Learning to control the brain through adaptive closed-loop patterned stimulation

Sina Tafazoli1,6, Camden J MacDowell1,2,3,6, Zongda Che1, Katherine C Letai1, Cynthia R Steinhardt1 and Timothy J Buschman1,4,5,7
1 Princeton Neuroscience Institute, Princeton University, Princeton, NJ 08540, United States of America
2 Department of Molecular Biology, Princeton University, Princeton, NJ 08540, United States of America
3 Rutgers Robert Wood Johnson Medical School, Piscataway, NJ 08854, United States of America
4 Department of Psychology, Princeton University, Princeton, NJ 08540, United States of America
5 Lead contact and corresponding author
E-mail: tbuschma@princeton.edu
Keywords: brain stimulation, electrical stimulation, machine learning, closed-loop stimulation, neuromodulation

Supplementary material for this article is available online

Abstract
Objective. Stimulation of neural activity is an important scientific and clinical tool, causally testing hypotheses and treating neurodegenerative and neuropsychiatric diseases. However, current stimulation approaches cannot flexibly control the pattern of activity in populations of neurons. To address this, we developed a model-free, adaptive, closed-loop stimulation (ACLS) system that learns to use multi-site electrical stimulation to control the pattern of activity of a population of neurons. Approach. The ACLS system combined multi-electrode electrophysiological recordings with multi-site electrical stimulation to simultaneously record the activity of a population of 5–15 multiunit neurons and deliver spatially-patterned electrical stimulation across 4–16 sites. Using a closed-loop learning system, ACLS iteratively updated the pattern of stimulation to reduce the difference between the observed neural response and a specific target pattern of firing rates in the recorded multiunits. Main results. In silico and in vivo experiments showed ACLS learns to produce specific patterns of neural activity (in ~15 min) and was robust to noise and drift in neural responses. In visual cortex of awake mice, ACLS learned electrical stimulation patterns that produced responses similar to the natural response evoked by visual stimuli. Similar to how repetition of a visual stimulus causes an adaptation in the neural response, the response to electrical stimulation was adapted when it was preceded by the associated visual stimulus. Significance. Our results show an ACLS system that can learn, in real-time, to generate specific patterns of neural activity. This work provides a framework for using model-free closed-loop learning to control neural activity.

1. Introduction

The ability to control neural activity is an important tool for understanding the brain and for treating brain disorders. In neuroscience, causal manipulation of neural activity has been critical for testing hypotheses about how neural activity relates to behavior (e.g. decision making, Salzman et al 1992, Tye et al 2013) and for understanding how neural circuits function (e.g. the role of oscillations, Cardin et al 2010). Clinically, stimulation of neural activity has emerged as a treatment option for neurological diseases: reducing tremors in Parkinson’s patients (Groiss et al 2009), improving mood in patients with severe depression (Mayberg et al 2005), and interrupting epileptiform activity (Zangiabadi et al 2019).

Typically, stimulation acts on the entire population of neurons in the same way (e.g. high vs. low stimulation intensity). However, there is growing evidence that the brain represents information...
in the pattern of activity across the population of neurons. For example, sensory stimuli are represented in the high-dimensional vector of neural activity encoded in a population of neurons in visual cortex (Dicarlo and Cox 2007). Similar high-dimensional representations carry information about the contents of working memory (Rigotti et al 2013) and reward signals (Dabney et al 2020). Given the distributed nature of representations, improving the efficacy of brain stimulation will require precise control over the pattern of neural activity across a population of neurons (Grosenick et al 2015, Mardinly et al 2018).

Here, we present an adaptive closed-loop stimulation (ACLS) approach that learns the pattern of multi-site stimulation needed to generate a specific pattern of neural activity in a population of neurons.

Given the heterogeneous and non-linear nature of the brain, precise control of neural populations requires a stimulation system that can learn an optimal stimulation pattern. Without learning, one must have a priori knowledge of the stimulation pattern that produces a desired response. For some scientific applications, this may be known, such as when replaying a previously observed pattern (using optical or electrical stimulation, Berger et al 2011, Hampson et al 2012, Emiliani et al 2015, Zhang et al 2018, Marshel et al 2019). However, in many scientific applications the stimulation pattern is unknown, such as when wanting to achieve a neural or behavioral state that has not previously been observed. Furthermore, in clinical applications, the stimulation pattern for precisely controlling populations of neurons is unknowable. This is due to two limitations. First, the desired pattern of activity across a neural population cannot be known, as it differs between individuals and is disrupted by disease (which motivates the need for clinical stimulation). Second, the heterogeneity and non-linearity of the brain makes it impossible to predict how a population of neurons will respond to patterned stimulation.

To circumvent these issues, stimulation approaches have begun to incorporate the ability to learn new stimulations patterns. Broadly, learning systems can be divided into two approaches: model-based and model-free. Model-based stimulation relies on learning an accurate model of the input–output relationship between stimulation and downstream neural responses for a given network. Once this model is learned, then one can use it to optimize the stimulation pattern that produces a desired neural response (Ahmadian et al 2011, Choi et al 2016). This approach has effectively predicted the response to stimulation patterns in well-characterized biological neural networks (e.g. sensory thalamus and basal-ganglia pathways; Choi et al 2016, Beauchamp et al 2020) and artificial neural networks (Ahmadian et al 2011). One advantage of such a system is that, once learned, the model can predict the stimulation pattern needed to generate a new, previously unseen, neural response (Choi et al 2016). However, this approach has several disadvantages. First, model-based approaches require detailed knowledge of the target network and its response to stimulation (i.e. they require a large training set; Cho et al 2016). Even with this, models struggle to capture complex, non-linear responses (Bristow et al 2006) and often cannot extrapolate beyond their training set (Hasson et al 2020). These problems are particularly pronounced in complex systems in which large numbers of neurons are controlled (Rao 2019). This complexity is readily seen in stimulation studies. For example, although stimulation of a single site in visual cortex leads to a percept of a phosphene at a specific location (Tehovnik and Slocum 2007), stimulating multiple sites simultaneously does not generate the expected form (Beauchamp et al 2020). This shows how a simple model learned on single site stimulation fails to extrapolate beyond its training set. In this way, the complexity of the brain limits the scientific and clinical application of model-based learning.

In contrast, model-free learning uses optimization algorithms to identify the stimulation patterns that generate a desired response. Such approaches iteratively evaluate the response to stimulation, continuously updating the stimulation pattern in order to learn the optimal stimulation pattern (Madadi and Söffker 2018). In this way, model-free learning requires no pre-existing knowledge of the underlying neural system; enabling stimulation to generalize to unseen states and navigate complex non-linearities. Recent work in the domain of control theory has shown model-free learning can effectively control complex systems (e.g. iterative learning control; Bristow et al 2006), including in silico neural networks (Mitchell and Petzold 2018). Thus, model-free learning may offer a complementary approach to model-based and classic, non-learning, approaches to controlling neural activity.

Here, we describe an ACLS system that combined model-free learning and patterned, multi-site stimulation to control populations of neurons in silico and in vivo. A closed-loop machine learning algorithm iteratively updated the stimulation pattern, allowing ACLS to learn to induce a desired pattern of activity in the neural population. In vivo experiments in awake mice, showed learning was reliable (85.42% success) and typically occurred in ∼15 minutes. Importantly, because the system used a model-free approach, it did not require a model of the neural system and was robust to noise and drift in the response of neurons. In this way, ACLS offers an effective approach for independently adjusting many stimulation sites and flexibly controlling the activity of neurons over time. Such model-free learning approaches will allow us to test open scientific questions relating neural
population representations to cognition and lays the groundwork for improving clinical applications of brain stimulation.

2. Methods

2.1. Implementation of adaptive closed-loop stimulation

An ACLS system was implemented in MATLAB (see supplemental methods for details; available online at https://stacks.iop.org/JNE/11/056007/mmedia). Pseudocode outlining ACLS framework is shown in box S1.

ACLS works by iterating through three steps (figure 1). First, ACLS observes the evoked response to a specific pattern of stimulation (figure 1(A)). Here, we use neural activity as the evoked response. For in silico experiments, the evoked response was the activity of one of the hidden layers of a convolutional neural network (CNN; detailed below). For in vivo experiments, the neural response was the number of spikes from recorded multiunits during a predefined window (i.e. the ‘firing rate’; detailed below). However, the ACLS approach can generalize to other evoked responses, such as local field potentials, physiological markers, or behavior.

