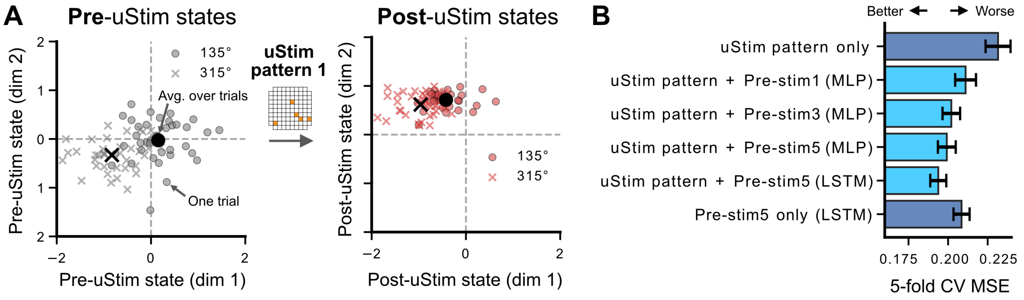


Figure 2: **The pre-uStim brain state affects the post-uStim brain state.** (A) An example 5-electrode uStim pattern. The orange cells in the array map indicate the location of the stimulated electrodes. Each small circle (135° memory condition) or cross (315° memory condition) indicates the brain state on a single trial, and the large black circle and cross indicate the mean across trials within each memory condition. (B) Post-uStim state prediction performance across models (see Fig. S1 for model architectures). Error bars indicate ± 1 standard error across cross-validation folds (Section 2.3).

3 Results

3.1 The brain's response to stimulation depends on the pre-uStim brain state

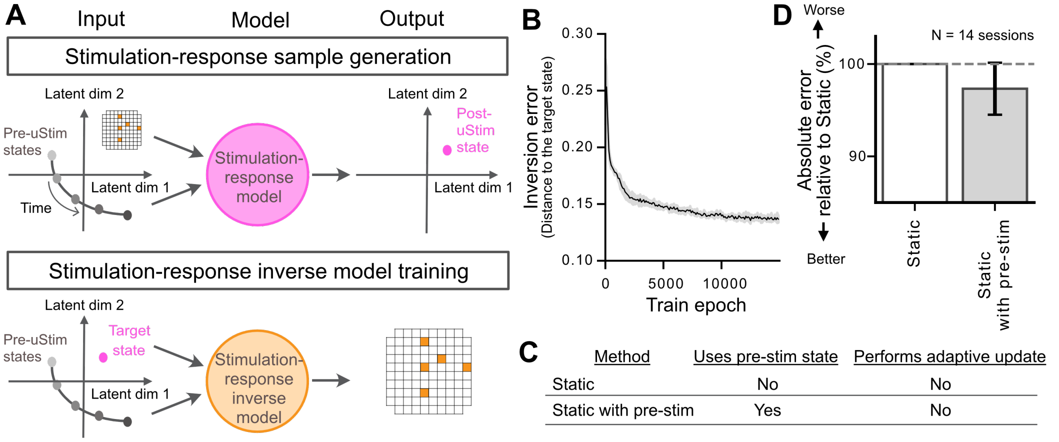
Does neural population activity prior to brain stimulation affect the brain's response to stimulation? To assess this, we measured the impact of the pre-uStim state on the uStim response in a behavioral experiment where different memory conditions (Section S3) were used to create distinct pre-uStim brain states in a low-d space identified using FA (Fig. 2A, left). We observed that uStim shifted neural activity within the low-d space (Fig. 2A, right; see Fig. S3 for additional examples) away from the location where the activity would have been without uStim (cf. Fig. S3, leftmost column, bottom panel). This demonstrates that uStim perturbed the neural population activity. Furthermore, the induced post-uStim activity states appeared to depend on the memory condition (Fig. 2A, right; see Fig. S3 for additional examples). This suggests that brain states induced by uStim depend on the pre-uStim brain state.

To quantify whether knowledge of this state dependency helps improve the prediction of brain responses to uStim, we compared the post-uStim state prediction performance of the stimulation-response model with and without pre-uStim state information (Fig. S1). For a model without pre-uStim state information, we used a CNN which predicted post-uStim states only as a function of the uStim patterns applied (as in [10]), termed "uStim pattern only". For models with pre-uStim state information, we tested a combination of a CNN and a MLP [21] with up to 5 pre-uStim time bins, termed "uStim pattern + Pre-stim{1,3,5} (MLP)", and a combination of a CNN and an LSTM with 5 pre-uStim time bins, termed "uStim pattern + Pre-stim5 (LSTM)". To collect training data to fit these models, we applied randomly-chosen 5-electrode uStim patterns in 6 experimental sessions.

The models with pre-uStim state information (Fig. 2B, four middle bars in light blue) outperformed the model without this information (Fig. 2B, top dark blue bar, $p < 0.05$ for all four models, one-tailed t-test). Among the models with pre-uStim state information, uStim pattern + Pre-stim5 (LSTM) achieved the best prediction performance by incorporating 5 time bins of pre-uStim state information. A model using only pre-uStim state information, termed "Pre-stim5 only (LSTM)" (Fig. 2B, bottom dark blue bar), performed worse than uStim pattern + Pre-stim5 (LSTM) ($p < 0.05$, one-tailed t-test). Thus, both the uStim pattern and pre-uStim state information were important for prediction. The pre-uStim state affects where the activity evolves after uStim, and this can be leveraged to improve the prediction of brain response to uStim.