Second, the ACLS evaluates an error function that determines how much the evoked response differs from the desired ‘target response’ (figure 1(B)). Here, the error function was defined as the Euclidean distance between the evoked response ($r$) and the target response ($t$):

$$
\varepsilon = \sqrt{\sum_{i=1}^{n} (r_i - t_i)^2}
$$

where $r$ and $t$ are vectors in $N$-dimensional space (where $N$ is the number of measured variables; here, number of simultaneously recorded multiunits). However, ACLS is compatible with any error function (e.g. Euclidean distance, correlation, cosine similarity; see figure S2(A) and (B) for example of ACLS learning with a cosine similarity error function).

Third, ACLS uses a machine learning algorithm to automatically update stimulation parameters to reduce the error function on the subsequent iteration (figure 1(C)). Here, this was achieved using a ‘greedy’ stochastic learning algorithm. On each block of trials, the algorithm generated 5–10 new stimulation patterns by randomly selecting patterns from a distribution around the previous best stimulation pattern (gaussian distribution for in silico, uniform bounded distribution for in vivo experiments). The previous best stimulation pattern was the stimulation pattern that evoked the lowest error during the previous four blocks. During the block of trials, the 5–10 new stimulation patterns and the previous best stimulation pattern were each repeated 5 times. At the end of the block the new best stimulation pattern was chosen, and the process was repeated. By iteratively progressing through these steps, the algorithm learns a stimulation pattern that minimizes the error function and, thus, produces the target response (see box S1 for pseudocode).

To find global minima while avoiding local minima, the spread of the distribution of new stimulation patterns was determined by an ‘annealing factor’, $\lambda$. The value of $\lambda$ changed after each block, according to:

$$
\lambda_{i+1} = \lambda_i \times \epsilon \\
\lambda_{i+1} = 1.118, 
$$

where $\lambda_i$ and $\lambda_{i+1}$ are the current and next $\lambda$ values respectively, and $\epsilon$ is a weighting scalar. If a new ‘best’ (e.g. minimum error) stimulation pattern was found on the current block, then $\epsilon = 1.118$ to increase the search space. If no new best was found, then $\epsilon = 0.8$ to settle to the minima. $\lambda$ was initialized at 3.3 and capped at 10 for in silico and in vivo recordings. For example, given equation (2), the new set of $n$ stimulation patterns for in silico experiments would be:

$$
\mathbf{s}_{n}^{i+1} = \lambda^{i+1} \mathbf{N}(0, 1) + \mathbf{s}_{n}^{best}
$$

where $\mathbf{s}_{n}^{best}$ is the previous best stimulation pattern and $\mathbf{s}_{n}^{i+1}$ are $n$ new stimulation patterns.

ACLS has several parameters, including the number of stimulation patterns generated in each block, the number of repetitions of each stimulation pattern, and the scaling of the annealing factor with learning ($\epsilon$ from equation (2)). In the Results and supplemental text, we explore different values for these parameters. Based on this, we found 5–10 new stimulation patterns per block balanced the speed of learning with the degree of exploration in stimulation space. Increasing the number of stimuli increased the block length, which decreased the rate of learning. This could be problematic if the features of the system being controlled change more rapidly than the algorithm can learn (i.e. it would no longer be able to compensate for drift). However, if the system is stable, then more stimulation patterns will improve the ability of ACLS to explore stimulation space and reliably find a globally optimal stimulation pattern (seen as decreased variance in learning curves when more patterns are used in each block). Thus, the number of repetitions of each stimulus should be chosen to mitigate the impact of noise while avoiding unnecessary resampling (that will slow overall learning). We found averaging over 3–5 repetitions worked well in vivo. We did not systematically explore different scaling factors of the annealing ($\epsilon$ in equation (2)). However, based on sparse sampling, we found setting $\epsilon$ to [0.8, 1.118] was reasonable and allowed for exploration while still settling to a minimum in a reasonable amount of time. Increasing $\epsilon$ would broaden
exploration, while decreasing ε will minimize exploration and allow ACLS to collapse to a local minima more quickly.

2.2. Convolutional neural network simulations

We first tested ACLS in silico on a deep CNN. Recent work has shown that CNNs effectively model the representations and functional transformations of the ventral visual stream (Yamins et al 2014, Bashivan et al 2019, Ponce et al 2019, Walker et al 2019, Bao et al 2020). Thus, CNNs offer a convenient, biologically relevant, in silico benchmark for testing the ability of ACLS to control neural activity.

The CNN had six layers: an image input layer (9 × 9 pixels); three feature extraction layers (each consisting of a convolutional layer, a batch normalization layer, a rectified linear layer, a max pooling layer), a penultimate fully connected layer, and a softmax classification layer (see Supplemental Methods). Unless otherwise noted, the activity of the fully connected layer (e.g. the final hidden layer) was the response used to test ACLS.

For initial in silico tests of the ACLS, CNNs were trained to classify numeric images using the MNIST dataset (7500 images of 0 to 9 digits; 750 samples from each digit; downsampled to 9 × 9 pixels). As expected, CNN performance was high (92% on average).

To investigate the impact of network complexity on ACLS learning rate, a CNN was trained to classify 2, 10, 18 or 26 alphabets from EMNIST dataset (91 650 images of 26 English alphabets; 3525 samples from each letter). To achieve a similar classification accuracy (>87%) for this extended dataset, a second fully connected layer was added before the classification layer. ACLS was used to control the first fully connected layer. For these simulations, white Gaussian noise (µ = 0, σ = 0.5) was added to the response of the first fully connected layer.

2.3. Experimental model and subject details

Adult female (N = 7) and male (N = 2) mice of at least 12 weeks of age were used for awake acute experiments (see Supplemental Methods). A subset of the mice (N = 4) used for awake acute experiments were also used for adaptation experiments. All experiments and procedures were carried out in accordance with the standards of the Princeton University Animal Care and Use Committee and the National Institute of Health.

Surgery was performed under general anesthesia and all animals were provided post-surgical pain management (see Supplemental Methods). To allow for neural recordings, a 3–4 mm diameter craniotomy was centered over visual cortex (2.2 mm lateral and 1 mm anterior to lambda; see Supplemental Methods).

2.4. Acute neural recordings in awake mice

Acute recordings were done 5–28 days after surgery. For recordings, mice were head-fixed in a 1.5 inch diameter, 4 inch long polycarbonate tube. Before experiments, mice were habituated to handling and to the recording setup for 2–4 days to reduce animal stress and movement artifacts.

Figure 1. Schematic of adaptive closed-loop stimulation. Adaptive closed-loop stimulation (ACLS) works by iterating through three steps. (A) First, the system observes the evoked response to a specific pattern of stimulation. Here, we record multiunit neural activity (see Methods), but this could be any measurable variable (e.g. bold signal, local field potential, single units, or behavior). (B) Second, the ACLS evaluates an error function that determines the difference between the evoked response and the desired target response. Here, we use Euclidean distance to a target pattern of neural activity, but this can be any error function (e.g. cosine similarity, correlation) to a target neural or behavioral state. (C) Third, a machine learning algorithm updates the stimulation pattern in order to reduce the error function. By iteratively progressing through these steps, the algorithm learns a stimulation pattern that minimizes the error function and, thus, produces a desired neural or behavioral response. ACLS pseudocode is supplied in box S1.
Stimulation and recording used a 64-site NeuroNexus silicon probe (A2x32-5 mm-25-200-177-A64). The probe consisted of two shanks (200 µm apart). Each shank had 32 electrodes (area of 177 µm²) arranged in a line and separated by 25 µm. All electrodes were activated with iridium oxide. The 32 electrodes on each shank were separated into 16 recording electrodes and 16 stimulation sites, as schematized in figure S5(B). Probes were inserted into primary visual cortex.

Extracellular electrophysiological signals were acquired using an RZ3 processor (Tucker-Davis Technologies, TDT) with a sampling rate of 25 kHz and band-pass filtered from 0.5 to 5 kHz. Electrical stimulation was delivered by an IZ2 stimulator (TDT) and consisted of a series of ten cathode-leading, bi-phasic pulses (200 µs pulse width) delivered at 300 Hz. During learning, the ACLS algorithm only changed the amount of electrical charge delivered to each stimulation site; the stimulation sites and other stimulation parameters were kept constant throughout the experiment.

Action potentials (spikes) were detected by thresholding the filtered electrophysiological signal (see Supplemental Methods). Waveforms were manually inspected by the experimenter and channels with artifacts were excluded from the entire experiment (inspected waveform characteristics included inter-spike-interval, waveform shape, amplitude, etc.; Hill et al 2011). To facilitate real-time, closed-loop control, we did not sort spikes into single neurons. As such, all neural activity is ‘multiunit’ in that it could contain contributions from multiple neurons near the recording electrode.

The multiunit response to electrical stimulation was taken as the number of spikes during a 200 ms window. The window was automatically adjusted for each experiment to maximize information about electrical or visual stimuli (see Supplemental Methods) but always started at least 10 ms after the last stimulation pulse to avoid stimulation artifacts. To ensure our electrically-evoked multiunit responses were physiologically meaningful, we used ACLS to reproduce the multiunit response to visual stimuli. Therefore, only visually-responsive multiunits were used in these experiments. Visual responsive multiunits were defined as carrying ≥0.05 bits of information about the presented visual stimuli (see Supplemental Methods). In total, 5–15 multiunits were recorded in any experiment. Note that, although the target pattern did not include non-visually responsive multiunits and the ACLS did not consider these multiunits in its operation, these multiunits may have been affected by electrical stimulation.

Stimulation was delivered through the stimulation sites closest to the recording electrodes with target multiunits. If a recording electrode had two neighboring stimulation sites, then both were used. Similarly, one stimulation site may neighbor two recording electrodes. Typically, the number of recorded multiunits and stimulation sites were roughly equal. For example, if an experiment used 10 multiunits from 10 recording electrodes, then the set of ~10 closest stimulation sites were controlled by the ACLS. Once the recording electrodes and stimulation sites were defined, they remained constant throughout the experiment.

2.5. Mapping visual response space and identifying a target response

For in vivo experiments, the achievable visual response space was mapped by presenting natural stimuli from the Caltech 101 database (Fei-Fei et al 2004). Between 4 and 20 images were randomly presented in a sequence (stimuli lasted 200 ms, followed by 1000 ms of blank gray screen, each stimulus was presented 20 times; see Supplemental Methods). The response of recorded multiunits to each image created an N-dimensional vector (where N is the number of multiunits, which ranged from 5–15). Principal components analysis (PCA) was used to reduce the dimensionality of this space — the first two principal components (PCs) were used to define the ‘visual response space’.

For each experiment, the ‘target’ of the ACLS algorithm was chosen as the multiunit response to a stimulus. As detailed below in Results, experiments simultaneously learned to produce two different target responses. Targets were chosen to be in different parts of the visual response space, controlling for drift in the state of the network or animal.

2.6. Computing error of evoked multiunit responses

Like visual stimuli, each ACLS-controlled electrical stimulus generated a response vector; taken as the activity of recorded multiunits during a 200 ms window immediately after electrical stimulation (see Supplemental Methods). Response vectors were then projected into the visual response space using the first two PCs (defined by responses to visual stimuli, see above). The error for ACLS learning was taken as the Euclidean distance between the 2D coordinates of the stimulation-evoked response and the target response. This error is termed the ‘reduced dimensional error’. Similarly, error was also computed in the full dimensional space, defined as the Euclidean distance between the stimulation-evoked response vector and the target response in the full N-dimensional space (where N is the number of recorded multiunits).

2.7. Adaptation experiment

Adaptation experiments immediately followed a subset (n = 4) of the awake recording experiments. Probe placement, electrical stimulation, and multiunit recordings were all performed as in awake acute experiments.
Prior to adaptation experiments, ACLS was used to learn two electrical stimulation patterns (ES1, ES2) that reproduced the population-level response of two target responses of visual stimuli (VS1 and VS2; learning was performed as in awake experiments). ES1 and ES2 were taken as the stimulation patterns during the last block of learning with the smallest error to the target response of visual stimuli VS1 and VS2, respectively. Each trial of the adaptation paradigm consisted of two visual stimulus presentations (either VS1 or VS2), followed by electrical stimulation (either ES1 or ES2). This yielded eight different trial types: VS1→VS1→ES1, VS2→VS2→ES1, VS1→VS2→ES1, VS2→VS1→ES1, VS2→VS2→ES2, VS1→VS1→ES2, VS1→VS2→ES2, VS2→VS1→ES2. Given constraints on the duration of awake animal experiments, we limited our investigations to this subset of possible sequence combinations, because they allowed for adaptation to be measured between visual and electrical stimulation and matched previous work (Winston et al 2004). All stimuli were followed by a 100 ms interstimulus interval (ISI). Trials were separated by 4500–5000 ms intertrial interval (ITI). ISI and ITI values were chosen based on previous studies (Jin et al 2019). Each trial type was repeated 30 times and the trial order was random.

The adaptation effect was quantified using an adaptation index, \( I = \frac{R_{\text{same}} - R_{\text{same}}}{R_{\text{same}}} \times 100 \), where \( R_{\text{same}} \) was the average multiunit activity in response to a repeat of the same stimulus and \( R_{\text{different}} \) was the average multiunit activity in response to a different stimulus.

### 2.8. Correcting for drift and quantifying noise

To quantify global drift in the response space over time in the *in vivo* experiments, the initial stimulation pattern of each run was repeated twice per block. Drift was measured as the Euclidean distance between these repetitions and the target response. Because these stimuli were sampled infrequently, the moving mean of this drift was computed across six blocks. The error of these repetitions was not reported to the ACLS algorithm. To correct for this drift, the error of ACLS evoked patterns was divided by the drift for each block.

To quantify noise in the response to electrical stimulation, each stimulation pattern was repeated five times per block during *in vivo* experiments. Noise was computed as the pairwise Euclidean distance between repetitions of the same stimulus, averaged across all stimuli per block.

### 2.9. Quantifying the success of ACLS

Success of each ACLS recording was quantified in three ways. First, a decrease in error from the first to the last block was computed as: \( \Delta e = e_{\text{start}} - e_{\text{end}} \) where \( e \) is the error function (equation (1)) during the first and last block. \( \Delta e > 0 \) was considered a success. Second, a decrease in error over time was determined by fitting the first order exponential, \( f(x) = ae^{\lambda x} + c \) to the error trajectory. A negative value of \( \lambda \) (e.g. a decreasing function) was considered a success.

Third, we tested the ability of ACLS to selectively evoke a population-level neural response by calculating a selectivity index \( s = \frac{e_{\text{run1, target1}} - e_{\text{run2, target1}}}{\text{error in run2}} \) where \( e_{\text{run1, target1}} \) is the error of the current ACLS run to its target, and \( e_{\text{run2, target1}} \) is the error of the other, simultaneous (and independent), ACLS run to the same target. Alternatively, we calculated selectivity as \( s = e_{\text{run1, target1}} - e_{\text{run1, target2}} \) which tested whether a run was closer to its target than the other target (error was drift-corrected to compensate for a global shift towards either target). In both cases \( s < 0 \) was considered a success. These errors captured specificity of ACLS and ensured the reduction in error was not due to a general collapse to a common response space across runs.

### 3. Results

ACLS works by iterating through three steps (figure 1). First, the system observes the evoked response to a specific pattern of stimulation (figure 1(A)). Here, we measure the multiunit firing rates of a population of neurons. However, this response could be any measurable variable, such as distal neural activity, oscillations in the local field potential, blood-oxygenation signals, or behavior. Second, the ACLS evaluates an error function that determines how much the evoked response differs from the desired ‘target response’ (figure 1(B)). Here, we use Euclidean distance to a target response of neural activity, but ACLS could use different error functions to achieve other neural or behavioral states. Third, a machine learning algorithm updates the stimulation pattern in order to reduce the error value (figure 1(C)). By iteratively progressing through these steps, the algorithm learns a stimulation pattern that minimizes the error function and, thus, produces a target response (see Methods for details and box S1 for pseudocode).