3.2 A stimulation-response inverse model selects uStim patterns based on the pre-uStim state

OMiSO's goal is to choose a uStim pattern that can create a targeted neural population state. While the stimulation-response model trained in Section 3.1 can predict brain responses to any input combination of a uStim pattern and a pre-uStim state (Fig. 3A, top), they can not do the inverse - start with a desired brain state and find a uStim pattern that will drive neural activity to that state. One



Method	Uses pre-stim state	Performs adaptive update
Static	No	No
Static with pre-stim	Yes	No

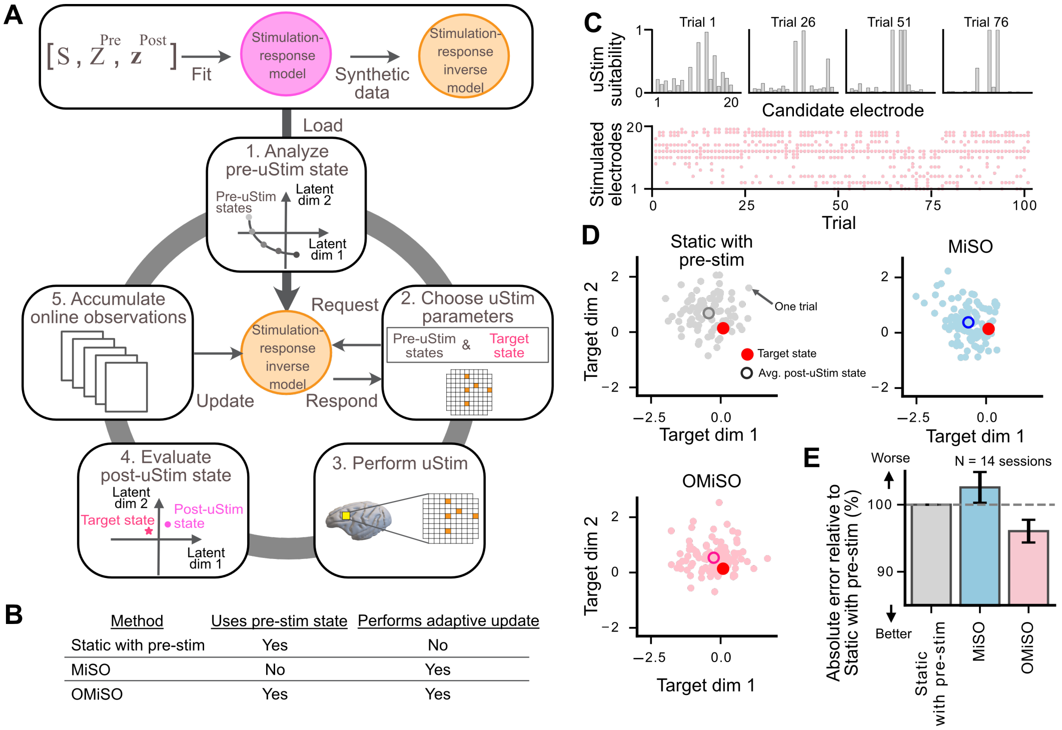
Figure 3: **Inverting the stimulation-response model to choose a uStim pattern based on the pre-uStim state.** (A) Schematic of the inversion procedure. (B) Inversion performance. The error shading indicates the standard deviation across 5 inverse models. (C) Methods used for performance comparison to evaluate the model inversion quality. (D) Absolute error relative to the "Static" method computed in a 2-d target latent space. Error bars indicate ±1 standard error across sessions.

potential approach is to construct a lookup table of predicted post-uStim states for all possible input combinations and find the uStim pattern expected to produce the closest post-uStim state to the target state. However, this approach is infeasible with a continuum of pre-uStim states. Another approach is to mathematically invert the stimulation-response model to choose a uStim pattern. However, this is challenging because the stimulation-response model could map multiple combinations of uStim patterns and pre-uStim states to the same post-uStim state (i.e., the mapping is not one-to-one).

Instead, we approximated the inverse mapping of the stimulation–response model by generating synthetic data (Section 2.4). More specifically, we generated synthetic data using the stimulation-response model, where we considered the model-predicted post-uStim states as potential target states (Section S5). The generated synthetic data allowed us to train the inverse model, which receives a pre-uStim state and a potential target state and returns a suitability of stimulating each candidate electrode (Fig. 3A, bottom). For the stimulation-response model, we used uStim pattern + Pre-stim5 (LSTM), which achieved the best prediction performance (Fig. 2B). For the stimulation-response inverse model, we used an MLP (Fig. S1), since it is computationally fast and could complete all necessary computations within 50 ms (Fig. 1B, computation period) following the observation of pre-uStim states, allowing uStim to be delivered immediately afterward.

To assess inversion quality, we evaluated whether the inverse model could output uStim patterns that induce the appropriate post-uStim neural states (i.e., the post-uStim state predictions). We found that with more training epochs (each of which consists of 15,504 synthetic samples), the inverse model improved at choosing uStim patterns that induced post-uStim states close to the target state (Fig. 3B, Manhattan distance in the 2-d latent space, Eq. 11). This improvement saturated around 100,000 epochs with a distance (i.e., inversion error) of around 0.13 a.u. (error for perfect inversion is 0). For comparison, the trial-by-trial standard deviation in brain states (in no-uStim trials within the same memory condition) was 0.7 a.u.

To understand the impact of imperfect model inversion on achieving target brain states, we experimentally compared the performance of two methods, one with perfect inversion and another that could only be approximately inverted (Fig. 3C). The first method (termed the "Static" method) constructed a lookup table of predicted post-uStim states for all possible uStim patterns (15,504 patterns) using the uStim pattern only model, which does not consider pre-uStim state information. It then found the uStim pattern expected to produce the closest post-uStim state to the target state. Having a complete lookup table implies a *perfect* inversion of the uStim pattern-only model. The second method (termed the "Static with pre-stim" method) used the inverse model of the uStim pattern + Pre-stim5 (LSTM) model trained on the generated synthetic data (which only achieves an *imperfect* inversion). Neither the prediction lookup table nor the inverse model were updated during the experiments, so we refer to them as static methods. To compare the two methods, we performed brain stimulation experiments



Method	Uses pre-stim state	Performs adaptive update
Static with pre-stim	Yes	No
MiSO	No	Yes
OMiSO	Yes	Yes

Figure 4: **Performance of OMiSO in brain stimulation experiments.** (A) Schematic of OMiSO. OMiSO trains stimulation-response and stimulation-response inverse models using previously collected stimulation-response samples (top box). On each uStim trial, OMiSO performs five steps: (1) analyzes the pre-uStim brain state across five time points (gray dots), (2) inputs these pre-uStim states and the target state into the inverse model to choose a uStim pattern (represented as a grid), (3) applies the chosen uStim pattern, (4) compares the resulting post-uStim state (dot) to the target state (star), and (5) updates the inverse model every five trials using the new stimulation-response samples. (B) Methods used for performance comparison. (C) Examples of stimulation suitability for each electrode (top) and selected uStim patterns (bottom) by OMiSO during an example session using the same target state across trials. (D) Induced post-uStim states by each method during an example session (the same session as in panel C). Different colors (gray: static with pre-stim; blue: MiSO; pink: OMiSO) indicate the method used to select uStim patterns. Each small dot shows the post-uStim brain state from a single trial, whereas each large hollow dot shows the mean post-uStim state across trials. The red dot in each panel is the target brain state, shared across all three panels and trials. (E) Absolute error of different methods relative to "Static with pre-stim", computed in the 2-d target latent space. Error bars indicate standard error across sessions.

in which we applied uStim patterns chosen by each method in an attempt to achieve a 2-d target brain state. If the effect of the inversion error on the uStim pattern selection performance is small, the "Static with pre-stim" model, which leverages pre-uStim information, should outperform the "Static" method. We found that the mean error to the target state of "Static with pre-stim" across sessions was smaller than for the "Static" method, although the difference was not statistically significant (Fig. 3D, $N = 14$ sessions, $p = 0.3$, Wilcoxon signed-rank test). This provides support for incorporating the inverse model into an adaptive brain stimulation framework to choose uStim patterns based on pre-uStim state information.

3.3 OMiSO adaptively updates the inverse model during a brain stimulation experiment

During each experimental session, new stimulation-response samples become available. Leveraging these samples may help us adaptively update the stimulation-response inverse model to account for inversion errors, as well as changes in the stimulation-response relationship within and across sessions. To test this, we developed an adaptive brain stimulation framework, OMiSO (Fig. 4A, Section 2.1) where we used a PPO-inspired objective function to adaptively update the stimulation-response inverse model (Section 2.6). We tested OMiSO during brain stimulation experiments and compared its performance against an inverse model that was not adaptively updated ("Static with pre-stim", tested in Section 3.2) (Fig. 4B). OMiSO had limited observations (104 trials on average after the latent state identification trials within a session in 14 experimental sessions) to update the inverse model (after every 5 trials), which could put it at a disadvantage due to overfitting compared to the "Static with pre-stim" method. We also compared OMiSO's performance against our previously proposed method, MiSO [10], which adaptively updates its predictions but does not leverage pre-uStim states. OMiSO adaptively updated the stimulation suitability for each electrode over time (Fig. 4C, top; example session), changing the uStim pattern applied for achieving the same target state (Fig. 4C, bottom). Changes in pre-uStim brain states could also result in different uStim patterns being selected over time, even in a non-adaptive method. However, the induced brain states produced by the adaptive method were closer to the target state than those produced by the "Static with pre-stim" method (for which the inverse model was not adaptively updated) (Fig. 4D). This indicates that the adaptive model updates were essential for improving the accuracy in inducing the target state. Across multiple sessions, OMiSO drove neural activity closer to the target state than the two alternative methods (Fig. 4E, $N = 14$ sessions, $p = 0.05$ for Static with pre-stim and $p = 0.01$ for MiSO, Wilcoxon signed-rank test) by taking into account the pre-uStim state and by adaptively updating the stimulation-response inverse model.

4 Discussion

We propose a brain stimulation framework, OMiSO, which involves two key methodological advances for shaping brain states: incorporation of the pre-stimulation brain state and adaptive updating of the model used to choose a stimulation pattern. In brain stimulation experiments, OMiSO outperformed methods that do not incorporate these advances. While the neural activity used in this study consists of spiking responses recorded with a planar grid of electrodes implanted in the brain, and each stimulation pattern specifies the location of the stimulated electrodes within the grid, OMiSO can be readily applied to other stimulation and recording protocols (e.g., holographic optogenetics [22] or other uStim parameters such as frequency and amplitude).

In comparison to MiSO [10], OMiSO introduces several key technical advances. First, MiSO does not account for the pre-stimulation brain state when predicting brain responses, whereas OMiSO explicitly incorporates it, enabling more accurate prediction of brain responses. Second, during online adaptation, MiSO relies on a tabular approach that stores the predicted post-stimulation state for each candidate stimulation pattern and updates only the entry corresponding to the tested pattern. This approach is sample inefficient as it does not propagate the knowledge from a new stimulation-response observation to improve the predicted response of other stimulation patterns. OMiSO instead employs an inverse model that updates its parameters online, enabling learning beyond the tested stimulation pattern. Finally, the tabular method in MiSO becomes impractical as the size of the stimulation parameter space grows. OMiSO's model-based approach enables better scaling to larger parameter spaces, expanding the search from 4,560 ("96 choose 2 electrodes") stimulation patterns [10] to 15,504 ("20 choose 5 electrodes") stimulation patterns, with improved performance over MiSO.

Although we experimentally demonstrated the advantages of OMiSO, there are scenarios where it might be less beneficial. First, in response to strong stimulation, the brain might be driven to a similar post-stimulation state regardless of the pre-stimulation activity. Second, in clinical applications, neural states can exhibit substantial non-stationarity. For example, in individuals with epilepsy, the neural state distributions might be quite different during periods with seizures (when neurons tend to fire in bursts) versus periods with no seizures [23]. When using OMiSO, one should ensure that the neural state distributions are similar between the training period and online use.

OMiSO can be adapted to both clinical and basic neuroscience applications, in which the goal is to drive neural activity to a target brain state (Eq. 1). In clinical applications (e.g., DBS), the target could be defined as neural activity corresponding to a healthy brain state. OMiSO can then identify stimulation patterns that move neural activity towards this desired state. Alternatively, OMiSO could be extended to drive neural activity away from unhealthy brain states. For basic neuroscience, OMiSO can increase the accuracy with which one perturbs neural activity to elucidate the neural mechanisms underlying sensory, cognitive, and motor function ([5, 24]).