#### 3.1. ACLS control of neural representations in silico

We first implemented ACLS *in silico*, testing its ability to control activity within a deep CNN. CNNs have representations similar to those found in visual cortex (Yamins et al 2014, Ponce et al 2019) and neurons in CNNs have many of the properties of neurons in the brain (e.g. non-linearities and complexity of responses). Therefore, controlling CNNs faces many of the same obstacles as controlling the brain, making it a good starting point for testing ACLS.

Here, we used ACLS to control a five-layer CNN trained to classify ten numeric digits from the MNIST database (figure 2, see Methods). ACLS delivered ‘stimulation’ inputs to the input layer of the CNN (i.e. a grayscale image, figure 2(A)). The ACLS learned the amplitude of stimulation input to each neuron in
that layer (i.e. the pixel value, bounded in the grayscale range). The response of the CNN was taken as the pattern of activity in the last hidden layer (i.e. immediately preceding output; the ‘response’ layer in figure 2(A)). Similar to the brain, stimulus representations become more complex along the hierarchy of the CNN (Mahendran and Vedaldi 2016). Indeed, the last layer of the CNN represented the categories...
of stimuli in a distributed manner, across the neural population (figure 2(D)). Given this, we chose to control the activity of the last hidden layer to best approximate the complexities of the brain.

Our goal was to generate specific patterns of neural activity in the last hidden layer. To this end, we began by mapping the response of this layer to 250 images of the numbers 0, 1, 2, and 3 (figure 2(B); 1000 total images). As expected for a distributed representation, the first two PCs captured the majority of the variance in activity of the target layer (68.14%; figure 2(C)). Similar low-dimensional representations are seen in the visual cortex, as shown below and in previous work (e.g. Cohen and Maunsell 2010).

Therefore, to reduce the impact of noise and to aid visualization, we projected the CNN response onto a 2D space defined by the first two PCs (figure 2(D), colored points).

A target response was randomly selected from the set of responses to images of the number ‘1’ (figure 2(D), orange star; note, this choice was arbitrary and other targets could have been chosen without detriment). ACLS learned to produce this target response by minimizing an error function, taken as the Euclidean distance between the response of the CNN to the current stimulation pattern and the target response. Euclidean distance was calculated in the 2D PC space (although similar results were observed in the full dimensional space, as shown below).

To learn to minimize this error function, ACLS used a stochastic learning algorithm (figure 2(E)). On each block of trials, the algorithm generated a set of new stimulation patterns by randomly perturbing the current ‘best’ stimulation pattern (see Methods). Each of these new patterns was then evaluated according to the error function and the stimulation pattern with the lowest error was taken as the new best stimulation pattern (if none of the new patterns improved performance, then the previous best stimulation pattern was kept). This process was repeated for each block, allowing the algorithm to iteratively move closer to a stimulation pattern that minimized the error function. In the example shown in figure 2, ACLS was initialized with a random image of the number 0 from which to generate its first set of stimulation patterns. Note that the ACLS did not receive any information about the response to the images which generated the 2D PC space or about the stimulation that produced the target response. The only input to the ACLS was the random initial stimulation pattern and the error of each stimulation response.

One concern with such ‘greedy’ algorithms is that they are prone to finding local minima in the error function. To reduce the likelihood of this happening, initial blocks of learning used large perturbations, allowing ACLS to broadly explore stimulation space. The magnitude of the perturbations was then slowly reduced over time, allowing the algorithm to identify a locally optimal stimulation pattern (see Methods and box S1). The stochastic learning algorithm was chosen because it does not require a model of the system and because it does not have to estimate the gradient of the error function (avoiding strong assumptions about the nature of the response manifold). This is important, since neural responses may follow complex topologies in neural networks (Dícarlo et al 2012). In this way, the stochastic learning algorithm avoids the obstacles that make it difficult to control complex neural networks.

As seen in figure 2(F), ACLS successfully minimized the Euclidean distance between the stimulation-evoked neural response and the target pattern. As described above, the variance of the stimulation patterns was initially high, as the algorithm explored the broad stimulation space (figure 2(G)). Over time, the variance decreased, allowing the algorithm to settle on the (locally) optimal stimulation pattern. After 50 blocks of stimulation, ACLS discovered a stimulation pattern that was 92.32% closer to the target than its initial starting point in this example (figure 2(H); 2500 total trials, 50 trials in each block, consisting of 5 repetitions of 10 stimulus patterns, see Supplemental Methods). This was consistent across replications of the experiment (91.51% ± 2.57% STD closer to target, N = 50). Despite learning in a reduced dimensional PC space, ACLS also reduced distance in the full dimensional space, moving 48.25% closer in this space (figure 2(I); 53.67% ± 17.01% STD for N = 50 replications). When ACLS minimized error in the full dimensional space (rather than the reduced dimensional space), it moved 80.87% closer in this space (figure 2(I), dotted orange line; 83.08% ± 2.76% STD for N = 50 replications). Interestingly, the stimulation patterns identified by ACLS were classified by the CNN as belonging to the target number (figure 2(J)). This categorization was not part of the error function but reflects the overlap between the stimulation evoked response and the original stimulus-driven response.

It is important to note that ACLS is not necessarily re-discovering the initial input that generated the target pattern (figure S1). Instead, ACLS is often discovering new stimulation patterns that produce the same response in the last hidden layer. Such ‘adversarial images’ are a known phenomenon in deep CNNs and reflect the convergent nature of the CNN (Szegedy et al 2014). The goal of the CNN is to map varied inputs onto similar representations and so it is not surprising that the algorithm is identifying one of these alternative stimulation patterns. Indeed, if ACLS is tasked with controlling earlier hidden layers (e.g. convolution layer 1 or 2) the discovered stimulation pattern closely resembles the initial stimulation pattern (figure S1). This suggests the adversarial images are not a by-product of the ACLS approach but reflect the convergent nature of CNNs.

ACLS was robust to variability in neural responses. Previous work has shown neural responses
Figure 3. ACLS is robust to noise in the network, drift in the state of the network, and the complexity of the network. (A) Traces showing the impact of different noise magnitudes on the repeated presentation of the same stimulus pattern. On each trial, white Gaussian noise ($\mu = 0$, $\sigma = 0, 0.5, 1, 3, \text{ or } 5$) was added to the response of the final hidden layer (equates to an average SNR of $\sim \infty, 18, 9, 3, \text{ and } 2$, respectively). Traces show the block-averaged error of evoked response to the same stimulation pattern across blocks (e.g. the initial pattern given to ACLS; $N = 2$ repetitions per block). $N = 50$ replications for each noise level. (B) Noise in response impaired ACLS learning. Noise levels match those in A. Repeating stimuli mitigated the impact of noise: light, medium, and dark gray lines show the error across blocks with 1, 5, and 10 repetitions of each stimulus when noise was held constant at $\sigma = 3$. Lines show the median block-averaged error of evoked response, normalized to the average error of the first block ($N = 50$ runs for each noise level). Shaded regions show inter-quartile range (IQR). (C) As in A, but shows how drift in the weights of the fully connected layer affects the response to the repeated presentation of the same stimulus pattern. On each trial, white Gaussian noise was accumulatively added to the weights of the fully connected layer ($\sigma$ of noise distribution was 0, 0.93, 9.3, or 93% of the initial standard deviation in weights). $N = 50$ replications for each drift level. (D) As in B, but shows the ACLS can compensate for most levels of drift. Colors match drift levels in C. (E) Bar plot showing that increasing the number of image classes to be classified by the CNN increased the dimensionality of the final hidden layer. Dimensionality was measured as the number of PCs needed to capture $\geq 90\%$ of the variance. (F) As in B and D, but shows increasing the complexity of the representational space slows ACLS learning, yet does not prevent learning. Colors match image classes in E.

are highly variable, possibly due to a combination of random noise and drift in the state of the animal (Calhoun et al 2019). Here, we show ACLS is robust to both random noise and systematic drift. As seen in figures 3(A) and (B), ACLS was robust to random noise. Noise was modeled by adding white noise to the response to a stimulation pattern (figure 3(A)). At low levels of noise (a signal to noise ratio, SNR $\approx 18$) ACLS learned at approximately the same speed as when there is no noise (figure 3(B), purple and green lines, respectively). Intermediate levels of noise (SNR $\approx 9$; yellow line) slowed the speed of learning,
but ACLS still converges on a stimulation pattern that approximates the target response. Finally, high levels of noise (SNR < 4, gray and blue lines), disrupted learning significantly. This is to be expected, as the same stimulation led to drastically different responses from the CNN. In a high noise environment, averaging the responses across repeated stimulation helped to compensate for the impact of noise (figure 3(B), light, medium, and dark gray lines show SNR ≈ 3 but with 1, 5, and 10 repetitions, respectively). This makes sense, as random noise averages to zero, but comes at the cost of slowing learning as each stimulation must be repeated. However, it suggests averaging the response to repeated stimulation can help ACLS learn in a high-noise environment, something we take advantage of when controlling neural representations in vivo (detailed below).