5 Acknowledgments

This work was supported by Japan Student Services Organization fellowship, NIH CRCNS R01 MH118929, NSF NCS DRL 2124066, and Simons Foundation NC-GB-CULM-00003241-05. The authors are inventors on pending International Patent Application No. PCT/US25/47445, which relates to the methods developed in this paper. We are grateful to Samantha Schmitt for assistance with data collection, and to our animal care staff.

References

[1] Bruno B Averbeck, Peter E Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5):358–366, 2006.

[2] Shreya Saxena and John P Cunningham. Towards the neural population doctrine. *Current opinion in neurobiology*, 55:103–111, 2019.

[3] Saurabh Vyas, Matthew D Golub, David Sussillo, and Krishna V Shenoy. Computation through neural population dynamics. *Annual review of neuroscience*, 43(1):249–275, 2020.

[4] R Becket Ebitz and Benjamin Y Hayden. The population doctrine in cognitive neuroscience. *Neuron*, 109(19):3055–3068, 2021.

[5] Marlene R Cohen and William T Newsome. What electrical microstimulation has revealed about the neural basis of cognition. *Current opinion in neurobiology*, 14(2):169–177, 2004.

[6] Marlene R Cohen and Adam Kohn. Measuring and interpreting neuronal correlations. *Nature neuroscience*, 14(7):811–819, 2011.

[7] John H Reynolds and Leonardo Chelazzi. Attentional modulation of visual processing. *Annu. Rev. Neurosci.*, 27(1):611–647, 2004.

[8] David A McCormick, Dennis B Nestvogel, and Biyu J He. Neuromodulation of brain state and behavior. *Annual review of neuroscience*, 43(1):391–415, 2020.

[9] Sina Tafazoli, Camden J MacDowell, Zongda Che, Katherine C Letai, Cynthia R Steinhardt, and Timothy J Buschman. Learning to control the brain through adaptive closed-loop patterned stimulation. *Journal of Neural Engineering*, 17(5):056007, 2020.

[10] Yuki Minai, Joana Soldado-Magraner, Matthew Smith, and Byron M Yu. MiSO: Optimizing brain stimulation to create neural activity states. *Advances in Neural Information Processing Systems*, 37:24126–24149, 2024.

[11] Yuxiao Yang, Shaoyu Qiao, Omid G Sani, J Isaac Sedillo, Breonna Ferrentino, Bijan Pesaran, and Maryam M Shanechi. Modelling and prediction of the dynamic responses of large-scale brain networks during direct electrical stimulation. *Nature biomedical engineering*, 5(4):324–345, 2021.

[12] Arthur Guez, Robert D Vincent, Massimo Avoli, and Joelle Pineau. Adaptive treatment of epilepsy via batch-mode reinforcement learning. In *AAAI*, volume 8, pages 1671–1678, 2008.

[13] Simon Little, Alex Pogosyan, Spencer Neal, Baltazar Zavala, Ludvic Zrinzo, Marwan Hariz, Thomas Foltynie, Patricia Limousin, Keyoumars Ashkan, James FitzGerald, et al. Adaptive deep brain stimulation in advanced parkinson disease. *Annals of neurology*, 74(3):449–457, 2013.

[14] Qitong Gao, Stephen L Schmidt, Afsana Chowdhury, Guangyu Feng, Jennifer J Peters, Katherine Genty, Warren M Grill, Dennis A Turner, and Miroslav Pajic. Offline learning of closed-loop deep brain stimulation controllers for parkinson disease treatment. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, pages 44–55, 2023.

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

[16] Alan D Degenhart, William E Bishop, Emily R Oby, Elizabeth C Tyler-Kabara, Steven M Chase, Aaron P Batista, and Byron M Yu. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nature biomedical engineering*, 4(7):672–685, 2020.

[17] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[18] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[19] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

[20] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

[21] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[22] Hillel Adesnik and Lamiae Abdeladim. Probing neural codes with two-photon holographic optogenetics. *Nature neuroscience*, 24(10):1356–1366, 2021.

[23] Roland D Thijs, Rainer Surges, Terence J O'Brien, and Josemir W Sander. Epilepsy in adults. *The lancet*, 393(10172):689–701, 2019.

[24] Mehrdad Jazayeri and Arash Afraz. Navigating the neural space in search of the neural code. *Neuron*, 93(5):1003–1014, 2017.

[25] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

Supplementary material

S1 Summary of hyperparameters

The table below summarizes the hyperparameter values used in this study. The latent dimensionality m for the FA model was chosen by first computing the optimal dimensionality separately for each session by maximizing the cross-validated data likelihood, then finding the mode of the distribution of optimal dimensionalities across sessions. The number of electrodes used for latent space alignment was chosen to be smaller (by 1 to 8 electrodes) than the number of common usable electrodes to increase the robustness of alignment. For the stimulation-response model training, the learning rate, weight decay, and the number of training epochs (i.e., number of complete passes through the entire training dataset) were chosen based on a grid search. For the model inversion, we did not perform extensive tuning of the hyperparameters, such as a max epoch, since the model yielded similar performance (Eq. 11) even with different hyperparameters. Batch size per epoch refers to the number of synthetic data samples generated for each epoch. The parameters for uStim pattern selection and the PPO-inspired adaptive model update were chosen by running simulations (Section S6). uStim parameters (amplitude, frequency, and duration) were set to avoid causing overt behavioral changes or strong post-uStim activity inhibition across the entire array.

Method	Parameter name	Description	Value
Latent space identification & alignment	m	FA latent dimensionality	5
	n_{stable}	Num. of stable electrodes for alignment	40
Stimulation-response model fitting	R	Num. of training experimental sessions	5
	t	Num. of pre-uStim state time bins	1-5
	l	Target state dimensionality	2
	-	Optimizer	AdamW
	-	Learning rate	0.0006
	-	Weight decay	0.001
	-	Training epoch	20
Model inversion	-	Batch size per epoch	15,504
	-	Max epoch	100,000
	-	Num. of validation samples	1,000
	-	Validation frequency (epoch)	100
	-	Early stop patience (epoch)	20
uStim pattern selection	-	Num. of latent space identification trials	120
	$\varepsilon_{\text{init}}$	Initial value of $\varepsilon_{i,k}$	0.5
	$\varepsilon_{\text{floor}}$	Minimum value of $\varepsilon_{i,k}$	0.1
	γ	Discount factor of $\varepsilon_{i,k}$	0.95, 0.96
Adaptive model update	$\varepsilon_{\text{clip}}$	Clip value of model update	0.15, 0.2
	b	Baseline for advantage computation	0.6, 0.7
	-	Batch size	5
	-	Optimizer	AdamW
	-	Learning rate	2×10^{-5}, 1×10^{-4}
	-	Weight decay	0.001
uStim parameters	u	Num. of uStim candidate electrodes	20
	n_{Stim}	Num. of uStim electrodes	5
	-	Total num. of possible uStim patterns	15,504
	h	Horizontal size of electrode grid	10
	v	Vertical size of electrode grid	10
	-	uStim amplitude (uA)	25
	-	uStim frequency (Hz)	50
	-	uStim duration (ms)	40

S2 Spiking activity preprocessing

To identify latent dimensions of the neural population activity using FA, we computed binned spike counts during the 1.5 s memory period of the first 120 trials recorded in each session (termed the "latent space identification trials"). We used 50 ms bins, yielding 30 bins per trial. The total number of time bins used to fit the FA model on each session was 3,600 (120 trials × 30 bins). The same trials were used to extract a list of usable electrodes $\boldsymbol{e}_i$ for the ith session based on the following three criteria: mean firing rate >1 Hz, Fano factor <8, and <20$\%$ coincident spiking with each of the other electrodes. The FA model was fitted using only the usable electrodes.

S3 Details of uStim experimental paradigm

Experimental procedures were approved by the Institutional Animal Care and Use Committee of Carnegie Mellon University. In each experimental session, the monkey performed a memory-guided saccade task. On each trial, the monkey first fixated on a dot at the center of the screen. After establishing fixation, a target appeared on the screen for 100 ms. This was followed by a memory period, after which the center dot turned off (go cue) and the monkey performed an eye movement to the remembered target location to receive a liquid reward. The fixation duration (i.e., the time from when the monkey acquired fixation to when the go cue was given) was set at either 1.65 or 1.95 seconds for the latent space identification trials and 1.25 or 1.55 seconds for the other trials (randomly chosen, with probabilities 0.8 and 0.2 for the shorter and longer durations, respectively). The location of the target was chosen from four peripheral targets ([45°, 135°, 225°, 315°] or [0°, 70°, 135°, 270°]) on the latent space identification trials and from two peripheral targets ([135°, 315°] or [0°, 70°]) on the other trials. These target directions were chosen based on the mapped receptive fields of the recorded PFC neurons so that diverse brain states could be induced.

There were two types of memory-guided saccade trials: "uStim trials", in which we applied uStim, and "no-uStim trials", in which we did not apply uStim. The experimental system randomly chose which type to perform on each trial. On uStim trials, we applied uStim for 40 ms during the memory period. The stimulation was biphasic with each square pulse in the biphasic pair being 250 μs in duration. We set the current amplitude low enough not to induce any eye movements (25 μA for each electrode we stimulated). On each trial, we changed the locations of $n_{\text{Stim}} = 5$ stimulated electrodes selected among $u = 20$ candidate electrodes (Fig. S2), while keeping other parameter values such as current amplitude (25 μA) and frequency (50 Hz) fixed. The candidate electrodes for stimulation were selected based on three criteria: (1) they were spatially distributed across the array to prevent excessive current concentration when stimulating with multiple electrodes, (2) they were positioned near electrodes with consistent recordings of spiking activity across sessions to ensure reliable evaluation of brain responses to stimulation, and (3) they induced changes in brain activity when stimulated. We selected the u candidate electrodes from the full set of electrodes, rather than restricting selection to the n_{stable} stable electrodes, to allow greater spatial diversity in stimulation configurations.

Although we could have defined the target state in the $m = 5$ dimensional latent space, we instead defined the target state within a subset ($l = 2$) of those dimensions. We made this choice because not all latent dimensions are equally suitable for reliable perturbation of neural activity. For example, the activity in some dimensions might not be easily modifiable by uStim, and some dimensions might not be well-aligned across sessions. Therefore, we chose the two target dimensions based on two criteria: 1) diverse brain states could be induced along these dimensions with uStim (evaluated by visualizing the induced brain states across different dimensions), and 2) they showed good alignment across multiple sessions (as determined by the similarity of the aligned dimensions across sessions, cf. Eq. 3). The dimensions that passed the criteria were not necessarily the top FA dimensions that explained the greatest covariance among the neurons.

To select which uStim pattern to apply, we chose the n_{Stim} electrodes with the highest predicted suitabilities, with $1 - \varepsilon_{i,k}$ probability (Section 2.5). For simplicity, we stimulated using the same number of electrodes ($n_{\text{Stim}} = 5$) when evaluating the performance of OMiSO as during the training sessions. Alternatively, one can use other electrode selection strategies, for example applying a suitability threshold (e.g., 0.8) and stimulating all electrodes whose predicted suitability exceeds that threshold.

For the adaptive model update, OMiSO only used new observations without using a replay buffer (i.e., it performed on-policy model updates), as in [18].

S4 Summary of experimental sessions

We conducted 31 brain stimulation experimental sessions in total. Of these sessions, 7 sessions were used for offline analyses ("Offline analysis sessions"), 10 sessions with randomly selected uStim patterns were used to train the stimulation-response model and the stimulation-response inverse model ("Training sessions"), and 14 sessions were used to test the performance of different methods ("Test sessions"). Among the 14 test sessions, 6 of them were obtained by running 2 test sessions per day for 3 days. The table below summarizes the number of sessions used for each result reported in the manuscript.