ACLS was also able to compensate for drift in the responses of the CNN. Drift was modeled by changing the weight matrix of the fully connected layer within the CNN (weights drifted by 0%–93% per trial). Changing the weight matrix meant the same input (i.e. an image of a ‘0’ or ‘1’) led to a different response over time (figure 3(C)). Despite this drift, ACLS was able to learn to generate the target pattern, unless drift levels were such that the network was highly unstable (figure 3(D)). This is an advantage of the learning nature of ACLS; as the system state shifts, ACLS can learn to compensate for changes in the response.

Next, we investigated how the complexity of the CNN impacted the ability to learn. To manipulate complexity, we trained a new network to classify either 2, 6, 10, or 26 different letters (from EMNIST database, see Methods). Despite changes in the read-out layers, all other characteristics of the network remained the same. As seen in figure 3(E), increasing the number of stimuli to be classified increased the dimensionality of the hidden target layer (as measured by the number of PCs necessary to explain ≥ 90% of the variance in activity). In all cases, ACLS was able to learn to control the CNN. However, learning speed was slower in higher dimensional networks, suggesting more time may be needed to learn in complex networks (figure 3(F); all ACLS parameters were held constant).

Finally, as noted above, one of the strengths of the ACLS approach is that it can use different error functions to achieve different goals. To show this, we used ACLS to learn to control CNN activity with an error function that increased the cosine similarity between the evoked response and the target response. Learning with this alternative error function was similar to previous results (figures S2(A)-(B)), suggesting ACLS can be used as a general tool for minimizing different error functions.

Altogether, our in silico simulations showed ACLS can learn to control a complex neural network. It was robust to noise, shifts in neural state, changes in network complexity, and could minimize multiple cost functions.

3.2. ACLS control of neural representations in vivo

Next, we tested the ability of ACLS to control neural populations in vivo. ACLS was able to produce neural responses in both anesthetized and awake animals. Here we focus on awake animals as they provide the more complete test of ACLS; details of the anesthetized results are provided in the supplemental information (figures S3(A) and S4).

As with the in silico experiments, we were interested in testing whether ACLS could produce ‘natural’ neural responses. To this end, we began by mapping the response of visual cortex neurons to different visual stimuli (each presented for 200 ms, see Supplemental Methods and figure S5(A)). Both electrophysiological recording and electrical stimulation were done through a single silicon probe inserted into primary visual cortex (V1; 32 electrodes for recording, 32 sites for stimulation, see figures 4(A) and S5(B)-(C) and Methods). This allowed us to record from small populations of neurons (5–15, all multiunit activity, see Methods), while stimulating at nearby sites. The response of a multiunit to a visual stimulus was taken as the number of spikes during a 200 ms window after stimulus onset. The precise timing of the window varied across recordings to best capture the evoked response but was typically 40–240 ms after stimulus offset (see Methods). Multiunits that were not selectively responsive to the visual stimuli were excluded from further analysis, including ACLS learning (see Methods). The response of selective multiunits was then used to define a population response vector to each stimulus (see Methods).

Figure 4(A) shows an example recording in which an animal received 20 repetitions of 10 different isoluminant visual stimuli, resulting in 200 stimulus responses (figure 4(B), response vector is across 7 multiunits). To define the response space of the recorded visual cortex multiunits, we used PCA to create a 2D space that captured the majority of variance in responses (73.53%, figures 4(C) and (D); across N = 24 recordings the first two PCs captured 81.54% ± 1.77% SEM of the variance). Projecting responses into this reduced dimensionality space mitigated the effect of noise and emphasized the effect of visual stimulation. It also allowed us to define the space of ‘achievable’ patterns of neural responses.

Once the visual response space was defined, we used ACLS to learn electrical stimulation patterns that could reproduce the neural response to visual stimuli. Electrical stimulation was delivered through 32 dedicated stimulation sites (see figure S5(B) for example map of stimulation sites on the probe). Although 32 stimulation sites were available, several of them fell outside of cortical tissue. So, stimulation was limited to sites that were local to recording
Figure 4. ACLS can control neural representations in visual cortex of awake mice. (A) Experiment schematic. Awake, head-fixed mice were presented with visual stimuli (from Fei-Fei et al 2004) while recording multiunit neural activity in the primary visual cortex (V1). Visual stimuli were 200 ms in duration, interleaved with 1000 ms presentations of a gray screen. Black trace shows example raw trace of visual stimulus-evoked spiking activity (red dots). (B) Example multiunit neural activity response to the purple (top) and orange (bottom) outlined images in A. (C) Distribution of responses to 10 different visual stimuli, each presented 20 times (gray dots indicate response to one stimulus). Purple and orange stars show the target response to the purple and orange-outlined stimuli. Shaded regions denote STD of 20 visual stimulus presentations of each stimulus. (D) Explained variance in neural activity captured by the principal components. (E) Schematic of ACLS learning to reproduce visual stimulus-evoked neural responses. Electrical stimulation was applied with a gray screen to match baseline visual input. An example stimulation pattern across seven stimulation sites is shown in purple. Black trace shows example raw trace of stimulation followed by spiking activity (red dots). (F) Example time course of learning. Orange and purple stars indicate the two target responses (same as B and C; dotted lines show STD of visual response). Targets were learned by two simultaneous, but independent, runs of the ACLS algorithm. Orange and purple lines show the 2D trajectory of learning over blocks. Shaded region indicates SEM of responses in that block (N = 12 blocks, each with five repetitions of six unique stimulation patterns). Individual purple and orange points show response to repetitions of the initial stimulus pattern (one repetition per block); transparency scales with time (light → dark). The movement of these points shows drift in the system over time. Black dotted line denotes borders of space shown in C. Inset shows region denoted by gray box. (G-H) Error between ACLS-evoked responses and target response across blocks in (G) 2D response space and (H) full dimensional space. Orange and purple lines show mean error per block (Euclidean distance; N = 30 stimulations); shaded regions show SEM. Dotted black line denotes noise floor, quantified as the average pairwise Euclidean distance between repetitions of the same stimulation pattern (N = 5 repetitions of six unique stimulation patterns per block). Dashed purple and orange lines show drift in the response of the initial stimulation, measured as the distance to the target response. Lines for initial stimulation repetitions and noise lines reflect sliding window mean of six blocks. (I-J) Error over time, corrected for drift of initial stimulation response over blocks (see Methods). As in G-H, solid lines show error of learned responses across blocks and dashed lines show the corresponding noise floors (also corrected).

electrodes with a multiunit response and, therefore, were likely in or near cortex (typically 6–16 stimulation sites; see Supplemental Methods for exclusion criteria). Electrical stimulation patterns were defined as the vector of stimulation amplitude across the set of stimulating sites. Each site delivered a specific amount of charge (between 0–10 nC), that was updated by the ACLS system from trial to trial. All stimulations were delivered in a sequence of ten cathode-leading, high-frequency bi-phasic pulses (200 µs pulses delivered at
300 Hz). The neural response to electrical stimulation was measured in the same window as visual stimulation (e.g. 200 ms window starting 40–240 ms post stimulation offset), which avoided artifacts from the electrical stimulation (see supplemental methods and figure S5(C) for schematic of spike detection).

Before learning, two target responses were manually selected from the visual stimulus response space (figure 4(C), orange and purple stars). Selection was done blind to the exact response pattern, but targets were chosen such that they were separated in the neural response space. The ACLS algorithm then learned to produce each target response by reducing the error between the evoked response and the target (see figure S6 for example stimulations and responses along the learning trajectory). As for in silico experiments, the error function was taken as the Euclidean distance in 2D PC space between the evoked response and the target response. The ACLS system had no a priori knowledge of the stimulation space or the relationship between stimulation and responses. ACLS was only provided with (a) a randomly generated initial stimulation pattern (figure 4(E); each site was randomly drawn from a uniform distribution, bounded to prevent tissue damage), and (b) the error of each evoked response. The two selected targets were learned simultaneously by two independent runs of the ACLS algorithm (alternating randomly between the two targets; no information was shared between runs).

As with the in silico experiments, ACLS learned to produce the target pattern of neural responses. Figure 4(F) shows ACLS learning during our example session (see supplemental information and figure S7 for additional examples). Across 12 blocks, the evoked responses systematically moved through the 2D PC space, moving 79.90% and 95.20% closer to their respective targets (i.e. orange towards orange and purple towards purple; 30 trials are in block, consisting of 5 repetitions of 6 stimulus patterns, see Methods). Accordingly, the error between the evoked response and the target response decreased over time (figure 4(G); orange and purple solid lines, shading denotes SEM). Although the system was attempting to minimize error in the 2D PC subspace, error was also reduced in the full dimensional neural space (figure 4(H)). Stimulation occurred approximately every 2.1 seconds, resulting in a total learning duration of ~12.5 minutes for each run (~25 minutes in total for the two simultaneous runs).

As noted in our analysis of the in silico model, both noise and drift in neural responses influence the ability of ACLS to learn to evoke the target response. Here, noise was estimated by the average pairwise distance between the neural response to repetitions of the same stimulation pattern. This estimate of noise is shown in figures 4(G)-(H) (black line). Such noise can impact ACLS in two different ways. First, it can lead to noise in the estimate of the error of a particular stimulation pattern (as shown in silico, figure 3(A)). To compensate for this, ACLS averaged over five repetitions of each stimulation pattern. Second, the noise in the response also places a fundamental limit on how close ACLS can get to the target response pattern — it is impossible to get closer than the ‘noise floor’. This can be seen in figures 4(G)-(H); the error between the evoked response and the target response reduces across blocks but asymptotes close to the noise floor (figures 4(G)-(H), black dashed line).

Drift in the system is reflected in a systematic change in the response to the same stimulation over time. Such changes could reflect adaptation to stimulation, movement of the probe, or changes in the animal’s physiological state (i.e. arousal). To understand the nature of drift in vivo, we repeated a fixed set of ten stimulation patterns in an anesthetized mouse over a period of 1 hour. As seen in figure S8, the response to the same stimulation pattern changed dramatically over time, suggesting drift can be a significant issue. Interestingly, the magnitude of drift was highly correlated across stimulation patterns, suggesting drift reflects a global state change, rather than pattern-specific changes.

It is important to account for drift when evaluating the success of ACLS, as drift could either be towards or away from a particular target. Therefore, to measure and correct for drift, we repeatedly delivered the initial stimulation pattern for both targets throughout the recording (note: the ACLS algorithm was not aware of these stimulations or the responses). In our example recording, the evoked response to the initial stimulation patterns drifted over time (figure 4(F), orange and purple dots), which decreased the ‘error’ of the initial stimulation pattern relative to the target responses (figures 4(G)-(H), orange and purple dashed lines). To correct for this drift, we calculated the ‘drift-corrected error’ of each response by dividing the error of each response by the ‘error’ of repeats of the initial stimulation pattern (see Methods). As seen in figures 4(I)-(J), even following this correction, ACLS is able to reduce the error between the evoked and target response in both the reduced and full dimensional spaces.

A decrease in drift-corrected error was observed for the majority of recordings. Figure 5(A) shows the error traces of all ACLS recordings (48 ACLS runs, across 24 recordings in 9 animals), colored by whether ACLS successfully reduced error. Across all successful runs, ACLS decreased the Euclidean distance by 36.91% ± 4.90% SEM by the end of the 5th block and by 45.57% ± 3.75% SEM by the end of the 10th block (often approaching the noise floor, figure 5(B)). To quantify learning of each run, we measured the difference in error between the evoked and target response on the first and last block of learning. Over 48 total runs, this error was reduced in 85.42% of the runs (figure 5(C); 41/48 runs, $p = 10^{-7}$, binomial test vs. 50% chance). Similarly, we can quantify the
Figure 5. Quantifying ACLS success in awake animals across multiple sessions. (A) Corrected error over time for all ACLS runs (N = 48) in all recording sessions (N = 24), across all mice (N = 16). Display follows figure 4(I). Individual orange and gray lines show error over time of individual successful and failed ACLS runs, respectively. Runs were considered successful if the mean error during the final block was less than the first block. Red line shows mean trajectory across successful runs. For visualization, the x-axis truncated at 15 blocks (N = 6 runs were longer than 15 blocks, all continued to decrease in error). (B) Pairwise comparisons of drift-corrected error (orange dots) and noise (black dots) at blocks 1, 5 and 10 across all successful ACLS runs (N = 41). ∆N is the mean difference between the error and the noise of each run (i.e. gray lines). (C) Table of success rate of the ACLS runs across multiple sessions, quantified by a reduction in error over learning (left column), a negative exponential fit to the learning curve (middle column), and specific learning towards one target (relative to the other; right column). See text and Methods for details of the statistics.

change in error with learning by fitting an exponential curve to the error over the blocks in each run ($f(x) = ae^{\lambda x} + c$, see Methods). Using this metric, 81.25% of the runs successfully reduced error ($\lambda < 0$ for 39/48 runs, $p = 10^{-6}$, binomial test). In the full dimensional space, ACLS reduced error on 81.25% of the runs (quantified as decreased error in the last block compared to the first; 39/48, $p = 10^{-6}$, binomial test). Although ACLS learned to reduce the Euclidean distance between the evoked and target neural patterns, it also improved other metrics of similarity between the evoked and target responses. For example, the angular distance ($d = \arccos \left( \frac{A \cdot B}{\|A\| \|B\|} \right)$) decreased in the reduced dimensional space in a significant majority of runs (figure S9, 36/48 runs, 75.00%, $p = 10^{-4}$, binomial test; with a trend towards significance in the full dimensional space: 29/48 run, 60.42%, $p = 0.097$, binomial test).

One concern is that these success metrics may reflect ACLS learning to navigate from the initial stimulation pattern to the ‘cloud’ of visually-evoked responses, rather than learning to produce a specific pattern of firing activity. To test the ability of ACLS to selectively evoke population-level neural responses, we calculated whether a given run ended closer to its target than the other simultaneous run (e.g. $\varepsilon_{\text{run1, target1}} - \varepsilon_{\text{run2, target2}} < 0$; drift-corrected to compensate for a global shift towards either target). Using this metric, 68.75% of runs were selective for their own target in both the reduced and full dimensional space (33/48, $p = 0.0066$ by binomial test for both spaces). One may expect selectivity values to be lower than overall success rates since they are more susceptible to high noise levels blurring discriminability between the targets. Indeed, logistic regression found the signal-to-noise ratio of the run was predictive of selectivity ($p = 0.018$, see Methods). Supplemental experiments further validated the selectivity of ACLS in anesthetized experiments; it could selectively navigate between two electrically-evoked neural responses (figures S3 and S4; see supporting information).

3.3. Using adaptation to test the physiological relevance of electrically evoked patterns
As noted above, our goal was to test whether ACLS could produce ‘natural’ neural responses. So far, we have measured similarity between the visually-evoked and electrically-evoked as the Euclidean distance in neural space. One concern is that this may result in
neural responses that are not physiologically meaningful, either because Euclidean distance may not be an accurate measure of response similarity or because learning occurred in a reduced dimensional space (even though error in full dimensional space was also reduced).

Therefore, to test the physiological similarity between electrically-evoked and visually-evoked responses, we tested whether electrically-evoked responses adapted to visual stimuli. Previous work has shown that the neural response to a visual stimulus is reduced when the stimulus is repeated (Jin et al 2019). This ‘adaptation’ of the neural response has been used to measure the similarity of neural responses to visual stimuli (Leopold et al 2001). Here, we used this approach to test if neural responses to electrical stimulation were physiologically similar to visually-evoked responses by testing if the response to electrical stimulation was adapted by the presentation of the associated visual stimulus.