Figure	Offline analysis sessions	Training sessions	Test sessions
Fig. 2A	1 session	-	-
Fig. 2B	6 sessions	-	-
Fig. 3, Fig. 4	-	10 sessions	14 sessions

S5 Details of stimulation-response model inversion

To train the stimulation-response inverse model (Fig. S1D) with synthetic data, OMiSO iterates the data generation process and model parameter updates over multiple epochs. In each epoch, OMiSO generates 15,504 synthetic samples, each of which consists of a pre-uStim state $\breve{\boldsymbol{Z}}^{\text{Pre}} \in \mathbb{R}^{l\times t}$, a uStim pattern $\boldsymbol{s} \in \{0,1\}^u$, and a predicted post-uStim state averaged across R stimulation-response models, which is used as the target state $\breve{\boldsymbol{z}}^{\text{Targ}} \in \mathbb{R}^l$. The number of synthetic samples need not match the total number of uStim patterns, although we chose that as our hyperparameter value. To generate one synthetic sample, OMiSO selects one of the experimentally observed pre-uStim states $\breve{\boldsymbol{Z}}^{\text{pre}}_{i,k}$ from the stimulation-response model training data as $\breve{\boldsymbol{Z}}^{\text{pre}}$ and pairs it with a uStim pattern chosen randomly among the 15,504 possible patterns. OMiSO then uses the paired pre-uStim state and uStim pattern to generate a post-uStim state prediction that can be used as the target $\breve{\boldsymbol{z}}^{\text{Targ}}$. With these synthetic data, OMiSO trains a stimulation-response inverse model $\pi_\theta(\hat{\boldsymbol{s}}|\breve{\boldsymbol{z}}^{\text{Targ}}, \breve{\boldsymbol{Z}}^{\text{Pre}})$ by minimizing the binary cross entropy loss between the chosen uStim patterns $\boldsymbol{s}$ and the predicted patterns $\hat{\boldsymbol{s}}$ with respect to θ:

$$\mathcal{L}_{\text{BCE}} = -\textstyle\sum_{v=1}^{u}\left[s_v \log(\hat{s}_v) + (1-s_v)\log(1-\hat{s}_v)\right] \tag{10}$$

where $\boldsymbol{s}$ and $\hat{\boldsymbol{s}} \in \{0,1\}^u$, and s_v and $\hat{s}_v$ are the vth entry of these vectors, respectively. Then the next epoch begins, until the stopping criterion is reached.

To evaluate the inversion quality for the stopping criterion, OMiSO uses 1,000 validation samples, which are separately generated using the R stimulation-response models. The kth validation sample consists of a pre-uStim state $\breve{\boldsymbol{Z}}^{\text{Pre}}_k \in \mathbb{R}^{l\times t}$, a uStim pattern $\boldsymbol{s}_k \in \{0,1\}^u$, and the target state $\breve{\boldsymbol{z}}^{\text{Targ}} \in \mathbb{R}^l$. OMiSO first obtains the predicted stimulation suitability of each candidate electrode using the inverse model, $\pi_\theta(\hat{\boldsymbol{s}}_k|\breve{\boldsymbol{z}}^{\text{Targ}}, \breve{\boldsymbol{Z}}^{\text{Pre}}_k)$. It then chooses n_{Stim} electrodes with the highest predicted suitabilities in $\hat{\boldsymbol{s}}_k$. Using a grid format representation of the selected uStim pattern $\hat{S}_k \in \mathbb{R}^{h\times v}$ (locations of the stimulated electrodes indicated with a 1, and all other electrodes have a value of 0), OMiSO obtains the predicted post-uStim state using the stimulation-response model $f(\hat{\boldsymbol{z}}^{\text{Post}}_k|\hat{S}_k, \breve{\boldsymbol{Z}}^{\text{Pre}}_k)$ where $\hat{\boldsymbol{z}}^{\text{Post}}_k \in \mathbb{R}^l$. Finally, OMiSO evaluates the error between the predicted and target states across all validation samples:

$$\mathcal{L}_{\text{inv}} = \tfrac{1}{1000}\textstyle\sum_{k=1}^{1000} ||\breve{\boldsymbol{z}}^{\text{Targ}} - \hat{\boldsymbol{z}}^{\text{Post}}_k||_1 \tag{11}$$

If the inversion with the synthetic data were perfect, this error would equal 0. OMiSO performs the validation every 100 epochs (instead of every epoch) to reduce the computational cost of inverse model training. This validation frequency is a tunable hyperparameter that can be adjusted based on the available computing resources. This validation performance was used for the early stopping of the model training to avoid overfitting. OMiSO stops the inverse model training when Eq. 11 shows no improvement for 20 consecutive validation performance evaluations.

S6 Simulations to determine the hyperparameter values for the brain stimulation experiments

To set the values of the hyperparameters for the adaptive brain stimulation experiments (listed in the *Adaptive model update* section in Section S1), we ran simulations by using the R trained stimulation-response models (Section 2.3) and the stimulation-response inverse model (Section 2.4). In each simulation, we generated 100 uStim trials with the trained stimulation-response inverse model and one of the R stimulation-response models. We set the target brain state to be one of the target states to be used in the subsequent experiment. On each trial, the simulator randomly sampled one pre-uStim state from the R experimental session datasets and passed it to the stimulation-response inverse model together with the target state. The stimulation-response inverse model returned the stimulation suitabilities for each candidate electrode, and we chose the uStim pattern to "apply" using the method described in Section 2.5. The sampled pre-uStim state and selected uStim pattern were then passed to the stimulation-response model to obtain the post-uStim state. Finally, the pre-uStim state, the selected uStim pattern, and the post-uStim state were accumulated to perform the PPO-inspired adaptive update of the stimulation-response inverse model, as described in Section 2.6.