To this end, we first mapped the visual response of multiunits and then used ACLS to learn stimulation patterns that generated neural responses similar to two different visual stimuli (as in figures 4 and 5). Specifically, electrical stimulation pattern ES1 and ES2 were taken as the stimulation patterns with the smallest error to visual stimuli VS1 and VS2, respectively (from last block of learning). To ensure the ACLS was able to reproduce the visual response, we only included the experiments in which ACLS
successfully decreased the error towards both targets (N = 4 total sessions).

Next, we tested the effect of adaptation on both visually-evoked and electrically-evoked responses. Animals were presented with a sequence of three stimuli; two visual stimuli, followed by an electrical stimulation (figure 6(A); see Methods). The sequence always started with visual stimulus 1 (VS1) or visual stimulus 2 (VS2). This was followed by either the same stimulus (e.g. VS1→VS1) or the other stimulus (e.g. VS2→VS1). Finally, the electrical stimulation could either match the second visual stimulus (e.g. VS1/2→VS1→ES1) or be different (e.g. VS1/2→VS2→ES1). Figure 6(B) shows the effect of adaptation for two example multiunits. As expected, repeated presentation of the same visual stimulus led to adaptation, reflected by a reduction in firing rate to the repeated stimulus (multiunit 1: 15.16% in 100–150 ms and 9.47% in 150–200 ms, figure 6(B), upper-left; multiunit 2: 7.94% in 100–150 ms and 9.83% in 150–200 ms, figure 6(B), bottom left). The response to ACLS-learned electrical stimulation patterns was also adapted when preceded by the matching visual stimulus (multiunit 1: 4.79% in 100–150 ms and 0.84% in 150–200 ms, figure 6(B), upper-right; multiunit 2: 20.13% in 100–150 ms and 54.21% in 150–200 ms, figure 6(B), bottom right).

Adaptation was consistently observed across sessions and multiunits (figure 6(C); N = 4 sessions, N = 26 multiunits). The evoked response to a repeated visual stimulus was reduced by 5.04% in the 50–150 ms following stimulus onset (p ≤ 5.5*10^{-8} in 50–100 ms and p ≤ 5.6*10^{-8} in 100–150 ms, t-test with Bonferroni-corrected p-value, figure 6(C)). Similarly, the response to electrical stimulation was reduced by 3.26% in the 150–200 ms following stimulus onset if it was preceded by the matching visual stimulus (p = 0.0092, t-test with Bonferroni-corrected p-value). Note that this effect followed the time course of the response to electrical stimulation (and not visual stimulation). This is to be expected, as adaptation is modulating the response to visual/electrical stimulation, not reshaping its time course. Together, these results suggest the neural responses evoked by ACLS are physiologically similar to natural neural responses.

4. Discussion

Here we present an ACLS system that can learn to produce specific neural responses. The system uses a stochastic learning algorithm to reduce the error between the evoked neural response and a target neural response. Importantly, the model-free learning algorithm allows the ACLS to be agnostic to the underlying structure of the neural circuit and avoids needing initial knowledge of how neurons respond to electrical stimulation.

We show ACLS learns to generate target responses in both in silico and in vivo neural networks (figures 2, 4, 5 and 6). In awake animals, learning was successful for the majority of the recording sessions (∼80% of runs reduced their error). The error between the initial and target pattern was typically reduced by ∼50% (often approaching the noise floor). This reduction was achieved within ∼15 min (∼10 blocks with ∼300 total stimulations).

In addition to reducing the error in ‘neural space’, we found electrically evoked responses were physiologically similar to natural responses evoked by visual stimuli. This was seen in the adaptation of electrically-evoked responses to a preceding, matching visual stimulus. Importantly, ACLS was robust to noise and drift in the state, as shown in both artificial and real neural networks.

Here we present the ACLS using electrophysiological recordings of neural activity and electrical stimulation. However, the ACLS framework is flexible. It can be easily adapted to work with a variety of stimulation approaches (e.g. electrical, optical, or magnetic), a variety of ways to measure the response of the brain (e.g. single units, bold signal, local field potentials, or behavior), and a variety of error functions to achieve a desired neural or behavioral state. We believe this flexibility will allow ACLS to have several unique scientific and clinical applications.

4.1. Scientific applications of ACLS

ACLS complements existing model-based and non-learning brain stimulation approaches. In comparison, ACLS allows for (1) improved control of neural populations and (2) optimization of any error function. Both of these characteristics will enable new scientific applications.

Recent experimental work has shown information in the brain is represented in a distributed manner, across the neural population (Yuste 2015, Saxena and Cunningham 2019). These population responses are thought to reside on a low dimensional manifold (Cunningham and Yu 2014), with the position of neural activity on this manifold representing the value of a cognitive variable (e.g. visual perception, Dicarlo and Cox 2007). However, this hypothesis is difficult to causally test because the stimulation patterns that drive activity to specific locations in this manifold are challenging to know a priori. Similarly, a model of the system requires detailed knowledge of single cell activity, population level encoding, and behavioral output. The ACLS could address these concerns by actively learning stimulation patterns that evoke different neural responses along the manifold. This would allow one to test how cognitive variables evolve over the manifold (as suggested by Jazayeri and Afraz 2017). In particular, because ACLS can learn to reproduce any achievable neural response, one could directly compare the behavioral response to neural activity that is either on or off the
manifold (Sadtler et al 2014, Wärnberg and Kumar 2019).

In addition, the ACLS approach is well-suited to using electrical stimulation to change other characteristics of neural activity. This simply requires changing the error function. For example, an error function based on the oscillatory synchrony between neural populations would let ACLS learn stimulation patterns that synchronize neural populations, allowing one to test the role of synchrony in cognition (Danzl et al 2009, Fries 2015, Helfrich et al 2018, Knudsen and Wallis 2020). This highlights the power of a model-free approach—this is feasible with ACLS despite the fact that we do not have an accurate generative model of synchrony.

4.2. Clinical applications of ACLS

Clinically, ACLS may improve the precision and efficacy of clinical neuromodulation devices. By learning the appropriate stimulation patterns, the system will be able to optimize stimulation protocols for neurodegenerative and neuropsychiatric disorders. In particular, by defining a disease-specific error function, ACLS could optimize the stimulation pattern to reduce a specific neural or clinical symptom. For example, one might want to treat Parkinson’s disease by stimulating such that pathological beta-band oscillations are reduced (Little et al 2013). Alternatively, one could stimulate in a way that reduces motor tremors (Groiss et al 2009). In both cases, the same ACLS approach is used; only the error function is updated. Furthermore, because error functions can be combined, the best stimulation pattern may be one that minimizes both beta-band oscillations and tremors. In this way, ACLS could provide the ability to extend existing stimulation treatments to allow for greater flexibility in the stimulation pattern and treatment options.

Furthermore, the adaptability of ACLS could help to individualize treatments. Recent work suggests that the efficacy of deep-brain stimulation depends on physiological differences between patients and the exact placement of stimulating electrodes (Horn 2019, Greene et al 2020). The adaptability of ACLS could compensate for electrode placement or allow a greater number of electrodes to be implanted, with the system searching for the optimal stimulation pattern across those electrodes. Furthermore, current neurostimulators cannot quickly adapt to changes in state (i.e. changes in wakefulness or progression of a disease). Because the ACLS continuously tracks its own performance, it can automatically compensate for such state changes. Again, these examples highlight the power of model-free learning; model-based approaches would require infeasibly detailed, individualized, biophysical models to predict the effect of stimulation.

5. Conclusion

We demonstrate the utility of ACLS to control neural networks, both in in silico models and in in vivo mouse cortex. We found that ACLS could learn (in ~15 min) to induce a specific pattern of neural activity in a population of 5–15 multiunits in awake mouse cortex. This learning was reliable (85.42% success) and robust to common challenges of brain stimulation such as drift in state. Furthermore, learned stimulation patterns evoked neural responses that were similar to the natural responses evoked by visual stimuli. Altogether, our results show ACLS can learn, in real-time, to generate specific patterns of neural activity, providing a framework for using model-free, closed-loop learning to control neural activity.