For each combination of hyperparameters, we performed $R \times T$ simulations (where each of the R stimulation-response models was used to produce post-uStim states for each of T different target states) and computed the average euclidian distance in l-dimensional space between the predicted post-uStim state and the target state over all simulations. We performed Bayesian Optimization (using Optuna [25]) with this error objective and selected the hyperparameter values that achieved the smallest error. The ranges of hyperparameter values explored during Bayesian Optimization are summarized in the table below.

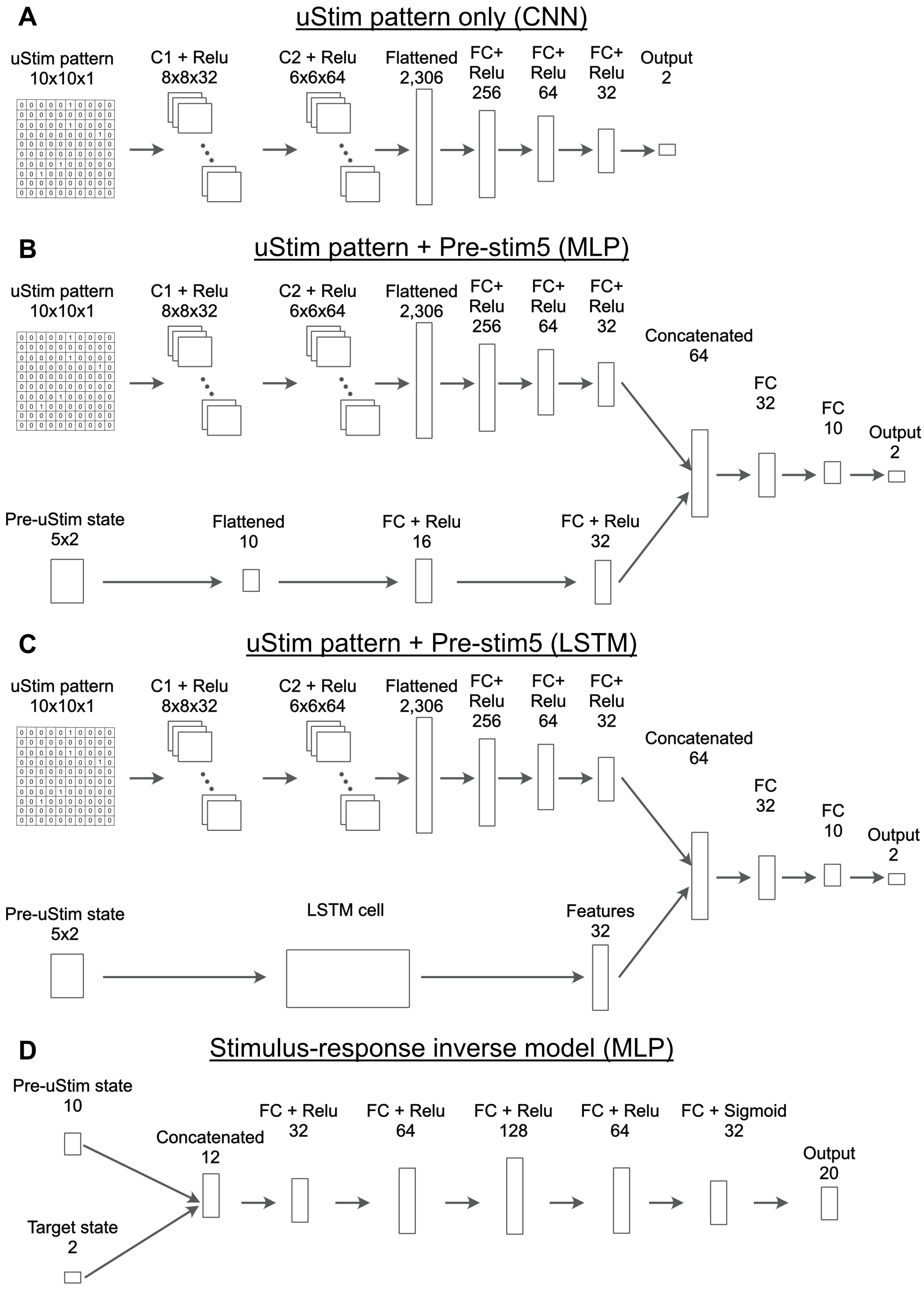
Method	Parameter	Description	Range of values
uStim pattern selection	$\varepsilon_{\text{init}}$	Initial value of $\varepsilon_{i,k}$	0.3-0.7
	$\varepsilon_{\text{floor}}$	Minimum value of $\varepsilon_{i,k}$	0.05-0.15
	γ	Discount factor of $\varepsilon_{i,k}$	0.95-0.99
Adaptive model update	$\varepsilon_{\text{clip}}$	Clip value of model update	0.1-0.3
	b	Baseline for advantage computation	0.5-1.5
	-	Batch size	3-10
	-	Learning rate	0.00001-0.001

S7 Details of model fitting and computing resources

The model architectures used in this study are depicted in Fig. S1. Hyperparameters are listed in Section S1. All models were implemented in PyTorch [26] and fit using the AdamW optimizer. We trained all models on a local computing cluster using 4 NVIDIA GeForce RTX GPUs and 11GB of RAM. The same local computing cluster was used to run PPO-inspired adaptive model updates of the stimulation-response inverse model during uStim experiments. Our experiments also involved three additional computers dedicated to specific tasks: one for experimental trial control (Intel Core i5-4590 @ 3.30 GHz × 4, 15.5 GB RAM, Intel HD Graphics 4600, Ubuntu 20.04.2 LTS), one for neural data recording (Intel Core i7-7700 @ 3.60 GHz, 4 GB AMD Radeon R7 450 + Intel HD Graphics 630, Windows 10 Pro), and one for real-time neural data processing and uStim pattern selection (11th Gen Intel Core i5-11500 @ 2.70 GHz × 12, 15.4 GB RAM, AMD Radeon Pro WX 3200 + Intel Graphics, Ubuntu 20.04.4 LTS). In our OMiSO implementation, the stimulation-response inverse model was adaptively updated every 5 trials using a PPO-inspired objective function on a local computing cluster, which was different from the machines used to run the experimental code. In this way, the adaptive model updates did not interfere with the stimulation pattern selection process, which is time-sensitive.