In its current form, ACLS has several limitations that motivate future work. The flexibility offered by the ACLS’ model-free approach comes at the cost of requiring the system to independently learn the stimulation pattern for each target response. In comparison, model-based methods take a large amount of time and data to learn but, once learned, can facilitate generalization. Future work can mitigate the cost of model-free learning by developing optimization algorithms that improve the success rate and speed of learning.

The real-time nature of our experiments limited us to using a simple threshold detection to identify neural activity. This likely increased the level of noise in recorded responses and limited the selectivity of ACLS learning. Thus, real-time spike sorting may improve future ACLS performance.

Future work is also needed to test the ability of ACLS to learn over longer periods, control more complex stimulation systems (greater than the ~10 stimulation sites used here), or control different types of stimulation parameters (e.g. the pulse shape or frequency). Similarly, future work is needed to adapt ACLS to other brain stimulation paradigms, including non-invasive techniques (e.g. transcranial magnetic stimulation), and to test the ability of ACLS to improve clinical outcomes in patients.

Acknowledgments

The authors thank Alex Libby, Caroline Jahn, Matt Panichello, and Sarah Henrickson for their feedback. We also thank the Princeton Laboratory Animal Resources staff for their support. This work was funded by NIH DP2 EY025446.

Author contributions

Conceptualization: TJB, ST, CJM, KCL; Data curation: ST, CJM, TJB; Formal Analysis: ST, CJM, TJB; Funding acquisition: TJB; Investigation: ST, CJM, DC, KCL, TJB; Methodology: ST, CJM, TJB; Project
Dabney W, Kurth-Nelson Z, Uchida N, Starkweather C K, Cunningham J P and Yu B M 2014 Dimensionality reduction for solving brain-computer interfaces: An overview. J. Neural Eng. 11 335–41

Dicarlo J J, Zoccolan D and Rust N C 2012 How does the brain solve visual object recognition? Neuron 73 415–34

Emiliani V, Cohen A E, Deisseroth K and Hausser M 2015 All-optical interrogation of neural circuits J. Neurosci. 35 13917–26

Fei-Fei L, Ferguson R and Perona P 2004 Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. 2004 Conf. Comp. Vis. Pattern Recognition Workshop pp 178

Fries P 2015 Rhythms for Cognition: communication through Coherence Neuron 88 220–35

Greene D J et al 2020 Integrative and network-specific connectivity of the Basal Ganglia and Thalamus defined in individuals Neuron 105 742–58

Gross S J, Wojtsekh L, Sudmeyer M and Schnitzler A 2009 Review: deep brain stimulation in Parkinson's disease Ther. Adv. Neurol. Disord. 2 379–91

Grosicki J, Marszel J H and Deisseroth K 2015 Closed-loop and activity-guided optogenetic control Neuron 86 106–39

Hampson R E, Gerhardt G A, Marmarels V, Song D, Opriš I, Santos L, Berger T W and Deadwyler S A 2012 Facilitation and restoration of cognitive function in primate prefrontal cortex by a neuropsychosis that utilizes minicolumn-specific neural firing J. Neural. Eng. 9 056012

Hasson U, Nastase S A and Goldstein A 2020 Direct fit to nature: an evolutionary perspective on biological and artificial neural networks Neuron 105 416–34

Helfrich R F, Fiebelkorn I C, Szczepanski S M, Lin J J, Parvizi J, Knight R T and Kastner S 2018 Neural mechanisms of sustained attention are rhythmic Neuron 99 854–65

Hill D N, Mehta S B and Kleinfeld D 2011 Quality metrics to accompany spike sorting of extracellular signals J. Neurosci. 31 8699–705

Horn A 2019 The impact of modern-day neuroimaging on the field of deep brain stimulation Curr. Opin. Neurol. 32 511–20

Jazayeri M and Afraz A 2017 Navigating the Neural Space in Search of the Neural Code Neuron 93 1003–14

Jin M, Beck J M and Glickfeld L I L 2019 Neuronal adaptation reveals a suboptimal decoding of orientation tuned populations in the mouse visual cortex J. Neurosci. 39 3867–81

Knudsen E B and Wallis J D 2020 Closed-loop theta stimulation in the orbitofrontal cortex prevents reward-based learning Neuron 106 537–47

Leopold D A, O’Toole A J, Vetter T and Blanz V 2001 Prototype-referenced shape encoding revealed by high-level aftereffects Nat. Neurosci. 4 89–94

Little S et al 2013 Adaptive deep brain stimulation in advanced Parkinson disease Ann. Neurol. 74 449–57

Madadi E and Söffker D 2018 Model-free control of unknown nonlinear systems using an iterative learning concept: theoretical development and experimental validation Nonlinear Dyn. 114 1151–63

Mahendran A and Vedaldi A 2016 Visualizing deep convolutional neural networks using natural pre-images Int. J. Comput. Vis. 120 233–55

Mardiny A R, Oldenburg J A, Pégar N C, Sridharan S, Lyall E H, Chesnok K, Brohawn S G, Waller L and Adesnik H 2018 Precise multimodal optical control of neural ensemble activity Nat. Neurosci. 21 881–93

Marshel J H et al 2019 Cortical layer-specific critical dynamics triggering perception Science 365 eaaw5202

Mayberg H S, Lozano A M, Voon V, McNeely H E, Seminowicz D, Hamani C, Schwalb J M and Kennedy S H 2005 Deep brain stimulation for treatment-resistant depression Neuron 45 651–60

Mitchell B A and Petzold L R 2018 Control of neural systems at multiple scales using model-free, deep reinforcement learning Sci. Rep. 8 10721

Ponce C R, Xiao W, Schade P F, Hartmann T S, Kreiman G and Livingstone M S 2019 Evolving images for visual neurons Neurophysiol. 103 056012
using a deep generative network reveals coding principles and neuronal preferences Cell 177 999–1009
Rao R P 2019 Towards neural co-processors for the brain: combining decoding and encoding in brain–computer interfaces Curr. Opin. Neurobiol. 55 142–51
Rigotti M, Barak O, Warden M R, Wang X-J, Daw N D, Miller E K and Fusi S 2013 The importance of mixed selectivity in complex cognitive tasks Nature 497 585–90
Saddler P T, Quick K M, Golub M D, Chase S M, Ryu S I, Tyler-Kabara E C, Yu B M and Batista A P 2014 Neural constraints on learning Nature 512 423–6
Salzman C, Murasugi C, Britten K and Newsome W 1992 Microstimulation in visual area MT: effects on direction discrimination performance J. Neurosci. 12 2331–55
Saxena S and Cunningham J P 2019 Towards the neural population doctrine Curr. Opin. Neurobiol. 55 103–11
Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I and Fergus R 2014 Intriguing properties of neural networks (arXiv:1312.6199)
Tehovnik E J and Slocum W M 2007 Phosphene induction by microstimulation of Macaque V1 Brain Res. Rev. 53 337–43
Tye K M et al 2013 Dopamine neurons modulate neural encoding and expression of depression-related behaviour Nature 493 537–41
Walker E Y, Sinz F H, Cobos E, Muhammad T, Froudarakis E, Fahey P G, Ecker A S, Reimer J, Pítkow X and Tolias A S 2019 Inception loops discover what excites neurons most using deep predictive models Nat. Neurosci. 22 2060–5
Wärnberg E and Kumar A 2019 Perturbing low dimensional activity manifolds in spiking neuronal networks PloS Comput. Biol. 15 e1007074
Winston J S, Henson R N A, Fine-Goulden M R and Dolan R J 2004 fMRI-adaptation reveals dissociable neural representations of identity and expression in face perception J. Neurophysiol. 92 1830–39
Yamins D L K, Hong H, Cadieu C F, Solomon E A, Seibert D and DiCarlo J J 2014 Performance-optimized hierarchical models predict neural responses in higher visual cortex Proc. Natl Acad. Sci. 111 8619–24
Yuste R 2015 From the neuron doctrine to neural networks Nat. Rev. Neurosci. 16 487–97
Zangiabadi N, Ladino L D, Sina E, Orozco-Hernández J P, Carter A and Téllez-Zenteno J F 2019 Deep brain stimulation and drug-resistant epilepsy: a review of the literature Front Neurol. 10 601
Zhang Z, Russell L E, Packer A M, Gould O M and Häusser M 2018 Closed-loop all-optical interrogation of neural circuits in vivo Nat. Methods 15 1037–40