S8 Code availability

Python code for OMiSO is available on GitHub at https://github.com/yuumii-san/OMiSO.git.



0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Figure S1: **Model architectures.** (A) Stimulation-response model architecture, uStim pattern only [10] (Fig. 2B). C1 and C2 denote convolutional layers, and FC denotes a fully connected layer, with ReLU activations applied throughout. (B) Stimulation-response model architecture, uStim pattern + Pre-stim5 (MLP) (Fig. 2B). (C) Stimulation-response model architecture, uStim pattern + Pre-stim5 (LSTM) (Fig. 2B). (D) Stimulation-response inverse model architecture, MLP (Fig. 3, Fig. 4). The model architectures in (A)-(C) involve the same CNN to process the uStim pattern input.

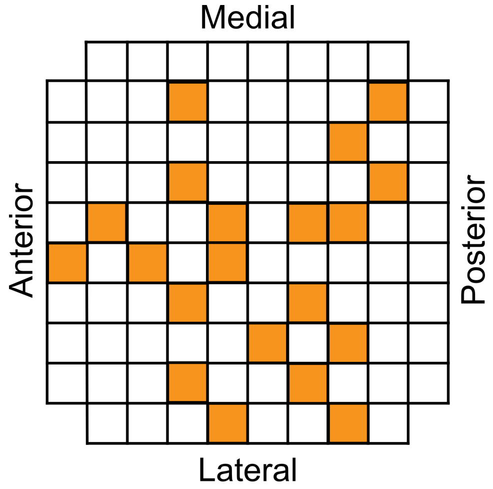


Figure S2: **Candidate uStim electrodes.** The orange cells indicate the locations of the 20 candidate electrodes chosen based on the criteria described in Section S3.

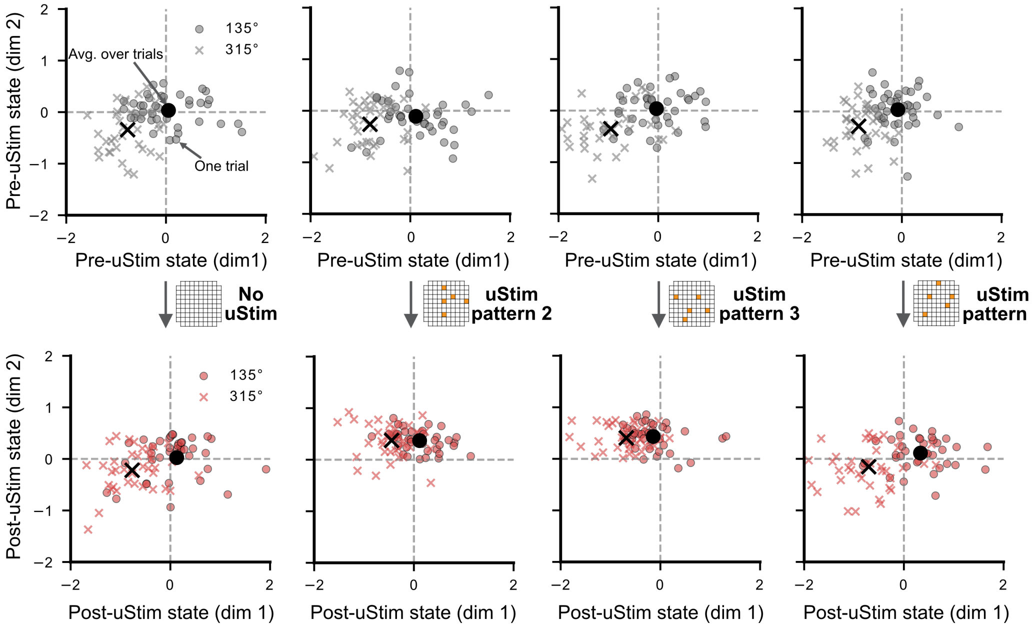


Figure S3: **Pre-uStim states and subsequent post-uStim states induced by example uStim patterns.** Pre-uStim and post-uStim states with additional uStim patterns tested during the same experimental session as in Fig. 2A. The top row shows pre-uStim states, and the bottom row shows post-uStim states. The leftmost column shows brain states from no-uStim trials, while the remaining three columns show brain states from uStim trials with three distinct uStim patterns. Each uStim pattern is represented as a grid, with the stimulated electrodes indicated by orange cells. Same conventions as in Fig. 2A. As in Fig. 2A, uStim shifted neural activity within the low-d space, and the induced post-uStim activity states depended on the memory condition with the degree varying by the uStim pattern.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide a summary of the current literature and knowledge gap, and explain the contributions of our work in the abstract and introduction. Our claims are supported by the experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of this work in the third paragraph of the discussion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose the details of the proposed framework, experiments, data analyses, and modeling approaches needed to reproduce the main results. We also have released the code to reproduce the tested framework on github.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have published the code and sample data needed to reproduce the tested framework on github.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The training and test details are provided in Section 2.3 and Section S7.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We include error bars in all figures and provide the outcomes of statistical tests for the main results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details for the compute resources in Section S7.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed the NeurIPS Code of Ethics and our work is in compliance. All experimental procedures involved in this work were conducted in accordance with the United States National Research Council's Guide for the Care and Use of Laboratory Animals, and were approved by the Institutional Animal Care and Use Committee of Carnegie Mellon University.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential societal impacts in the last paragraph of the discussion section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The data used in this study do not pose a risk for release as it was not acquired from human subjects and does not contain images or identifiable information.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all assets used in this paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The methods section provides the detail necessary to reproduce the proposed framework. We have also published the code and sample data on github.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core components of this study do not involve the use of LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.